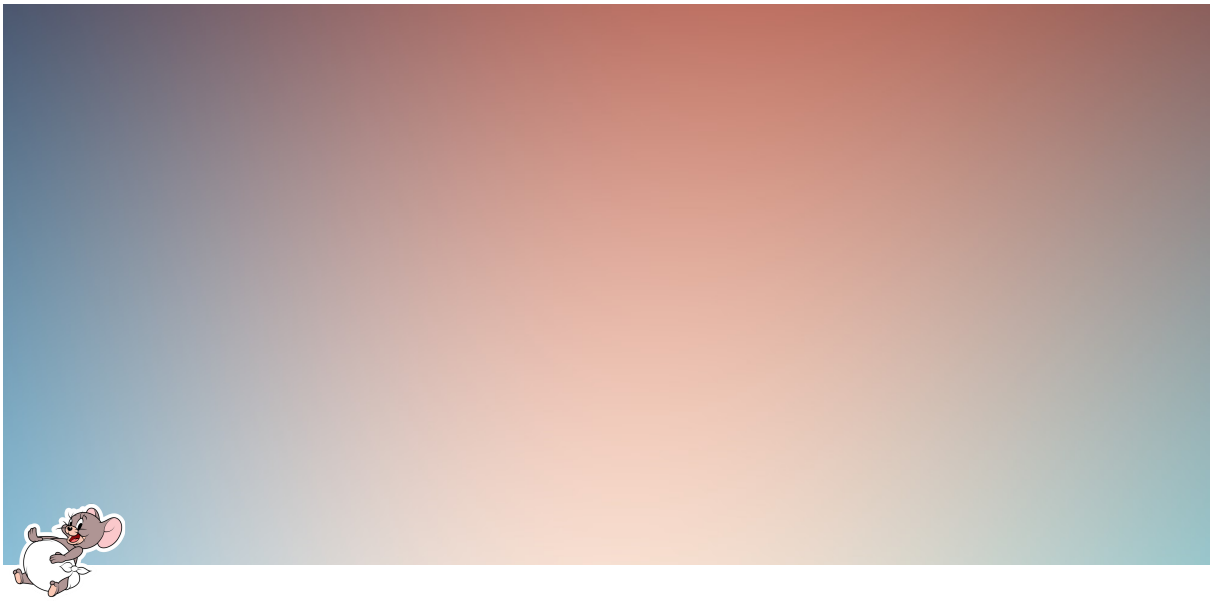
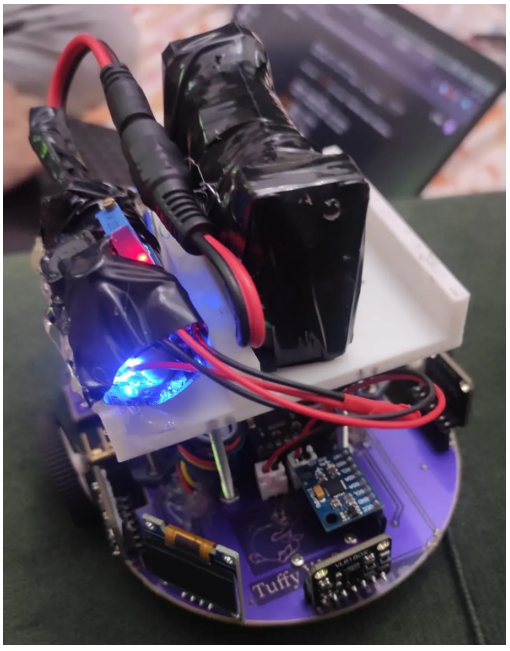


Micromouse 2024 - Tuffy V1.0

> Technical Report

Introduction

This report provides a comprehensive overview of the Micromouse project,which aims to build an autonomous robot capable of solving a maze. The project involved both hardware and software development, combining various sensors, motors,and control systems with an efficient algorithm to navigate the maze.



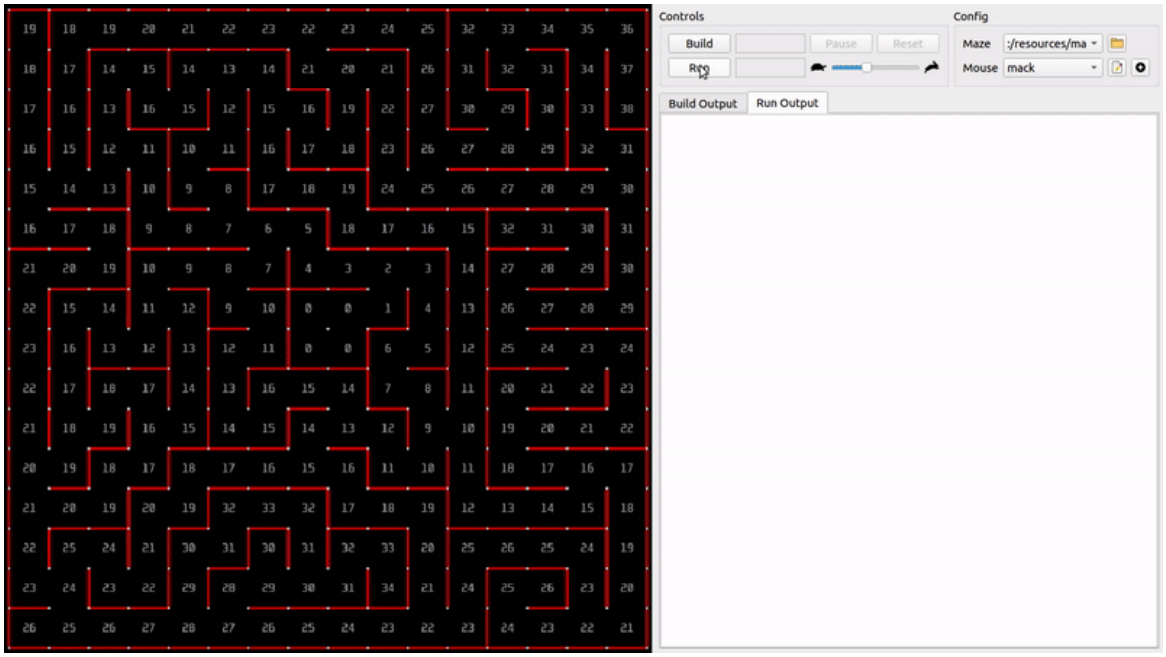
Say hi to Tuffy :)

▼ Software

The software implemented on the Micromouse robot is based on the Floodfill Algorithm.This algorithm is designed to efficiently explore and map the maze while searching for the optimal path to the goal. The Floodfill Algorithm is known for its ability to find the shortest route in a dynamic environment by incrementally filling the cells with values that represent the distance from the goal. The robot continuously updates its understanding of the maze and adjusts its path accordingly.

Also there is a simulator to test our algorithm:

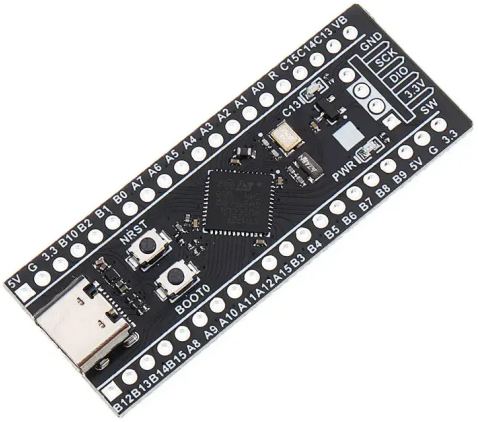
> https://github.com/mackorone/mms



▼ Hardware

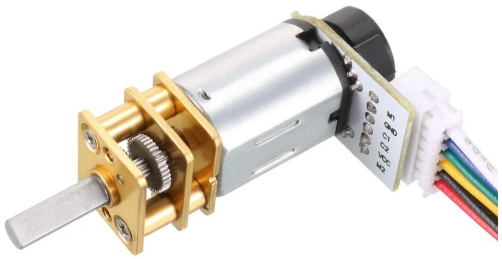
Microcontroller Unit (MCU):

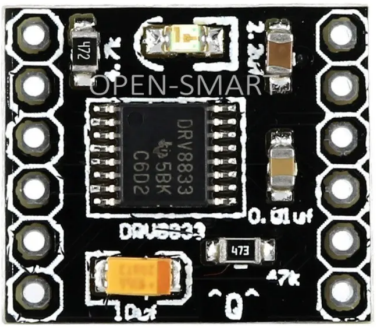
The Micromouse robot uses a Blackpill MCU (STM32F401RCT6), which is based on the ARM Cortex M4 architecture. This microcontroller provides sufficient computational power and peripherals to handle sensor inputs, motor control, and communication with the various components of the robot.



Motors and Motor Driver:

The robot is driven by N20 motors with Hall effect encoders, which provide precise feedback on the motor's speed and position. The motor driver used is the DRV8833, which allows for efficient control of the motors, enabling smooth and accurate movement of the robot.





Sensors:

Three VL6180X sensors are used for distance measurement and wall detection, allowing the robot to navigate the maze without colliding with obstacles. Additionally, the MPU6050 sensor provides gyro and accelerometer data, helping the robot maintain its orientation and balance during motion.

Indicators:

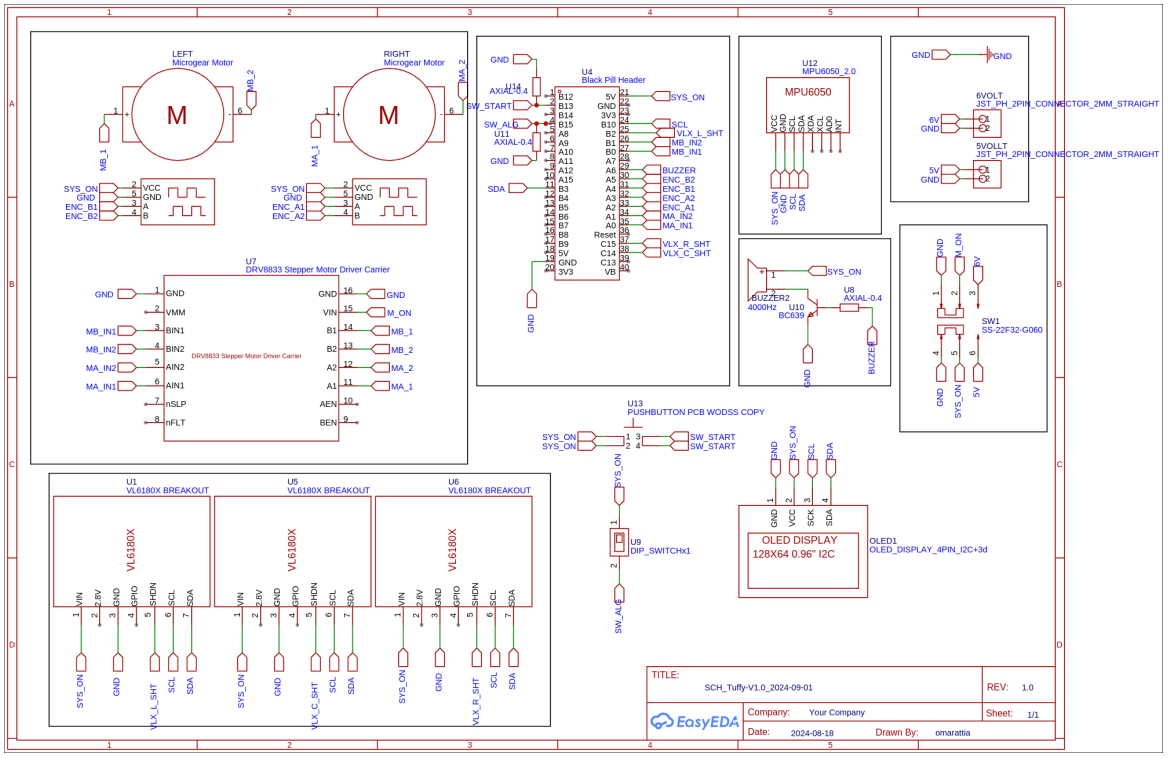
We used a buzzer and OLED display as indicators for debugging purposes.

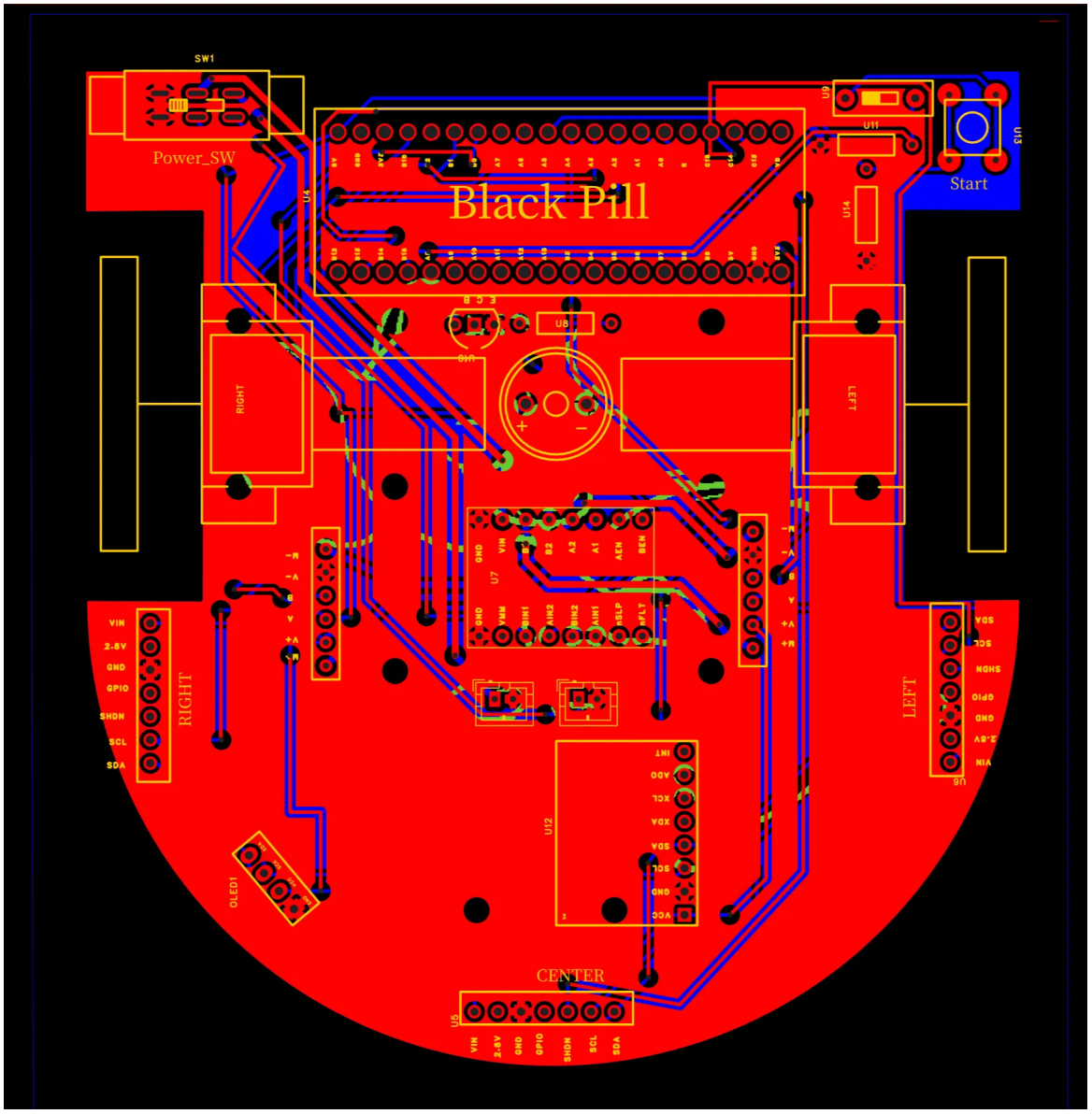
Power Supply:

The robot is powered by two 3.7V lithium batteries, providing the necessary voltage and current to operate the motors and sensors. Two buck converters are used to step down the voltage to 5V for the sensors and 6V for the motor driver.

PCB:

A custom-designed PCB (**Double Layer**) was used to integrate all the components of the robot. The PCB ensures proper connections between the sensors, motors, and the MCU.





▼ Issues and Challenges

During the development of the robot, several issues were encountered.

- **Holes dimensions in the PCB were incorrect and The footprint of the MCU was wrong, and there was a wire in the schematic that was not connected.** We fixed all these issues in a new PCB, but due to limited time, we could not print it.
- **The caster wheel was higher than the back wheels**, so the robot was not 100% stable and we did not fix this problem.
- **We could not control the right motor's speed**: The issue was with the PB0 pin of the MCU, which was connected to the IN1 pin of the motor driver. We resolved the issue by using another pin (PA7).

- **Motor bracket and wheel dimensions:** We used an open-source 3D design for the motor bracket and wheel. To address sizing issues, we modified the dimensions and reprinted these parts.
- **Two of our VL6180X sensors burned out.** We were unable to determine the cause of this issue.
- **Noisy readings from MPU6050:** We solved this problem by using a Kalman filter to tune mpu readings.
- **Centering the car inside the cell while moving:** We attempted to use a PID controller to solve this problem, but due to time constraints, we couldn't complete the implementation.
- **Measuring the distance moved by the robot:** When we used encoders directly to measure distance, the readings were noisy and inaccurate. To address this, we implemented a PID controller to estimate the actual distance moved by the robot. Although the resulting measurements still had some error, they were within an acceptable range.
- **MCU Power Problem:** The MCU would occasionally power off unexpectedly. We were unable to determine the exact cause, but suspect it may have been due to the MCU's pin headers not being properly fixed to the PCB.

▼ Future Improvements

- Implement a start button to initiate the robot's movement
- Include the Buck converters inside PCB.
- Add a hardware switch to toggle between algorithms for the first and second runs, or implement a system to save the maze layout after navigation and autonomously return to the start point for the second run
- You may choose another Algorithm to insure that you get the shortest path.
- Use MPU9250 instead of MPU6050 for its higher efficiency and reduced noise
- Design an improved mounting system for the battery unit and buck converters to enhance the robot's stability during movement

For more details , Check Our Github Repo:

https://github.com/MahmoudSamy511/Micromouse-2024