Toronto Metropolitan University
Department of Electrical, Computer, & Biomedical Engineering
Faculty of Engineering & Architectural Science

Course Title:	**Microprocessor Systems**
Course Number:	**COE538**
Semester/Year (e.g.F2016)	**S2024**

Instructor:	**Arghavan Asad**

Assignment/Lab Number:	**6**
Assignment/Lab Title:	**Project**

Submission Date:	**August 11, 2024 9:00 PM**
Due Date:	**August 14, 2024 9:00 AM**

Student LAST Name	Student FIRST Name	Student Number	Section	Signature*
Yashotharan	Ashviny	[redacted]	1	[redacted]
Kuganendra	Aananth	[redacted]	1	[redacted]
Qureshi	Maheen	[redacted]	1	[redacted]

*By signing above you attest that you have contributed to this written lab report and confirm that all work you have contributed to this lab report is your own work. Any suspicion of copying or plagiarism in this work will result in an investigation of Academic Misconduct and may result in a "0" on the work, an "F" in the course, or possibly more severe penalties, as well as a Disciplinary Notice on your academic record under the Student Code of Academic Conduct, which can be found online at: https://www.torontomu.ca/content/dam/senate/policies/pol60.pdf

Introduction

This report covers the Lab Project: Robot Guider Challenge for COE538: Microprocessor Systems. In this project, the robot in use, *eebot*, is required to navigate itself, from start to finish, in the maze shown in Figure 1.

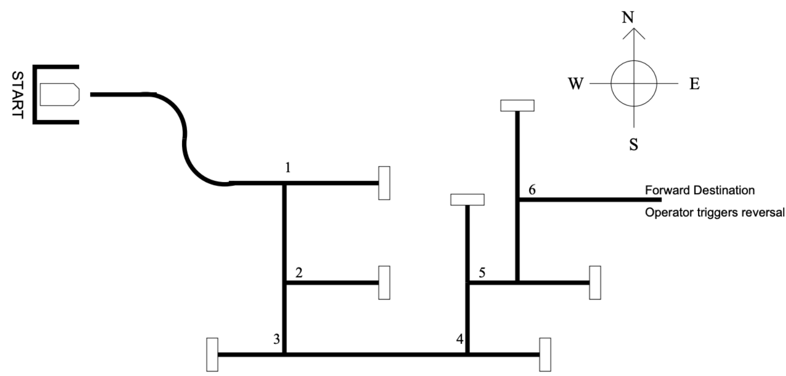

Figure 1: Robot Guidance Challenge

Successful navigation of the *eebot* was accomplished by programming the HCS12 microcontroller using Assembly language. This comprehensive project tested technical skills and knowledge while also providing valuable insight into system integration and troubleshooting. Through this endeavor, we gained hands-on experience in microprocessor programming and hardware interfacing, solidifying our understanding of the concepts included in this course.

Description

The HCS12-based assembly code in this project consisted of several main coding blocks. The equates section defines constants for various components, including the LCD display, timers, and robot states like "START," "FWD," and "ALL_STOP." These constants provide a foundation for the robot's operations by ensuring consistent and accurate control signals throughout the program. The data section establishes baseline and variance values for the robot's sensors, ensuring accurate readings for decision-making. It also includes memory storage for the display lines and sensor readings, which are crucial for monitoring and controlling the robot's actions in real-time.

In the variable section, two critical variables are introduced: "TOF_COUNTER," for tracking real-world time, and "CRNT_STATE," for storing the robot's current operational state. Additionally, the section includes variables for number storage, a hex table for conversions, and spare memory for future use, enhancing the program's flexibility and robustness.

The movement programming section builds on these initialized constants, handling the logic for moving the robot and updating the display to reflect current states and actions. This part of the code required extensive troubleshooting to ensure the robot performed as expected.

Finally, the last blocks of code, adapted from previous labs, include routines for BCD to ASCII conversion, display updates, delays, and sensor readings, all of which are essential for the robot's functionality.

Troubleshooting

When working on any technical project, challenges inevitably arise and must be addressed to ensure the program functions as intended. Despite the assembly program compiling seamlessly, several issues were encountered during the testing of the *eebot*. Each *eebot*s' conditions may vary, in particular, the scanner condition. The program must use different base variance variables to account for this. Threshold values for each sensor had to be calculated for the variance. The average of the highest (on-path) and lowest (off-path) values of the sensor provided the threshold value. Without the variance values, the robot could steer off-course.

The threshold values needed to be applied correctly to keep the robot on the path. An iterative approach was used, testing multiple variance values until the robot functioned properly. At first, the threshold values were used as the variance. The robot steered off the maze. Then, the threshold was reduced to 80% of the average producing even worse results; the robot began spinning in circles. At one point, the robot began turning back when it encountered a crack in the wooden maze, bringing into question a flaw in the software, hardware, or even the maze itself was flawed. Once the variance was set as the difference between the high and threshold value of the sensor, the robot navigated itself through the S turn component of the maze. However, the robot began to struggle at intersections, turning back and moving away from the finish.

This time, the focus was on the robot's turning speed. Reducing the speed allowed the sensor more time to make decisions, enabling the robot to follow the black line on the map more accurately. However, the robot's turns were still too rapid, causing it to veer off course entirely. To address this, we programmed the turns to be executed more quickly, believing that this would allow the robot to follow the map more effectively. By reducing the turn time from 8 to 4 units, we finally achieved the correct balance, and the robot was able to navigate the maze as intended.

This project was relatively small in scale, the code was still quite extensive. Most of the code branched off the skeleton code in lab 5, while the read-guider.asm file was used for the *eebot* sensors. This project enforced industry-level skills such as coding using frameworks, time-management, problem solving and testing/debugging – all vital for managing large software projects..

Conclusion

In conclusion, our team gained valuable experience in programming the HCS12 microcontroller using Assembly language and implementing sensors on the *eebot* robot to navigate a maze. We learned a great deal about testing hardware and software, and working together to troubleshoot complex issues. This project has equipped us with practical skills that will be beneficial in our future endeavors, both in the industry and beyond.