SYSC4805A_W20_L1

PROGRESS REPORT

Project Group: B'Dazzled Blue

Group members:

Samy Ibrahim (101037927)

Muhammad Tarequzzaman (100954008)

Jacob Martin (10100849)

Ahmad Chaudhry (101003005)

Table of Contents

Abstract

This document outlines the proposal for our SYSC 4805 maze solving project, throughout the document we will be discussing our objective, deliverables, and our approach to the solution. We will also present our timeline for expected tasks and milestones through a Gantt chart and a responsibility assignment matrix. Diagrams depicting the hardware used throughout the project as well as a sequence diagram showing the flow of code is also presented.

Overall Objective

The overall object of this project is to design and implement a robot that can solve a simple maze with multiple paths to get to the end objective using sensors fusions for robot guidance and the robot will print out the successful path upon completion the objective.

Overall Deliverables

The overall deliverables of this project will be detailed in this section. The robot is using multiple sensors to allow it to autonomously navigate through a maze. The mentioned maze consists of multiple paths on the ground drawn out with black tape leading to an objective at the end. The required sensors to complete the maze are photodiode and ultrasonic sensors attached to an Arduino (and robot assembled in Lab 1: consisting of the body, motor, and wheels). Currently, we are using four photodiode sensors and one ultrasonic sensor to perform the task at hand. Two of the photodiode sensors are placed at the front of the robot, and one photodiode sensors is placed on the both sides of the robot. The photodiode sensors on the front of the robot is used to navigate the robot forwards and keeping it on the specified black line. The sensors on either side of the robot are used to determine if a right/left turn is coming. The ultrasonic sensor is used to determine if there are any obstacles in the path of the robot, if the robot notices an obstacle, it will turn around and try different routes. While solving the maze, the robot saves the path it took and once the robot finds the end of the maze (finds the objective), it will then proceed to output its recorded path from memory onto an LCD screen attached to the robot.

Success Criteria

Multiple test Maze's will be developed as testing scenarios for the robot, the Maze's will be designed in a way where the "objective" can be moved around so that the same Maze can be used multiple times for testing. In general, we plan on conducting

- Unit Testing:
    1. Test Technical and Functional aspects of the robot separately
    2. Write separate tests for every component in the system (just assert sensors are reading data and assert to 1)
    3. Line Following Test: create a test that ensures the robot remains on track
        - Set a time limit for the test, and assert that both motors are never off, the robot should always be following the line (if it is adjusting, one motor may be off but the other must remain on)
    4. Sensing Objects Test: create a test to ensure the ultrasonic sensor can determine an object is in its way. We will be using varying heights and distances of obstacles to test further.
    5. Determining End of Maze Test: The end of the maze will be specified with white tape on the ground, so this test asserts that if both front photodiode sensors get triggered by the white tape, the robot stops and prints its output to the LCD.
    6. Printing Successful Path Test: This test asserts that once the robot solves the maze, it indeed prints out the path, the successful (expected) path can be hard coded as a string and then compared to the output of the robot
- Performance Testing:
    1. Response Test: we must ensure that the quick output of our sensors doesn't get overridden by another aspect of the system taking too long (for example, a serial print takes a full millisecond, is it needed?). The test will consist of timers (modules) recording the execution time of all methods and loops in the system. These will be analyzed and important methods (such as turning) will need to be optimized.
    2. Full Maze Performance Test: this test will record the time it takes to solve the full maze and report which aspects of the robot are taking the longest (so that those areas can be optimized).
- Stress Testing:
    1. Big Maze Test: if the maze is S times the size of our current testing one (S times the intersections and turns), how will the robot perform, will it be able to solve the maze in less than (or equal to) S*(usual maze solving time).
    2. No Objective Test: This will test the ability of the robot to handle no Objective found at the end of the maze (to know and not keep looping and looking).

- Integration Testing:
    i. Test the entire system working together
    ii. A full run of the robot testing a simple Maze and ensuring that it can be solved successfully, and that the correct path is printed.
- If the system can pass all these test (with acceptable results in non-unit tests), then it behaves as expected and should successfully find the objective.

Trigger Type

The Maze Solver we are building is an Event Triggered system meaning that the main flow is controlled by specific events rather than relying on time. The main reason for “Events” in our system is the robot getting off its course (getting off the black line/ maze), if that specific event occurs, then the robot immediately fixes itself (gets back onto the line). Another big event is coming across an intersection (places where the robot can turn), if that happens, the robot will have to remember what decision it made and turn into one of the paths.

Requirements

Based on the overall deliverables of the project the following hardware components will be required to develop the maze solving robot:

- 1 x Robot Physical Hardware With 2 Motor and Wheels x 2
- 1 x Arduino nano
- IR Optical Sensors x 4
- Ultrasonic Sensor x 1
- L9110S Dual H-Bridge motor driver x 1
- Motor Shield board x 1
- Connection Wires
- Switch x 1
- LCD Screen x 1
- Battery Holder x 1
- Battery x 4 + x2

WBS (Work Breakdown Structure)

1. Project Design
   a. Coming up with project idea
   b. Design the project and write the Proposal
2. Implement Simple Robot
   a. Assemble Robot body with the frame, motor, and wheels.
3. Add Maze Solver Specific Hardware and Line Following
   a. Add ultrasonic and photodiode sensors
   b. Design and Implement an algorithm to follow a black line
4. Design Algorithms for Maze Solving
   a. Turn around if obstacle or end of path is found
      i. Implement ultrasonic sensor to sense obstacles
      ii. Add turning functionality
   b. Keep trying paths until objective is found
      i. Avoid already used paths
5. Design Algorithm for Printing Successful Path
   a. Store successful/unsuccessful paths in 2D arrays
   b. Display successful path on LCD screen
6. Functional Testing of Robot Sensors and Functionality
   a. Unit Testing
   b. Integration Testing
7. Non-Functional Testing of Robot
   a. Performance Testing
      i. Time critical system maintained
      ii. Watchdog Timer activation to avoid miss deadline.
      iii. Battery Sense
   b. Stress Testing
8. Verification & Validation
   a. Code verification
   b. Integration of verified code
   c. Functionality and Code validation

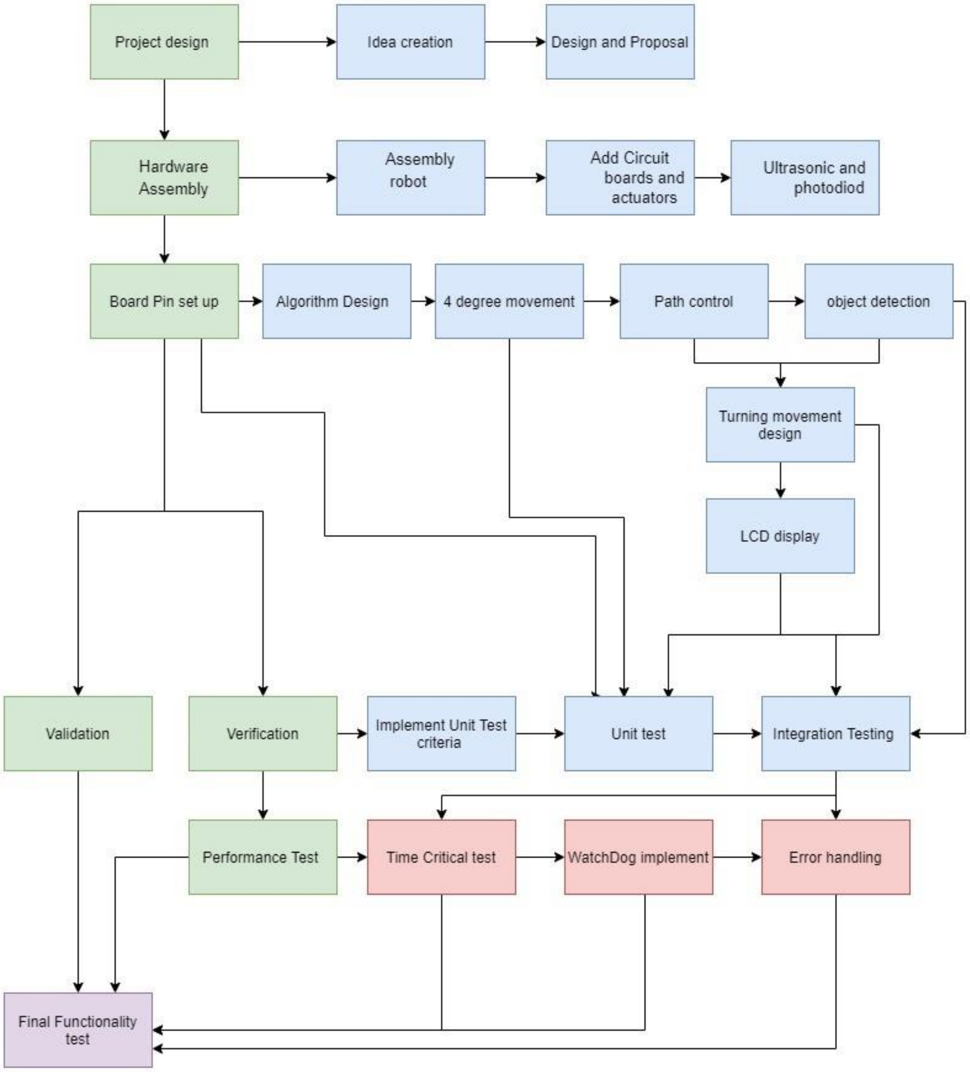

Figure 1: WBS Summary Diagram

Schedule Network Diagram

This diagram in fig.1 outlines our schedule for the defined task in respect to each other.

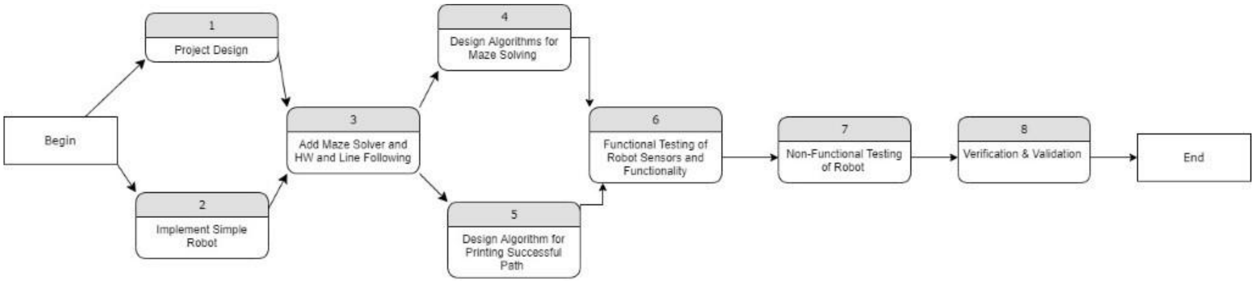
Begin
1
Project Design
2
Implement Simple Robot
3
Add Maze Solver and HW and Line Following
4
Design Algorithms for Maze Solving
5
Design Algorithm for Printing Successful Path
6
Functional Testing of Robot Sensors and Functionality
7
Non-Functional Testing of Robot
8
Verification & Validation
End

Figure 2: Schedule Network Diagram

Gantt chart

Gantt chart diagram in fig. 2 out lining our task with respect to labs and dates.

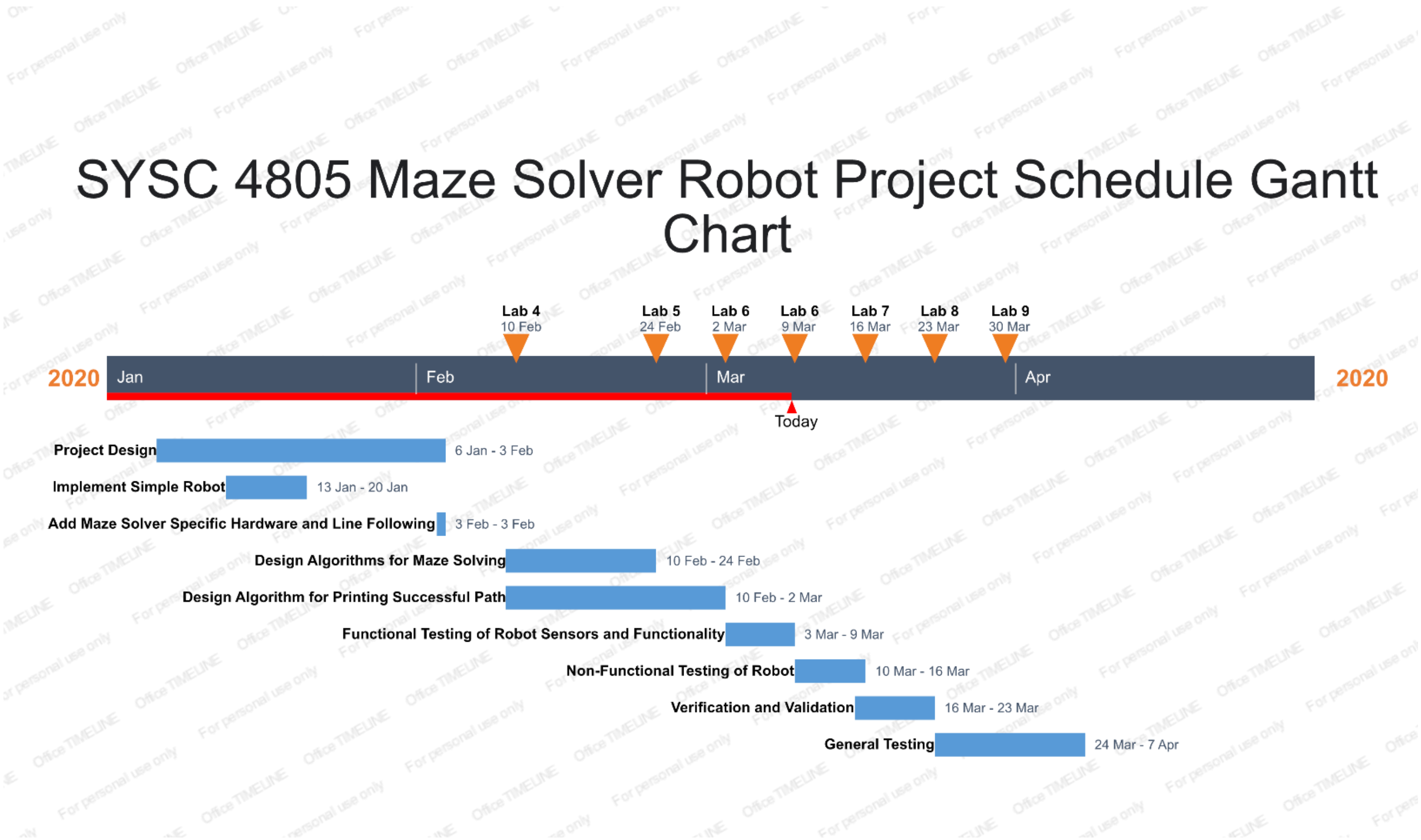

Figure 3: Gantt Chart

Responsibility Assignment Matrix

List the key personnel who will be responsible for completion of the project, as well as other personnel involved in the project.

MT: Muhammad Tarequzzaman	AC: Ahmad Chaudhry	SI: Samy Ibrahim	JM: Jacob Martin

ACTIVITY	RESPONSIBE	APPROVER	CONTRIBUTION	Informed
1.a -Coming up with project idea	JM	SI	AC	MT
1.b -Design the project and write the Proposal	SI	AC	MT	JM
2.a -Assemble Robot body with the frame, motor, and wheels.	AC	JM	MT	SI

3.a -Add ultrasonic and photodiode sensors	JM	AC	SI	MT
3.b -Design and Implement an algorithm to follow a black line	JM	SI	MT	AC
4.a -Turn around if obstacle or end of path is found	JM	SI	MT	AC
4.b -Keep trying paths until objective is found	SI	JM	MT	AC
5.0 – 2d Array Implementation	SI	JM	AC	MT
5.a -Store successful/unsuccessful paths in 2D arrays	SI	JM	AC	MT
5.b -Display successful path on LCD screen	AC	MT	JM	SI
6.a -Unit Testing	MT	SI	AC	JM
6.b -Integration Testing	AC	MT		
6.c – Watchdog	MT	SI		
6.d - Battery Sense	MT			
7.a -Performance Testing	MT	SI		
7.b -Stress Testing	AC	MT		
7.b.i- Proper turning during obstacle detection				
7.b.ii - Controlling its path while on the line				
8.a -Code verification	MT	SI	AC	JM
8.b -Integration of verified code	MT	AC	JM	SI
8.c-Functionality and Code validation	SI	MT	AC	JM

Table 1: Responsibility Assignment Matrix

Circuit Diagram

Below is a detailed circuit diagram of the maze solving robot:

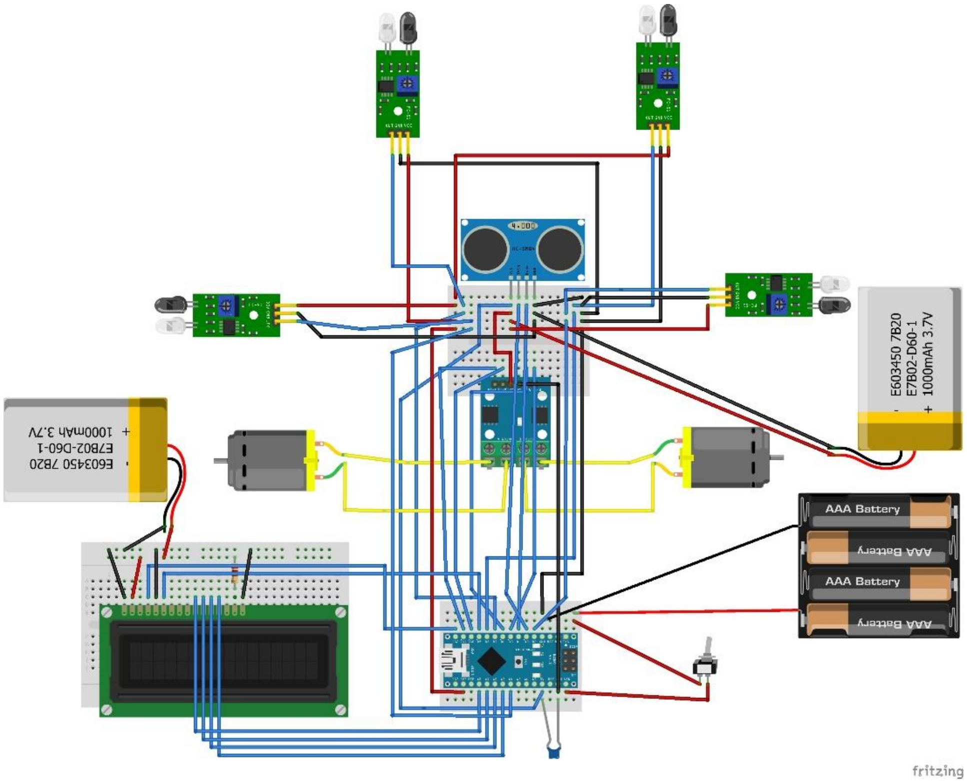

Figure 4: Circuit Diagram

Circuit Schematics

Below is a detailed circuit schematic diagram of the maze solving robot:

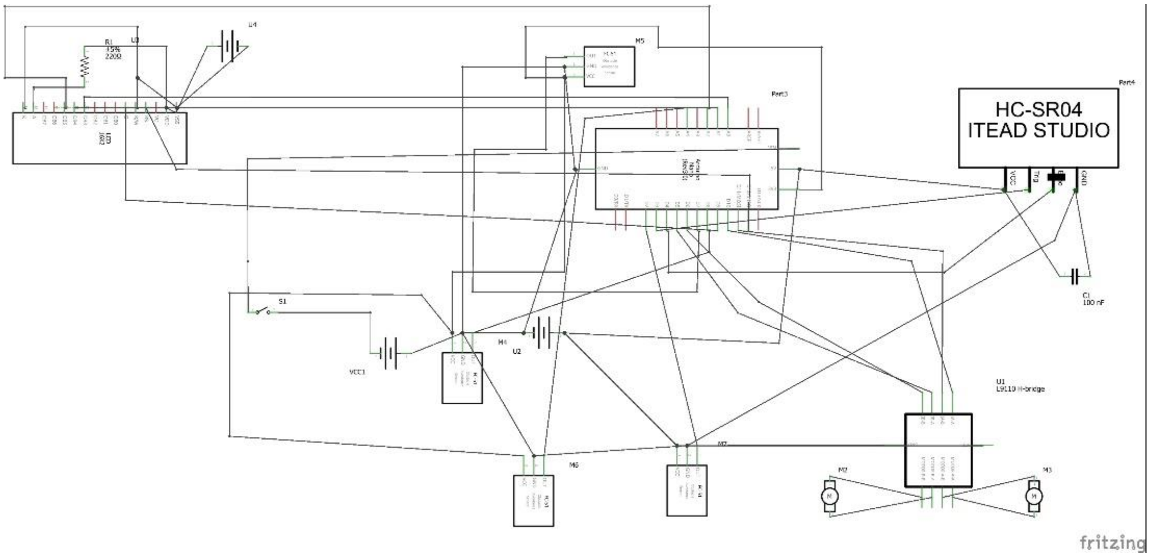

Figure 5: Circuit Schematics

Sequence Diagram

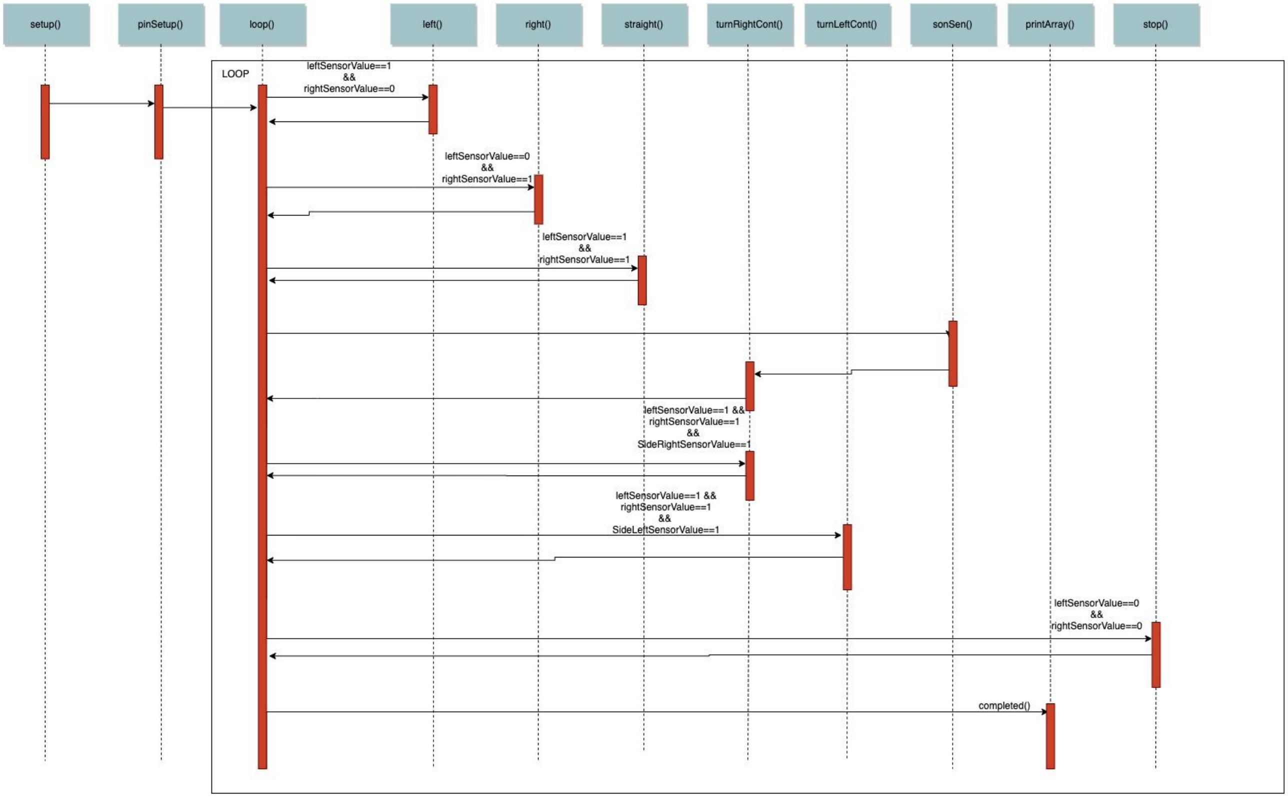
setup()
pinSetup()
loop()
left()
right()
straight()
turnRightCont()
turnLeftCont()
sonSen()
printArray()
stop()
LOOP
leftSensorValue==1 && rightSensorValue==0
leftSensorValue==0 && rightSensorValue==1
leftSensorValue==1 && rightSensorValue==1
leftSensorValue==1 && rightSensorValue==1 && SideRightSensorValue==1
leftSensorValue==1 && rightSensorValue==1 && SideLeftSensorValue==1
leftSensorValue==0 && rightSensorValue==0
completed()

Figure 6: Sequence Diagram

Appendix

Provide supporting material for your proposal here. It may be:

Component Cost Reference:

- Robot framework x 1 (Comes with kit)
- Arduino nano x 1 (Comes with kit)
- IR Optical Sensors x2 (Comes with kit) + x2 (bought)
- Ultrasonic Sensor x 1 [https://www.sparkfun.com/products/15569] ($3.95)
- L9110S Dual H-Bridge motor driver x 1 (Comes with kit)
- Motor drive board x 2 (Comes with kit)
- Wires (Comes with kit)
- Switch x 1 (Comes with kit)
- Wheels x 2 (Comes with kit)
- LCD Screen x 1 (Pre-Owned)
- Battery Holder x 1 (Comes with kit)

Total Cost:

$10.95

Datasheets:

- Arduino nano x 1 [https://culearn.carleton.ca/moodle/pluginfile.php/3697721/mod_resource/content/1/ATmega168Data-Sheet.pdf]
- IR Optical Sensors x 2 (Comes with kit)
- Ultrasonic Sensor x 1 [https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf]
- L9110S Dual H-Bridge motor driver x 1 [https://culearn.carleton.ca/moodle/pluginfile.php/3697718/mod_resource/content/1/Motor_control_driver_chip-l9110.pdf]

Research materials:

1. This resource provides us with an example on how to interface with the ultrasonic sensor and how the physical component works providing us with a better understanding of how to use the component within the project. https://howtomechatronics.com/tutorials/arduino/ultrasonic-sensor-hc-sr04/

2. Below are few possible references that we can use as possible solutions for the maze solving algorithms that can be implemented into the robot.
    - https://www.cs.bu.edu/teaching/alg/maze/
    - http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.302.4944&rep=rep1&type=pdf
    - https://www.researchgate.net/publication/331481380_A_Review_of_Various_Maze_Solving_Algorithms_Based_on_Graph_Theory
3. For testing we will use "ArduinoUnit" Eclipse IDE to develop Unit test for Arduino Hardware functionality and code functionality, Bellow is the documentation,
    - **ArduinoUnit**: https://github.com/mmurdoch/arduinounit
    - **AUnit:** https://github.com/bxparks/AUnit
    - Example: https://maker.pro/arduino/tutorial/using-unit-test-frameworks-with-arduino