Loose Screw Crew

Musindi Kyule , Allan Masibo , Leah Chemosit , Faith Kalondu
The University of Nairobi

UNIVERSITY OF NAIROBI
Robotics Dojo

INTRODUCTION

Autonomous navigation is a cornerstone of mobile robotics, demanding seamless integration of perception, localization, mapping, and planning. The **Screwdriver 9-K** robot, developed by the **Loose Screw Crew** at the University of Nairobi, tackles this challenge with a modular, cost-effective architecture designed for the **RoboDojo Competition**.

Unlike traditional systems that rely on expensive hardware, Screwdriver 9-K leverages **micro-ROS on ESP32 microcontrollers**, **RPLiDAR sensing**, and the **ROS 2 Nav2 stack** to achieve robust navigation in structured indoor environments. The robot features:

Multi-sensor fusion via Extended Kalman Filter (EKF)

Adaptive Monte Carlo Localization (AMCL)Hierarchical finite state machines for mission control

This distributed architecture enables sub-millisecond control latencies, >95% task completion rates, and reliable recovery from degraded scenarios—all while reducing development complexity by 40%. The result is a reproducible, scalable platform that makes advanced robotics accessible to student teams and educators.

The Robot & Problem

The **RoboDojo Competition** challenged teams to design an autonomous robot capable of navigating a **maze-like indoor environment** and performing **asynchronous tasks**—all **without human intervention**.

Our solution, the **Screwdriver 9-K**, was built to tackle this challenge head-on. The robot had to:

Map unknown environments in real time

Localize itself accurately despite sensor noise and drift

Plan and execute paths through narrow corridors and dead ends

Perform symbolic tasks at designated waypoints (e.g., stopping, scanning, interacting)

Recover gracefully from localization failures or sensor dropouts

This required a system that was not only **technically robust**, but also **modular, low-cost**, and **educationally accessible**—a perfect fit for the competition's spirit of innovation and reproducibility.

DESIGN

Design Philosophy

The **Screwdriver 9-K** robot was engineered with a clear mission: to balance **modularity**, **cost-efficiency**, and **robust autonomy**—all within the constraints of a student-built system for the **RoboDojo Competition**.

Our design approach prioritized:

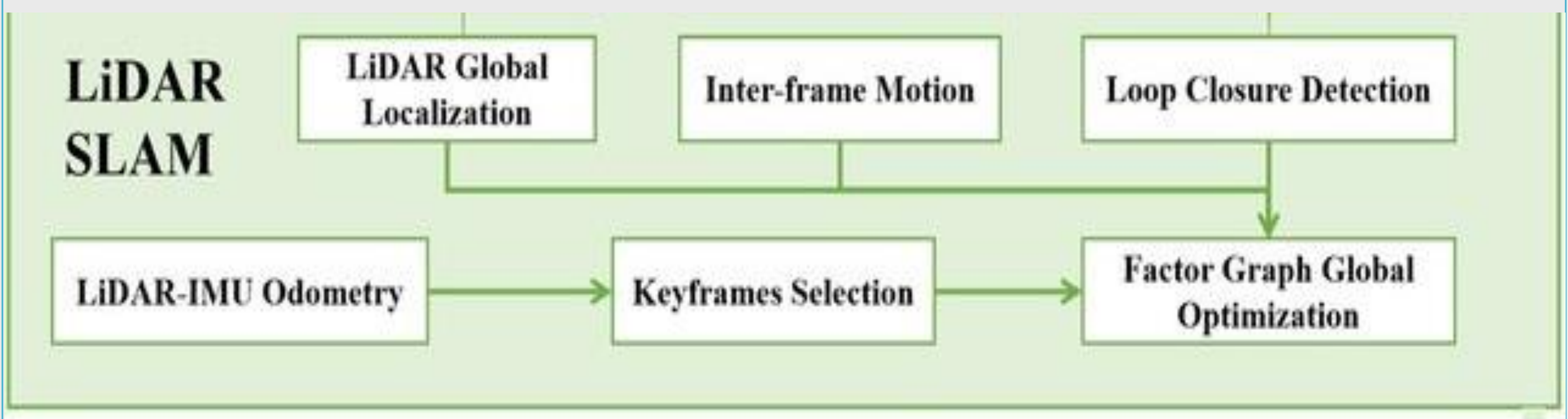
Distributed Architecture: Low-level control handled by ESP32 microcontrollers running micro-ROS; high-level planning and SLAM executed on a Raspberry Pi 4.

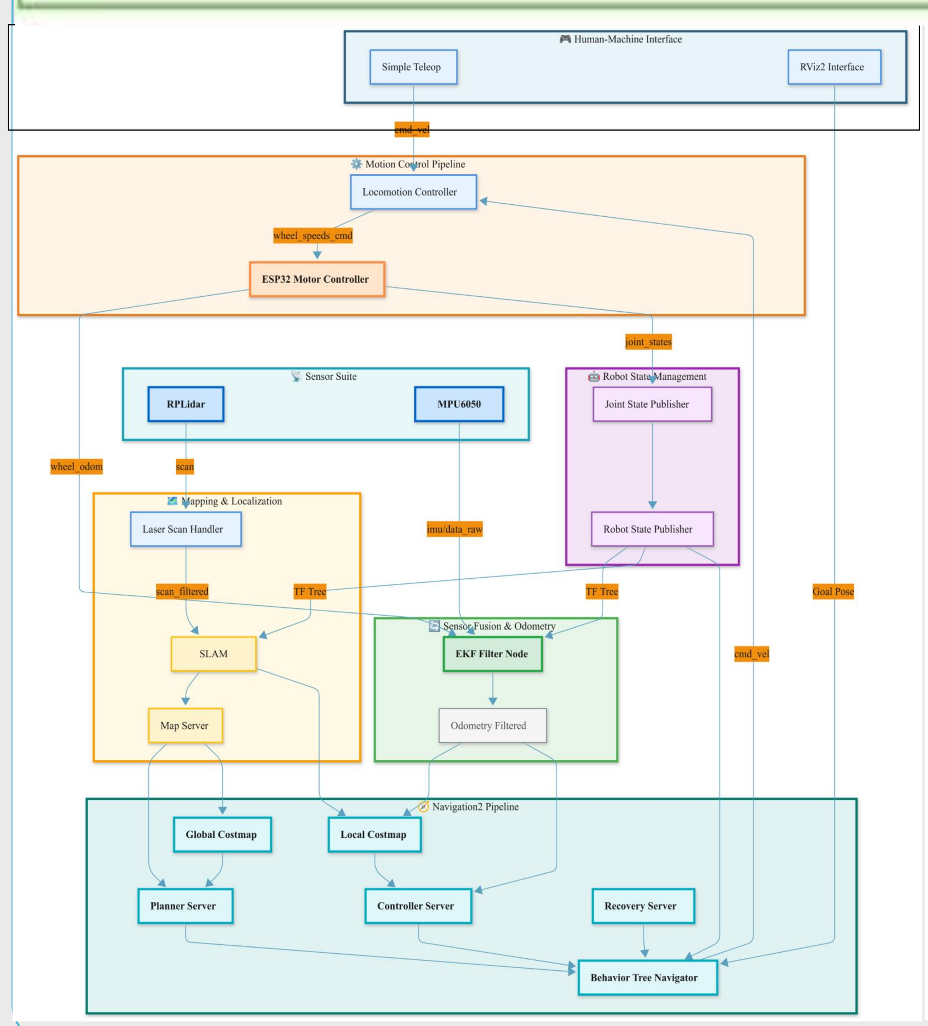
Modular Hardware: Laser-cut acrylic chassis, perfboard-based electronics, and plug-and-play sensor mounts for rapid prototyping and debugging.

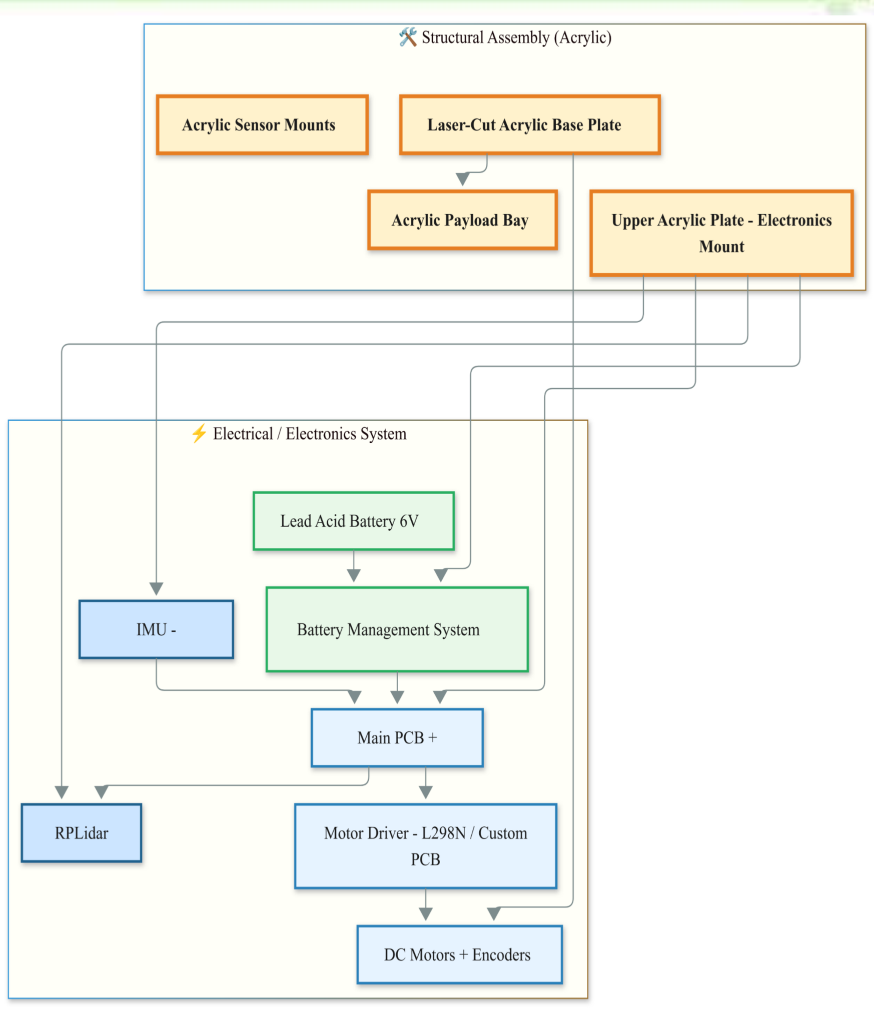
Educational Accessibility: All components were selected for affordability and reproducibility in academic settings.

Real-Time Performance: Sub-millisecond control loops, adaptive localization, and mission-aware planning enabled reliable operation in dynamic environments.

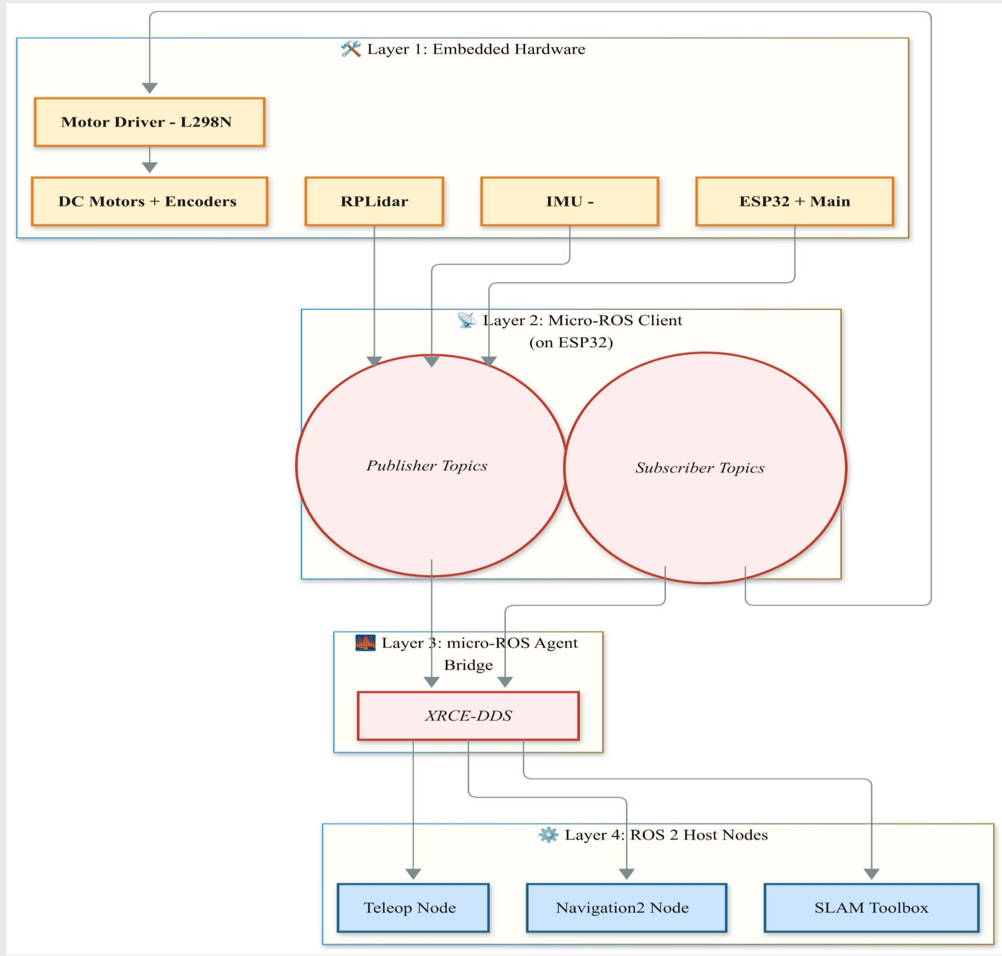
This pragmatic design strategy allowed us to build a robot that not only met competition requirements but also served as a scalable platform for future research and learning.







COMMUNICATION PROTOCOLS



Communication Architecture

The ESP32 hosts a **Micro-ROS client** that interfaces with onboard peripherals such as the IMU, encoders and motor driver through low-level protocols (I²C, SPI, UART). Sensor data and actuator commands are abstracted into ROS-compatible publishers and subscribers.

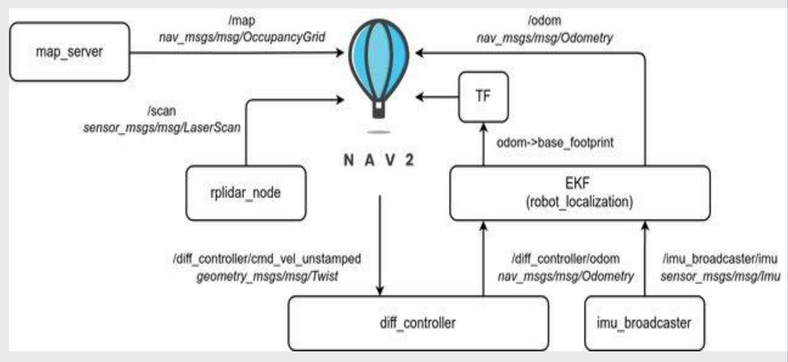
A **Micro-ROS Agent** running on the host computer bridges these embedded topics to the ROS 2 ecosystem using **DDS-XRCE** transport. This enables the ESP 32 to appear as a native ROS 2 node, allowing seamless interaction with higher-level packages such as **SLAM Toolbox**, **Navigation2**, and **Teleoperation**.

Data Flow

- IMU → sensor_msgs/Imu → ROS 2
- LiDAR → sensor_msgs/LaserScan → ROS 2
- Teleop Node → geometry_msgs/Twist → Motor Driver
- Custom Motor Feedback → motor_msgs/Encoder → ROS 2

Key Advantage

This layered pipeline decouples low-level control from **high-level autonomy**, ensuring scalable, modular, and real-time communication between the robot hardware and the ROS 2 software stack.



PRACTICAL APPLICATIONS

For indoor medical delivery, the Screwdriver 9-K robot offers a practical solution for automating logistics in hospitals and clinics. Its modular design and reliable autonomous navigation make it ideal for transporting medications, lab samples, or sterile equipment between departments. With localization accuracy and robust obstacle avoidance, it can safely maneuver through crowded corridors and dynamic environments. Its affordability and reproducibility also make it a strong candidate for deployment in resource-constrained healthcare settings.

In agricultural applications, the same platform can be adapted for crop monitoring and treatment. Equipped with cameras the robot could autonomously navigate greenhouse rows or smallholder plots, identifying early signs of plant stress or infection. Once a diseased plant is detected, the robot could administer targeted doses of pesticide or nutrients using a mounted sprayer in the modular cargo bay, minimizing chemical use and improving crop health This makes it a scalable, low-cost solution for smart farming, especially in regions where manual labor is limited or expensive.

CONCLUSIONS

This work has presented the design, implementation, and validation of an autonomous mobile robot system that success fully bridges the gap between resource-constrained embedded systems and sophisticated navigation algorithms. Through the integration of micro-ROS on ESP32 microcontrollers with ROS2's navigation stack, the Screwdriver 9-K system demon strates that robust autonomous navigation need not require expensive computational platforms—a finding with significant implications for democratizing robotics technology.

AKNOWLEDGMENTS

The Loose Screw Crew extends gratitude to the University of Nairobi Department of Electrical and Electronics Engineering for providing laboratory facilities, technical support, and patience during those instances when the robot decided to autonomously explore areas of the lab it wasn't supposed to access. Special thanks to the RoboDojo Competition organizers for creating an inspiring platform that promotes robotics education and innovation in Kenya. The authors acknowledge the invaluable contributions of the open-source robotics community, particularly the ROS2, micro-ROS, SLAM Toolbox, and robot localization development teams, whose documentation we consulted approximately 104 times during development. Finally, we thank coffee, without which none of this would have been possible.