Milestone 2 Design Report

EEE3097/8/9S
Group 63

Prepared by:

Chris Scheepers - SCHCHR077
Nang'alelwa Sitwala - STWNAN001

Prepared for:
EEE3097/8/9S
Department of Electrical Engineering
University of Cape Town

September 24, 2024

Declaration

1. We know that plagiarism is wrong. Plagiarism is to use another's work and pretend that it is one's own.
2. We have used the IEEE convention for citation and referencing. Each contribution to, and quotation in, this report from the work(s) of other people has been attributed, and has been cited and referenced.
3. This report is our own work.
4. We have not allowed, and will not allow, anyone to copy our work with the intention of passing it off as their own work or part thereof.

[illegible]

Sitwala

September 24, 2024

Chris Scheepers, Nang'alelwa Sitwala

Date

Contents

Chapter 1

Introduction

1.1 Problem Description

This report refers to the second milestone in the overarching project from the UCT engineering design course, EEE3097/8/9S, that requires designing a solution for a robotic micromouse to solve a maze autonomously. Milestone 2 refers to the project's initial calibration and localisation aspect using IR sensors and the basic use of the motors. This course focuses heavily on the software design of this problem. Simulink Stateflow was used to generate the control logic for sensor data integration and motor movement, which helped streamline code generation and provided efficient debugging capabilities.

1.2 Scope and Limitations

As this is milestone two, the scope of this report is limited to and focuses on the following tasks:

- Calibrating the mouse to ensure accurate sensor readings.
- Performing initialization routines to set the micro-mouse into a ready state.
- Identifying and holding a line, enabling the mouse to follow black lines in the maze.
- Achieving surrounding awareness by detecting walls and lines through the sensor subsystem.

Turning, mapping, and decision-making are not in the scope of this report.

Design Limitations

- Time constraints due to deadlines
- Processing speed and component choices
- Battery life
- Bottle-necking due to hardware used to program in MatLab

Testing Limitations

- Error handling and edge Cases
- Calibration and tuning

Chapter 2

Requirements Analysis

2.1 Requirements

The requirements for the milestone 2 micromouse maze-solving project are described in Table 2.1.

Table 2.1: Functional requirements for milestone 2.

Requirement ID	Description
R01	Calibration of sensors
R02	Line identification
R03	Surrounding Awareness
R04	Calibration of Motors

2.2 Specifications

The specifications, refined from the requirements in Table 2.1, for the milestone 2 micromouse maze-solving project are described in Table 2.2.

Table 2.2: Specifications of the milestone 2 project derived from the requirements in Table 2.1.

Specification ID	Type	Details
SP01	Sensors	Infrared (IR) Sensors
SP02	Detection Coverage	Front, Down
SP03	Real-time processing	Yes
SP04	Customisation	Adjustable calibration for different maze environments
SP05	Motors	Left and Right
SP06	Battery	3.7V Li-ion

Chapter 3

Solution Design

3.1 Design Decisions

3.1.1 Sensor Subsystem Integration

The micromouse's ability to navigate and perform line detection depends on its sensor subsystem. Out of the 8 infrared (IR) sensors available, only 4 were chosen for use: 2 forward-facing sensors for wall detection and 2 downward-facing sensors for line following. The left and right sensors, as well as the motor sensors, were not used because they were redundant in the current design and caused interference between sensors when used.

The system's selective use of sensors simplifies its design while retaining adequate functionality for the tasks required in this milestone. The forward-facing sensors detect proximity to walls, allowing the micromouse to avoid collisions, while the downward sensors help the mouse stay on the black line. Unused sensors can be re-integrated in future milestones if needed to improve performance or enhance features.

3.1.2 Methodology for Sensor Calibration

A crucial part of reaching this milestone was figuring out how to calibrate the sensors for accurate performance. To accommodate the changing light conditions throughout the day, on-site calibration was performed in the actual maze environment. The calibration process involved taking multiple readings to establish threshold values that differentiate between a black line and other surfaces. During this process, the forward-facing sensors were calibrated to detect walls at a certain distance ahead and we calibrated the downward sensors to more reliably recognize the black line.

The system was able to achieve more consistent readings by averaging sensor data over multiple trials, which helped filter out noise. This calibration methodology ensured that the micromouse could adapt to its environment without losing accuracy in its wall detection and line-following functions.

3.1.3 Code Development with Simulink Stateflow

Simulink Stateflow was used to design the logic that controls sensor input, motor operation, and decision-making processes in the micromouse, as well as to generate the code. This environment allowed for the systematic design of the micromouse's states, which govern different behaviors such as moving forward, stopping, detecting walls, or following a line. Stateflow's visual approach to state machine design made it easier to debug and refine the system's logic.

By structuring the code into distinct states, each with clearly defined entry and exit conditions, the system can respond dynamically to sensor inputs in real-time. For example, when the forward-facing sensors detect an obstacle, the micro-mouse transitions into a stop state, signaling the motors to halt and triggering surrounding awareness actions. Debugging within Simulink also allowed for the identification and correction of issues in real time, improving overall system reliability.

3.1.4 Thresholds for Sensor Readings

During the design phase, establishing effective thresholds for the sensor readings was crucial for ensuring correct behavior. The forward-facing sensors needed a threshold that would allow the micro-mouse to recognize walls at just the right distance to make timely adjustments. The downward sensors, on the other hand, required a threshold that could accurately differentiate between the black line and the surrounding floor of the maze.

Through trial and error during calibration, these thresholds were fine-tuned. For example, the system was programmed to recognize a black line when the downward sensor readings dropped below a certain value, indicating the presence of a dark surface. Similarly, forward sensor readings that exceeded a specific value indicated an approaching wall, triggering the mouse to stop or change direction.

3.1.5 Redundancy Management and Design Simplification

In the initial design, all 8 sensors were considered for use. However, it became apparent that using both the left and right sensors, in addition to the forward and downward ones, added complexity without providing significant additional functionality for the current milestone. Therefore, the design was simplified to use just the 4 most critical sensors.

This decision not only reduced the complexity of the system but also minimized the processing load on the microcontroller. By focusing on the most relevant sensor data, the micro-mouse can operate more efficiently and respond faster to critical events like detecting walls or navigating lines.

3.2 Failure Management

Table 3.1: Milestone 2 Failure Management

Name	**Description**
Redundancy	Multiple IR sensors are used to detect front or down directions.
Loop detection	Logic has been implemented to detect repetitive patterns and break if needed.
Modulation	Code has been subdivided into manageable and independent subsystems for easy debugging.

Chapter 4

Acceptance Testing

4.1 Tests

Table 4.1: Milestone 2 acceptance tests

Test ID	Description	Testing Procedure	Pass/Fail Criteria
AT01	Calibration of sensors	• Place wall in front of micromouse • Rotate micromouse in a closed pixel of the maze	• Front wall detection and black line detection.
AT02	Line identification	• Place calibrated micromouse on black line • Initiate motors	• Micromouse does not stray from the black line it is following.
AT03	Surrounding awareness	• Place calibrated micromouse in the maze • Initiate motors	• Micromouse does not hit the side of the maze and stops in front of a wall.
AT04	Calibration of motors	• Initiate calibration	• Micromouse moves in a circle and then in a straight line once attempting AT02.

4.2 Critical Analysis of Testing

Table 4.2: Milestone 2 acceptance test results

Test ID	Description	Result
AT01	Calibration of sensors	Pass
AT02	Line identification	Pass
AT03	Surrounding awareness	Pass
AT04	Calibration of motors	Pass

4.2.1 AT01 - Calibration of sensors

The calibration of our sensors worked well and, during our testing, we found that environmental conditions did not affect the calibration as data at that moment of calibration would be used. The micromouse passed the other tests due to the calibration of sensors which shows the working nature of our solution. As noted in the next section, the line-following has room for improvement. We suspect the lowered sampling time could cause the mouse to stray from the line before an output could be registered so increasing sampling time could mitigate this issue.

4.2.2 AT02 - Line identification

The line-following subsystem worked successfully about 80-90% of the time. The mouse could stay on the black line and adjust its movement based on sensor input. However, there were occasional errors where the mouse struggled to maintain its path or got caught on the black tape as it would drag at the back, particularly at crossroads, where detection became slightly less reliable. Adjusting the speed of the motors or perhaps the sampling time of our solution could mitigate this issue. Additionally, adding a tiny sled or wheel at the back would fix the problem of getting caught on the tape.

4.2.3 AT03 - Surrounding awareness

After calibration, the wall detection subsystem performed almost perfectly. The forward-facing Infrared sensors reliably detected walls, helping the mouse to avoid obstacles and adjust its path. This showcased the effectiveness of the calibration process. Although not a requirement, the micro-mouse was able to detect crossroads within the maze, although with slightly less accuracy than the wall and line detection. These crossroads require the mouse to make decisions on whether to turn or continue straight and while the detection worked, it would sometimes miss the crossroad or get stuck in a loop of detecting the crossroad and would stay in the 'stopping state'. This issue, although for milestone 3, could be mitigated by taking the average of the down-sensing data and adding a timer that forces a certain amount of time before the crossroad function can be called again.

4.2.4 AT04 - Calibration of motors

This worked as expected and was the easiest part to implement in our solution. Changing our inputs to percentages in MatLab made tinkering and debugging issues simplier.

Chapter 5

Conclusion

The most difficult part of this milestone was primarily centered around calibrating the sensors to account for environmental changes, particularly fluctuations in light intensity throughout the day. These variations affected how the IR sensors perceived the maze, leading to inconsistencies in wall and line detection. The choice to calibrate within the maze just before a run allowed the sensors to adapt to real-world conditions, ensuring more reliable performance and helping mitigate this issue. Simulink Stateflow played a crucial role in debugging by providing clear visual feedback on sensor states and motor actions, enabling quick identification and correction of issues.

The results from Milestone 2 demonstrate a strong foundation for future progress. The success in wall detection and line following shows that the core systems are functioning as expected, though further refinement is necessary for crossroad detection and line-holding precision. The experience gained during this milestone, especially in dealing with sensor calibration and debugging, provides a solid basis for the upcoming milestones, where full maze navigation will be implemented. The current state of the micro-mouse is promising, with room for further optimization and improvement.

5.1 Recommendations

Speed was not a requirement for this milestone but this is one area of our project that could improve - by increasing motor speed and decreasing sampling time, the micromouse could increase drastically in efficiency. Although not a massive issue, the back of the micromouse would occasionally get caught on the black tape on the floor, causing it to get stuck. we would recommend adding a small sled or wheel that raises the back to mitigate this issue. Finally, the crossroad detection of the micromouse needs slight optimisation as it would occasionally get stuck in a loop of detecting the crossroad and so would get stuck in the ‘stopping state’. This is again a simple issue to fix and we recommend adding a timer that does not allow the crossroad function to be called after a certain amount of milliseconds so the micromouse can leave the current crossroad.