Project Eternal: End Term Documentation

Team 58

December 11, 2025

Contents

1 Detailed system architecture diagram **3**

2 Software architecture and flowcharts **4**

3 User Manual and Maintenance Guide **5**
3.1 User Manual . . . 5
3.1.1 Startup and Initialization . . . 5
3.1.2 Navigation and Operation . . . 5
3.1.3 Data Logging and Database Management . . . 5
3.1.4 System Shutdown . . . 6
3.2 Maintenance Guide . . . 6
3.2.1 Electrical Maintenance . . . 6
3.2.2 Mechanical Maintenance . . . 6

4 Testing and Validation Report **7**
4.1 X–Y Motion Hardware Testing . . . 7
4.1.1 Objective . . . 7
4.1.2 Methodology . . . 7
4.1.3 Results . . . 7
4.1.4 Validation Summary . . . 7
4.2 Z-Motion Hardware Testing . . . 8
4.2.1 Objective . . . 8
4.2.2 Methodology . . . 8
4.2.3 Results . . . 8
4.2.4 Validation Summary . . . 8
4.3 Camera Scan and Data Registration Testing . . . 8
4.3.1 Objective . . . 8
4.3.2 Methodology . . . 9
4.3.3 Results . . . 9
4.3.4 Validation Summary . . . 9
4.4 SLAM and Navigation Testing . . . 9
4.4.1 Objective . . . 9
4.4.2 Methodology . . . 9
4.4.3 Results . . . 11

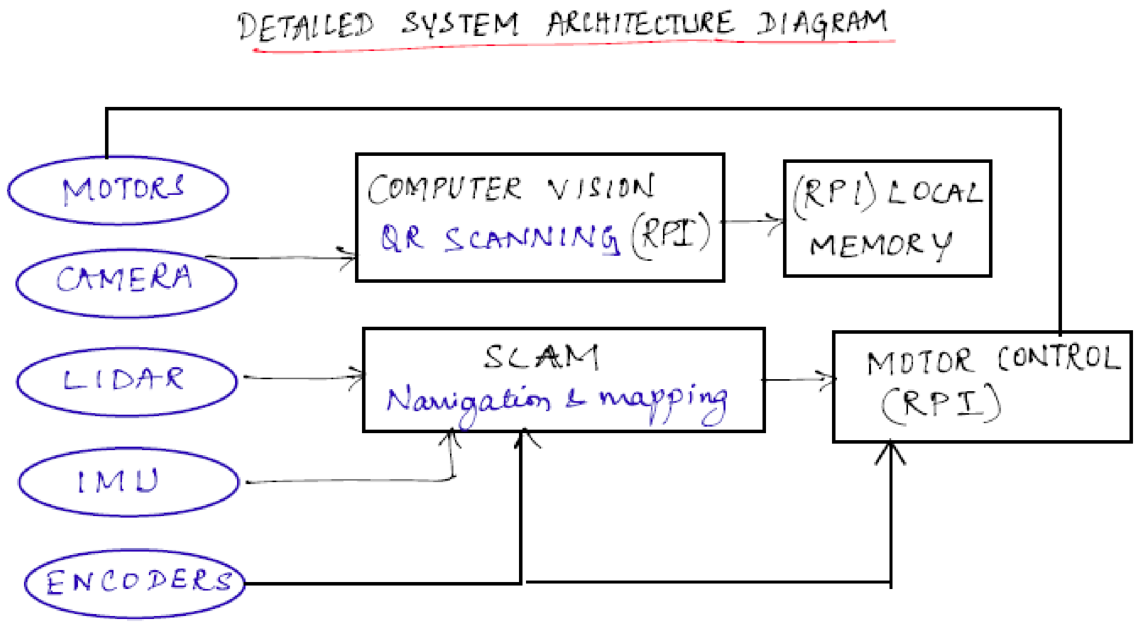
4.4.4 Validation Summary 11
4.5 Overall Conclusion 11
4.5.1 Github Link containing the code files: 11

5 Bill of Materials (BOM) with Cost Breakdown 11

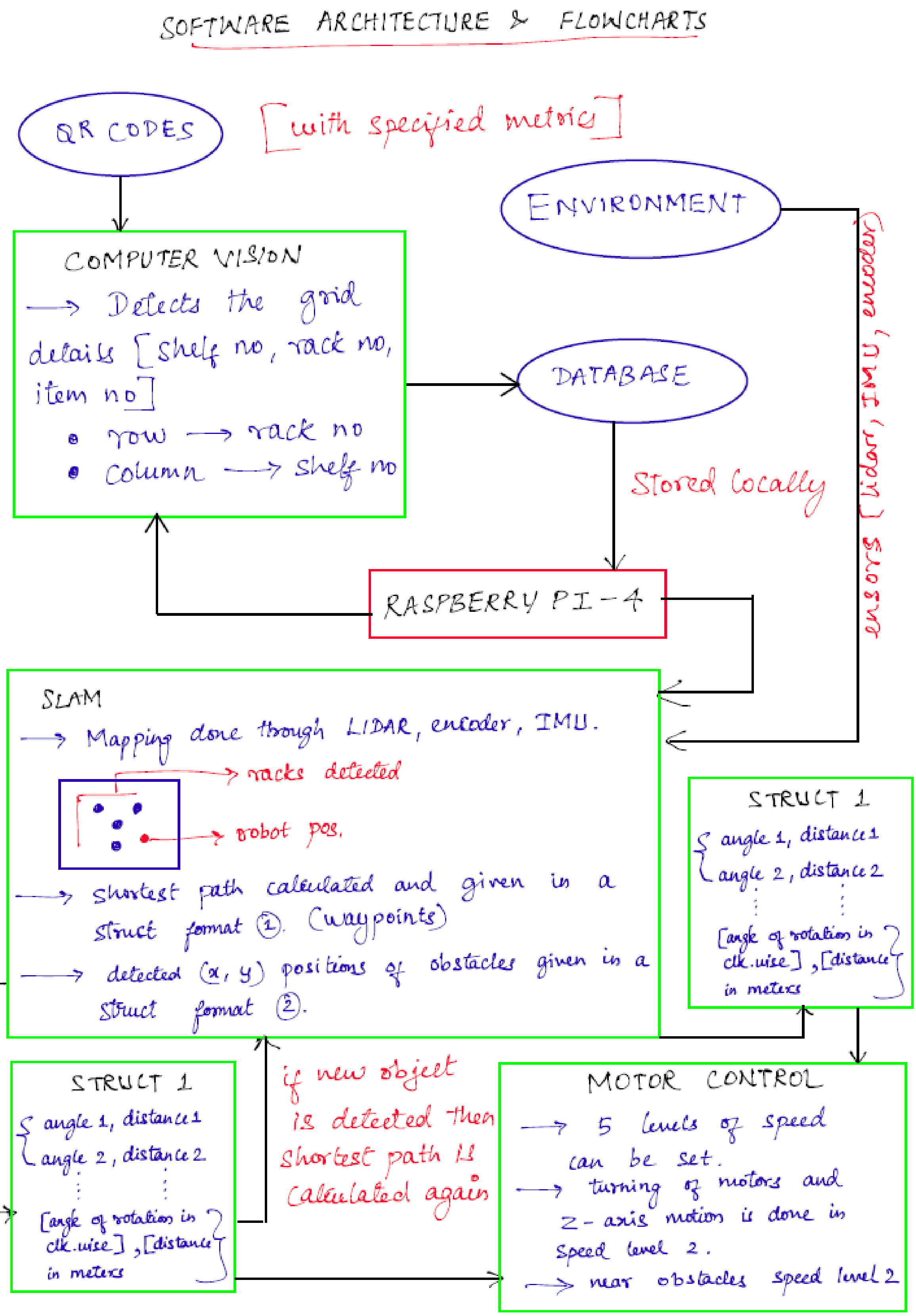
6 Circuit schematics and PCB designs 12
6.1 Hardware Design and Implementation 12
6.2 Circuit Schematic 12
6.3 PCB Design and Layout 13
6.3.1 Design Considerations 14
6.4 Bill of Materials (BOM) 14

7 3D CAD models and assembly drawings 15

1 Detailed system architecture diagram



2 Software architecture and flowcharts



3 User Manual and Maintenance Guide

3.1 User Manual

The "Eternal" rover is designed for autonomous warehouse mapping and inventory tracking. The system operates through a streamlined workflow involving mapping, navigation, and data logging.

3.1.1 Startup and Initialization

To initiate the system, power on the rover using the main switch. Upon startup, the onboard Raspberry Pi will boot the primary control software.

- **Status Indication:** The rover is equipped with a status LED. A **solid blue light** indicates that the system is initializing and the mapping module is active.
- **Mapping Phase:** The rover will autonomously traverse the warehouse environment to construct a 2D occupancy grid map using its LiDAR and IMU sensors.

3.1.2 Navigation and Operation

Once the initial mapping phase is complete, the generated map is rendered on the Graphical User Interface (GUI).

- **Autonomous Navigation:** The system switches to navigation mode automatically after mapping. The operator can view the real-time location of the bot on the GUI map.
- **Live Feed QR Detection:** During navigation, the GUI displays a live video feed from the Raspberry Pi Camera Module V2. The computer vision subsystem actively scans for QR codes on warehouse racks.
- **Visual Feedback:** When a QR code is detected, a bounding box is drawn around the code in the video feed to confirm detection. The decoded text is simultaneously displayed in the command terminal.

3.1.3 Data Logging and Database Management

The core function of the Eternal rover is inventory digitization.

- **Data Structure:** Detected information is parsed into three categories: *Item Number*, *Rack Number*, and *Shelf Number*.
- **Storage:** This data is automatically appended to a `database.csv` file located in the local directory.
- **Redundancy Check:** The software includes logic to handle duplicate scans. New items are added immediately, while repetitive scans of the same QR code are updated to reflect the most recent timestamp or location data.

3.1.4 System Shutdown

To safely halt operations, the user must execute the interrupt command `Ctrl+Shift+C` in the terminal. This ensures the `database.csv` file is saved correctly before the program terminates. The file can then be retrieved from the working directory for analysis.

3.2 Maintenance Guide

Routine maintenance is required to ensure the longevity and accuracy of the Eternal rover.

3.2.1 Electrical Maintenance

- **Battery Charging:** The rover is powered by a Lithium-Polymer (LiPo) battery pack. Ensure the battery is charged using the provided balance charger before every operation cycle. Do not let the battery voltage drop below the critical threshold (3.5V per cell).
- **Sensor Calibration:** Periodically check the IMU and LiDAR connections. If drift is observed in mapping, recalibrate the IMU on a flat surface.

3.2.2 Mechanical Maintenance

- **Assembly and Modularity:** The chassis is constructed using standard 20x20 aluminium extrusions. This modular design allows for easy assembly and disassembly using standard M4/M5 t-nuts and bolts.
- **Belt Tensioning:** Check the tension of the GT2 6mm timing belts on the Z-axis mechanism. If the belt is loose, loosen the motor mounting bracket, pull the belt taut, and retighten.
- **Wheel Alignment:** Ensure the castor wheels are free of debris (dust/thread) which can impede movement. Check the planetary gear motors for any abnormal noise indicating gear wear.

4 Testing and Validation Report

This section summarises the tests carried out on all major subsystems of the rover. Each subsystem was validated against the performance requirements stated in the problem statement. Standard testing practices were followed, including repeated trials, controlled test environments, and comparison with expected outputs. The overall objective of the testing phase was to confirm that the rover performs reliably, safely, and consistently under typical warehouse-like conditions.

4.1 X–Y Motion Hardware Testing

4.1.1 Objective

To verify that the rover can move smoothly and accurately in the X–Y plane, maintain stable trajectories, avoid obstacles, and stop safely when required.

4.1.2 Methodology

The rover uses a differential-drive configuration with two independently controlled motors. We implemented PID control for both motors to achieve accurate wheel velocity tracking. Testing included:

- Straight-line motion tests at various speeds.
- 90° and 180° turning tests.
- Repeated point-to-point navigation between fixed coordinates.
- Obstacle detection and avoidance trials.
- Emergency-stop response measurements.

All tests were conducted on a flat indoor surface similar to a warehouse floor. A layout with predefined checkpoints was created to validate repeatability.

4.1.3 Results

- **Horizontal positioning accuracy:** Within ±10 cm in repeated trials.
- **Path tracking:** Smooth motion with minimal deviation under PID control.
- **Obstacle detection:** Consistently triggered within the 5 m detection range.
- **Emergency stop:** Average response time remained under the 500 ms requirement.
- **Repeatability:** The rover was able to reach the same target point reliably across repeated runs.

4.1.4 Validation Summary

The X–Y motion subsystem met the required accuracy, responsiveness, and safety conditions. PID-based control gave stable performance, and the system reliably avoided obstacles while maintaining smooth navigation.

4.2 Z-Motion Hardware Testing

4.2.1 Objective

To validate the vertical scanning mechanism and ensure that it can move the camera assembly smoothly, safely, and within the required precision.

4.2.2 Methodology

The Z-axis system is based on a timing-belt and toothed-pulley arrangement. The following tests were carried out:

- Full vertical traversal from bottom to top and back.
- Stability assessment to check for wobbling or lateral play.
- Vertical positioning accuracy measurements.
- Load-holding tests to ensure no slippage under static conditions.
- Motion-smoothness evaluation during scanning.
- Over-travel and safety-limit validation.

Multiple cycles were performed to verify long-term consistency.

4.2.3 Results

- **Vertical accuracy:** Within the required ± 2 cm.
- **Traversal time:** Consistently under the expected duration for a full scan cycle.
- **Stability:** No noticeable wobble or drift during movement or while stationary.
- **Repeatability:** Vertical positions were reached reliably across cycles.
- **Safety:** Limit switches and software checks prevented over-extension.

4.2.4 Validation Summary

The timing-belt mechanism performed reliably and provided steady, vibration-free motion suitable for high-quality image capture. All Z-axis performance requirements were met.

4.3 Camera Scan and Data Registration Testing

4.3.1 Objective

To ensure that the camera can scan the entire rack, capture clear images, detect QR codes reliably, and register the captured data correctly.

4.3.2 Methodology

We performed repeated scanning runs on racks containing known ground-truth QR codes. The tests covered:

- Full vertical scan of the rack.
- Image clarity checks at different lighting levels.
- QR code detection and decoding accuracy.
- Comparison of the detected information with the ground-truth dataset.
- End-to-end data registration verification.

The scans were performed while the Z-axis was in motion to confirm that motion blur remained minimal.

4.3.3 Results

- The camera produced sharp, high-resolution images suitable for recognition.
- QR codes of size 5 cm $\times$ 5 cm were detected accurately and consistently.
- Data stored after each scan matched the ground-truth values.
- Vertical scanning took less than three minutes per rack.

4.3.4 Validation Summary

The camera and scanning pipeline worked reliably under repeated trials. Data extraction and registration were accurate, and all required KPIs for scan speed and quality were satisfied.

4.4 SLAM and Navigation Testing

4.4.1 Objective

To validate mapping, localization, and autonomous navigation using ROS-based tools and multi-sensor fusion.

4.4.2 Methodology

The following workflow was used:

1. **Environment creation:** A CAD model of the warehouse layout was prepared, exported as a .dae file, and loaded into a Gazebo world. This allowed controlled simulation and repeatable testing.
2. **SLAM mapping:** We used SLAM Toolbox with input from:
    - A 2D RPLIDAR
    - IMU data
    - Odometry computed from wheel encoders

3. **Sensor fusion:** An Extended Kalman Filter (EKF) node fused lidar, IMU, and odometry data to obtain a stable estimate of the robot's pose.

4. **Map generation:** After several mapping runs, the map was saved for use during navigation.

5. **Navigation testing:** Using the Nav2 stack, we defined checkpoints in the map and commanded the robot to navigate autonomously between them. Validation included path-planning latency, successful arrival at checkpoints, collision avoidance, and recovery behaviors.

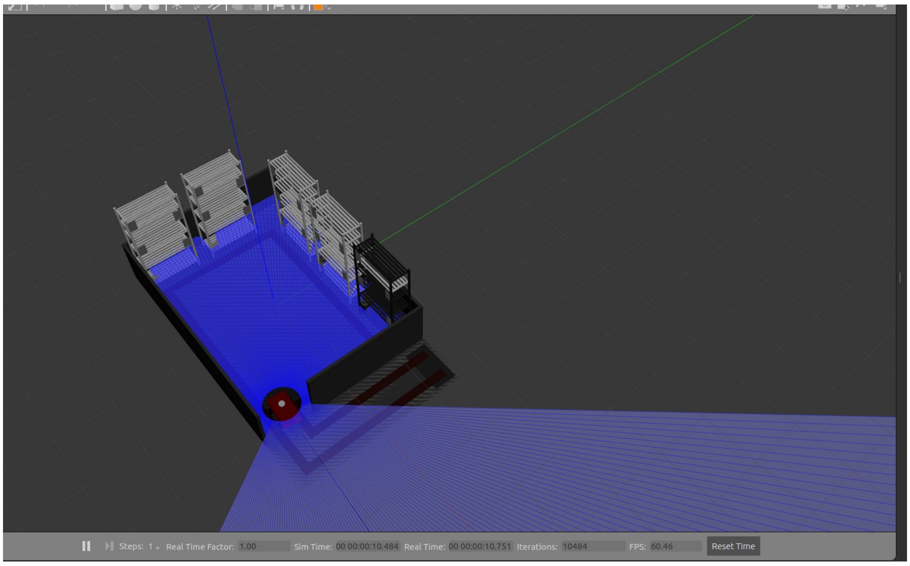


Figure 1: Gazebo World File

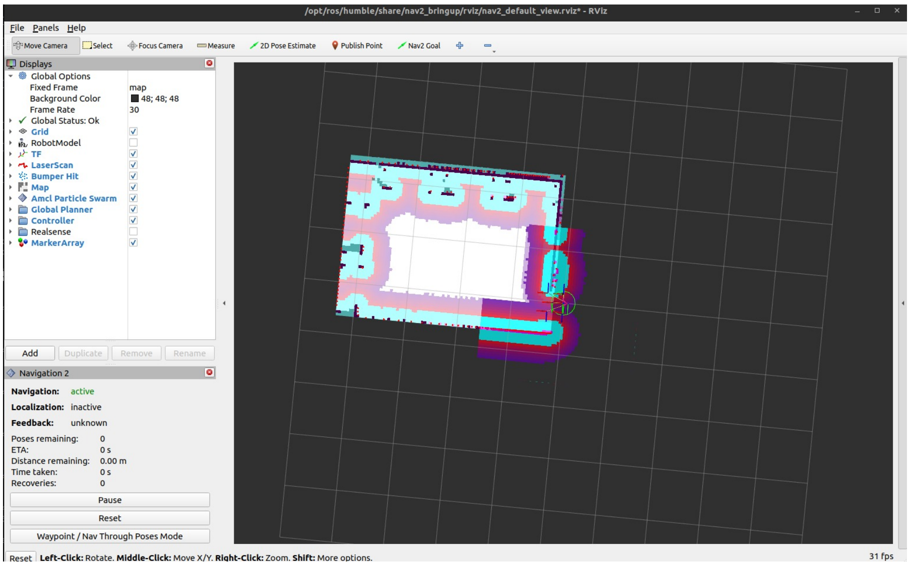


Figure 2: Slam Map

4.4.3 Results

- SLAM produced consistent and stable maps with no major distortions.
- Localization remained stable even during turns and near obstacles.
- Nav2 successfully planned and executed paths between all checkpoints.
- The robot maintained safe distances from obstacles throughout operation.
- Recovery actions (e.g., re-planning, rotating in place) worked correctly when intentionally disturbed.

4.4.4 Validation Summary

The SLAM and navigation system met all functional expectations. The combination of lidar, IMU, and odometry produced robust localisation, and Nav2 handled autonomous navigation reliably within the mapped environment.

4.5 Overall Conclusion

All major subsystems—X–Y motion, Z motion, scanning and data registration, and autonomous navigation—were tested thoroughly following standard engineering validation practices. The rover consistently met the key performance requirements specified in the problem statement and demonstrated reliable operation across repeated trials.

4.5.1 Github Link containing the code files:

https://github.com/Aech-7/ros2-slam-auto-navigation

5 Bill of Materials (BOM) with Cost Breakdown

The following table details the components utilized in the construction of the Eternal rover, including actuation, sensing, processing, and structural elements.

Item Category	**Component Name / Specification**	**Qty**	**Unit Cost ()**
Actuation (Locomotion)	Industrial Grade IG45 Planetary DC Geared Encoder Servo Motor (50W, 500RPM, 12V, 5.7KGCM)	2	4,307
Motor Driver	Cytron 20Amp 6V-30V DC Motor Driver (2 Channels) MDD20A	1	3,091
Actuation (Z-Axis)	DC 12V 200RPM High Torque Quad Encoder Motor	1	1,770
Driver (Z-Axis)	Cytron DC Motor Driver MD13S 13Amp	1	1,178
Hardware	DC Motor Mounting Bracket	2	177
Hardware	Mounting Clamp for Z-axis motor	1	91

Continued on next page

Item Category	Component Name / Specification	Qty	Unit Cost ()
Wheels	Castor Wheels	4	64
Wheels	High-Traction Wheels	2	2,478
Transmission	Timing Belt GT2 6mm (4 meters)	1	480
Transmission	16 Tooth 6mm Bore GT2 Timing Aluminum Pulley	2	80
Sensor (LiDAR)	RPLIDAR A1M8 - 360 Degree Laser Scanner (12m range)	1	7,500
Sensor (IMU)	Inertial Measurement Unit (9-DOF)	1	2,300
Sensor (Proximity)	IR Sensor Module	4	120
Processing	Raspberry Pi 4 Model B (8GB RAM)	1	*Owned
Microcontroller	ESP32 Development Board	1	460
Vision	Raspberry Pi Camera Module V2 (5MP)	1	296
Storage	32 GB Micro SD Card (Class 10)	1	*Owned
Indicators	3W LED Module	1	76
Sensor (Distance)	Ultrasonic HC-SR04 Sensors	5	60
Safety	Emergency Stop Kits	2	350
Structure	Aluminium Extrusions 20x20	5	660

**Note: Items marked as "Owned" were sourced from existing lab inventory and did not incur direct project costs during this phase.*

6 Circuit schematics and PCB designs

6.1 Hardware Design and Implementation

The hardware architecture of the autonomous robot is designed to ensure robust power distribution and low-latency communication between the central processing unit (Raspberry Pi 4) and the peripheral sensors and actuators. This section details the circuit schematics, the custom Carrier Board (HAT) design, and the component specifications.

6.2 Circuit Schematic

The electrical system is divided into three subsystems: the High-Power Drive System (12V), the Logic Regulation System (5V), and the Sensor Interface. Figure 3 illustrates the interconnections.

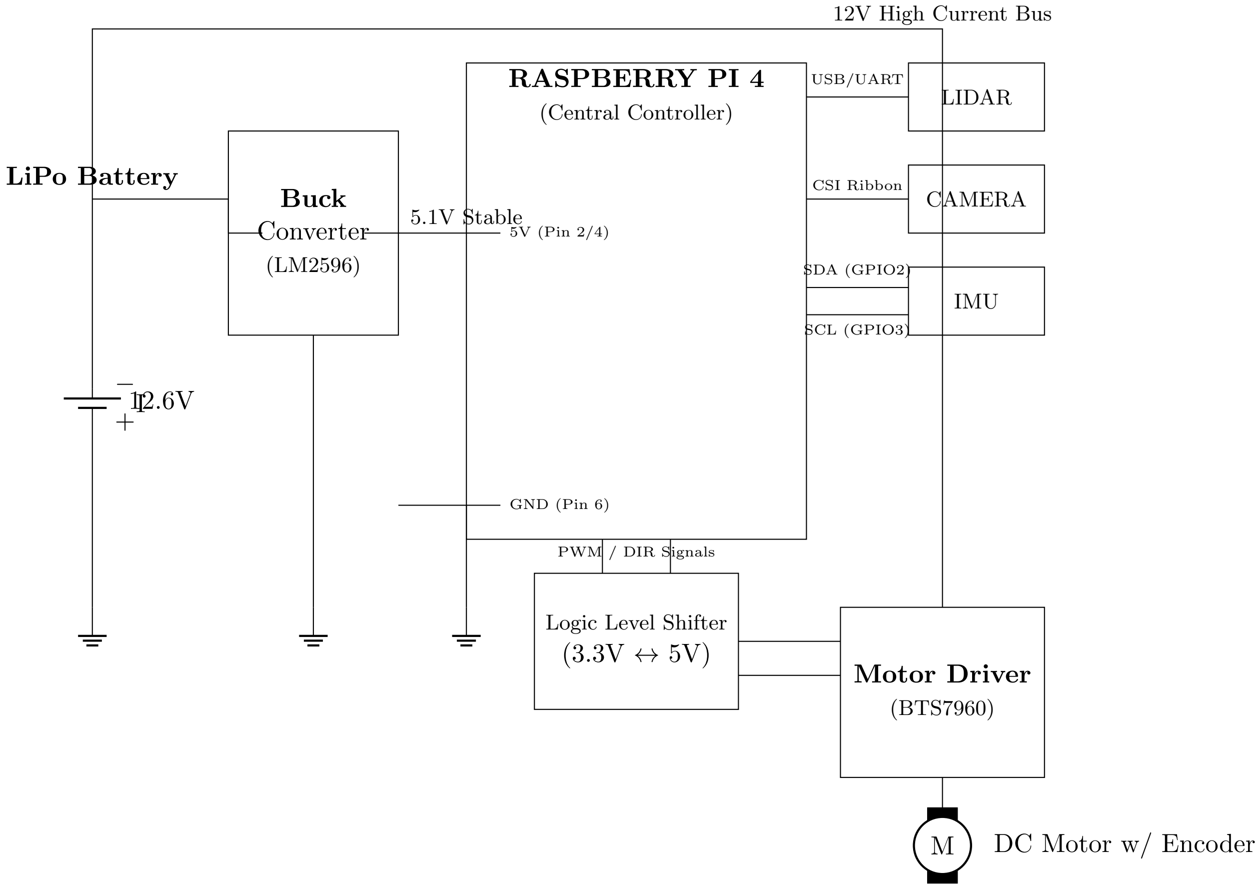


Figure 3: System-level circuit schematic detailing power distribution and signal flow.

The power system utilizes a high-discharge LiPo battery regulated via an LM2596 buck converter to provide a clean 5.1V supply to the Raspberry Pi, preventing brownouts during motor current spikes. The motor control logic utilizes a bi-directional logic level shifter to bridge the Raspberry Pi's 3.3V GPIO logic with the 5V logic required by the high-power BTS7960 motor drivers.

6.3 PCB Design and Layout

To eliminate unreliable jumper wires and ensure mechanical stability in a moving robot, a custom Carrier Board (HAT) was designed. This 2-layer PCB mounts directly onto the Raspberry Pi via the 40-pin GPIO header.

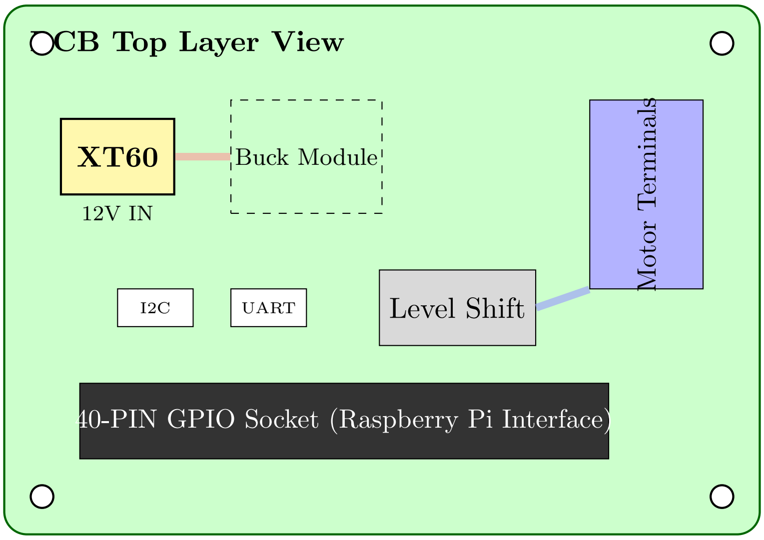


Figure 4: Conceptual layout of the Custom Carrier HAT showing component placement.

6.3.1 Design Considerations

The PCB layout adheres to the following design rules to minimize electromagnetic interference (EMI):

- **Power Planes:** A solid ground plane on the bottom layer acts as a shield for signal lines and ensures a common reference voltage.
- **Trace Widths:** High-current motor tracks (12V) utilize 30-mil width traces to handle loads up to 5A, while logic signals utilize standard 10-mil traces.
- **Connector Placement:** High-vibration connectors (Screw terminals and XT60) are placed at the board edges for strain relief.

6.4 Bill of Materials (BOM)

Table 2 lists the core components selected for the hardware implementation.

Table 2: Hardware Bill of Materials

Component	**Qty**	**Specification / Role**
Raspberry Pi 4 Model B	1	Main Controller (4GB RAM). Handles SLAM and CV.
RPLIDAR A1M8	1	360-degree Laser Range Scanner (12m range).
BTS7960 Driver	2	43A High Power H-Bridge for DC Motors.
LM2596 Module	1	DC-DC Buck Converter (12V to 5V 3A).
BNO055 IMU	1	9-DOF Absolute Orientation Sensor (I2C).
LiPo Battery	1	3S (11.1V) 2200mAh 30C Discharge.
Level Shifter	1	4-Channel Bi-directional (3.3V to 5V).

7 3D CAD models and assembly drawings

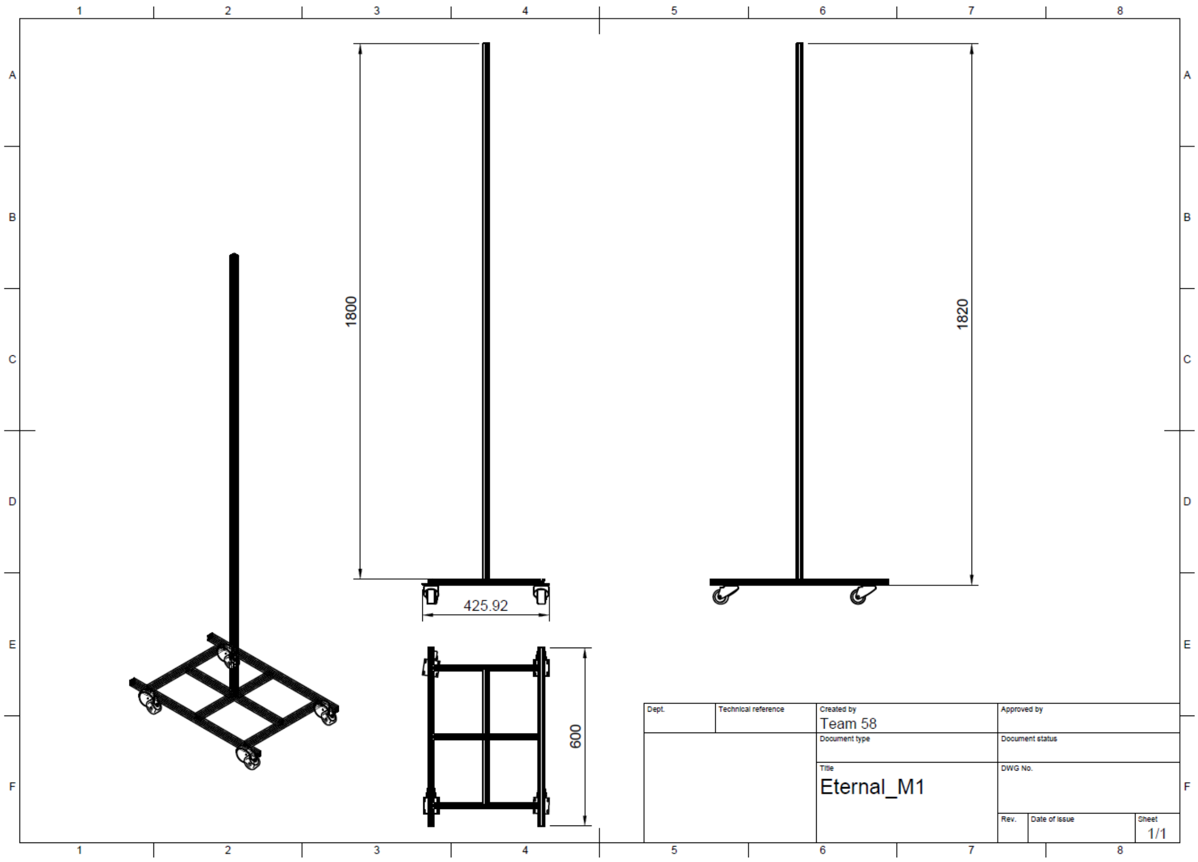


Figure 5: raw chasis