SkyRocket Project Report

CAZAUBON Lorenz & MAUVOISIN Paul

This document symbolizes the end of the SkyRocket project. It is a report that summarizes the work done during the project, including the design and testing of the autonomous chassis. More info can be found here: **github.com/Templatew/SkyRocket/**

April 14, 2025

Contents

1. Introduction

The development of this robot is part of Polytech Nice-Sophia's participation in the **French Robotics Cup**, an annual event where teams, primarily engineering students, design and build autonomous robots. The competition involves two robots competing on a 3-meter by 2-meter game table, where they must complete pick-and-place missions entirely autonomously, without human intervention, within a 100-second timeframe. At the end, points are awarded based on the tasks successfully performed.

Having competed last year, we identified a critical weakness in our robot's imprecise movements, which undermined the effectiveness of our mission systems. This experience guided our focus for this year's project, detailed in this report, which centers on improving the robot's mobility, orientation, and localization. The competition's regulations include specific constraints, such as height and perimeter limits, which define the robot's dimensions and design.

To address localization, we plan to use a system of three fixed beacons around the game table to triangulate the robot's position. This approach, permitted by the competition rules, is far simpler and more reliable than our previous attempt to use camera-based image detection paired with artificial intelligence. However, beacons only provide positional data, not the robot's orientation. To solve this, we considered placing two beacons on the robot itself but found this costly. Instead, we opted for a more affordable magnetometer to determine orientation accurately.

Additionally, the competition strictly prohibits collisions with other robots, making obstacle detection essential, even though the game table's elements are mapped out beforehand and our localization is precise. Most competitors, including us, rely on a LIDAR system for emergency obstacle detection to ensure safe navigation during matches.

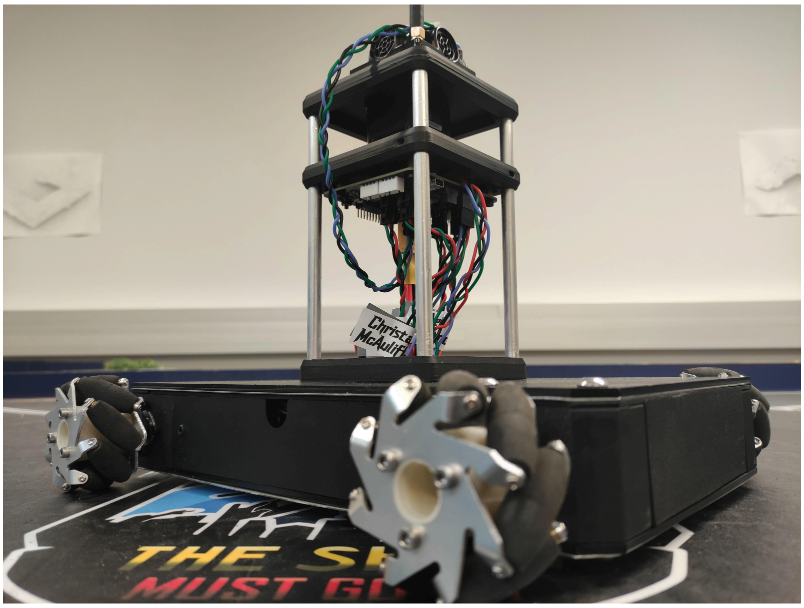

Figure 1: SkyRocket Robot

Figure 2: SkyRocket Logo

2. Modules

2.1. Beacons

The use of beacons is key to our project, enabling precise and reliable robot localization on the game table. After reviewing options, we chose Marvelmind beacons. Despite their high cost, they offer exceptional indoor accuracy, with errors of just a few centimeters, meeting our needs and avoiding past positioning issues.

We aim to build a robust system for future competitions, with an efficient mobile base that won't need redesigning. Marvelmind beacons use ultrasonic triangulation, with three fixed beacons around the table (per regulations) and a mobile beacon on the robot. Time-of-flight measurements calculate the robot's exact 3D coordinates.

This technology was preferred over last year's complex and unreliable camera-based vision system. The beacon's built-in IMU (accelerometer and gyroscope) provides additional data on orientation and movement, enhancing position and alignment accuracy for tasks like pick-and-place through filtering.

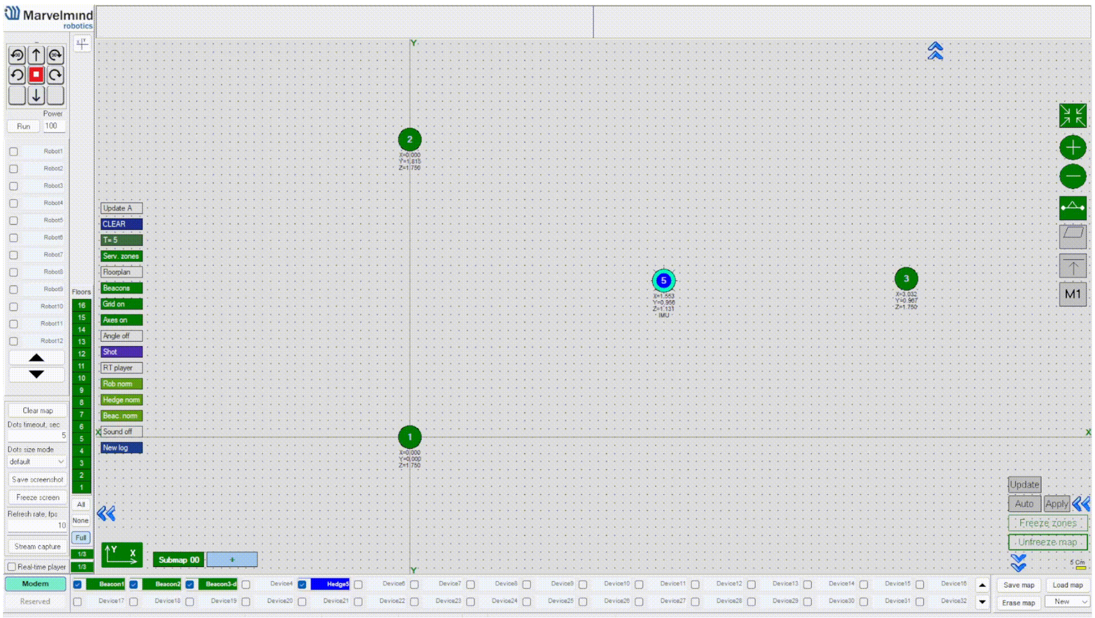

Figure 3: Beacon Software UI

2.2. Lidar

This year, our robot's second key module is a 2D LIDAR for obstacle detection, ensuring no collisions with opposing robots, as mandated by the French Robotics Cup rules.

Despite precise beacon-based localization and known game element positions for optimal path planning, the opposing robot's movement adds unpredictability. The LIDAR monitors the surroundings in real-time, triggering an emergency stop if needed.

It uses 360-degree lasers to measure distances to obstacles accurately. Our user-friendly LIDAR sends UART data in twelve-point packets, including angles, distances, and intensity.

The LIDAR's height is crucial, positioned high to detect only the opposing robot and avoid confusion with game elements, ensuring motors stop when necessary.

3. Mechanical Design

The entire robot design was done using SOLIDWORKS, a powerful CAD software. The design process was centered arround levereaging the omnidirectional wheels, which are a key feature of our robot. These wheels allow the robot to move in any direction without needing to turn, making it highly maneuverable.

Then we designed the chassis to be as compact as possible, while still allowing for all the necessary components to fit inside. The design was also optimized for stiffness and agility, ensuring that the robot could withstand the rigors of competition while still being able to move quickly and efficiently.

And finally the design had to follow the competition rules and at the same time be modular enough to allow for easy assembly, dissambly, maintenance and upgrades.

To follow all these constraints, the use of a powerful software like SOLIDWORKS was essential. It allowed us to create a detailed 3D model of the robot, which we could then use to test the design and make any necessary adjustments before moving on to the manufacturing phase.

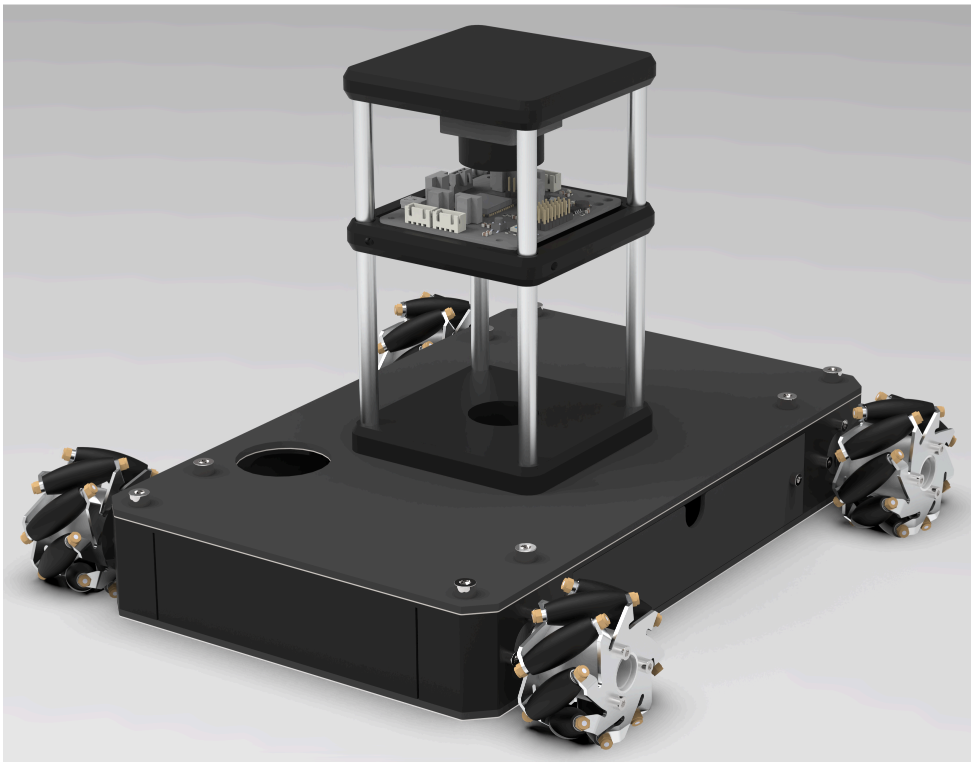

Figure 4: Render of the Robot

Figure 5: Render Close Up

4. PCB Design

4.1. Main PCB

Designing our own PCBs allowed us to master custom electronics for robotics, moving away from less capable boards like Arduino. We chose the ESP32-WROOM-S3 microcontroller, widely used in industry, with dual cores to handle mission programming and communication separately.

The ESP32 requires 3.3V, but the LIDAR needs 5V, so we added a buck converter to reduce the battery's 11V to 5V, powering the LIDAR and an LED for feedback when the robot operates independently. An LDO then converts 5V to 3.3V for the ESP32 and an integrated magnetometer, communicating via I2C.

The PCB also uses UART for data from the LIDAR and beacon, ensuring efficient communication across components.

4.2. Power Splitter PCB

Additionally, a design flaw in the control PCB led to the creation of a second power distribution PCB, which supplies energy to all the robot's modules and boards. When the control PCB was powered, the motors connected to it would briefly activate unintentionally.

To address this issue, adding a diode to prevent the control PCB from supplying power to the power PCB eliminated the unwanted motor activation.

Furthermore, it's beneficial to include a MOSFET on this power PCB, which supplies both the control PCB and the motors. The MOSFET acts as a programmable **kill-switch** to enable or disable the motors, allowing the robot to be powered on without the motors being activated. This is particularly useful for testing, remote control, and safety.

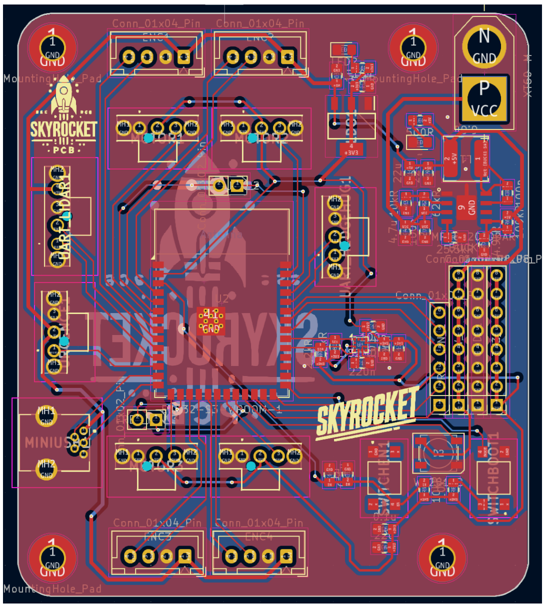

Figure 6: Main PCB Layout

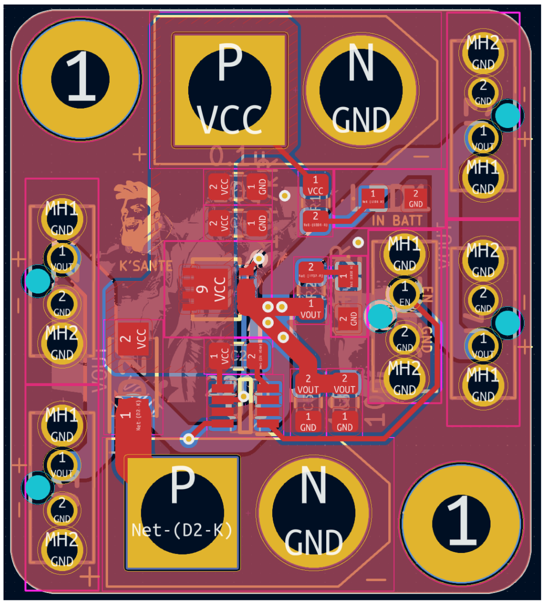

Figure 7: Power splitter PCB Layout

5. Programming

5.1. Embedded

The embedded programming was done using the PlatformIO IDE, which is a powerful and flexible IDE for embedded programming. It allows us to easily manage the code for the ESP32 microcontroller, including libraries and dependencies.

For now the code is not very complex, as we are still in the testing phase. The robot is controlled by the PC via Bluetooth Low Energy (BLE) and depending on the command received will launch different tasks, such as moving, yaw compensation or even the goto function that lets the robot go to a specific position on the game table on its own.

The complete implementation details and source code are available for review in our GitHub repository.

5.2. GUI using Qt

To control the robot during testing we developped a GUI using Qt, which is a powerful framework for creating cross-platform applications. The GUI allows us to easily control the robot and monitor its status in real-time. It provides a user-friendly interface for sending commands to the robot and receiving feedback on its status.

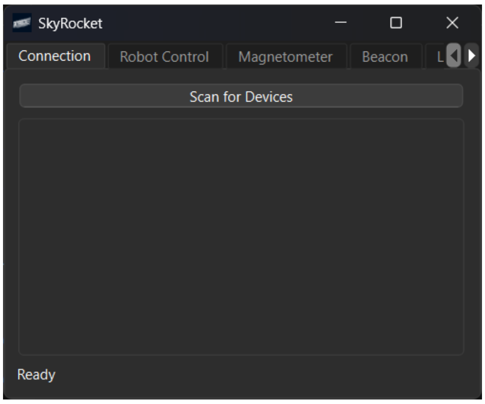

Figure 8: Connection Widget

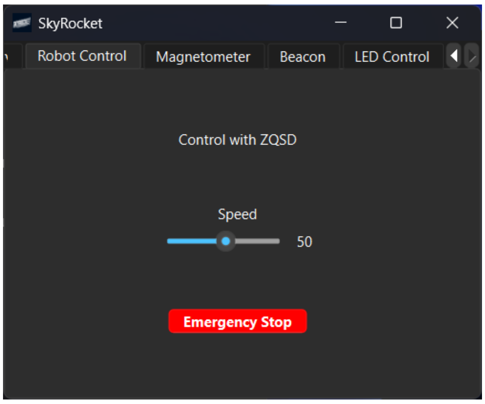

Figure 9: Manual Control Widget

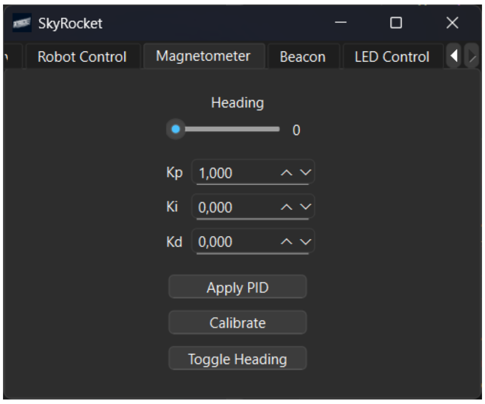

Figure 10: Magnetometer Widget

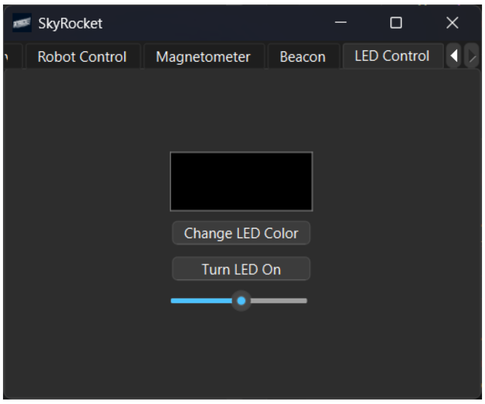

Figure 11: Led Control Widget

6. Issues, Solutions and Prospects for Improvement

6.1. Magnetometer Interference

During the robot's development, the primary challenge was the magnetometer's interference from the motors' magnetic field. Since the magnetometer was crucial for determining precise orientation to ensure accurate movements, fixing this was vital for the robot's performance. Our solution was composed of three steps:

1. Move the magnetometer as far away from the motors as possible.
2. Add steel plates to shield the magnetometer from the motors' magnetic field.
   Which also contributed to the robot's stiffness.
3. Program a more robust calibration algorithm.

6.2. Torque Transmission

To move our robot efficiently, we needed to ensure that the motors could transmit enough torque to the wheels. The initial design used a 3D-printed part to connect the motor shaft to the wheel, but this was not strong enough. We replaced those with forged carbon parts that were much more rigid and could withstand the torque generated by the motors.

6.3. Regulation Compliant Batteries

In the rule book, it is stated that the batteries must either be placed in a fire-proof bag or use a "safe" technology such as LiFePO4. We used LiPo batteries, which are not directly compliant with the rules. To solve this issue, we placed the batteries in a fire-proof bag and added a low-voltage alarm to the robot. This way, we could ensure that the batteries were safe and that we were compliant with the rules.

6.4. Prospects for Improvement

After this development phase, we identified ways to improve our robot's performance, particularly in movement precision, for future iterations or competitions.

First, we could use Kalman filters, combining the magnetometer, beacon and IMU data to improve the robot's localization. This would enhance the accuracy of the robot's position and orientation, allowing for more precise movements.

Secondly, we could make our own battery management system (BMS) to monitor the battery's state of charge and health. This would allow us to optimize the battery's performance and lifespan, ensuring that the robot operates at its best.

Finally, we could improve the robot's software by implementing more advanced algorithms for path planning and obstacle avoidance. This would allow the robot to navigate more efficiently and avoid collisions with other robots or obstacles on the game table.

7. Conclusion

At the conclusion of this project, we are fully satisfied with the results. The mobile base, our main focus this year, is now fully operational, meeting the precision, reliability, and robustness requirements for the French Robotics Cup. This was achieved through optimized localization with beacons, precise orientation via the magnetometer, and effective obstacle detection with the 2D LiDAR.

Beyond technical performance, we take pride in the chassis design. It not only complies with regulations but also stands out for its refined aesthetics and durability, ensuring long-term use.

This project provided a unique opportunity to gain skills not typically taught in robotics courses. Designing and producing our own electronic boards was a successfully met challenge, as was creating a custom Bluetooth application to simplify robot parameter adjustments.

The next step is to design the upper part of the robot, focusing on manipulating game elements and completing missions. With this solid foundation, we are confident in developing pick-and-place systems to maximize our performance at the competition.

Index of Figures