Introduction

The purpose of this project was to program a HCS12 eebot mobile robot using Assembly language to navigate through a maze by following a single black line and avoid dead ends (barrier) to move from the starting point of the maze to the end of the maze.

Project Description

The eebot uses a HCS12 microcontroller, and has multiple input devices that it utilizes to navigate through the maze including a bumper at the front and the back of the bot, and 5 line sensors and the bottom of the robot.

The bumpers are used to detect whether the bot has reached a dead end or not such that when triggered a subroutine is called that causes the robot to make a U-turn and take the other exit at the intersection (usually to turn the same direction as the one it did to get into the dead end). This was achieved by making the Robot pull the addresses of the correct commands from the stack as it had them stored and this would ensure the robot would not mess up and not go the wrong path.

The line sensors are 5 photoresistors that are used to follow the black line the maze is made off. They detect the line and move the robot forward as long as the line is there. When the robot reaches an intersection it turns towards the side of which it is closer to, as the line sensors are detecting a stronger signal there. For the robot to take a turn it calls the respective subroutine for left turn or right turn. This also ensured the robot would always stay on track so it does not wander off and go off the maze, however this made a robot navigate through the maze slowly, but it worked and achieved the goal of the project.

Most of the code was written in previous labs such as the right turn, left turn, and U-turn, however extra code had to be written to follow the line and call the respective subroutines for when certain events occurred, such as reaching a deadend and approaching an intersection.

Moreover, a section of the code was written to show the current states the robot is in and the current values in registers and accumulators.

Problems Encountered

The HCS12 microcontroller was a complicated chip that needed a lot of coding to get it to work, in the process we encountered multiple problems. A number of problems were solved through trial and error. Like the values of guidance variables were done through trial and error, the reason for that was because specific numbers depended on the robot as every robot had a specific way to set the numbers on the guidance variables, specifically the left, right, and U-turn distance. Since we didn't use the same robot we used for the last time, our numbers kept switching which was a huge problem and we had to redo the process of figuring out the exact value needed through trial and error every time we didn't use the same robot. The reason why we didn't use the same robot us cause while programing the first one we forgot to keep track of the number so the next time we grabbed another random robot it resulted in us changing the numbers in the code to get the right amount of movement from the robot in order for it to follow the tapped black line. Also another problem was the speed of the robot, it was moving very slow but that did not affect the outcome or the process of completing the full round from start to finish.

The parts that worked well was the data section which was easy and simple to work on. While parts we struggled in working the code was the state section, there were a lot of things that we had to fix and analyze to get the code working in its proper way. Another section that was a struggle was the sensor subroutine section, this part was vital and to work on since if the code

was not right that the robot would not follow the black taped line which is one of the main parts in the code.