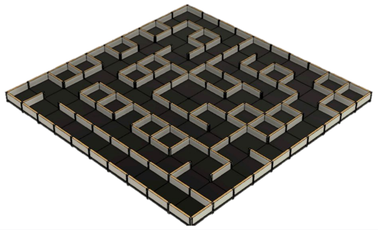
Micromouse

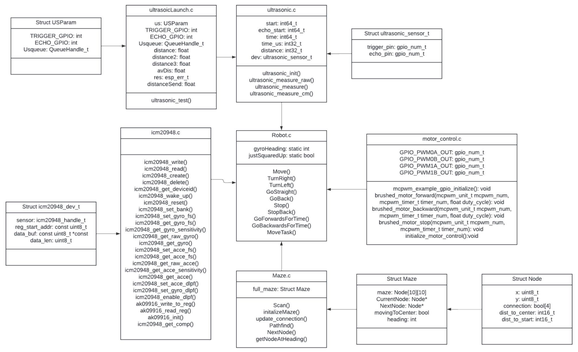
Brian Beerbower, Abigail DeCamp, Matthew Graber, Clay Wilson
Electrical & Computer Engineering
Advisor: Sean Carroll

Project Description

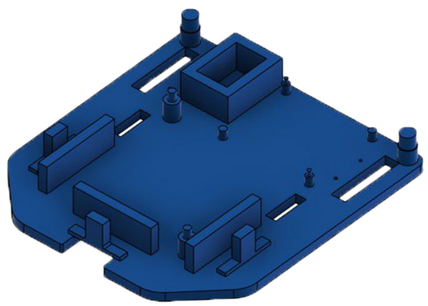
The National Robotic Challenge (NRC) hosts an annual competition in Marion, OH. The division we competed in is known as the micromouse competition. For this competition, you have to create a self-contained robot that can self navigate to the center of a maze in the shortest possible time. The rulebook states we are given 10 minutes to explore and navigate a maze set out in a 10x10 grid pattern. The quickest run time between the starting block of the maze and the center of the maze is recorded for scoring.



Class Diagram

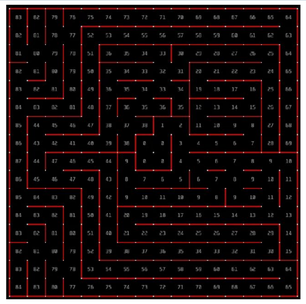


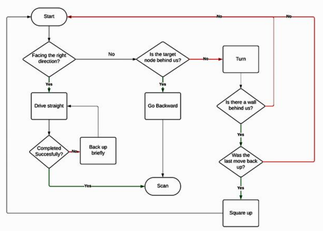
Chassis



Algorithm

Pathfinding Algorithm:
The robot uses a breadth first search algorithm to determine the distance each square is from the center of the maze. It then chooses the square that is closest to the center and easiest to move to.



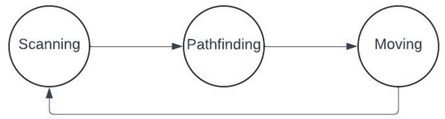


Control Algorithm:
After every move our robot makes, it analyzes the data it received to ensure that it moved correctly, and will attempt to correct itself if it didn't move correctly

Movement Algorithm:
The robot uses distance sensors to judge how far it has moved. It measures how far it is from walls on the sides to move straight, look for gaps in the wall to see when it should stop, and uses a gyroscopic sensor to turn accurately.

Error Correction:
If the robot ever takes too long to do something, the robot will realize that it's caught on something and adjust itself accordingly. Additionally, if it ever sees and impossible maze state, it will try to correct it, but if it is unable to it will simply clear its memory of the maze and begin learning again.

State Machine



1. The robot starts off by scanning its environment using its ultrasonic sensors and records that data.
2. After it has scanned the environment, the robot runs through its algorithms to determine the next course of action.
3. Once it knows where it wants to move to, the robots attempts to move to the desired grid space.
   a. This is accomplished using its ultrasonic sensors and its inertial measurement unit.

Hardware

Microcontroller:

We used an ESP32-S3 microcontroller in our design. The ESP32-S3 has 42 programmable general purpose input/output (GPIO) pins. We utilized its Vanilla FreeRTOS. The code for the microcontroller is in C language.

Sensors:

Our design utilizes two different types of sensors:ultrasonic sensors and an inertial measurement unit (IMU).

Our robot uses ultrasonic sensors to detect the environment around it. Ultrasonic sensors emit a high-frequency sound wave and then detects the waves as they get reflected back off an object.

The IMU is a 9 degrees of freedom (9-DoF) device, meaning that it is able to measure its orientation and motion in three-dimensions with 3 different sensors. The IMU incorporates an accelerometer, gyro, and magnetometer. Our robot utilizes the accelerometer and the gyroscope sensors. We use the accelerometer to detect collisions and the gyroscope to help orientation during the robot turning. The IMU communicates through I2C with the microcontroller.

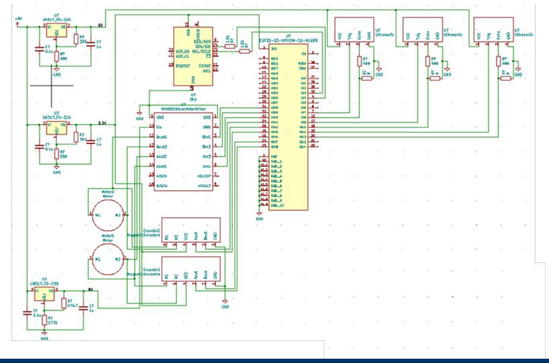
Motor and Motor Driver:

The motors that we use for our design are brushed DC Micro Metal Gearmotors. Geared motors are used to help get more torque out of a motor. The motors have a gear ratio of 29.86:1. These motors are controlled by pulse-width modulation (PWM) signals that are sent from the microcontroller through the motor driver.

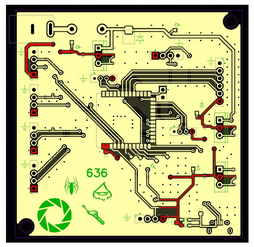
The motor driver is based around a dual H-bridge design. This allows us to individually control the motors' speed and direction of rotation.

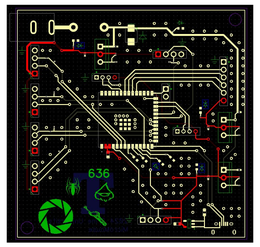
Printed Circuit Board (PCB):
The PCB was designed to decrease the number of wires needed and to gain access to all pins on the microcontroller.

Circuit Diagram

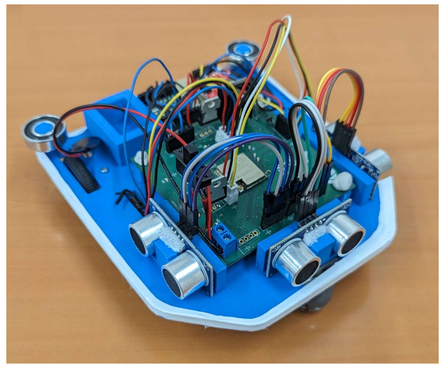


PCB





Final Product



Competition Results

Our team attended the 2024 NRC Micromouse competition on April 19. Unfortunately, our robot was not able to make it to the center of the maze within the modified time of 6 minutes. The robot had trouble moving grid space to grid space particularly in the long hallway in the maze. This caused the robot to be off in its calculation of where it was in the maze. We believe if allotted the full 10 minutes, our robot would have been able to reach the center.