

Lego Assignment - Michael

by Michael Davies

Submission date: 07-May-2023 12:18PM (UTC+0100)

Submission ID: 205062649

File name: 42249_Michael_Davies_Lego_Assignment_-_Michael_489678_1876447402.docx (531.89K)

Word count: 1366

Character count: 6893

Title: Lego EV3 (**Integrated Project**)

Author: Michael Davies

ID:2204766

Date: May 7th, 2023

Table of Contents:

1. Introduction
2. Research
3. Plan
4. Time Management
5. Design
6. Coding
7. Evaluation
8. Conclusion
9. References

Introduction

The Lego Mindstorms EV3 was utilized to build and program a robot for the purpose of navigating a line with an obstacle and completing a maze. The robot was designed, constructed, and programmed by the team, who ensured that it could follow the line, detect, and avoid obstacles, and maneuver through the maze. The Lego Mindstorms EV3 is a versatile and exciting tool that combines the creativity of Lego with the possibilities of robotics. This report will detail the team's experience of working with the Lego Mindstorms EV3, building, and programming the robot.

Research

To complete the project, we conducted extensive research into various tools and techniques that we could use to overcome the challenges of the problem. We investigated various sensors and how each one worked, as well as methods for detecting and avoiding obstacles, we also looked at how to solve a maze. Our research showed that using ultrasonic sensors and the Wall method worked the best and would be the most effective approach.

Ultrasonic sensors are capable of measuring distances and can detect objects up to a certain range. They work by emitting high-frequency sound waves and measuring the time taken for the waves to bounce back. This information is then used to calculate the distance between the sensor and the obstacle. We chose ultrasonic sensors as they are easy to use and reliable.[2]

During the process of designing and programming the robot to navigate the maze, the team investigated various methods of solving mazes. After exploring different possibilities, the team found that the most effective approach was to have the robot follow the left wall. This method ensured that the robot never accidentally missed a turn and avoided getting stuck at a dead end, which would cause it to waste time and energy.

By implementing the left-wall-following method, the robot was able to navigate through the maze in a systematic and efficient manner. As it moved along the left wall, it was able to identify all possible paths and make informed decisions on which direction to take. This method not only saved time and energy but also helped the robot to complete the maze more accurately.[3]

Coding language there are a lot of ways to program the EV-3 some of the ways are LEGO MINDSTORMS EV3 Software: This is the official software from LEGO that allows users to create programs for the ev3[4]

python: Python is a popular programming language that can be used to program the EV3. There are several Python libraries available for the EV3, including ev3dev and Pybricks.[5]

Java: Java is another popular programming language that can be used to program the EV3. The leJOS EV3 library provides a Java API for the EV3.[6]

Blockly: Blockly is a block-based programming language that can be used to

program the EV3. It is similar to Scratch but with a different interface and different benefits and drawbacks.[7]

Plan

Our approach to addressing the challenge was to prioritize the stability of the robot. We recognized the need to balance its size and weight to prevent it from toppling over during operation. With this in mind, we designed a compact robot with just two motors for propulsion and added two ball bearings to allow for pivoting on a single point. This configuration not only improves stability but also enables greater maneuverability, making our robot highly efficient and effective.

Time Management

when we were building the robot and coding it was very important to manage our time one way we did this was by tackling one challenge at a time with this we also made sure to test each part of the code once this was done it allowed us to put all the code together as one Photo[3] Photo[4] This allowed our robot to finish the line and the maze without the need to change what code is being used at each different part

Design

The design of our robot was a vital part of everything working and being modular and having this ability meant we were able to change whole areas very easily the 1st design of the robot had two wheels at the front and one centred free-moving wheel at the back this was unreliable since very sharp corners made the robot start to tip which made some parts touch the floor and had the possibility to fall off this was eliminated in the 2nd version of the design by having two balls at the back of the robot making it a lot more stable and doing this allowed for a taller robot which resulted in having a larger more stable robot on the front we designed a push bar which had a large surface area for objects to push on one of the down sides of this was that when it touched an obstacle it had a very small chance of activating the sensor so we changed to having a point at the front of the robot which we stuck with having the point had its downsides as well however the main one was when going around the line it would sometimes miss the obstacle but this was very rare less than 1 in 50 times from what we tested

Coding

when coding the EV-3 robot Rose and myself both focused on solving the maze since we thought it would be the hardest part we both worked on very similar solutions mine involved when ever going too close to a object it reversed a slight bit and then turned left and tried to go forward then kept doing this until a viable route was available Photo[1] we decided this solution had too many faults so we changed to this code Photo[2] this was done so we had more repeatability when doing the maze it also allowed us to have a more precise robot

Evaluation:

During the testing phase, we encountered several challenges, including sensor accuracy and robot stability. We noticed that sometimes the ultrasonic sensors would fail to detect obstacles in time, leading to the robot colliding with them. We also noticed that the robot's movements were not always stable, leading to it tipping over in some instances. We addressed these issues by refining the code and improving the stability of the robot.

One significant issue we encountered during testing was the robot getting stuck in certain areas of the maze. We noticed that the ultrasonic sensors were not able to detect obstacles accurately in some areas, leading to the robot crashing into them. We solved this problem by changing the code we used to the one in [Photo\[2\]](#) this is because it was more stable and was the correct

Despite the challenges we faced, the robot performed well overall. It was able to navigate through the maze and avoid obstacles in most cases. We measured the robot's performance by timing how long it took to complete the maze and noting any errors or malfunctions.

Conclusion:

In conclusion, the project was a success in that we were able to design and program a robot to navigate through a maze while avoiding the obstacle. We achieved our goal by conducting research into the most effective sensors and testing and building the robot and programming the robot to navigate through the maze.

We encountered several challenges during the testing phase, including issues with sensor accuracy and robot stability. However, we were able to address these issues by refining the code and improving the stability of the robot one all this was done we could solve the majority of the mazes thrown at the robot and it solves the line easily every time.

References

- [1] Lego Mindstorms EV3 User Guide: <https://education.lego.com/en-us/support/mindstorms-ev3/user-guide>
- [2] Ultrasonic Sensor for Lego Mindstorms EV3: <https://education.lego.com/en-us/support/mindstorms-ev3/build-a-robot/ultrasonic-sensor>
- [3] Left Wall Following Algorithm: https://en.wikipedia.org/wiki/Maze_solving_algorithm#Wall_follower
- [4] EV3 Programming Software: <https://education.lego.com/en-us/downloads/mindstorms-ev3/software>
- [5] Pybricks. (n.d.). Pybricks. Retrieved from <https://docs.pybricks.com/>

[6] The LeJOS Project. (n.d.). LeJOS EV3 - Java for Lego Mindstorms. Retrieved from <https://lejos.org/ev3.php>

[7] Google Developers. (n.d.). Blockly. Retrieved from <https://developers.google.com/blockly>

Photo 1

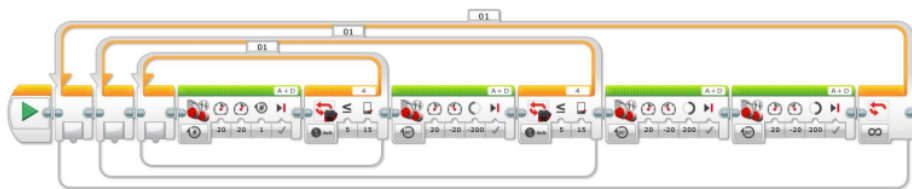


Photo 2

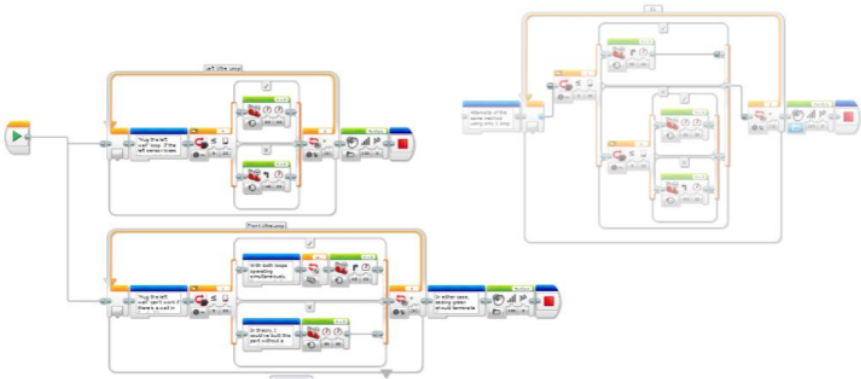


Photo 3

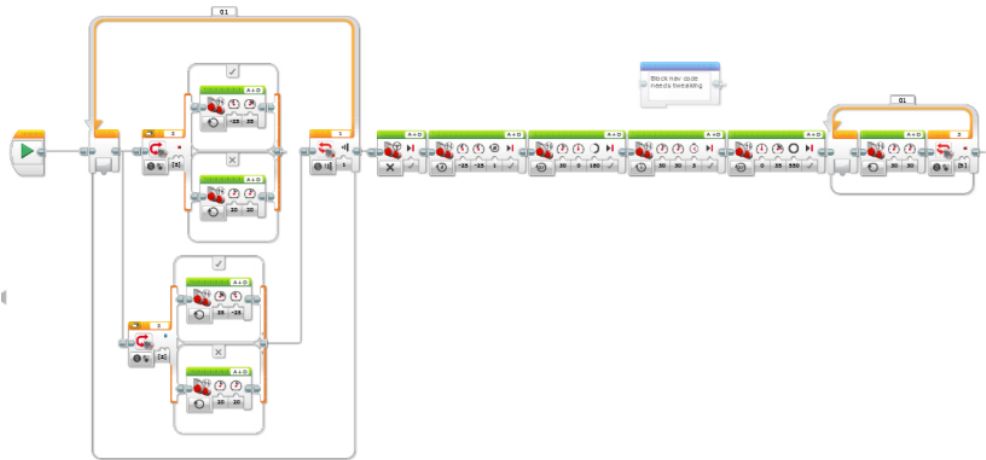
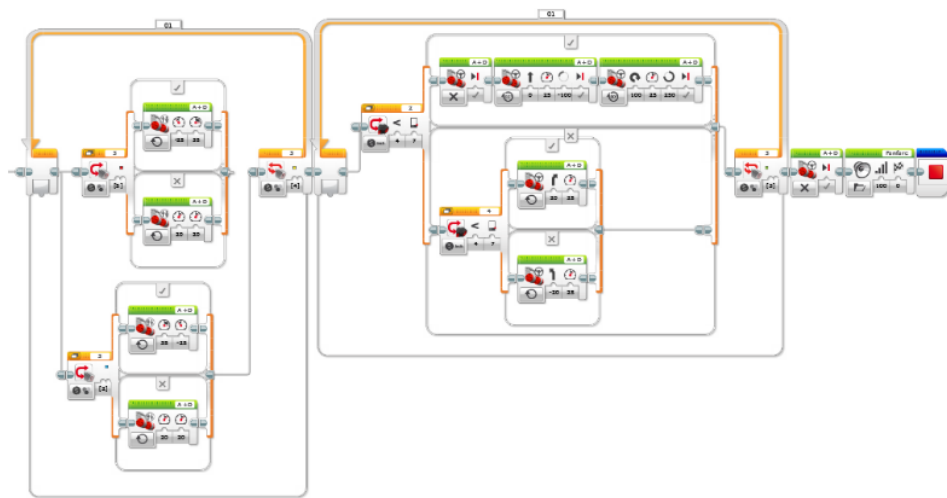


Photo 4



Lego Assignment - Michael

GRADEMARK REPORT

FINAL GRADE

72/100

GENERAL COMMENTS

Instructor

ASSESSMENT COMPONENT 1 (60%)

Project Introduction 4/5

Background research 8.5/15

Project plan 5/10

Design Process 25/40

Evaluation and results 19/30

Total: 61.5%

Comments:

The Introduction would have been improved by inclusion of a diagram, it's a bit hard to visualise!

Some research shown here, but not in much depth. I would have liked a bit more detail on potential maze-solving algorithms. Just include research here - what you actually chose comes later. Why are ultrasonic sensors a most effective approach? What else did you consider?

Design is a bit hard to follow. It would be helpful to include some pictures of potential/previous designs to assist the reader. Planning also needs more detail.

Evaluation is again a bit short. What would you change is you were going to start the project again? For

example, do you still think the ultrasonic sensors, left hand wall etc were the best approach, or would you chose a different approach?

There's some good work here, but try to improve your writing style to make things clearer - include diagrams and number the sections of your report would be a good edition. Remember not to reference Wikipedia for academic reports. This is because the pages can be edited by anyone so many not be reliable - they are a good starting point, but try to find the original reference and point at that instead.

ASSESSMENT COMPONENT 2 (40%)

Technical knowledge/skill 17/20

Analytical and problem-solving skills 17/20

Quality of result (performance of the finished application or suitability of the design) 54/60

Total: 88%

Comments: A strong demonstration, which met the requirements. There was a bit of improvement to be made around transition between line and maze, but this was a minor point. Well done.

OVERALL TOTAL FOR MODULE: 72%

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7
