NRC Micromouse Robot

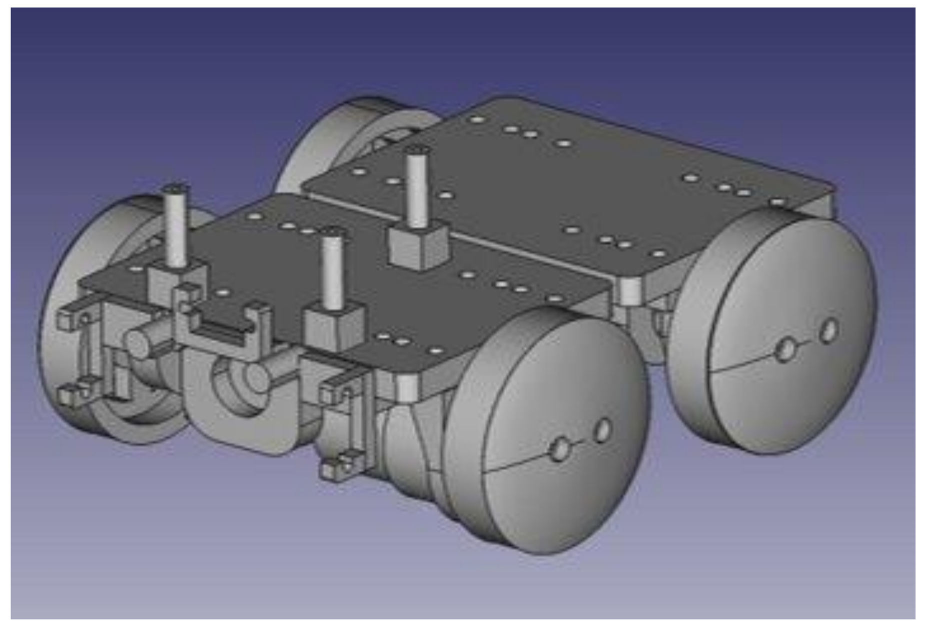
Kienan Miller, Loay Arrashoud, Marcus Miller
Electrical and Computer Engineering
Advisor: Sean Carroll, PhD

Abstract

The National Robotics Challenge hosts a group of robotics events yearly in Marion Ohio. What began as a manufacturing robotics competition grew into one that now includes combat, path finding, and other robotics events. Our team of two electrical engineers and one computer engineer have designed and assembled a maze solving robot to compete in the Micromouse event. A Micromouse is a small maze solving robot that navigates a maze made of 10x10 inch cells and then completes time trials in a race from the start to the end of the maze. Our robot uses several distance sensors to detect its surroundings and an encoder attached to a wheel to measure its travelled distance. From the data collected, the robot makes decisions on whether to turn, reverse, or continue forward.

Chassis

- Four-wheels
- Skid steer turning
  - One wheel per side
  - Two non-driven wheels
- 3-D printed components
- Disadvantages of the design
  - Friction dependent



Key Components

3 VL53L0x TOF Distance Sensors:
- Compact
- Measures up to 2m

FRDM KL25Z Microcontroller:
- ARM® Cortex™-M0+ Core
- 48MHz
- 16KB RAM

AMT102 Encoder:
- Optical disk measurements
- Adjustable pulse per revolution count

Driver Software

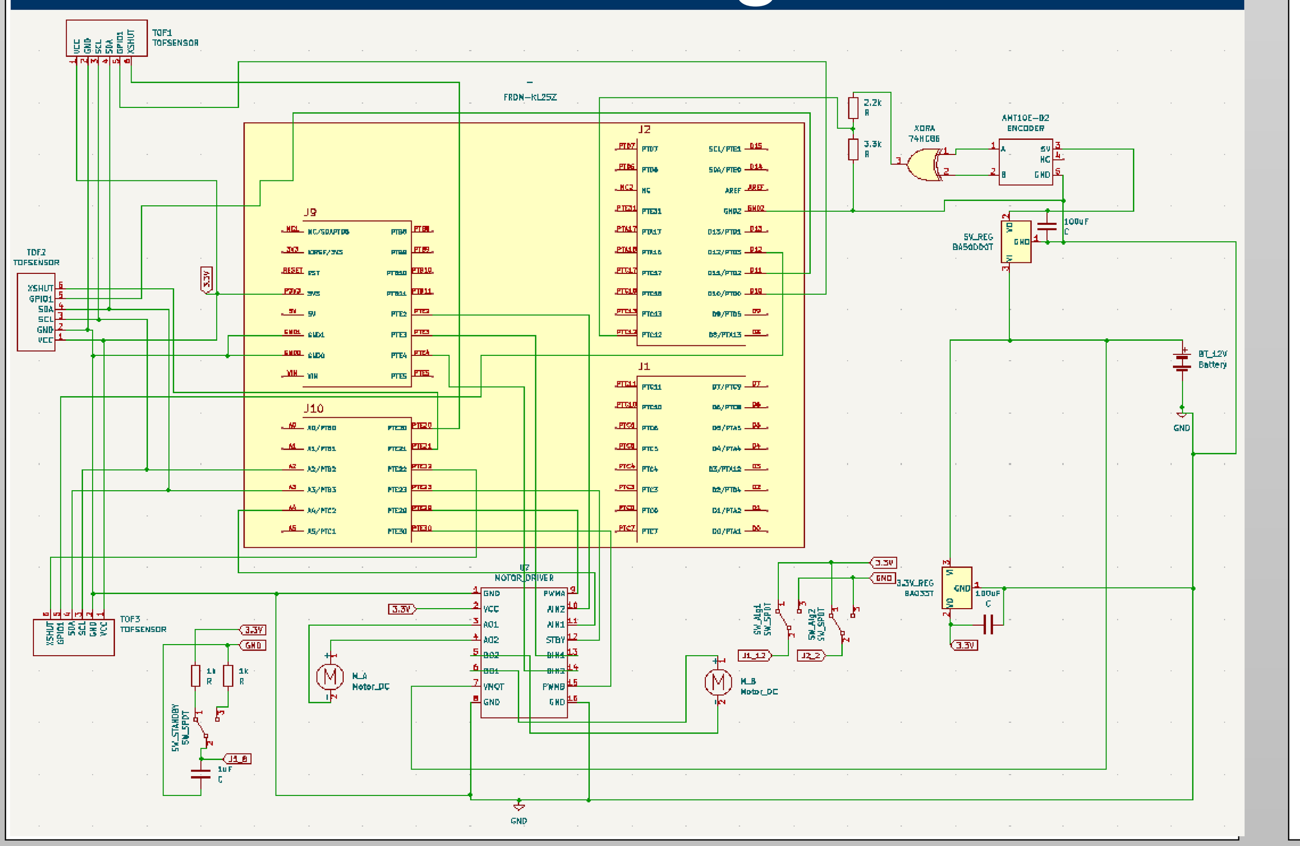
To communicate with the sensors on the robot, several drivers were required. Both the distance sensors and encoder required their own distinct drivers.

I2C is a two-wire communication bus that allows for data to be sent and received on the same set of lines. The I2C communication driver uses a queue to hold data packets to be sent. After a packet is transmitted, the program iterates to the next in the queue and the transmit/receive process is started again.

The distance sensor communication driver interfaces with the I2C driver to initialize and get distance measurements from the distance sensors. In the initialization portion, the address of each sensor is changed to a unique value, each sensor is calibrated to its temperature and input voltage, and the method of indicating complete measurements is set. In the measurement retrieval portion of the driver, a packet is sent to a distance sensor indicating that a new measurement should be started. The driver then waits until the sensor indicates that the measurement is complete. It then retrieves the data from the sensor.

The encoder outputs a square wave signal that produces a pulse after rotating a certain distance. The encoder driver takes this wave and runs it through a counter that is iterated on the rising edge of each pulse. After the pulse count reaches a specified value, a global distance variable is iterated by one. The global distance variable can be multiplied by the distance per count to get the total linear distance travelled.

Circuit Diagram

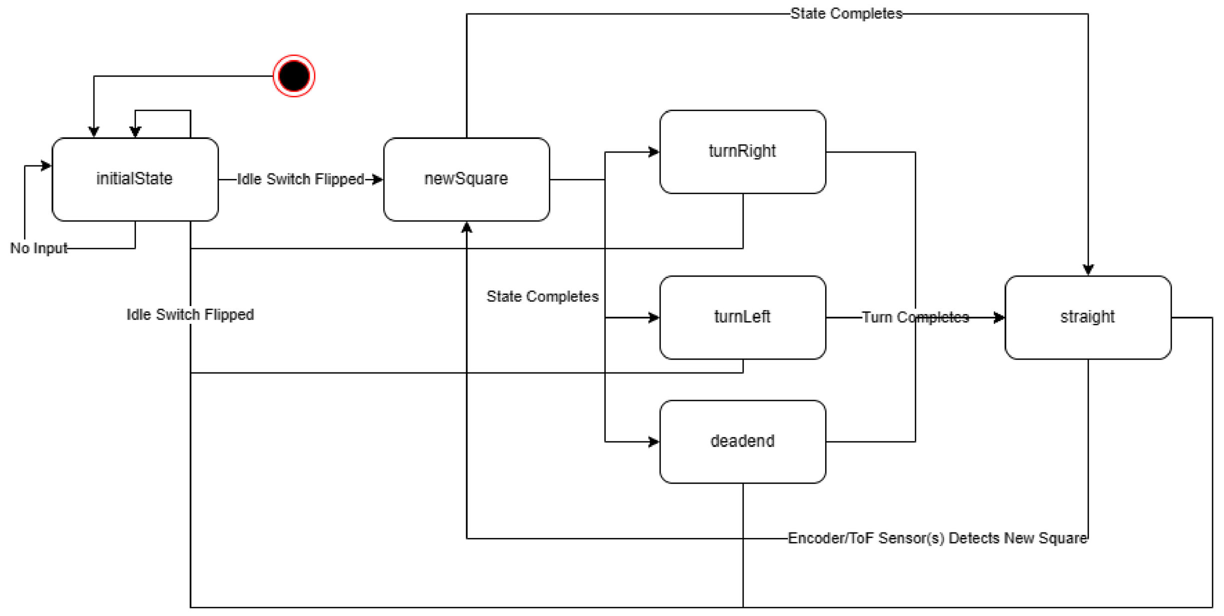


Algorithm & State Machine

Algorithm Overview:
To navigate the maze our robot utilizes data from the sensors to determine a list of possible moves in its current square. The robot's algorithm uses this information to determine the next state in our state machine. In cases of multiple turns, such as a crossroad section in the maze, the robot will prioritize right turns first, followed by straight, then left turns. Our robots turn biasing is done by a variable that is updated in our next state function to ensure the robot does not get stuck navigating the same section of the maze. In the case our robot get stuck, it uses the motor encoder to detect that it is no longer moving and will attempt to reverse to correct itself.

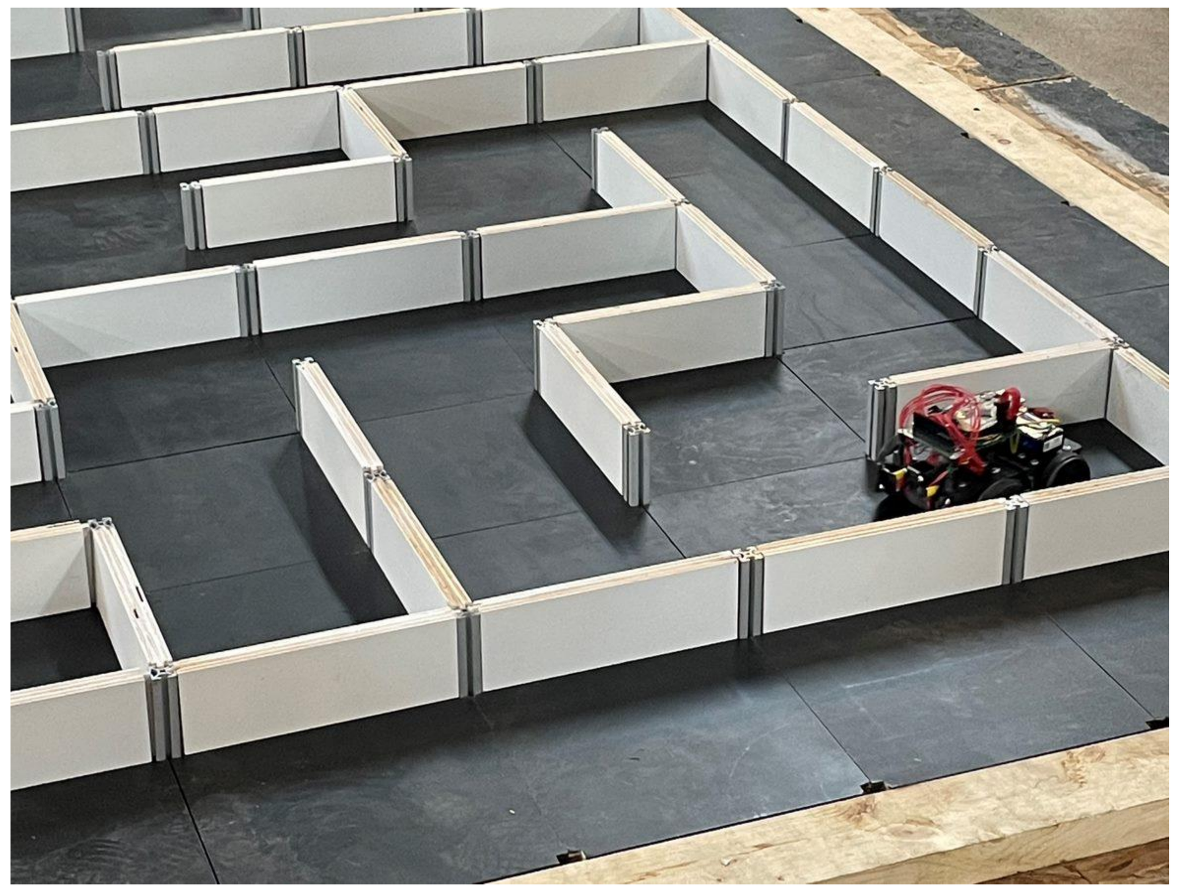
State Machine:
Our state machine consists of six different states that our robot uses to aid in maze navigation:

"InitialState" - Wait until told what to do by an interrupt connected to a switch.
"straight" - Wall follow mode, follows either right or left wall based on ToF sensor measurements.
"turnRight" - Sets motor speed and direction to turn right for set time, then sets current state to straight.
"turnLeft" - Sets motor speed and direction to turn left for set time, then sets current state to straight.
"deadend" - Sets motor speed and direction to turn left for set time until robot is turned around, then sets current state to straight.
"newSquare" - Collects ToF data and calculates next state with nextState() function.



Final Product

After completing ECE4003 - Senior Design, we delivered an autonomous robot capable of navigating a maze. This project was a test of our skills as engineers and challenged us to utilize our knowledge of the engineering principles and skills we have developed at Trine University.



Competition Results

On April 14th, 2023, we travelled to the NRC Micromouse competition and competed against other area universities. Unfortunately, due to the friction dependency of our robot and the distinct difference in the friction of the maze at the competition and our test environment, overturning caused our robot to fail to solve the maze. After the competition, retuning the robot to the friction of the maze greatly increased its performance. While these runs did not count for placement, they proved that our robot could navigate the competition maze and therefore fit our scope

Acknowledgements

Our team would like to acknowledge the following for support and advice during the completion of our senior design project:
- The Fawick extended family
- Dr. Sean Carroll
- EduEverything Inc. for hosting the National Robotics Challenge