

Bethan Moyle Integrating Project

by Bethan Moyle

Submission date: 05-Apr-2025 09:59AM (UTC+0100)

Submission ID: 254884846

File name: 57913_Bethan_Moyle_Bethan_Moyle_Integrating_Project_660190_121015311.docx (2.16M)

Word count: 1963

Character count: 10975



Application of Academic Tools

L0 – Integrating Project - COMPUTING

Module Code: ACHA4008

BSC4F Computing (Computer Networks and Cyber Security) with Foundation Year

Bethan Moyle (1707952)

1707952@student.uwtsd.ac.uk

Table of Contents

1	Introduction	2
2	Identifying Obstacles.....	3
2.1	Line.....	3
2.2	Obstacle.....	3
2.3	Maze	3
3	Hardware and Programming	3
3.1	Hardware	3
3.2	Programming	4
4	Project Plan.....	4
5	Design.....	5
5.1	Hardware	5
5.2	Program	6
5.2.1	When Program Starts	6
5.2.2	Define Follow Line	7
5.2.3	Define Object Detector	7
5.2.4	Define Move Around	8
5.2.5	Define Maze.....	9
5.2.6	Define Left	10
5.2.7	Define Right	10
6	Evaluation	11
7	References.....	12
8	Appendices	14

1 Introduction

The goal is to design and build a robot using the Lego Spike Prime set to navigate a maze. This requires identifying potential obstacles and developing strategies to

overcome them, including selecting appropriate components and programming the robot effectively. Robotics fosters creativity, critical thinking, and problem-solving skills [1], playing a significant role in improving efficiency, safety, and sustainability[2]. It combines knowledge from physics, mathematics, computer science, and engineering[3].

2 Identifying Obstacles

2.1 Line

The robot's first challenge is following a line to the maze entrance. A colour sensor [4] can be programmed to detect colour to adjust its movements [5] or respond to light variations to guide the robot [6].

2.2 Obstacle

The robot must detect obstacles and resume its path after bypassing them. Two sensors can be used: a Force Sensor [4] that activates upon being pressed [7], and a distance sensor [4] that detects walls or objects and helps the robot navigate [8].

2.3 Maze

To navigate the maze, the robot must be able to detect a wall from a distance, and adjust its direction, accordingly, continuing its path until it reaches the end of the maze. A suitable sensor for this task is the distance sensor [4]. The distance sensor operates using ultrasonic waves [8], featuring two components: one that emits a ultrasonic sound, while the other receive them [8]. By measuring the time taken for the sound to return, allowing the sensor calculate the distance from a wall ahead [8].

The robot's navigation strategy could follow the "Right then Left" rule [9]. In the robot's programming, this rule dictates that the robot will proceed forward until it detects an obstacle [9]. Upon detecting an obstacle[9], the robot records the distance to the nearest obstacle on the right and on the left [9]. If the right-side is farther away the robot will turn right [9]; alternatively, if the left-side obstacle is farther away the robot will turn left[9].

3 Hardware and Programming

3.1 Hardware

The robot base must be stable and support multiple sensors. The central hub [4] connects all components , communicates with the computer [10], and runs the program [10]. Motors will be used to power the wheels, while connectors [4] and

clips [4] hold the components together. At least three wheels should ensure stability.

3.2 Programming



The program controls motor speeds and positions. Conditional statements [11] allow the robot to respond to sensor inputs, and loops [11] help it continuously monitor its environment. For example, the distance sensor repeatedly checks distances, prompting adjustments when necessary.

4 Project Plan

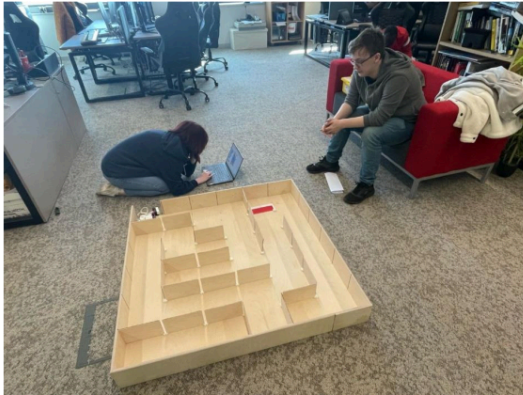


Figure 1: Test run, the Maze program

The obstacles identified moulded the robot's hardware and programming decisions.

For line detection, the team chose colour detection over light intensity due to inaccuracies with surface reflection and material. This method offers flexibility, allowing for easy adjustments based on the colour detected [7]. Testing this alone took approximately 3 hours.

For obstacle detection, the team selected a Distance Sensor [4] or  Force Sensor [4] to reduce size and weight. The sensor performs both obstacle detection and maze navigation. Testing this alone took approximately 5 hours.

For the maze, when a wall is detected, the distance sensor prompts the motors to turn toward the farthest obstacle. The robot makes 90-degree turns to avoid obstacles. This process alone took approximately 8 hours.

5 Design

5.1 Hardware



Figure 2: Advanced Driving Base

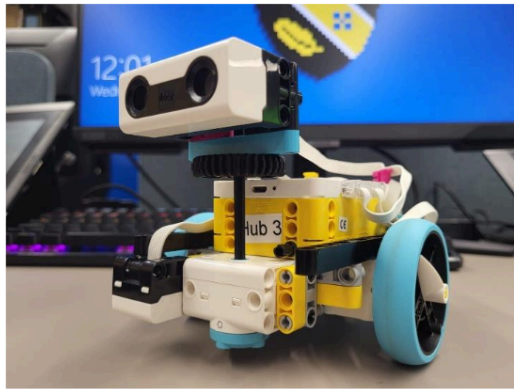


Figure 3: Final Robot Design using the Base 3 Model

The team initially attempted to use the Advanced Driving Base [12] (*Figure 2*) from the Lego Spike Prime set , but encountered issues with wheel size and missing parts. As a result, they switched to a simpler Base 3 model [13] (*Figure 3*) .

The sensors were placed at the front of the robot to detect direction and trigger the appropriate program for obstacle navigation. The colour sensor [4] was positioned in line with the wheel for optimal detection.

Wheels [4] were placed on either side of the hub[4], and a stabilizer [4] at the front helps support the sensor's weight, maintain balance, and ensure smooth rotation. Motors [4] are used to rotate the wheels [4], while another motor controls the movement of the distance sensor [4]. Various clips, connectors [4], and long strips [4] were used to attach the components and create the robot's structural frame.

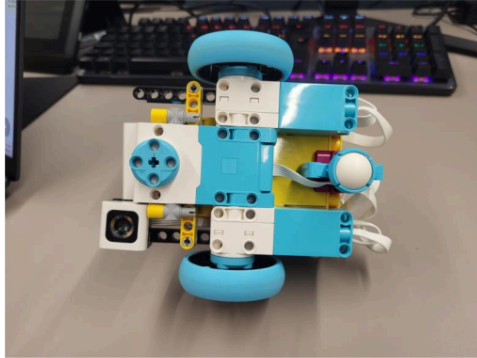


Figure 4: The underneath of the final Robot design

5.2 Program

5.2.1 When Program Starts

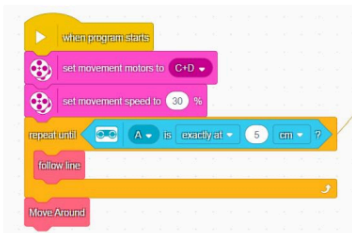


Figure 5: Start of Program Screenshot

This code initializes the motors for use. Upon starting the program, motors C and D, which control the wheels, are set to a speed of 30%. A loop is then activated with the condition that the distance sensor [4], connected to socket A, must detect a distance of exactly 5 cm from a wall. Once this condition is met, the program will activate the "Move Around" program. Until the distance sensor [4] detects this specific 5 cm distance, the robot will continue executing the "Follow Line" program.

5.2.2 Define Follow Line

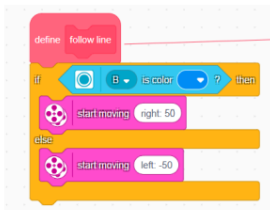


Figure 6: Follow Line Program Screenshot

The "follow line" definition is created to simplify the code. It sets specific parameters for the robot's movement: when the colour sensor detects blue, the robot will move right: 50; when it does not detect blue, the robot will move left:-50. This if statement causes the robot to continuously switch between moving left and right, allowing it to follow the line.

5.2.3 Define Object Detector

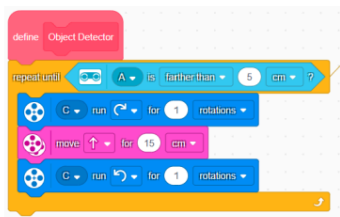


Figure 7: Object Detector Program Screenshot

The "Object Detector" program uses the distance sensor to check if the condition "farther than 5 cm" has been met in order to break the loop. Until this condition is satisfied, the robot will perform the following sequence: it will rotate motor C for one full rotation, move forward 15 cm, and then rotate motor C for another one rotation to return to its original facing position. This process allows the robot to recheck the distance and determine if the object it previously detected is still an issue.

5.2.4 Define Move Around

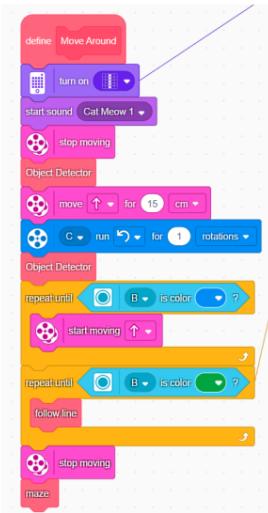


Figure 8: Move Around Program Screenshot

When this defined program is activated, the robot will produce a sound and display the number 1 on the Hub. The robot will then stop moving and begin running the "Object Detector" program. Once the condition in the "Object Detector" program is met and the loop is stopped, the robot will move forward 15 cm and then rotate motor C for one rotation. This allows the distance sensor [4] to recheck if there is still an obstacle. If an obstacle is detected, the "Object Detector" loop will run again until the condition is met.

Once the condition in the "Object Detector" program is satisfied, the robot will move forward until the colour sensor detects blue. It will then switch to the "Follow Line" program and continue until the colour sensor detects green. At that point, the robot will stop moving and begin executing the "Maze" program.

5.2.5 Define Maze

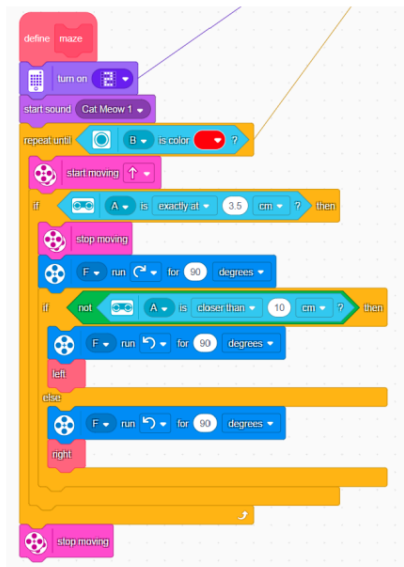


Figure 9: Maze Program Screenshot

When this defined program is activated, the robot will produce a sound and display the number 2 on the HWB. The robot will then begin moving forward until it detects the colour red, which will stop its movement. Within this loop, there are nested if statements. The first checks if the distance sensor detects a wall exactly 3.2 cm away; if true, the robot will stop and rotate motor F (with the attached distance sensor [4]) 90 degrees.

The next nested if statement checks whether the distance sensor [4] (socket A, motor F) is farther than 10 cm. If this condition is met, motor F will turn right, repositioning the sensor and allowing it to check the distance. The robot will then execute the defined "left" program if it is further than 10cm. If the distance sensor is closer than 10 cm, the robot will instead run the defined "right" program.

5.2.6 Define Left



Figure 10: Left Definition Program Screenshot

The "Left" program is used within the maze navigation routine. When activated, the program will wait for 1 second, then move motors C and D left: -100 for 0.5 rotations. After completing the movement, the robot will stop and wait for another second.

5.2.7 Define Right

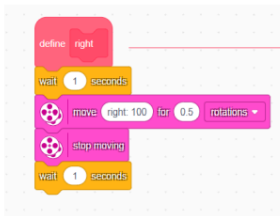


Figure 11: Right Definition Program Screenshot

The "Right" program is also used within the maze navigation routine. When activated, it will wait for 1 second, then move motors C and D right:100 for 0.5 rotations. After this movement, the robot will stop and wait for another second.

6 Evaluation

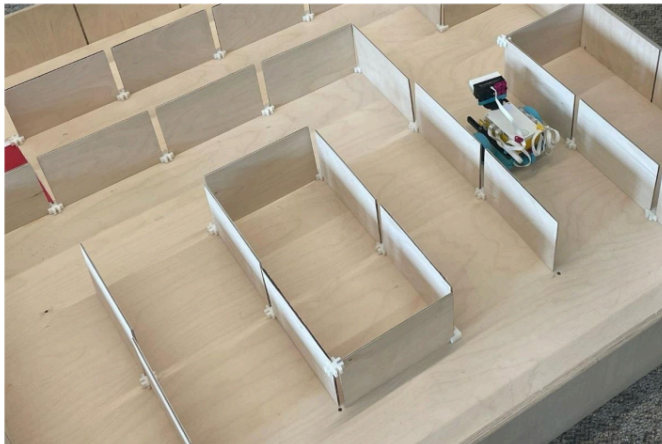


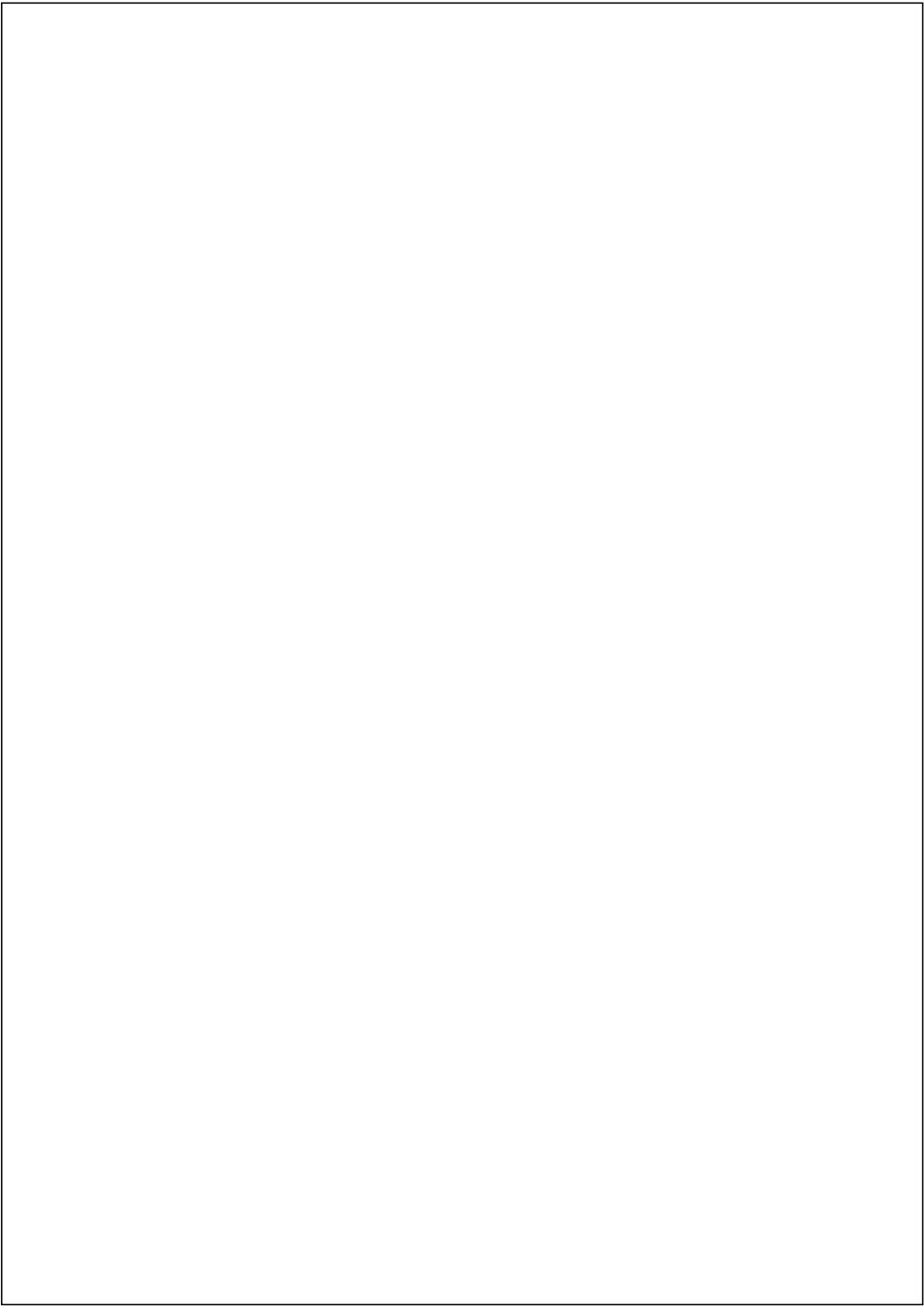
Figure 12: Robot Navigating the Final Maze

The robot successfully navigated the line and obstacle, encountering only one issue. During one of the turns, the distance sensor "eye" that ultrasonic waves [8] detected it was clear, while the other "eye", along with the rest of the robot, was obstructed. As a result, the robot became stuck and required manual intervention. This could have been avoided by either reducing the distance between each sensor check or increasing the coverage of each check.

At the start of the maze, the robot got stuck on the tape holding the ramp in place. This could have been prevented by using a stronger adhesive or a more structurally stable ramp. While the maze was navigated successfully, the robot struggled with oversteering on the 90-degree turns, meaning it was an inaccurate 90 degree. This caused the robot to move at a slight angle rather than directly forward, resulting in collisions with the wall. Reducing the turn angles and speed of the motors C and D could have minimized this issue.

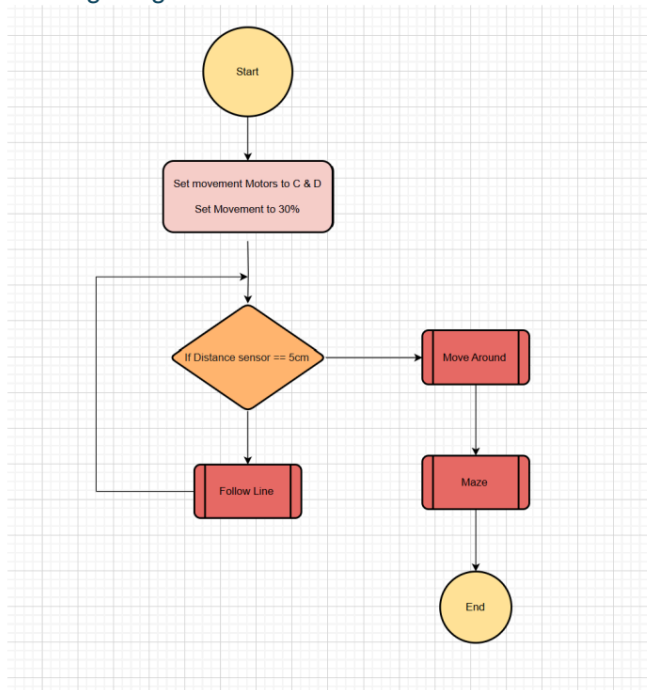
7 References

- [1] The Knowledge Hub, "Importance of Robotics In Education," The Knowledge Hub. [Online]. Available: <https://knowledge-hub.com/2022/05/20/importance-of-robotics-in-education/>
- [2] T. Djurickovic, "Revolutionizing the future: The power of robotics," Tech.eu. [Online]. Available: <https://tech.eu/2024/12/05/revolutionizing-the-future-the-power-of-robotics/#:~:text=From autonomous systems in defence,personalized medicine%2C and smart cities.>
- [3] Unknown, "Robotics," Formant. [Online]. Available: <https://formant.io/resources/glossary/robotics/#:~:text=In simple terms%2C robotics combines,completing tasks of varying complexity.>
- [4] L. Earnshaw, "Element Overview," *Interviewing*, pp. 4–5, 1987, doi: 10.1007/978-1-349-09528-5_3.
- [5] K. Briggs, "Line Follower," *Introd. to LEGO Mindstorms EV3*, 2019, doi: 10.1007/978-1-4842-5179-9_13.
- [6] ©2025 The LEGO Group, "Training Camp 3: Reacting to Lines," ©2025 The LEGO Group. Accessed: Mar. 02, 2025. [Online]. Available: <https://education.lego.com/en-gb/lessons/prime-competition-ready/training-camp-3-react-to-lines/>
- [7] R. Torok, "Exploring SPIKE Prime Sensors," Lego. Accessed: Apr. 03, 2025. [Online]. Available: [https://community.legoeducation.com/blogs/31/220#:~:text=Force sensor \(touch%2C force\),%25 corresponding to 10 newtons.](https://community.legoeducation.com/blogs/31/220#:~:text=Force sensor (touch%2C force),%25 corresponding to 10 newtons.)
- [8] FLLCASTS, "How does the distance sensor work," 2025 Axles Soft Ltd. Accessed: Mar. 02, 2025. [Online]. Available: <https://www.fllcasts.com/tutorials/2218-how-does-the-distance-sensor-work>
- [9] U. of York, "How do Maze-Solving Robots Work?," Future Learn. Accessed: Mar. 05, 2025. [Online]. Available: [https://www.futurelearn.com/info/courses/intelligent-systems/0/steps/247626#:~:text=and left wall\)-,a.,of the wall being followed.](https://www.futurelearn.com/info/courses/intelligent-systems/0/steps/247626#:~:text=and left wall)-,a.,of the wall being followed.)
- [10] ©2025 The LEGO Group, "LEGO Education SPIKE Prime Resources," Lego. [Online]. Available: <https://education.lego.com/en-us/product-resources/spike-prime/troubleshooting/spike-prime-faqs/#:~:text=The heart of the SPIKE,building and more time learning.>
- [11] Techmaster0, *LEGO Mindstorms Scratch Tutorial #03: repeat commands with loops (RI, EV3 & SPIKE) [English|HD]*, (2022). [Online]. Available: https://www.youtube.com/watch?v=yy4L_duedUU
- [12] ©2025 The LEGO Group, "Assembling an Advanced Driving Base," Lego. [Online]. Available: <https://education.lego.com/en-us/lessons/prime-competition-ready/assembling-an-advanced-driving-base/>
- [13] ©2025 The LEGO Group, "Driving Base 3," Lego. [Online]. Available: <https://spike.legoeducation.com/prime/models/blt9e7ffd6d4cc94c53>

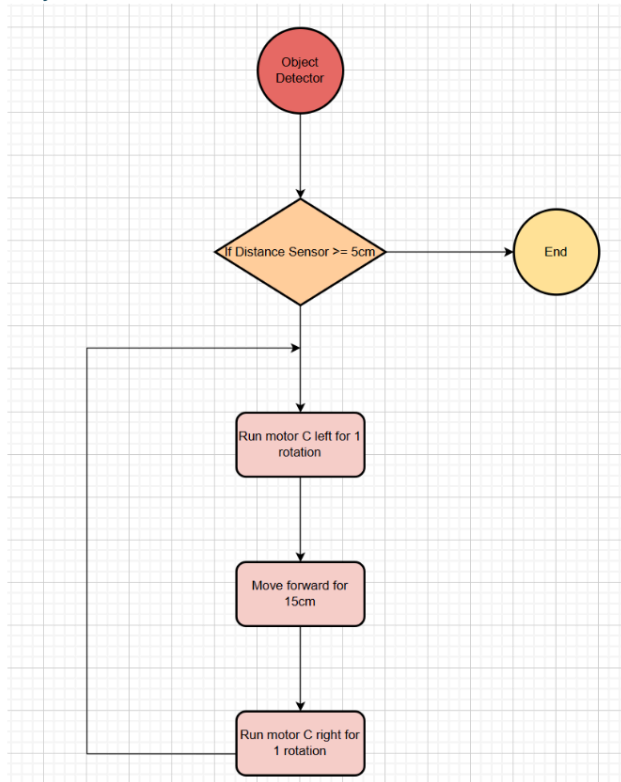


8 Appendices

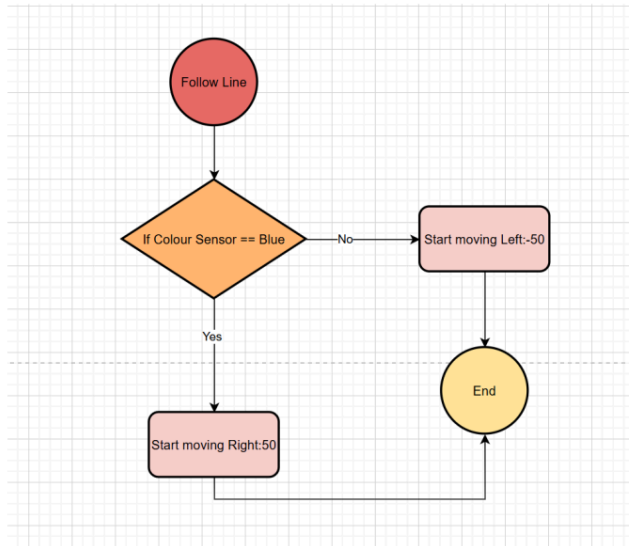
Starting Program Flowchart



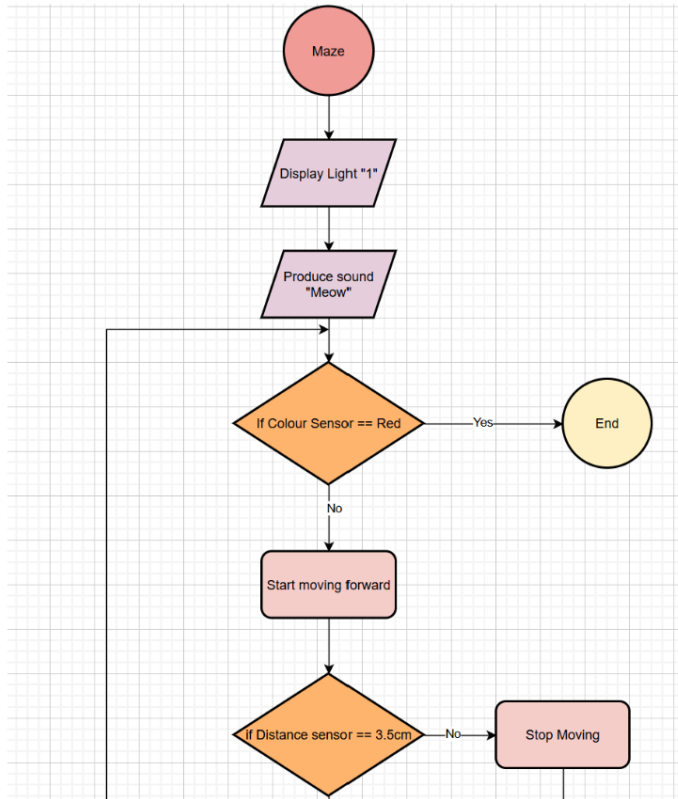
Object Detector Flowchart



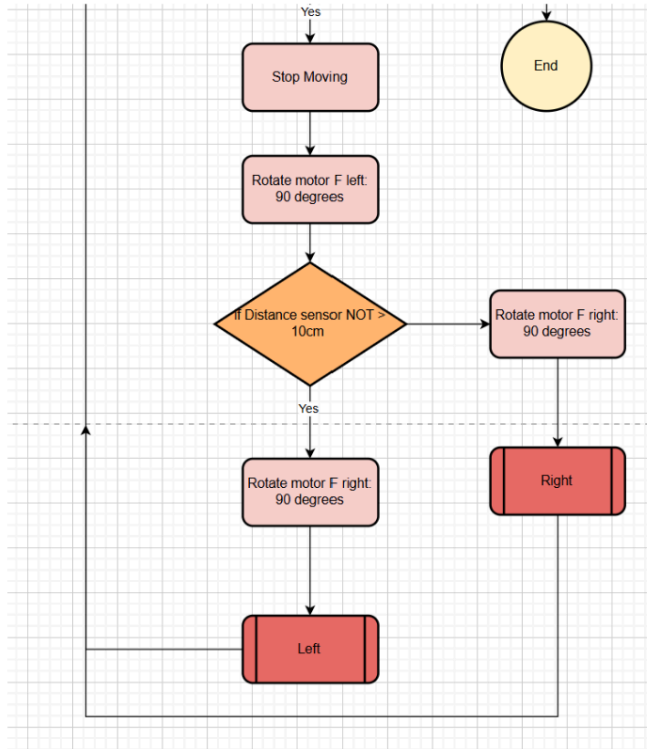
Follow line Flowchart



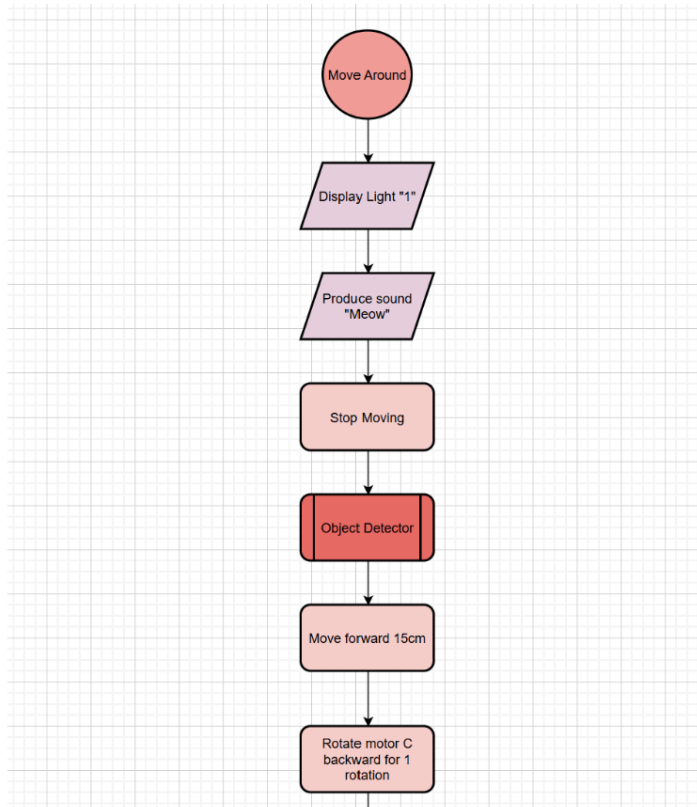
Maze Flowchart First Half



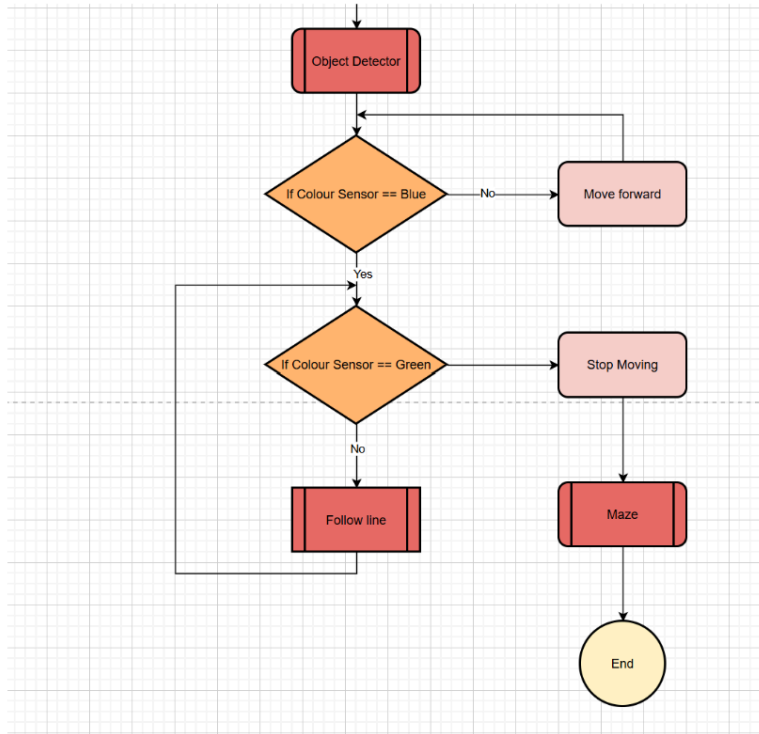
Maze Flowchart Second Half



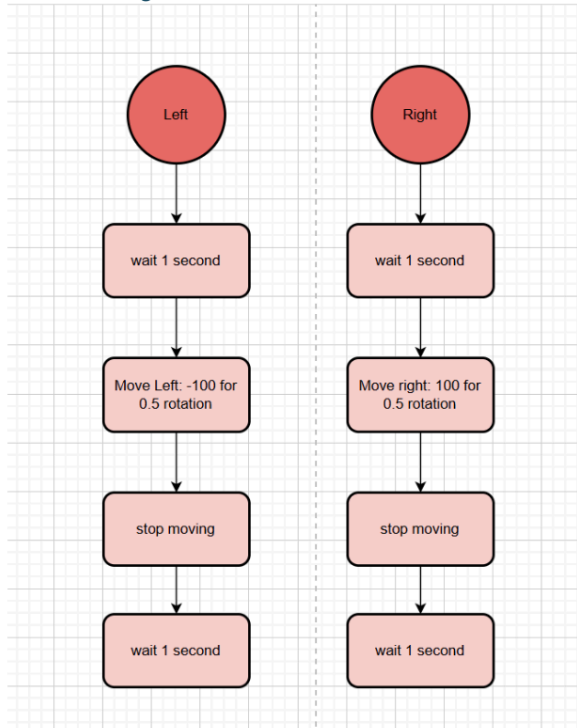
Move Around Flowchart First Half



Move Around Flowchart Second Half



Left and Right Flowchart



Bethan Moyle Integrating Project

GRADEMARK REPORT

FINAL GRADE

GENERAL COMMENTS

85/100

PAGE 1

PAGE 2



Comment 1 | Transferable Skills(15)

A clear ToC provided

PAGE 3



Comment 2 | Transferable Skills(15)

The introduction covers the aims and educational relevance well.



Comment 3 | Transferable Skills(15)

Task delegation and leadership in a specific stage could be clearly stated.

PAGE 4



Comment 4 | Engagement with Literature(15)

including a mix of references—LEGO docs, tutorials, and even some scholarly-style sources is commendable



Comment 5 | Knowledge and Understanding(40)

there is no much discussion of logic complexity or sensor calibration challenges.

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10



Comment 6 | Knowledge and Understanding(40)

personal contribution is not clearly highlighted.



Comment 7 | Cognitive and Intellectual(30)

real issues encountered during the demo and attempts thoughtful analysis of causes



Comment 8 | Cognitive and Intellectual(30)

clear screenshots of Word blocks and flowcharts, effectively explain the modular structure of the robot's logic.

GRADING FORM: DC-ACHA4008

BETHAN MOYLE

85

ENGAGEMENT WITH LITERATURE(15)



14

KNOWLEDGE AND UNDERSTANDING(40)



35

COGNITIVE AND INTELLECTUAL(30)



25

TRANSFERABLE SKILLS(15)



11