SYSC 4805 - Computer Systems Design Lab

Progress Report

Instructor: Mostafa Taha
Lab Section: L1
Group: 8

Team Name: Egyptian Blue

Date Completed: November 18th, 2022

Group Members:
Ahmed Abdelrazik - 101103048
Colton North - 101081850
Boshen Zhang - 101151224

Table of contents

1.0 Project Charter

1.1 Overall Objective

The overall objective of this project is to create a robot that is capable of clearing a $6m^2$ space of snowballs (plastic balls with diameters of 42.6mm). The robot utilizes an ultrasonic sensor to avoid both stationary and moving obstacles that are directly ahead of the robot. Line following sensors at the front left and front right of the robot ensure that it stays within the designated 2.5 x 2.5 meter area. Two time of flight (ToF) sensors are equipped at the front left and front right of the robot to detect any obstacles that are at 10 and 2 o'clock which the ultrasonic sensor may miss. An Arduino Due board uses the information gathered from said sensors, and informs the motor driver board to control the four motors of the robot accordingly. These motors do not exceed a maximum speed of 30cm/s so that the robot has ample time to react to obstacles in its path. The snowballs are pushed out of the testing arena with the use of a plow at the front of the robot. The end goal is to have the robot remove all of the snowballs out of the testing arena by the time 5 minutes have elapsed, while not hitting any obstacles while doing so.

1.2 Overall Deliverables

Deliverables & Milestones	**Due Date**
Project Proposal	October 14th
Movement Test	November 3rd
Progress Report	November 11th
Sensor Integration	November 17th
Mock Test	November 24th
In-Class Presentation	November 28th
In-Lab Demo	December 1st
Final Report	December 9th

Table #1: The deliverables, milestones and corresponding due dates.

1.3 Overall Architecture

The main target for this project is to implement the Arduino robot driving autonomously and reasonably avoiding obstacles within the specified time and specific area. In this case, there are five sensors applied including an Ultrasonic Distance Sensor, two Line Follower Sensors and two VL53L1X Time-of-Flight Distance Sensors. In general, the Ultrasonic Distance Sensor is placed at the middle front of the robot, its function is to detect whether there are obstacles (box) on the forward route of the car. If the distance between the car and the obstacle (in front) reaches a special value, such as 15cm, the motor drive board will receive signals and adjust the direction of the car according to different conditions. The Line Follower Sensors is used to detect whether the robotic vehicle reaches the boundary of the test arena since it can distinguish between white and black, so that the boundary line (black tape) can be detected according to this property. Once the sensor outputs “B”, which means black, the motor drive board will take control of the robot vehicle because the car has approached the black boundary. Time-of-Flight Distance Sensors are similar to the Ultrasonic Distance Sensor, they are also involved in distance measuring between obstacles and robot but left front and right front. Because sometimes obstacles may appear in the oblique front of the car, thus it is not accurate to judge obstacles only by the sensor in the center of the robotics. The figure below shows the locations of the sensors on the robot.

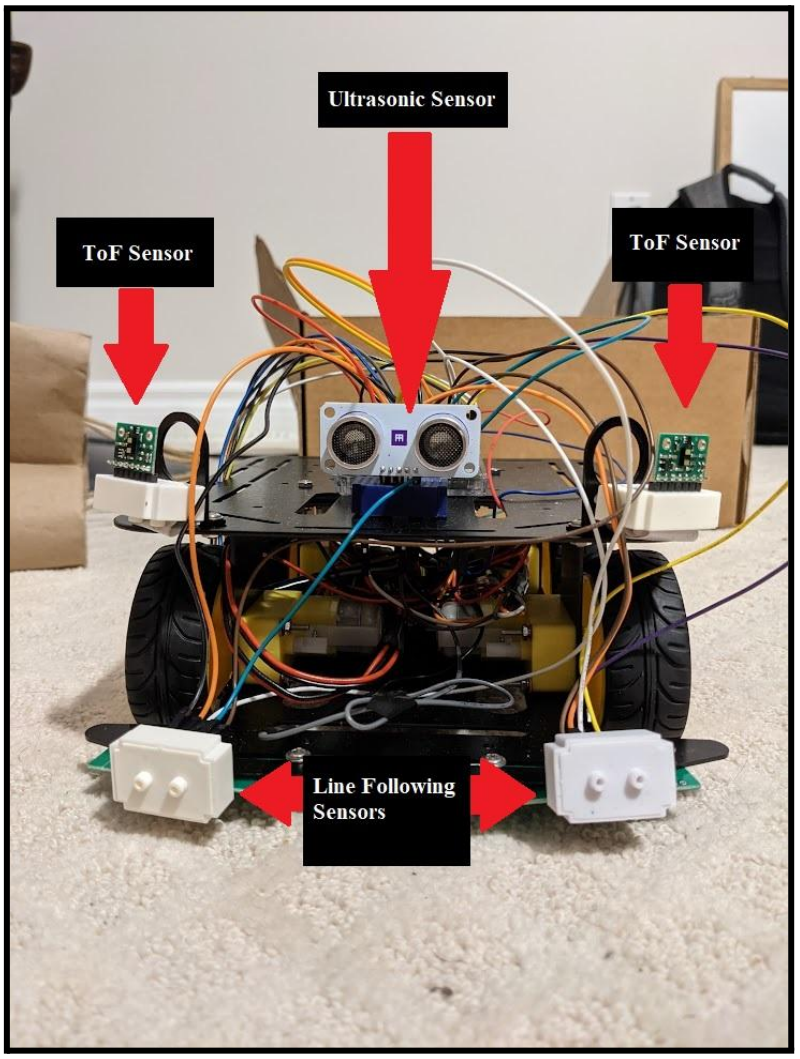

Figure #1: The sensors used in the snow remover robot.

	ULT	LToF	RToF	LLF	RLF	Situation	Motor Reaction
1	0	0	0	0	0	N	BR↑, BL↑
2	0	0	0	0	1	B	FR↑, BR↑, BL↑
3	0	0	0	1	0	A	FL↑, BR↑, BL↑
4	0	0	0	1	1	C	BR↓, BL↓ => BR↑, FL↑
5	0	0	1	0	0	B	FR↑, BR↑, BL↑
6	0	0	1	0	1	B	FR↑, BR↑, BL↑
7	0	0	1	1	0	D	BR↓, BL↓ => FL↑, BR↑, BL↑
8	0	0	1	1	1	C	BR↓, BL↓ => BR↑, FL↑
9	0	1	0	0	0	A	FL↑, BR↑, BL↑
10	0	1	0	0	1	E	BR↓, BL↓ => FR↑, BR↑, BL↑
11	0	1	0	1	0	A	FL↑, BR↑, BL↑
12	0	1	0	1	1	C	BR↓, BL↓ => BR↑, FL↑
13	0	1	1	0	0	DNE	Situation Do Not Exist
14	0	1	1	0	1	DNE	Situation Do Not Exist
15	0	1	1	1	0	DNE	Situation Do Not Exist
16	0	1	1	1	1	DNE	Situation Do Not Exist

- Ultrasonic Distance Sensor —— ULT
- VL53L1X Time-of-Flight (Left) —— LToF
- VL53L1X Time-of-Flight (Right) —— RToF
- Line Follower Sensor (Left) —— LLF
- Line Follower Sensor (Right) —— RLF

- “0” —— “OFF”
- “1” —— “ON"

- Situation A —— Turn Right
- Situation B —— Turn Left
- Situation C —— Reverse 1 Second, Then U Turn (Towards Right)
- Situation D —— Reverse 1 Second, Then Turn Right
- Situation E —— Reverse 1 Second, Then Turn Left
- Situation N —— Go Straight

- BR —— Back Right Wheel
- BL —— Back Left Wheel
- FR —— Front Right Wheel
- FL —— Front Left Wheel

Table #2: Truth table for sensors and corresponding action taken by motors.

	ULT	LToF	RToF	LLF	RLF	Situation	Motor Reaction
17	1	0	0	0	0	A	FL↑, BR↑, BL↑
18	1	0	0	0	1	B	FR↑, BR↑, BL↑
19	1	0	0	1	0	D	BR↓, BL↓ => FL↑, BR↑, BL↑
20	1	0	0	1	1	C	BR↓, BL↓ => BR↑, FL↑
21	1	0	1	0	0	B	FR↑, BR↑, BL↑
22	1	0	1	0	1	B	FR↑, BR↑, BL↑
23	1	0	1	1	0	A	FL↑, BR↑, BL↑
24	1	0	1	1	1	C	BR↓, BL↓ => BR↑, FL↑
25	1	1	0	0	0	B	FR↑, BR↑, BL↑
26	1	1	0	0	1	E	BR↓, BL↓ => FR↑, BR↑, BL↑
27	1	1	0	1	0	A	FL↑, BR↑, BL↑
28	1	1	0	1	1	C	BR↓, BL↓ => BR↑, FL↑
29	1	1	1	0	0	DNE	Situation Do Not Exist
30	1	1	1	0	1	DNE	Situation Do Not Exist
31	1	1	1	1	0	DNE	Situation Do Not Exist
32	1	1	1	1	1	DNE	Situation Do Not Exist

- Ultrasonic Distance Sensor —— ULT
- VL53L1X Time-of-Flight (Left) —— LToF
- VL53L1X Time-of-Flight (Right) —— RToF
- Line Follower Sensor (Left) —— LLF
- Line Follower Sensor (Right) —— RLF

- “0” —— “OFF”
- “1” —— “ON"

- Situation A —— Turn Right
- Situation B —— Turn Left
- Situation C —— Reverse 1 Second, Then U Turn (Towards Right)
- Situation D —— Reverse 1 Second, Then Turn Right
- Situation E —— Reverse 1 Second, Then Turn Left
- Situation N —— Go Straight

- BR —— Back Right Wheel
- BL —— Back Left Wheel
- FR —— Front Right Wheel
- FL —— Front Left Wheel

Table #3: Truth table for sensors and corresponding action taken by motors.

The two tables above display the robot’s reaction under different situations. There are 32 possibilities in total, however, they can be reduced to 5 + 1 scenarios. Directly turn right, directly turn left, reverse 1 second then U turn (towards right), reverse 1 second then right, reverse 1 second then left and go

straight, here we use ABCDE and N to represent them respectively. These 5 + 1 situations are sufficient for the testing requirements.

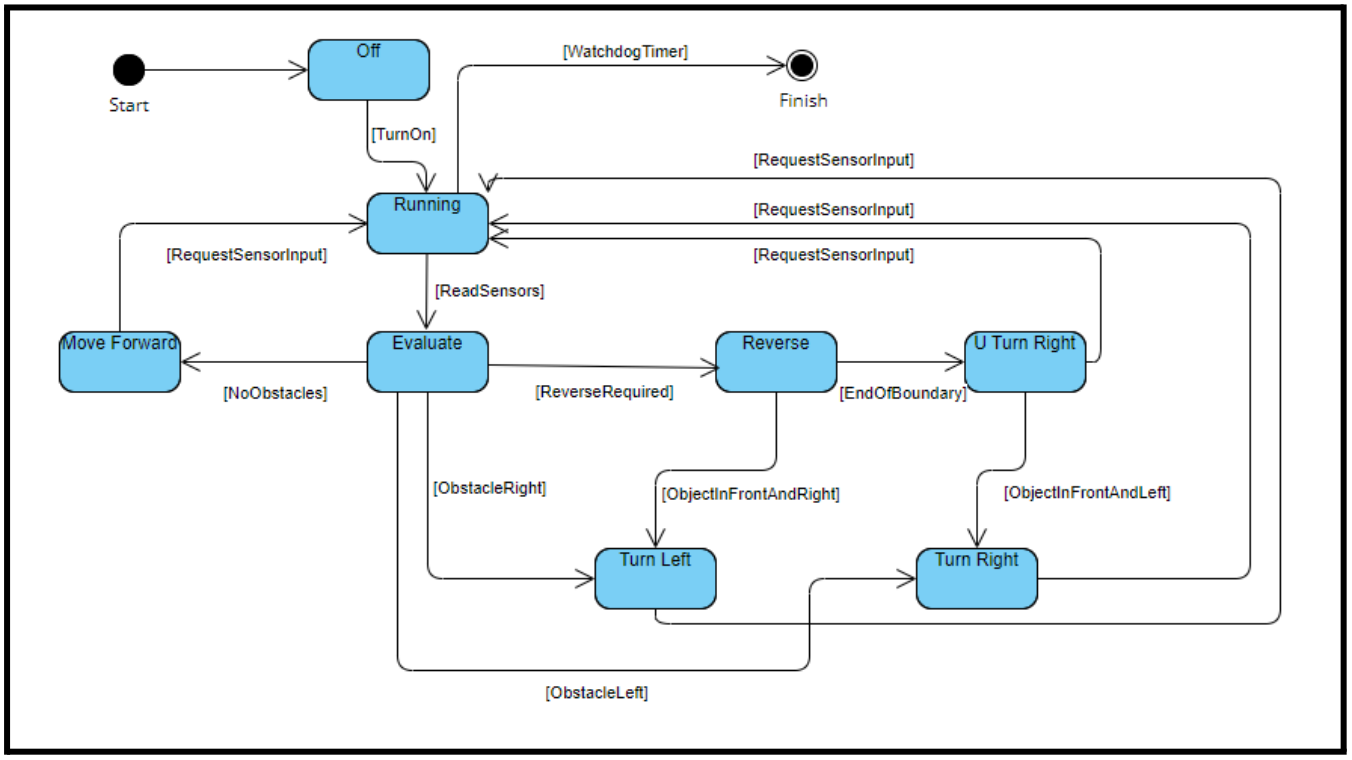

Figure #2: The state chart diagram of the system.

Above is the state chart diagram of the snow removal system. Initially the robot is turned off. Once turned on the robot will begin reading sensors to determine what it should do next. Then the robot begins evaluating what the sensors have read, and commands the motors to move accordingly. Should there be no obstacles detected from any of the sensors, the robot will proceed forward. If there is an obstacle to the right, the robot will turn left to avoid it and vice versa. If the robot needs to reverse, it will do so for one second, and then determine if it should turn left, right, or do a u turn right. After performing an action involving movement, the system goes back to request more information from the sensors to see if it needs to make more adjustments.

Finally, as can be seen above the watchdog timer should be implemented in our GitHub repository. The main purpose of a watchdog timer is to make sure the device performs a hard reset in case the hardware is stuck in an infinite loop. While in the running state, if the watchdog timer is ever set; meaning that the robot is stuck in an infinite loop the hardware will just turn off and turn on again and start operating normally once again.

2.0 Scope

2.1 List of Requirements

1. When a black line is detected by the left line sensor, the robot shall set the motors to turn right until the black line is no longer detected.
2. When a black line is detected by both line following sensors, the robot shall set the motors to reverse for 1 second, and then proceed to turn right.
3. When a black line is detected by the right line sensor, the robot shall set the motors to turn left until the black line is no longer detected.
4. When the left ToF sensor detects an object within 15cm, the robot shall set the motors to turn right until the object is no longer detected.
5. When the right ToF sensor detects an object within 15cm, the robot shall set the motors to turn left until the object is no longer detected.
6. When the ultrasonic sensor detects an object within 15cm, the robot shall set the motors to turn right until the object is no longer detected.
7. When no objects are detected, as well as no black lines are detected, the robot shall set the motors to move forward.
8. The robot shall move at a speed no greater than 30cm/s.
9. When 5 minutes have passed, the robot shall set the motors to stop moving.
10. The robot shall not exceed a maximum size of 216 x 252 x 150mm.

2.2 List of Activities

1. **Detection Strategy**
   - 1.1. Sensor Positioning
   - 1.2. Obstacle Detection Algorithm
   - 1.3. Boundary Detection Algorithm
   - 1.4. Moving Around Obstacles Algorithm
2. **Movement Strategy**
   - 2.1. Starting Positon
   - 2.2. Speed Controls
   - 2.3. Turning Controls
   - 2.4. Breaking Controls
   - 2.5. Most Efficient Route Algortihm
3. **Robot Aassembly**
   - 3.1. Sensor Wiring
   - 3.2. Build & Mount Snow Plow
4. **Testing**
   - 4.1. Sensor Range
   - 4.2. Integrate Sensors
   - 4.3. Movement Testing (without obstacles)
   - 4.4. Mock Test
5. **Documents**
   - 5.1. Progess Report
   - 5.2. In Class Presentation
   - 5.3. In Lab Demo
   - 5.4. Final Report

2.3 Work Breakdown Structure

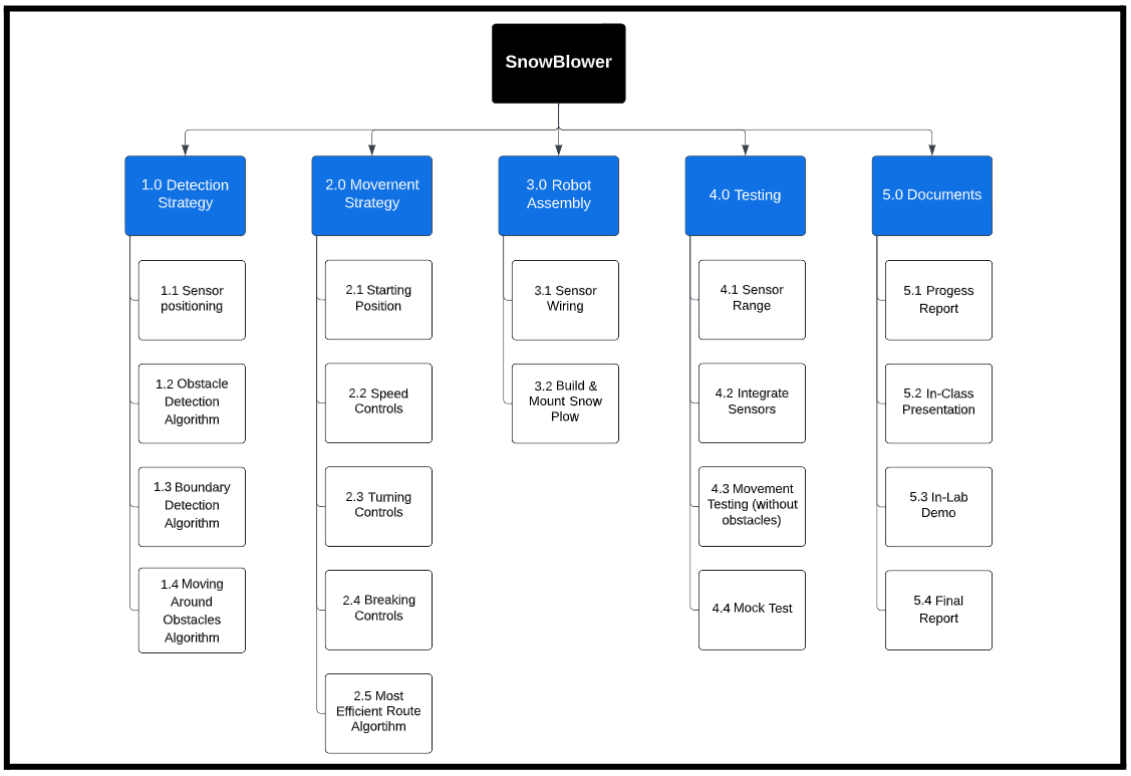

Figure #3: The work breakdown structure of the snow blower.

2.4 Testing Plan

Integration Test (Each Component Separately)

Item	Target	Success Criteria
VMA330 IR Obstacle Avoidance Sensor Module	Identify if the front of the car has approached the obstacles (four cartons).	The sensor can detect obstacles and send the signal to the computer if detected. Its LED can turn on meanwhile.
Line Follower Sensor	Identify test arena boundaries (black tape).	The sensor identifies the area directly below the car (the test arena or black border) and sends real-time data to the computer.
Ultrasonic Distance Senso	Identify obstacles (four cartons) and the distance between robotics and obstacles.	The sensor can send accurate measurements of the distance between the car and the obstacle in real time to the computer.
VL53L1X sensor	Detect the distance between the middle-left front (middle right-front) of the car and obstacles.	The two sensors can respectively detect obstacles within a specified distance (15 cm) on the middle-left and middle-right sides, and send signals to the computer.
MinIMU-9 v5 IMU Unit	Identify the rotation angle of the car for avoiding obstacles and boundaries of the test field.	The sensor can roughly send the angular difference before and after the car rotates, such as 180 degrees when it hits a boundary.
Wheel Encoders	Measure the real time speed of the robotic car.	The wheel encoders can calculate the speed of the car automatically by measuring the rotation speed of the wheel, and transmitted to the computer in real time.
Motor Driver Board	Drive the motor to ensure data communication between the motor and the computer.	The rear wheels on both sides of the car can control its speed and direction separately, and the computer can obtain more accurate real-time data.

Table #4: The testing plan for each component of the robot.

System Test (With Arduino)

Item	Target	Success Criteria
VMA330 IR Obstacle Avoidance Sensor Module	Notify the Arduino to stop when obstacles on the front of the car are detected.	The car can stop when obstacles appear from the front, the computer shows the reason for stopping (obstacles detected).
Line Follower Sensor	Inform Arduino to turn right, left or 180 degrees when the car reaches the boundary of the test arena.	When the robot car encounters the black boundary, it should notice the computer and turn the direction of travel to ensure that the car does not leave the test site.
Ultrasonic Distance Senso	When the distance of the obstacle in front is less than 15cm, send stopping signals to Arduino.	When there is an obstacle within 15cm in front of the car, stop moving forward and display the real-time distance between the car and the obstacle.
VL53L1X sensor	When the distance between the car and the obstacles on the left or right is less than 15cm, inform Arduino to turn the direction.	When there is an obstacle less than 15cm away from the car on the left or right side, stop and change the moving direction to the side without the obstacle.
MinIMU-9 v5 IMU Unit	Notify Arduino to stop reversing after 1 second, and then proceed to turn right until no boundaries are detected..	When the car meets the boundary of the test field,it reverses for 1 second, then proceeds to turn right; When the car turns to the direction of clear road condition due to obstacles, stop rotating and continue to move forward.
Wheel Encoders	Inform Arduino that the car's two rear wheels rotated speed and the forward speed of robotics.	When the car speed is close to 30cm/s, a warning should be issued and the speed should be reduced to a reasonable value.
Motor Driver Board	Notify Arduino when the car needs to speed up, slow down, stop or turn.	The Ardinuo can change the speed of the car in real time, stop the car, and turn the car by controlling the speed difference of two rear wheels.
Robotics	Within 5 minutes, the car can push away the white plastic ball within the range of the test site without encountering obstacles.	When driving to the boundary of the test area, when encountering obstacles in front and the distance is less than 15cm, when the distance between obstacles on both sides is less than 15cm, the car needs to turn the driving direction.

Table #5: The testing plan for the system as a whole with the use of the Arduino.

3.0 Schedule

3.1 Schedule Network Diagram

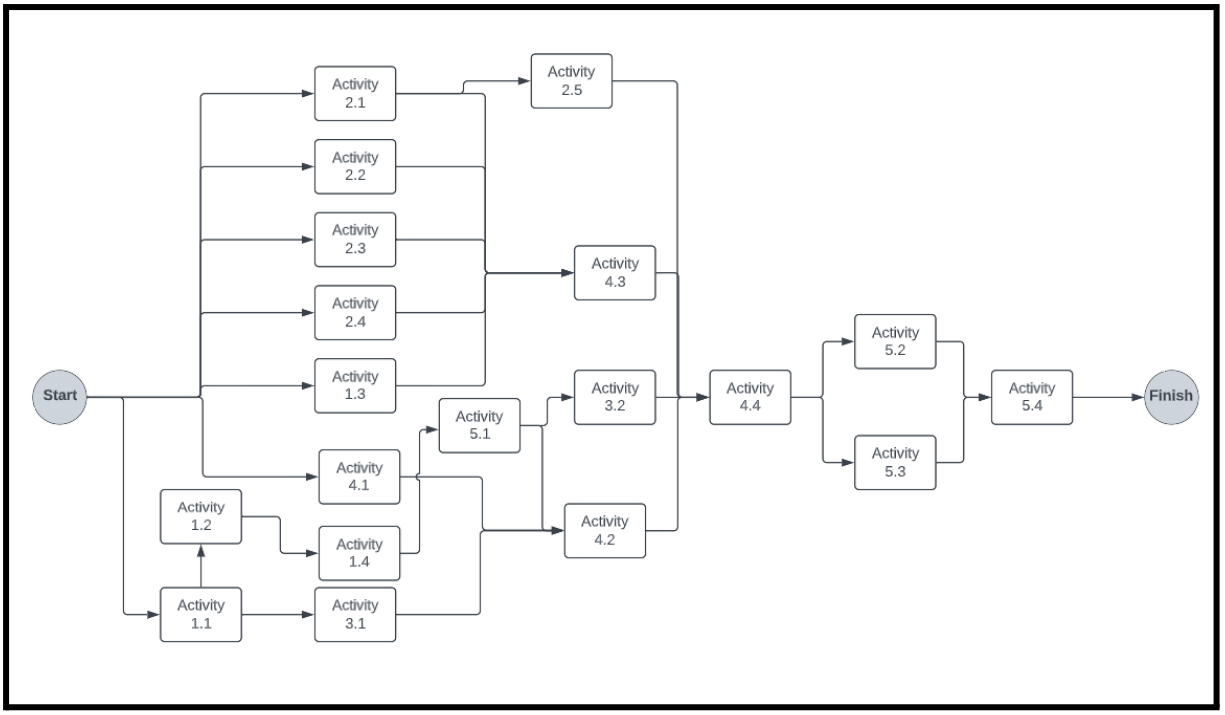

Figure #4: The Schedule Network Diagram displays the order in which activities are completed.

3.2 Gantt Chart

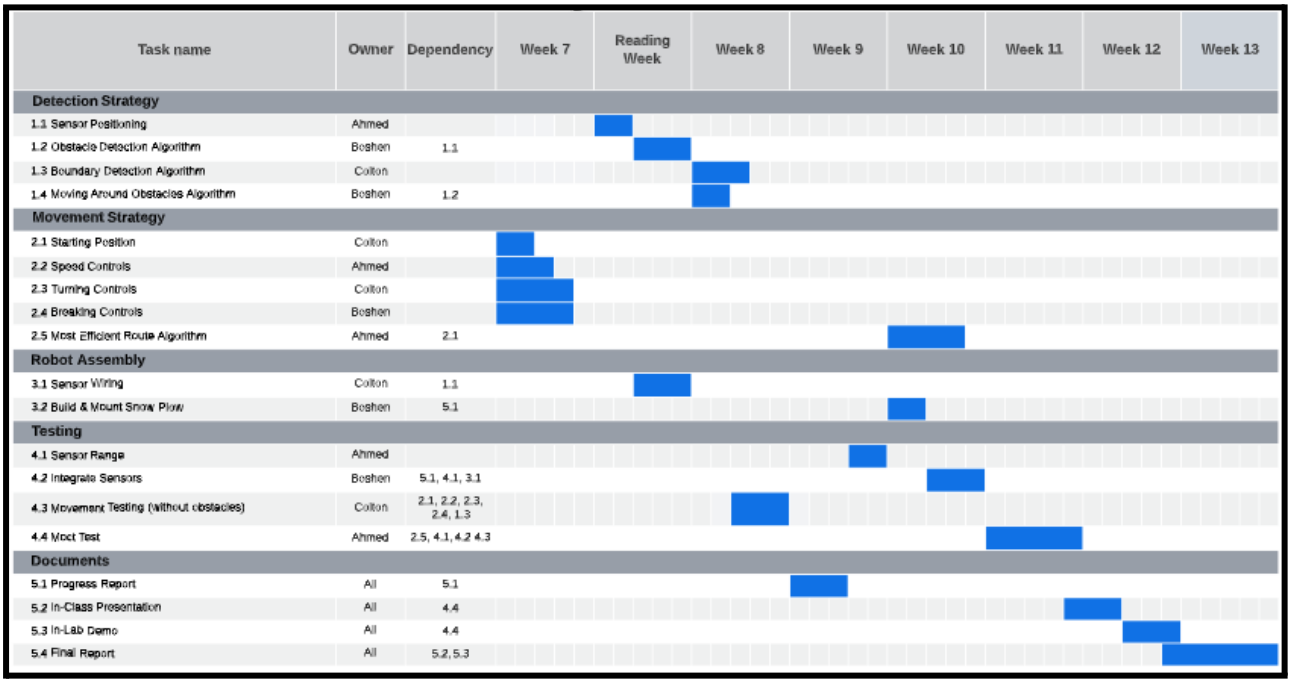

Figure #5: The Gantt Chart displays the time required to complete each activity throughout the duration of the remaining labs.

As seen above in the Gantt chart, the first week to start working on our project is week 7. During this week the starting position, speed controls, turning controls and breaking controls are to be completed. The following week is reading week which involves setting up the sensor positions, the obstacle detection algorithm, as well as the sensor wiring. Week 8 involves setting up the boundary detection algorithm, the obstacle avoidance algorithm, and then some movement testing. Week 9 is used mostly for the progress report, as well as some sensor range testing. Week 10 entails figuring out the most efficient route for the robot, the construction of the snow plow, and the final adjustments to the sensor integration. Week 11 involves a mock test to simulate the in-lab demo and necessary adjustments will be made. Week 12 is used for the in-class presentation, as well as the in-lab demo. Finally, week 13 involves writing the final report of the project.

4.0 Cost

4.1 Cost Control Plan

The expected timeline till the end of the project is 6 weeks. Each week should have 7 hours of work, 4 hours during the lab period and 3 extra hours. The second week is the reading week therefore the 4 hours of the lab section are missing. Moreover, Ahmed will be unavailable during weeks 4 and 5 hence the decrease in hours worked. The working hours recorded below is just an estimate and is subject to change if needed.

4.2 Cost Baseline Table

Week	Ahmed	Boshen	Colton	Total Hours	Cost	Accumlated Cost
1	7	7	7	21	$1050	$1050
2 (reading week)	3	3	3	9	$450	$1500
3	7	7	7	21	$1050	$2550
4	3	7	7	17	$850	$3400
5	3	7	7	17	$850	$4250
6	8	8	8	24	$1200	$5450

Table #6: The Cost Baseline Table displays the estimated work hours of each member.

4.3 Cost Baseline Figure

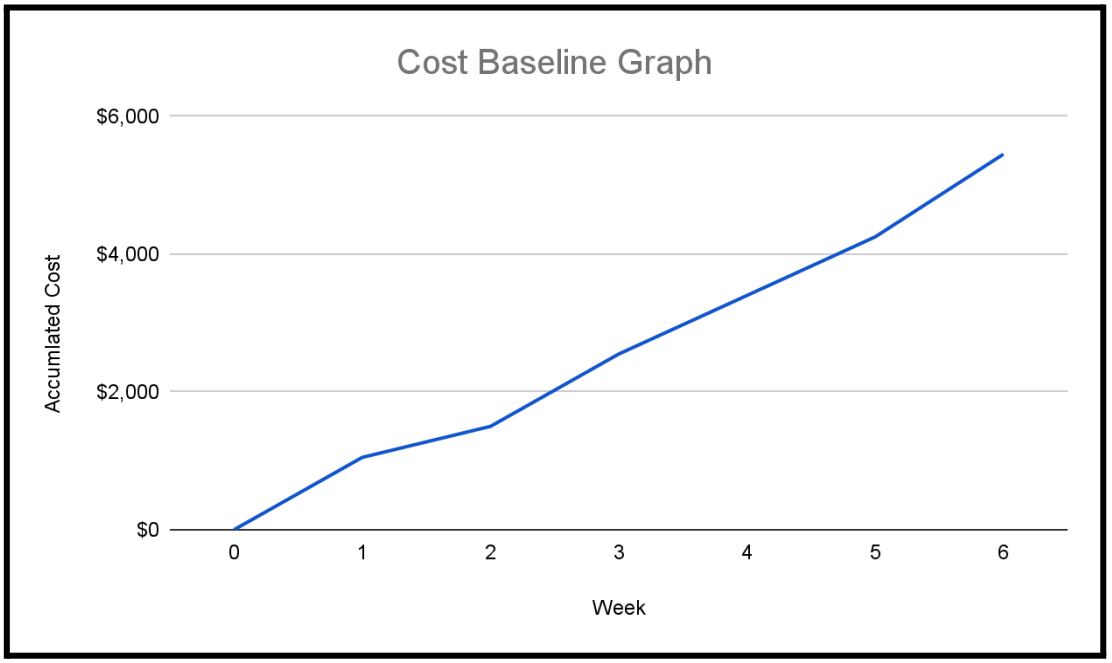

Figure #6: The Cost Baseline Graph depicts the accumulated cost over each week.

4.4 Planned Value Analysis

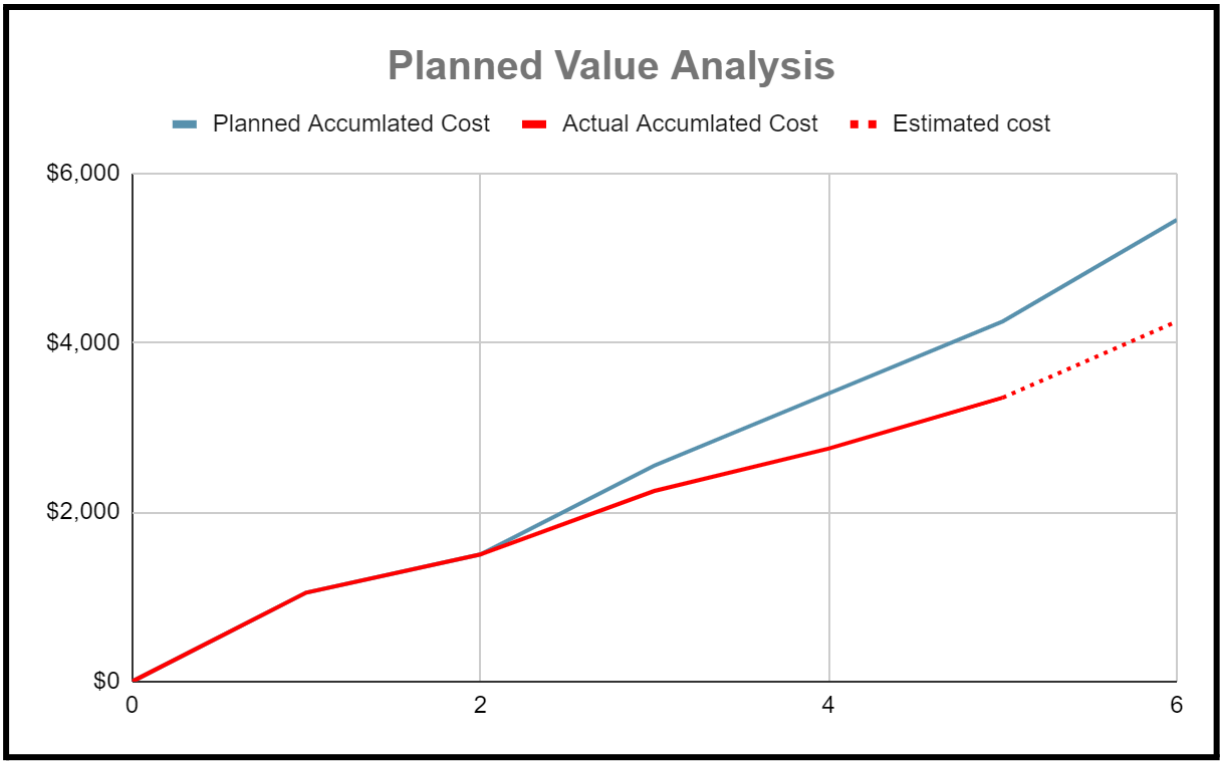

Figure #7: The Cost Baseline Graph depicts the accumulated cost over each week.

As can be seen from the analysis above, our estimation shows that we are behind schedule because actual cost, which is referring to our number of hours worked, is less than what was planned in the beginning. This is due to multiple reasons and roadblocks that we faced along the way. If we proceed with the same pace we will be behind according to our estimation. Therefore this shows us that we need to speed up as soon as possible in the last two weeks.

5.0 Human Resources

5.1 Responsibility Matrix

Activity		Ahmed	Boshen	Colton
1.0 Detection Strategy	1.1 Sensor positioning	Responible	Approver	
	1.2 Obstacle Detection Algorithm		Responible	Approver
	1.3 Boundary Detection Algorithm	Approver		Responible
	1.4 Moving Around Obstacles Algorithm		Responible	Approver
2.0 Movement Strategy	2.1 Starting Position	Approver		Responible
	2.2 Speed Controls	Responible	Approver	
	2.3 Turning Controls	Approver		Responible
	2.4 Breaking Controls		Responible	Approver
	2.5 Most Efficient Route Algortihm	Responible	Approver	
3.0 Robot Asembly	3.1 Sensor Wiring	Approver		Responible
	3.2 Build & Mount Snow Plow		Responible	Approver
4.0 Testing	4.1 Sensor Range	Responible	Approver	
	4.2 Integrate Sensors		Responible	Approver
	4.3 Movement Testing (without obstacles)	Approver		Responible
	4.4 Mock Test	Responible	Approver	
5.0 Documents	5.1 Progess Report	Responible	Responible	Responible
	5.2 In-Class Presentation	Responible	Responible	Responible
	5.3 In-Lab Demo	Responible	Responible	Responible
	5.4 Final Report	Responible	Responible	Responible

Table #7: The Responsibility Matrix displays who is responsible for each activity in the project.

6.0 GitHub

Below is an image of the commits to our GitHub repository over the duration of our project until this point.

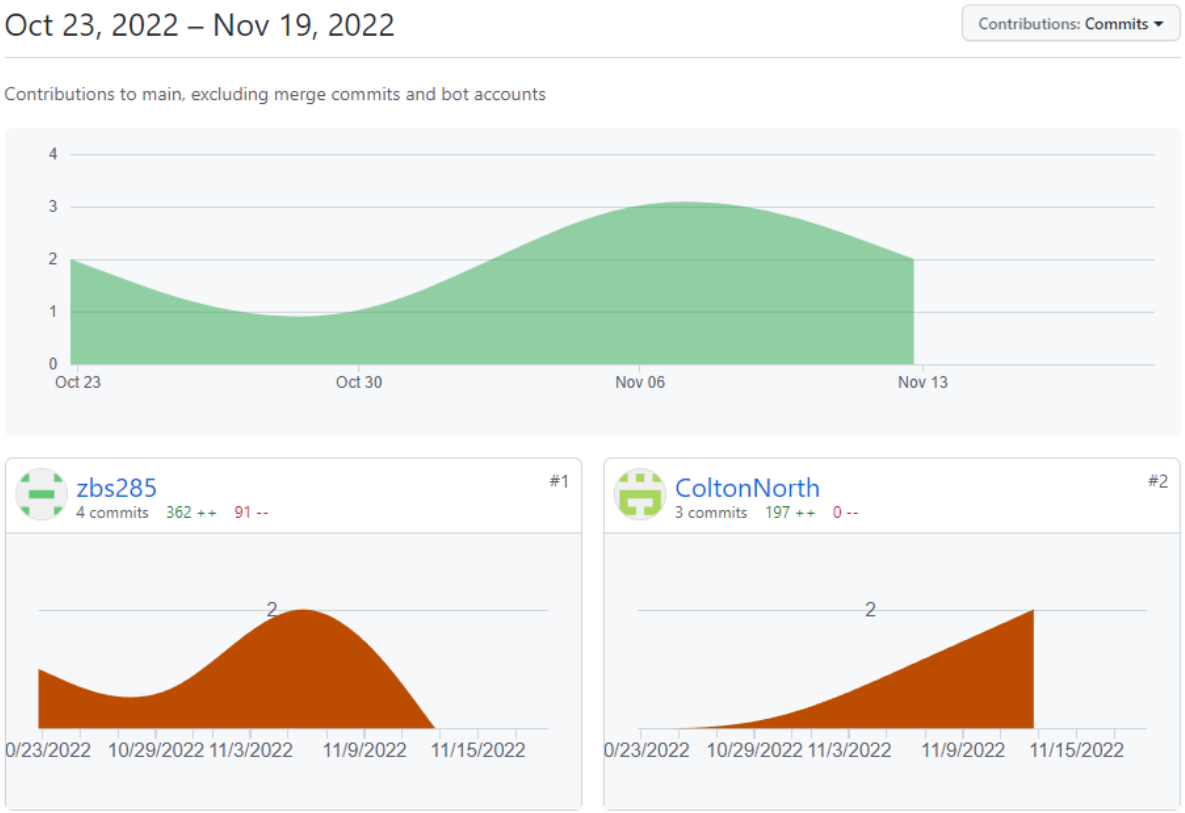

Figure #8: The activity on our GitHub.

There is still plenty of work to do before the end of this project, so the amount of commits to our GitHub is going to increase drastically over the next few days. So far most of the code is for testing various aspects of our system. The sensors and motors have to be working properly if this project is to succeed, so it is of utmost importance that they can be tested individually.