SYSC 4805 Computer Systems Design Lab

Progress Report

Team Fandango

Team 1 L1

Mostafa Taha

Flynn Graham

Laith Abdelrazeq

Faris Abo Mazeed

November 17th, 2022

Table of Contents

1.0 Project Charter

1.1 Overall Objective

The objective of this project is to develop a solution to the snow plow problem posed by Professor Taha. We are to build and develop a robot to clear a testing area of snow (represented by golf balls). The robot should clear this snow while avoiding all obstacles in the testing area. The robot should ensure it passes through the entire available testing area and clears as many if not all golf balls as possible. A variety of sensors will be used to complete a full sweep of the testing area while avoiding the mobile and stationary obstacles. The robot will need to be outfitted with these sensors and modified accordingly with a plow to aid in completing the objective. The solution will always stay within the given testing area, while moving the balls out.

To accomplish this, we set forth a clear set of development tasks to be accomplished that will allow us to achieve this goal. These development activities can be found in sections 2.2 and 5.1.

1.2 Overall Deliverables

- Project Proposal (October 14th, 2022)
    - (Project Charter, Scope, Schedule, Cost, Human Resources)
- Progress Report (November 11th, 2022)
    - (System Architecture, Statechart, Sequence Diagram, Value Analysis Figure)
- Group Presentation (November 21st, 2022)
    - (Introduce, Explain Background, Demonstrate)
    - (13 Min Presentation, 2 Min Question/Answer)
- Lab Demonstration (December 8th, 2022)
    - (Remove Maximum Number Of Snowballs Off The Perimeter Using Autonomous Snow Plow)
    - (10 Min Demo, 5 Min Code Review)
- Final Report (December 9th, 2022)
    - (Control Charts, System/Costumer Testing Results, Working System Code)

2.0 Scope

2.1 List of Requirements

2.1.1 Physical Requirements

1. Arduino DUE Board
2. Arduino IDE application to code the system's functionality
3. Robot Including Plow Maximum Size
    - Width x Length x High =216 x 252 x 150 mm
4. 4 Motors
5. 2 Ultra Sonic Distance Sensors
6. 1 Line Detector Sensor
7. Testing Zone Space
    - Approximately 2.5 x 2.5 meters square
8. Snow Balls (golf balls):
    - Several 42.6mm diameter balls

2.1.2 Project Requirements

9. A speed limit of 30 cm/s
10. Maximum allowable operation time
    - 5 mins, to move the most number of Snowballs out of zone
11. When an obstacle is detected, the robot shall adjust its path accordingly within the testing area to avoid collision
    - Move forward
    - Backward
    - Left and Right (90-degree turn)
    - 180-degree turn
    - Curved driving (left and right)
    - Hard stopping
12. The robot shall clear the most number of snowballs, by moving them out of the testing area in the fastest and safest way
13. When the area outline is detected, the robot shall change direction to stay inside the specified arena zone (black tape zone)

2.2 List of Activities

There are 5 main categories of activities that we have defined for our development. The vehicle body category represents the physical additions we must make to our robot chassis. The protocols define how we will design the overall movement decisions of our system. The movements themselves are a category as they must be programmed. The sensors category is responsible for programming the sensors in a way that we can use them to accomplish the requirements. Lastly, the testing category is of course responsible for conducting our end-to-end tests.

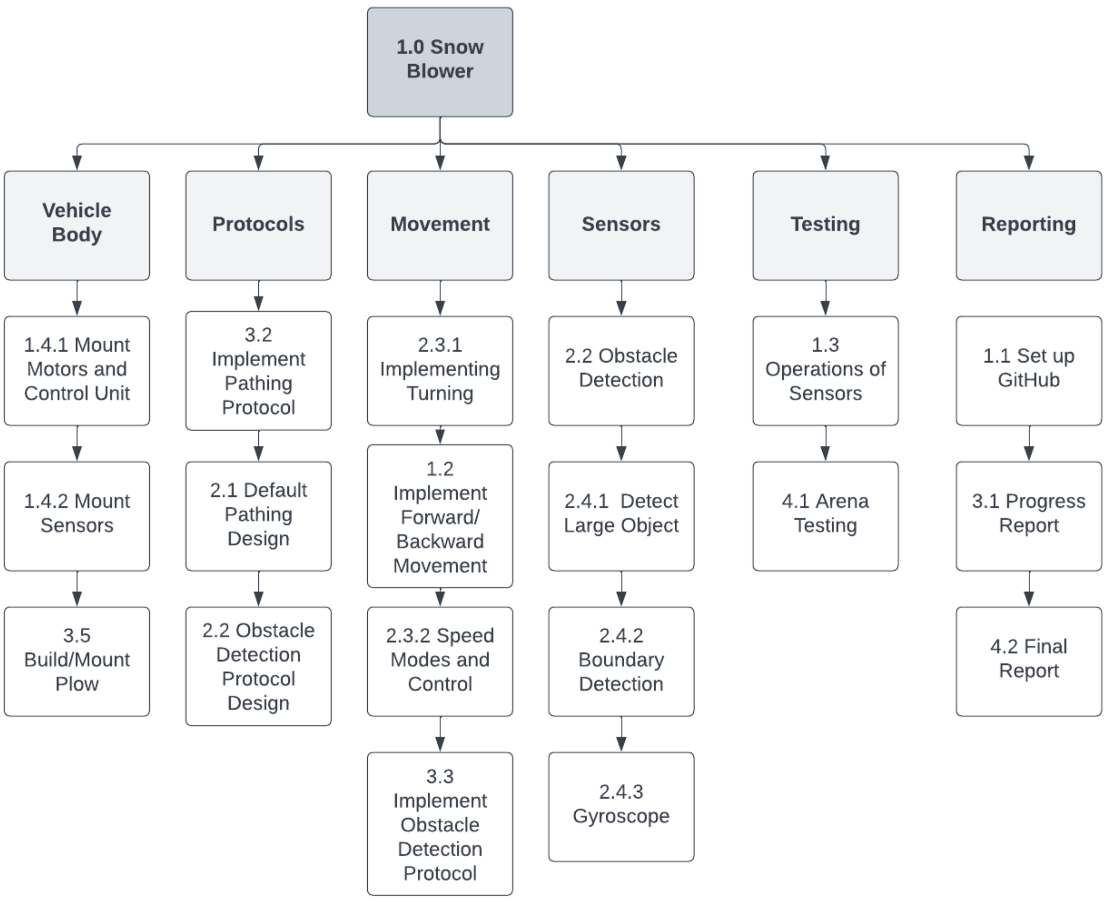

Figure 1: List of activities organized in a WBS

2.3 Testing Plan

The testing of our solution shall be done in a variety of stages, individual unit tests will be conducted on the sensors and motor functionality to ensure the base functions work as expected. Once those are conducted, integration tests will combine aspects together to ensure they can work in unison. Finally, an end-to-end test will be done to observe the functionality of the entire system against its objective.

2.3.1 Unit Tests

A variety of unit tests will have to be run to ensure the basic functionality of each component, these include:

- Testing each distance sensor at a variety of ranges. The success criteria will be if the sensors reading is accurate to the measured distance within ± 2cm
- Testing each motor to ensure it can move backward and forward. Each motor will be run, it will pass the test if the motor is observed to run.

2.3.2 Integration Tests

Integration tests will be conducted in order to gauge individual pieces abilities to begin functioning the way they are intended, these include:

- Testing a variety of movements made by the robot
  - 90 degree turn (left and right)
  - 180 degree turn
  - Straight line driving
  - Curved driving (left and right)
  - Hard stopping
- Testing the detection with the sensors
  - Testing detection of the outer barrier
  - Testing detection of obstacles

The success criteria of these tests will be if the robot can complete the specified action. For the movement tests, if the robot successfully performs the action, it will pass the test. For the sensor tests, we will put obstacles in front of it and take them away, if it detects the object while in front of it, and no longer when the obstacle moves, then the tests pass.

2.3.3 End to End Test

Final tests should be implemented to observe the robot completing its desired functionality, these include:

- Avoiding large stationary and mobile objects
- Moving golf balls out of the area
- Staying within the boundaries of a specified area

The success criteria of these is judged based on the final demo, the success is not a simple pass or fail, but more so a sliding scale.

3.0 Schedule

In this section the rough schedule for how we will develop our solution can be seen.

3.1 Schedule Network Diagram

Below is the schedule network diagram. See section 5.1 figure 1 for which activity corresponds to what activity number.

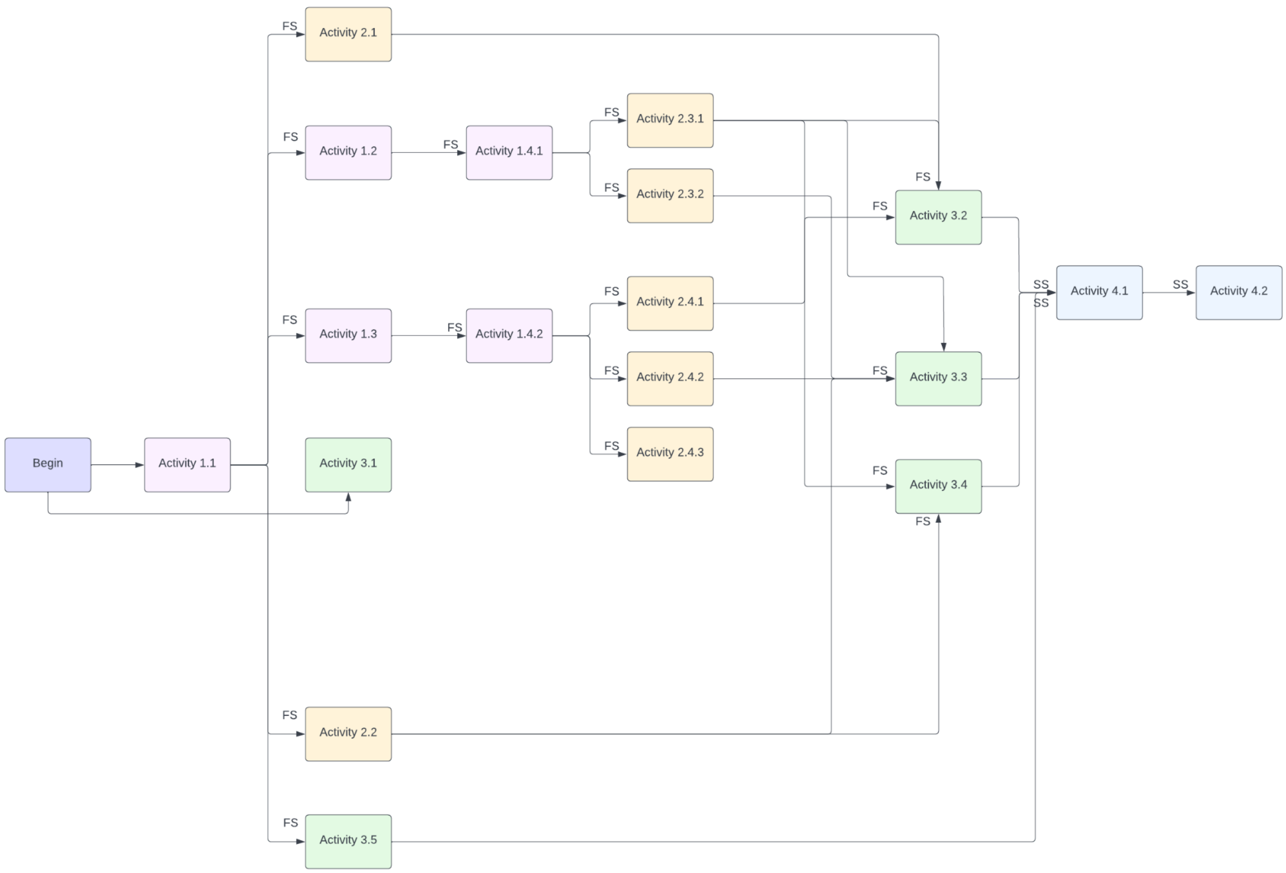

Figure 2: Schedule network diagram showing how each activity is connected

3.2 Gantt Chart

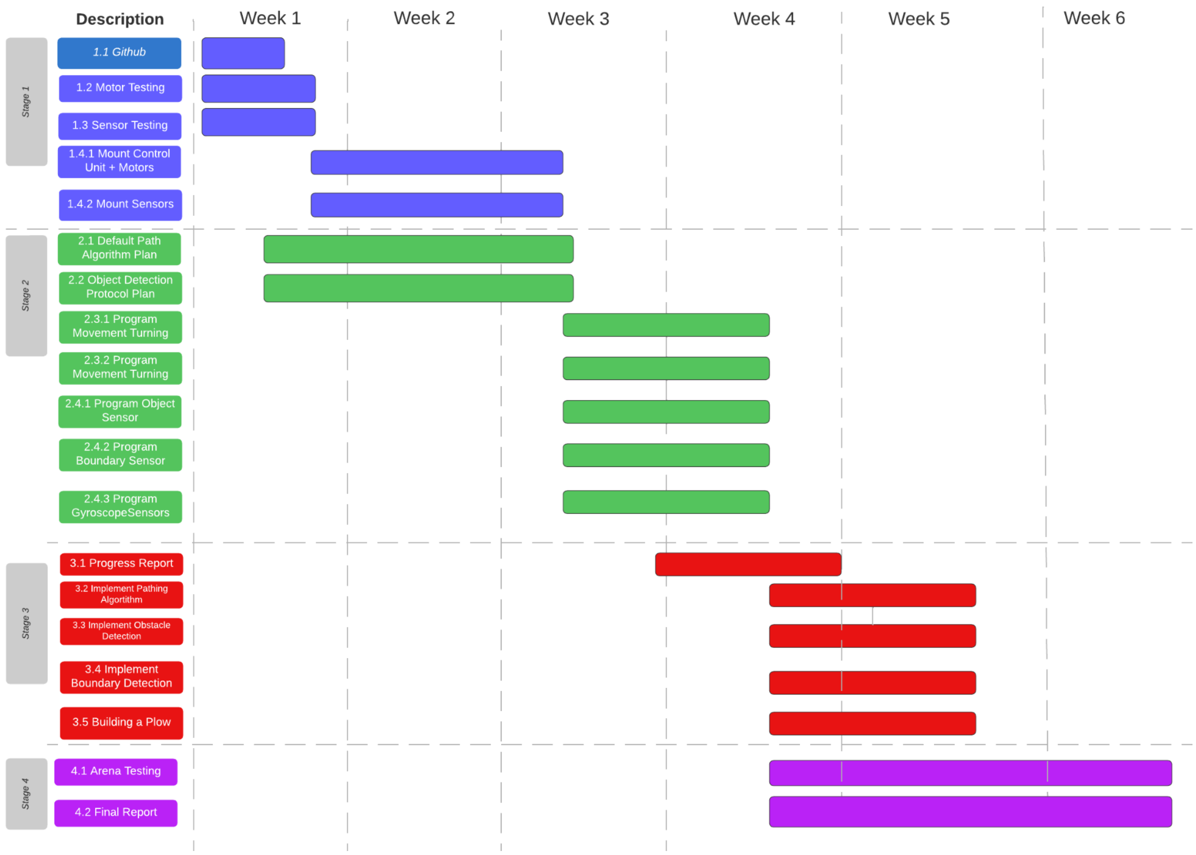

Figure 3: Gannt chart to show the approximate time required for each activity

4.0 Cost

4.1 Cost Description:

With 5 weeks of actual work, for the first 3 weeks, each member will work the 4 hours of the lab and 2 hours extra. For the last 2 weeks, each member will work the 4-hours of the lab and 4-hours extra. Each member is getting paid $50/hour.

4.2 Calculation of costs

Starting October 20th - November 24th, 2022

1. We had 6 weeks of working sessions (including reading week), for a group of three working members:
2. Weeks 1,3,4, each consisting of 18 hours, and weeks 5, 6, each consisting of 24 hours
3. Week 1,3,4: 6 hours/member x 3 members = 18 hours x 3 weeks = 54 hours for the group total
4. Week 5, 6: 8 hours/member x 3 members = 24 hours x 2 weeks = 48 hours for the group total
5. Weeks (1, 2, 3, 4, 5, 6) total group hours = 54 hours + 48 hours = 102 total hours for 3 members
6. 102 total hours for 3 members = 102/3 = 34 total hours/member
7. Each member's pay is $50/hour: Total Pay = 34 hours x $50 = $1700/member
8. Total pay on the project: $1700 x 3 members = $5100

4.3 Cost Baseline Table

Table 1: Approximate costs for our project

Week	Total Hours For Group	Total Accumulative Cost
1 (October 20)	18 (lab-4hrs, extra-2hrs)	$900
2 (Reading Week)	0	$900
3 (November 3rd)	18 (lab-4hrs, extra-2hrs)	$1800
4 (November 10th)	18 (lab-4hrs, extra-2hrs)	$2700
5 (November 17th)	24 (lab-4hrs, extra-4hrs)	$3900
6 (November 24th)	**24 (lab-4hrs, extra-4hrs)**	**$5100**

Cost Value Formula: Hourly Rate X # of Hours they'll work per person/week

4.4 Cost Baseline Graph

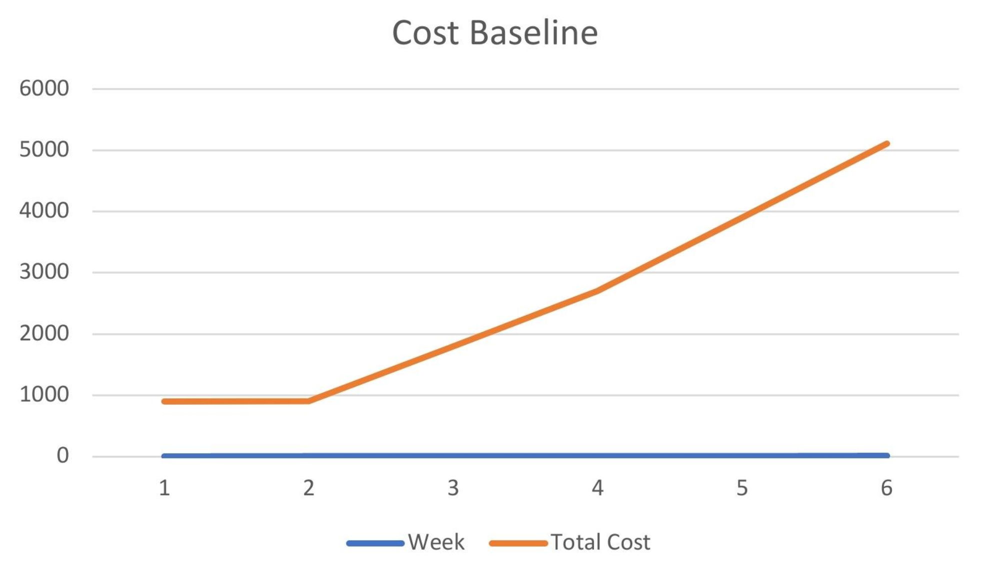

Figure 4: Costs organized into a line chart with time in weeks on the x-axis and cost in dollars on the y-axis

4.5 Planned Value Analysis Figure

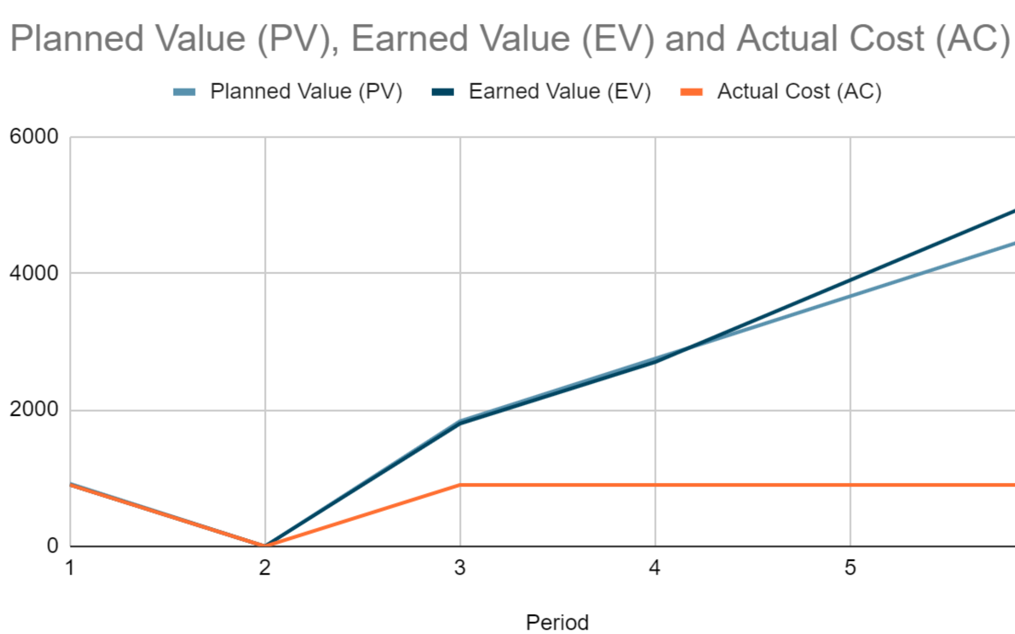

Figure 5: Planned Value Analysis Figure with time in weeks on x-axis and cost in dollars on y-axis

Table 2: Planned Value Analysis Table

Period	Planned Value (PV)	Earned Value (EV)	Actual Cost (AC)
1	916	900	900
2	0	0	0
3	1832	1800	900
4	2748	2700	900
5	3664	3900	900
6	4580	5100	900

Planned Value (PV) - The approved, time-phased budget allotted for carrying out the planned task

PV = 5500

Earned Value (EV) - Sometimes referred to as Budgeted Cost of Work Performed (BCWP), is the portion of the assignment that is actually finished.

PV = % Complete (Planned) x Task budget

PV = 75% x 5500 = 4125

Actual Cost (AC): It is sometimes referred to as Actual Cost of Work Performed (ACWP) and represents the true cost of the task, including every price required to perform it despite any difficulties. Included in it ought to be labor, supplies, tools, contractors, and other fixed costs.

Total Actual Cost = 5100

5.0 Responsibilities

With there being a variety of activities to complete, they must be distributed amongst the group members so that there is approximately 1 activity per group member per week. Below the assignment for these activities can be seen.

5.1 Human Resources:

Table 3: Responsibility assignment matrix

Activity Number	Activity	Faris	Flynn	Laith
1.1	Set up the GitHub	**R**	**A**	
1.2	Testing the operation of the motors (Forwards and backward)		**R**	**A**
1.3	Testing the operation of chosen sensors	**A**		**R**
1.4.1	Mounting motor controller with motors	**R**	**A**	
1.4.2	Mounting sensors on car		**R**	**A**
2.1	Default pathing of the final operation (Motion plan)	**A**		**R**
2.2	Obstacle detection protocol	**R**	**A**	
2.3.1	Program turns (90 degrees right and left, 180 degrees, slant left and right)		**R**	**A**
2.3.2	Program movement (Straight at a variety of speeds, hard stopping)	**A**		**R**
2.4.1	Detect large (not golf ball) object	**R**	**A**	
2.4.2	Detect boundary Sensor		**R**	**A**
2.4.3	Program gyroscope to create an internal map	**A**		**R**
3.1	Progress Report	**R**	**R**	**R**
3.2	Implement Pathing	**R**	**A**	
3.3	Implement Obstacle Detection		**R**	**A**
3.4	Implement Boundary Detection	**A**		**R**
3.5	Building a plow	**R**	**A**	
4.1	Arena Test	**R**	**R**	**R**
4.2	Final Report	**R**	**R**	**R**

(R- Responsible, A- Approver)

6.0 Overall System Architecture

The overall architecture of the system can be broken down into several pieces: The sensors used, the robots movement and the main driving algorithm.

6.1 Sensors

The sensors we are employing in our design are two ultrasonics, mounted in the front left and front right, both angled slightly outward by approximately 30 degrees. Then we are using the IR sensor right in the middle front of the vehicle. Below the front of the car we are also using one lane follower sensor. See the images below for reference:

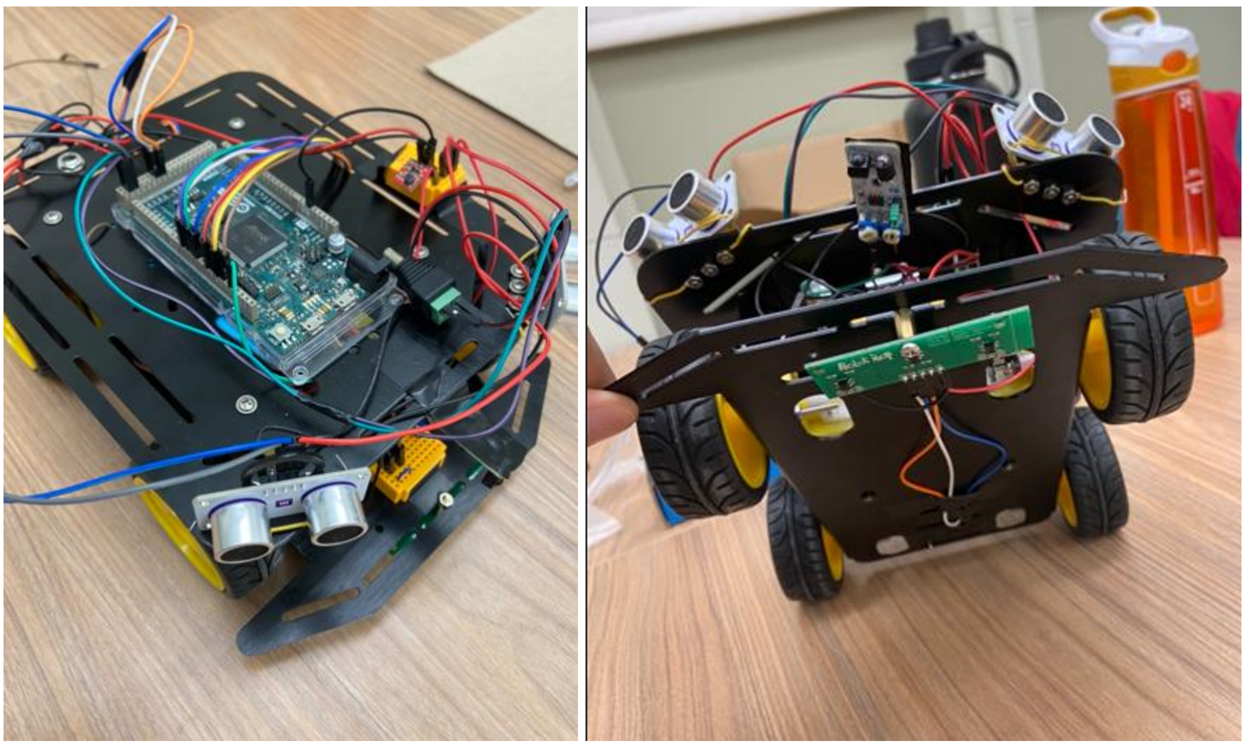

Figure 6: Images of our current robot implementation

6.2 Robot Movement

The movement of the robot has been defined by a variety of functions. The main movement operations we are employing are:

- Driving Forward
    - 4 wheel drive with all motors going forward
- Driving Backward
    - 4 wheel drive with all motors going backward
- Stopping
    - All wheels shut off
- Turning (Left and Right)
    - Both wheels of one side driving forward, the other two driving backwards. This causes the robot to turn in place.

These operations are defined very simply. One important thing to note is the signals we are sending to our motors have a relatively low duty cycle. This is because we believe that driving slowly allows our sensors and main loop more time to react and adjust to obstacles than if driving fast. Since the demo allows 5 minutes, we believe there is no rush, and thus are driving slow.

6.3 Main Driving Algorithm

The main driving algorithm that will be implemented is as follows. The robot will drive forward until the sensors detect something close by. The way we do this for the ultrasonic sensors is by giving each sensor a threshold. The threshold values are still being tested but the two thresholds we have are labeled minDistance and stopDistance. The minDistance is the first threshold we pass, if our reading from an ultrasonic is less than this minimum distance, an object is close by. If we detect an object close, we will begin turning away from that object until the distance is greater than minDistance. However if the sensor detects that an object is less than our defined stopping distance, we are too close to an object to properly turn. So in this case, we must drive backward a tiny bit before we begin turning. We will repeat this backward and then turn cycle until the ultrasonic distance value is no longer below the threshold.

The ultrasonics do most of the work in this regard, but we also employ a front IR sensor that aids in our detection. The IR sensor returns low if an object is within its programmed threshold. In our case, it is approximately 5 cm. If the IR sensor is low, it means that we have detected an object that is way too close. In this case, we employ the same measures as for when an ultrasonic detects something within its stop distance. It will back up, and then make a decision to turn either left or right based on which ultrasonic reading is higher (meaning we have more room that way). See the state diagram below for a visual representation of this algorithm.

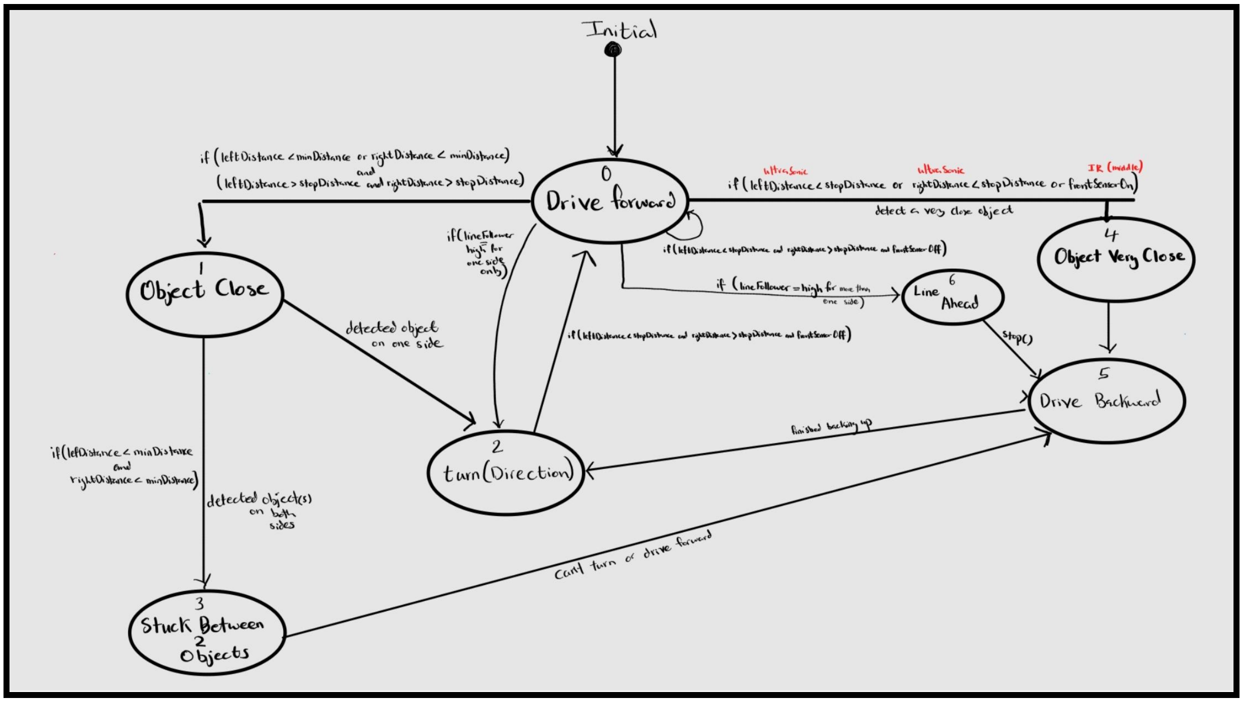

Figure 7: State Diagram of our driving algorithm

6.3.1 Watchdog Timer

We implemented a watchdog timer in our code which can be seen on the github page in the snowPlower.ino file. This watchdog timer simply resets at the end of our main loop. It is in place in case for some reason something executes infinitely, and we need to hard abort the system.

6.3.2 Unfinished Parts of the Algorithm

The last piece of sensor input we react to is the line follower. We currently haven't implemented that piece but it can still be conceptualized. We would do the same drive backward and turn cycle discussed above, but would do it in reaction to the lane follower detecting a 1. We would turn based on which of the 3 individual sensors is detecting the black line.

6.3.3 Sequence Diagram

Below you can find the sequence diagram of how the whole system will interact:

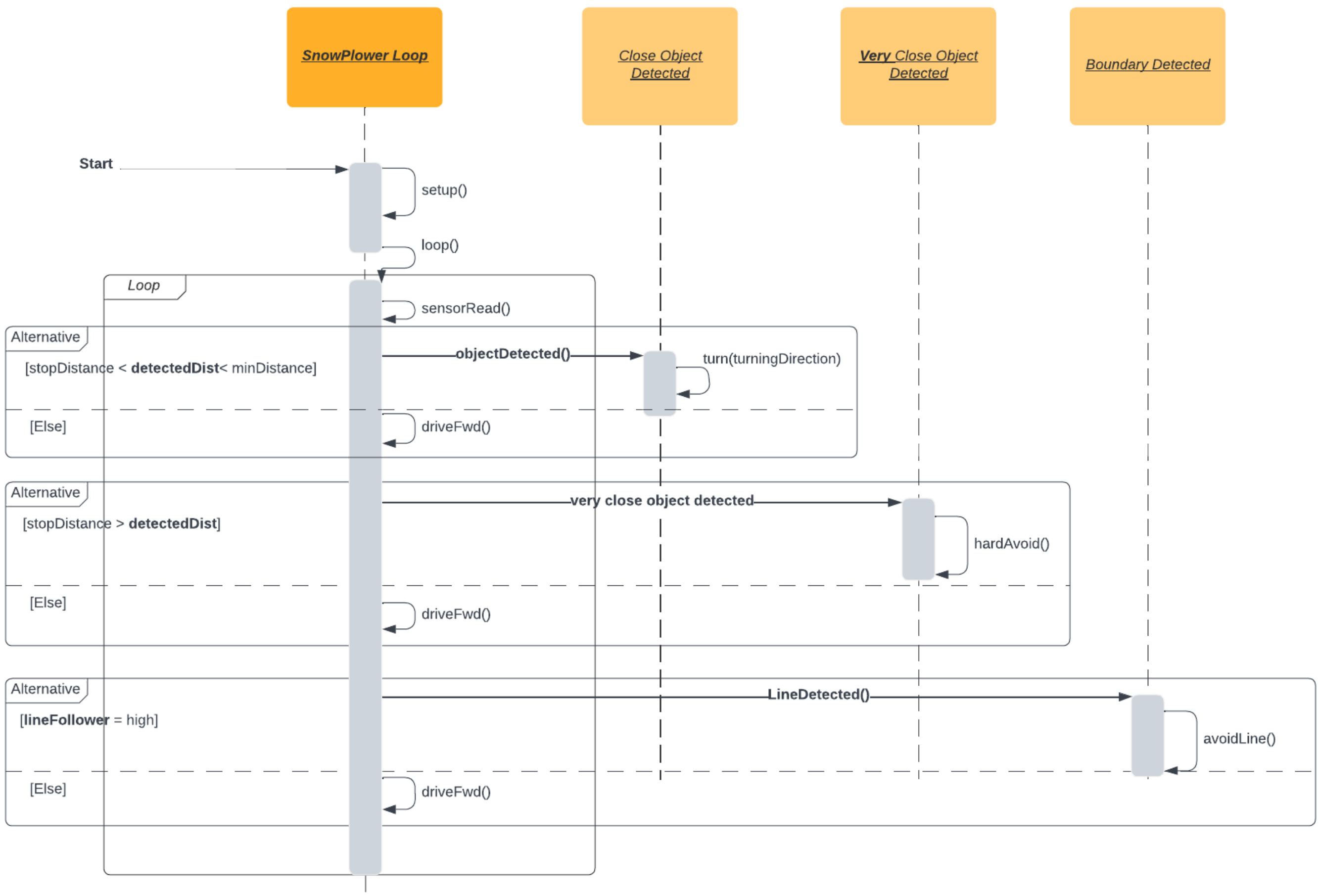

Figure 8: Sequence Diagram for the Snowplow Robot

7.0 Github

Below is the link to our project repo:

https://github.com/LaithRazeq/SYSC4805