Ryerson University

Department of Electrical, Computer, & Biomedical Engineering
Faculty of Engineering & Architectural Science

Course Title :	Microprocessor Systems
Course Number :	COE538
Semester :	F2021
Instructor	Dr. Vadim Geurkov
Assignment Number	Final Project
Lab Title :	Robot Guidance Challenge
Submission Date :	December 3rd, 2021
Due Date :	December 6th, 2021

Student Last Name	Student First Name	Student Number	Section	Signature*
Peerbocus	Luqmaan	500968131	6	L.P
Kirolos	Youssef	500968175	6	K.Y
Sonam	Datok	500893929	6	S.D

❖ Introduction:

To begin, this report analyzes the different processes of the COE538 final project. This project involved programming an HCS12 microprocessor mobile robot to follow a guided maze in the assembly language. This project would not have been easy if the previous labs were not completed successfully. In the previous 5 labs before completing this project, we got a strong base of understanding on how the assembly language interacts with our microcontroller using the simulated software Code Warrior. In the first couple of labs, we were introduced to the basics of how to use Code Warrior and complete simple assembly programs such as modifying memory locations and controlling peripherals (serial monitors). Moreover, towards the end of the semester, we started doing labs that were related to the eebot. During those labs, we were able to control motors and switches that gave the eebot semi-autonomous capability. In addition, we even utilized state machines to allow the eebot to self-navigate at a random route. We will use the knowledge attained in the previous labs while using some components of the previous labs in order to complete this project successfully.

❖ Objective:

The objective of this project is to program a robot to successfully navigate through a maze that meets the following requirements:

- The robot is started at the entry point and tracks down the guidance line.
- The robot must navigate the S turns to demonstrate that the guidance algorithm is working correctly.
- Whenever the eebot encounters a junction, it should decide which branch to take.
- If the branch comes to an ending road, the robot will encounter a barrier that actuates the front bumper. It should then execute a 180° turn, retrace the path, and take the other branch. It should also note that the branch taken was the incorrect one, and note the correct branch.
- If the robot does not encounter a dead end on that path, it should remember that the branch it chose was the correct one.
- This process continues until the robot reaches the maze forward destination point. The operator taps the rear bumper to indicate this to the robot.
- The robot should then retrace its path, back to the starting point, each time choosing the correct branch. In effect, the robot has learned or solved the maze.

This is a picture of the maze set up :

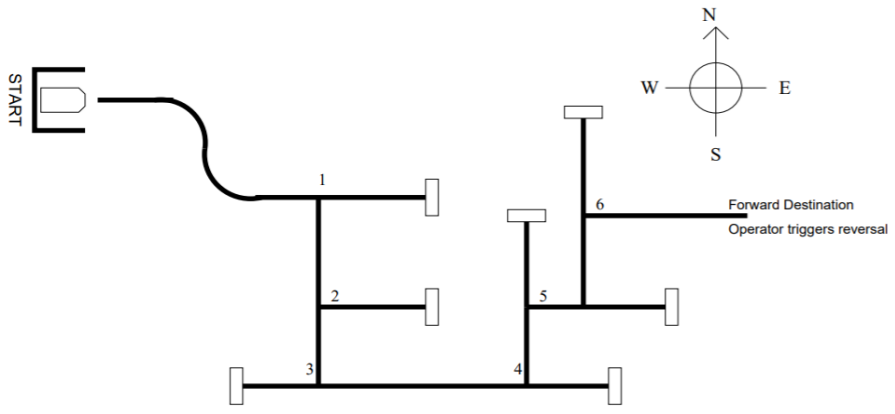

Figure 1: Robot Guidance Challenge

❖ Observations:

This project was completed successfully by breaking down the objectives into sections and focusing on each separately. Starting with understanding the directions which the eebot uses to navigate through the maze. We focused on working with the sensors and the LEDs to get the eebot to start making decisions. The sensors that were used are labeled A through F, as shown in Figure 2 below.

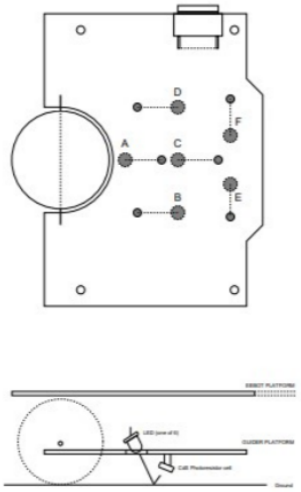

Figure 2. Guider Sensor locations, Top View

Next, we focused on Line tracking and pattern recognition. Working with the read-guider routine, we further modified it for the project. Furthermore, we analyzed the directions and how the eebot should turn either left or right, displaying states, and determining the direction it's facing.

❖ Problems Faced :

During the designing process, the problems we encountered were not being able to physically test our code using the eebot. Everything was theoretical so it was difficult to visualize everything happening and things like how long to turn for since we didn't know how fast the eebot moves. We weren't able to test the sensors to see how they react when not sensing a line underneath the device, this made it more troublesome when determining the thresholds for the sensors. Another problem was deciding what method was the best course of action to take when the eebot encountered a T-junction or an L-junction. The last major problem was remembering the correct path the robot had to take when it finished the maze. It was hard to tell it that it had reached a dead end and that it must remember that this was a bad turn. We weren't sure how to approach this condition at all and not being able to use the physical bot made things much more difficult.

❖ Recommendations/Conclusion:

To further improve the project design there were subroutines that did the same steps as others but with different names. We could reduce the risk of error by simply combining them and jumping to that one subroutine and returning once executed instead of creating multiple subroutines. Some subroutine names do not properly describe what the routine is, for example, “N_STATE1”. We could have used a more descriptive name for it so it is easier to understand when we look back at it in the future or for other readers to look at when debugging or evaluating. In addition, our program did not incorporate the path-saving algorithm to complete the maze and turn around and do it again. Adding in this save routine would allow us to finish the second part of the project and allow our robot to complete the same maze again without going through all the wrong steps.

❖ References:

- 1. Ryerson Electrical and Computer Dept., “COE 538 - Lab 1,” COE 538, 1999.
  - https://www.ee.rverson.ca/-courses/coe538/. [Date: 15-Sep-2021].
- 2. Ryerson Electrical and Computer Dept., “COE 538 - Lab 2,” COE 538, 1999.
  - https://www.ee.rverson.ca/courses/coe538/. [Date: 1-Oct-2021].
- 3. Ryerson Electrical and Computer Dept., “COE 538 - Lab 3.” COE 538, 1999.
  - https://www.ee.ryerson.ca/-courses/coe538/. [Date: 10-Oct-2021].
- 4. Ryerson Electrical and Computer Dept., “COE 538 - Lab 4.” COE 538, 1999.
  - https://www.ee.ryerson.ca/-courses/coe538/_ [Date: 29-Oct-2021].
- 5. Ryerson Electrical and Computer Dept., “COE 538 - Lab 5.” COE 538, 1999.
  - https://www.ee.ryerson.ca/courses/coe538/ [Date: 4-Nov-2021].
- 6. Ryerson Electrical and Computer Dept., “COE 538 - Lab Project.”COE 538,1999.
  - https://www.ee.ryerson.ca/~courses/coe538/ [Date: 18-Nov-2021].
- 7. Ryerson Electrical and Computer Dept., “COE 538 - Robot Guider.”COE 538,1999.
  - https://www.ee.ryerson.ca/~courses/coe538/. [Date: 18-Nov-2021].