Client: Dr. Phillip Jones
Advisor(s): Dr. Phillip Jones

MicroMouse

sddec20-28

Richard Anderson, Jorge Gomez, Joshua Wooi, Tyler Fuchs, Aaron Walter, Austin Chesmore

Problem Statement:

The goal is to demonstrate how the interconnection of hardware and software can function together to accomplish a task. In doing so raising interest in related majors.

Intended user

The intended user of the micromouse are potential students visiting Iowa State.

Design approach

Our design approach was to select components and hardware that would allow us to build software that would work native. To solve this we selected an Adafruit feather that would work on C++ firmware. And a User interface that would connect to the web server built on the micro controller.

Testing

Hardware

- Tested the individual hardware components , before testing the components together

Software

- googletest Unit Testing
- Console output for C++ and log files for JavaScript

Technical details:

GUI: HTML/JS via JSON/WebSockets
Framework: C++ (Async)
Floodfill: C++
A*: C++
PCB: Eagle

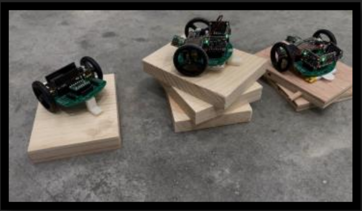


Figure 5: Micromouse prototypes

Block Diagram

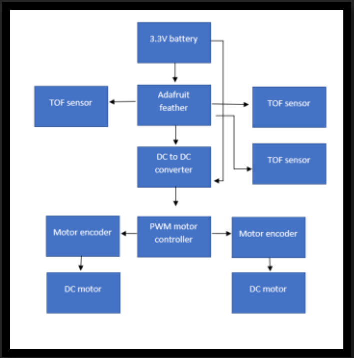


Figure 2: Block Diagram of physical components

Conceptual Sketch

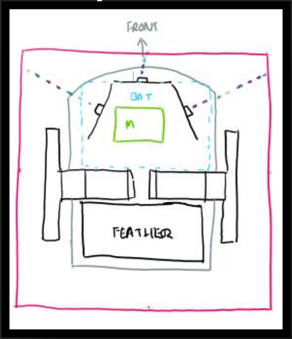
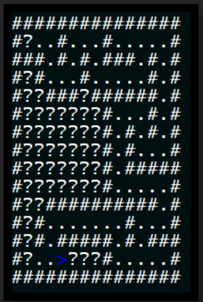


Figure 8: Sketch of the micromouse

Results

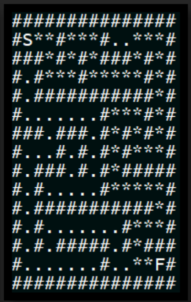
For the micromouse project the team dealt with the challenges of working remote during the covid-19 pandemic to deliver a full hardware functional micromouse. The prototype is fully gamepad controllable and can utilize Time of Flight sensors to see obstacles.

Additionally, the team has created functional software for A* and flood fill algorithms that can be implemented on the micromouse in the future



```
###############
#?..#...#.....#
###.#.#.###.#.#
#?#...#.....#.#
#??###?######.#
#???????#...#.#
#???????#.#.#.#
#???????#.#...#
#???????#.#####
#???????#.....#
#??##########.#
#?#.......#...#
#?#.#####.#.###
#?..>???#.....#
###############
```

Figure 6: Flood Fill



```
###############
#S**#***#..***#
###*#*#*###*#*#
#.#***#*****#*#
#.###########*#
#.......#***#*#
###.###.#*#*#*#
#...#.#.#*#***#
#.###.#.#*#####
#.#.....#*****#
#.###########*#
#.#.......#***#
#.#.#####.#*###
#.......#..**F#
###############
```

Figure 7: A*

Figure 1: micromouse team in the order of the names written above

Functional Requirements

- Be able to navigate a maze as fast as possible both autonomously and manually.
- Show the location of the mouse in the maze
- Be able to scan for nearby walls forwards and on its sides.
- Move forward, backwards, and turn

Non-Functional Requirements

- The path taken towards the maze goal should be optimized in order to be solved as fast as possible
- The mouse will use A* floodfill algorithms to solve the maze.

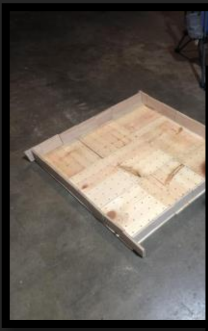


Figure 4: A wooden maze

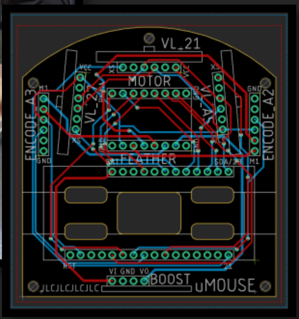


Figure 3: PCB prototype

Flood Fill & A* Traversals

The mouse runs through the Maze using modified flood fill (left), and then uses A* to find the shortest path using the filled map.