

Mindstorms Project Report

by Rose Plumley

Submission date: 06-May-2023 11:58PM (UTC+0100)

Submission ID: 205048003

File name: 40436_Rose_Plumley_Mindstorms_Project_Report_489678_739510193.docx (2.3M)

Word count: 1929

Character count: 9933

Lego Mindstorms Assignment

Integrating Project

ACHA4008

Rose Plumley (2218443)

Sophie Meyrick-Vernon

Michael Davies

1. Project Introduction	3
2. Problem Analysis and Background Research	3
3. Project Plan and Proposed Solution	5
4. Finished Designs	6
4.1 Hardware	6
4.2 Software	8
5. Results and Evaluation	9
6. References	10

1. Introduction

This report aims to showcase skills learned during the computing course foundation year, by implementing a solution to a given scenario, combining knowledge from several modules. Specifically, this group has been tasked with building and programming a robot using the Lego Mindstorms EV3 kit and programming environment to navigate a line, negotiate around an obstacle on this line, and solve a maze.

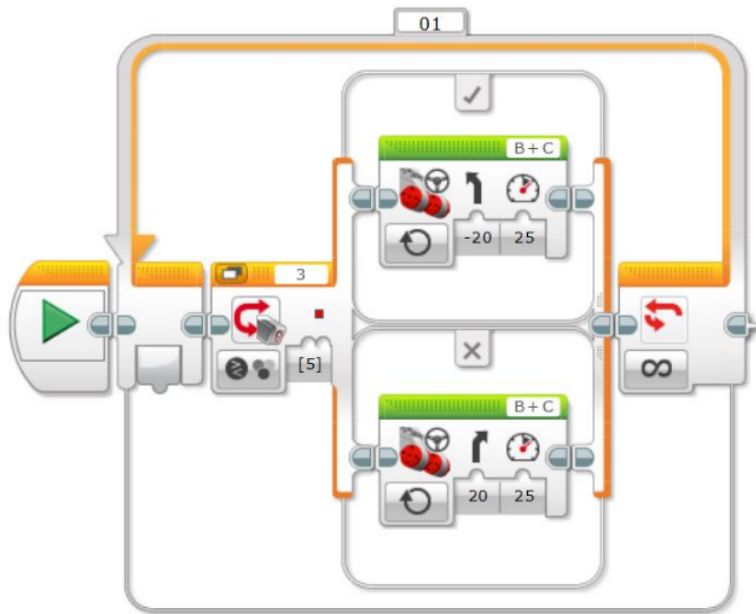
2. Problem Analysis and Background Research

For each part of the problem, the robot will need some form of input to deliver an appropriate output. The Lego Mindstorms kit the team had access to came with the following sensors to receive input: An infrared sensor, a colour sensor, a touch sensor, a gyro sensor, and an ultrasonic sensor.

The first part of the problem, navigating the line, can only provide input through colour. While shape could technically be applied as input, the line is a 2-dimensional shape, and thus cannot be read by the infrared, ultrasonic or touch sensors. For reference, a straight portion of the line looks like this:



Note the separation of the red and blue lines especially. While the robot can receive different colour inputs, it must be able to make decisions based on which colour it detects.



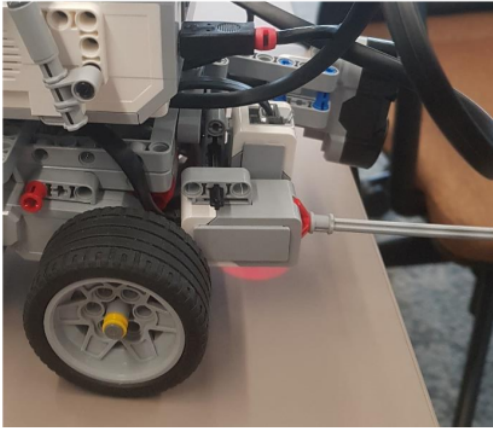
For example, the above programming infinitely loops a command that veers the robot to the left when it detects red and turns right if it does not. To this end, assuming the colour sensor is installed

facing downwards, and is placed roughly in the middle of the machine, it should advance along the line, albeit not in the smoothest motion.

It should be noted however, that the colour sensor works best at a right angle to whatever it is sensing, at a very close (but not touching) distance [1, p. 13].

Along the line, an obstacle (in this case a block) will be placed, which the machine will need to navigate around, after which it must reconnect with the line. For this, either the touch, ultrasonic or infrared sensor can be used. While the colour sensor could also be used to detect the block, the sensor would have to be very close to the object, meaning to avoid collision with the obstacle, the robot would have to be travelling at a much slower speed, to the point of inefficiency. Thus, the task is better suited to the former three sensors.

Conversely, the touch sensor would then require a kind of “collision” as well. However, as it functions as a binary switch (either on or off), it can react much more quickly to input. As well as this, the range at which it is triggered can be extended with other parts found in the EV3 kit. For example, Griffin [2, p. 26-28] demonstrates a design whereby a “bumper” can be used for this purpose. Alternatively, a simple crossbar can be inserted into the tip of the touch sensor’s button, like so:



If the surface area of this extended tip is an issue, a gear could be inserted onto the opposite end. However, the design of the touch sensor itself is somewhat loose, so too much weight on the end of the crossbar can cause it to bend downwards – this can be seen slightly in the above image as well.

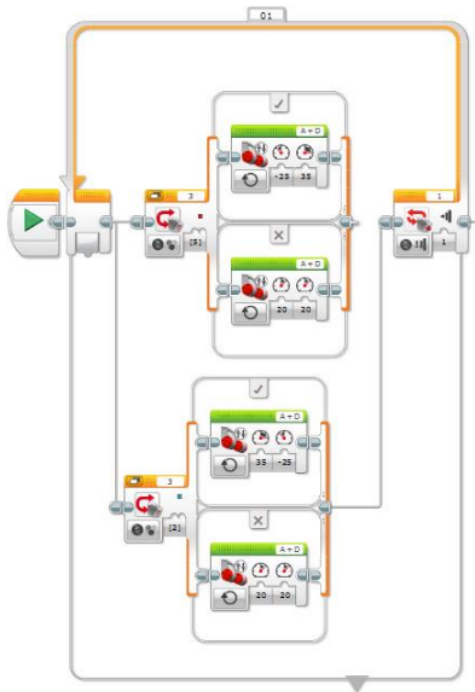
The infrared and ultrasonic sensors are best discussed with the maze problem in mind, as the sensors can be used to determine distance from walls and make positional movements accordingly. Both sensors function by measuring and comparing distances, however the former does so by reflecting infrared light [3], while the latter generates sound and detects the echo [4]. Thus, which sensor is more effective depends on the colour and material of the maze. For instance, if the maze walls were coloured white, and were made of a very glossy material, then it would be much more reflective, making infrared the more effective sensor [5]. Ultrasonic reflectivity, however, is mainly dependant on what the sound wave propagates through. For example, in clinical ultrasound, the propagation of ultrasonic waves is affected more by the density and elasticity of the material it passes through [6]. In air, for example, the speed of sound is roughly 340m/s, while through water it is upwards of 1500m/s.

Nevertheless, reflection is less the concern versus how quickly an echo can be received by the ultrasonic sensor's microphones. During testing against a maze material of black-coloured cardboard, the ultrasonic and infrared sensors performed similarly. If high positional accuracy is a concern, then ultimately ultrasonic would be more suited to the task, as propagation of ultrasonic waves is slower than infrared [7 p. 173]. More specifically, Lego also notes the ultrasonic sensor is accurate to +/- 1cm [4].

3. Project Plan and Proposed Solution

The robot needs to perform three main tasks: follow a line, navigate around an obstacle, and solve a maze. To do this, it will require motors, wheels, appropriate sensors, and the programming to make informed decisions based on input. Thus, following research above, a combination of touch, ultrasonic and colour sensors working in tandem should provide the best results.

Following from the given example in section 2, a more developed version on the same principles can be created:



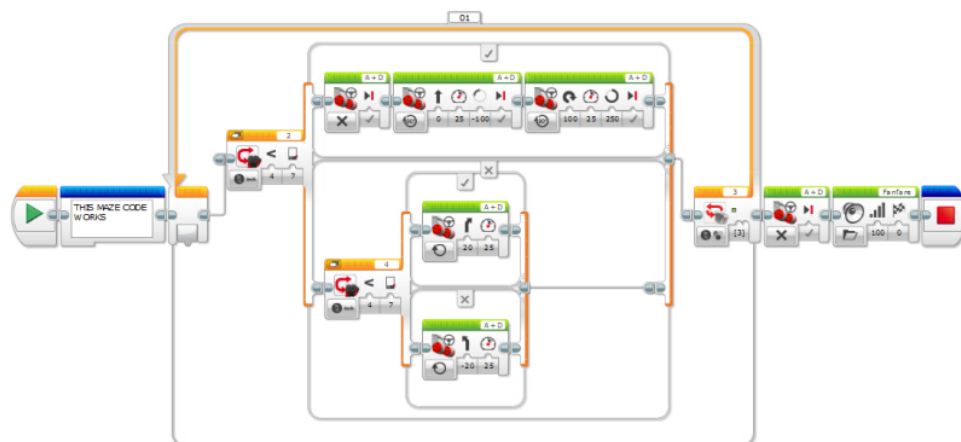
The programming works by looping two switch blocks, one for detecting red, and another for detecting blue. If red is detected, the machine should move forward, turning slightly to the left as well. This is done at under 50 speed to reduce the possibility of overcorrection. Too much movement, and the robot could overshoot the lines, leaving it stuck infinitely moving forward (i.e., the colour sensor does not detect red or blue). Vice-versa, should the colour sensor detect blue, the opposite should occur – slow, forward movement with a tendency to the right. In this way, the machine should “wobble” along the lines, while also being slow enough to allow the robot to follow any curves and turns.

This loop breaks when the touch sensor is triggered – at which point, as the obstacle is kept constant in terms of shape and size, the robot performs a pre-set of movement instructions, as follows:



This line of instructions halts the robot, reverses it, and has it pilot around the right-hand side of the obstacle. Once this is done, a smaller loop moves the robot forward slowly, until it detects red again. Once it does, the line program should subsequently repeat itself.

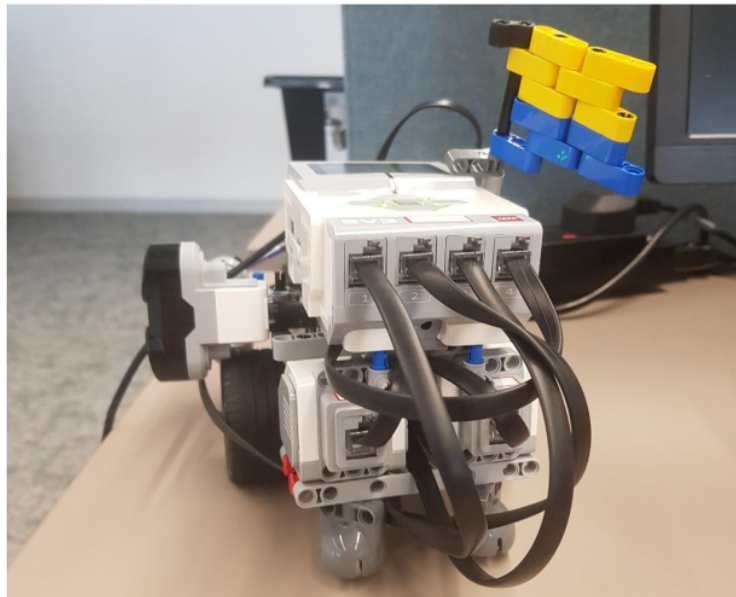
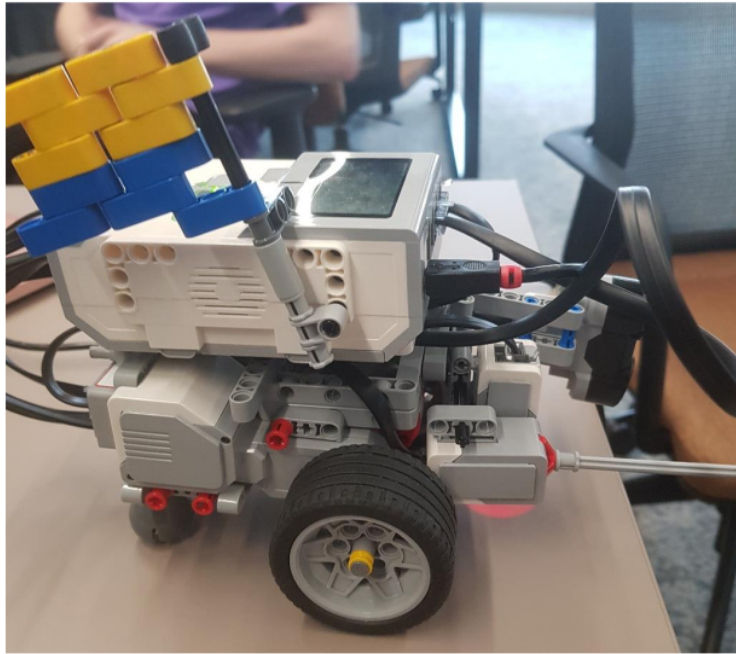
The line following program terminates once the colour sensor detects green. At this point, the code swaps to the maze portion of the programming. As the form of the maze is not a constant, the robot should be able to solve any maze, meaning the prevailing logic used to solve this was to “hug the left wall”. In this case, two ultrasonic sensors were used in tandem. The first, positioned near the front of the machine, functions through a switch. If the distance is under a given distance, in this case 7 inches, the robot should stop, reverse slightly, then turn somewhat heavily to the right. If the distance detected is not smaller than 7”, the logic flows to the next switch, an ultrasonic sensor positioned on the left side of the machine. Following the same distance logic, if the machine is too close to the wall it detects, it moves forward and to the right at a slow speed. If the distance is far enough to not trigger, the robot should move forward and to the left at an equally slow speed.

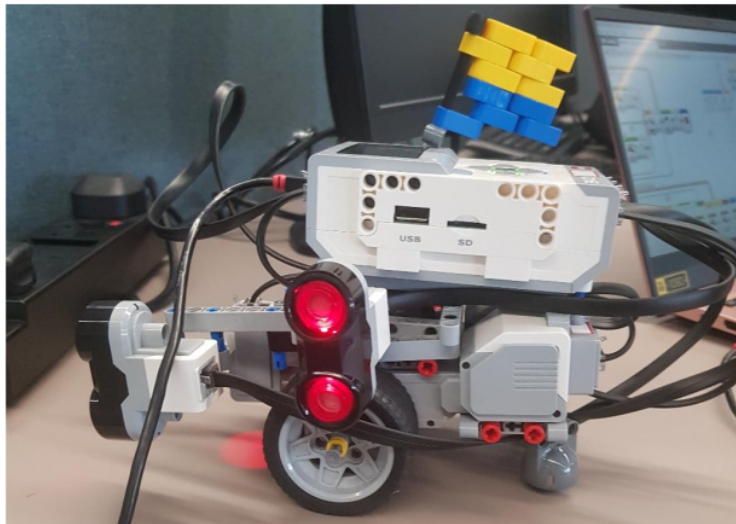
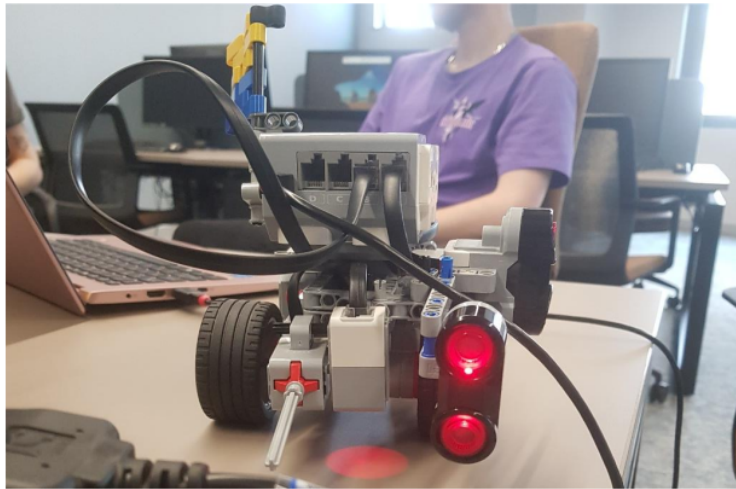


In this way, the logic used is somewhat similar to the line following programming, whereby the machine “wiggles” along a path. This design was also inspired and adapted from Griffin [2, p. 81-92], whose programming follows similar logic, including instructions for following a wall, turning a corner, and notably, going through an opening. The programming above does not need to account for this, however, as no openings were ever used during maze construction and testing. If any improvements were to be made in future iterations, this could nevertheless be a consideration.

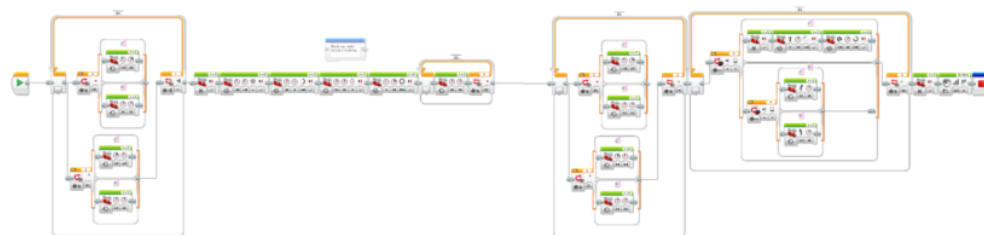
4. Finished Designs

4.1 Hardware





4.2 Software



Above is the full programming for solving each problem.

The first main block covers the line following programming, wiggling along the line and adjusting as needed. Once the touch sensor is triggered, the movement blocks in-between execute. This drives the robot backwards, and pivots around the obstacle. At the end of this

movement is a small loop that moves the machine forward, breaking once it detects red (as the robot moves around the right side of the block, red will be the first colour it can encounter). After this, the line programming is repeated until the colour sensor detects green, at which point the maze programming begins. Once green is detected at the end of the maze, the motors stop, the robot plays a fanfare, and the program terminates.

5. Results and Evaluation

During the line following portion, the programming worked completely, with the robot following along as intended.

At the obstacle, the touch sensor triggered as intended, navigated the block, and re-joined the line, however this only worked as the block was far enough away from the curve in the line. If it were any closer, the machine may not have navigated correctly. Solving this problem would be easiest by using a shorter crossbar at the touch sensor.

Once the robot detected green at the end of the line, it played a fanfare and stopped completely. To this end, it was not able to perform each part seamlessly in one program. This was assumed to be a problem with the logic flow, as green was used to detect both the end of the line portion of the programming, as well as the end of the maze portion. This would be solved easiest by having the colour sensor detect a different colour for either end of the programming sections (i.e., the line could finish at green, while the maze could instead finish once yellow is detected).

In a separate program, the maze was solved as intended for the most part. Tuning and calibration of the motors would be ideal however, as when the machine turns right, it does so in too large of a radius, meaning while it follows the same logic of following the left wall, it does so inefficiently, and in an unintended way. As well as this, it is a potential error that, in a different maze configuration, could have caused the robot to be unable to solve the maze from turning too much in one direction.

Nevertheless, the robot still performed each task, and to this end, the project was a success.

6. References

- [1] *Lego.com*, "Lego Mindstorms Ev3 User Guide" [Online]. Available: https://www.lego.com/cdn/cs/set/assets/bltbef4d6ce0f40363c/LMSUser_Guide_LEGO_MINDSTORMS_EV3_11_Tablet_ENUS.pdf. [Accessed: 05-May-2023].
- [2] T. Griffin, *The art of LEGO® Mindstorms® EV3 programming*. San Francisco, CA: No Starch Press, 2014.
- [3] J. Burfoot, "EV3 sensors," *LEGO Engineering*, 22-Mar-2018. [Online]. Available: <http://www.legoengineering.com/ev3-sensors/>. [Accessed: 06-May-2023].
- [4] "EV3 Ultrasonic Sensor- middle school - education," *LEGO® Education*. [Online]. Available: <https://education.lego.com/en-us/products/lego-mindstorms-education-ev3-ultrasonic-sensor/45504>. [Accessed: 06-May-2023].
- [5] Nagwa, "Lesson explainer: Infrared radiation," *Nagwa.com*. [Online]. Available: <https://www.nagwa.com/en/explainers/757124265845/>. [Accessed: 06-May-2023].
- [6] "FUSIC ultrasound physics. Focused ultrasound in Intensive Care (FUSIC)," *FUSIC-SY*, 04-Jan-2022. [Online]. Available: <https://fusic-sy.co.uk/education/ultrasound-physics/>. [Accessed: 06-May-2023].
- [7] A. Antunes dos Santos Júnior and A. Antunes dos Santos Júnior, *Ultrasonic Waves*. Place of publication not identified: IntechOpen, 2012.

Mindstorms Project Report

GRADEMARK REPORT

FINAL GRADE

81 / 100

GENERAL COMMENTS

Instructor

ASSESSMENT COMPONENT 1 (60%)

Project Introduction 4/5

Background research 7/15

Project plan 7/10

Design Process 34/40

Evaluation and results 25/30

Total: 77%

Comments:

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

There's a bit too much in the Research section about what you actually chose (eg red/blue line) - this would be better in the Design section. I would have liked a bit more on maze theory.

Design process is clearly shown with suitable accompanying photos of robot and code excerpts.

Some good consideration of the limitations and potential alternatives in the results and evaluation section.

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

OVERALL TOTAL FOR MODULE: 81%

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10
