Computer Systems Design Lab
Autonomous Snowplow
Final Report

SYSC 4805

Team Tickle Me Pink

L2-3
Andrew Kong, 101144536
Lizzy Prak, 101110748
Marc Guadagno-Laundy, 101149297

Table of Contents

1.0 Project Charter

The purpose of the proposal will be to define both the overarching objectives and key deliverables. The project's scope includes a precise breakdown of requirements, an organized Work Breakdown Structure, and a well-structured testing plan to ensure all criteria are met. The project's timeline forms another crucial component, necessitating a detailed activity list, a visual Schedule Network Diagram, and a Gantt chart to track progress effectively. Fiscal planning is addressed through a clear cost baseline, ensuring financial feasibility throughout the project's duration. The proposal also underscores the importance of human resource allocation, ensuring clarity in roles and responsibilities for every task, with each activity having a designated responsible individual and approver. Together, these elements provide a robust roadmap, guiding teams toward successful project completion.

1.1 Project Objective

The objective of this project will be the creation of a robot empowered by an Arduino microcontroller and a curated ensemble of sensors. Our team, "Tickle Me Pink," aims to proficiently clear lightweight wooden cubes that simulate snow. This task is intensified by the presence of varied obstacles, both static and dynamic. Harnessing the power of the Arduino platform, our robot will integrate custom-selected sensors to detect boundaries marked by the black lines, distinguish obstacles, and guide its snow-clearance operations. Factoring in the project specifications, the robot will also have scope for strategic expansions, particularly to accommodate a tailored plow. Initiating its task from a pre-defined corner, the robot will operate within set speed limits and aim to achieve its mission within 5 minutes.

1.2 Project Deliverables

- Project Proposal (October 20th, 2023)
  - Project Charter, Scope, Schedule, Cost, Human Resources
- Progress Report (November 14th, 2023)
  - System Architecture, Statechart, Sequence Diagram, Value Analysis Figure
- In-Lecture Presentation (November 16th, 2023)
  - 13 Minute Presentation Including Introduction, Background, Demonstration
- In-Lab Demonstration (November 28th, 2023)
  - 10 Minute Demo To Remove ‘Snow’ From Perimeter Autonomously
- Final Report (December 8th, 2023)
  - Control Charts, System & Costumer Test Results, Working Code

2.0 Scope

2.1 List of Requirements

1. The robot shall not exceed a speed of 30 cm/s.
2. The robot shall detect obstacles in its path and adjust its course accordingly.

3. When the plowing area boundary is detected, the robot shall pivot to remain within the allowable perimeter.
4. The robot shall push the snow to the exterior of the highlighted perimeter.
5. The robot shall be able to clear the snow in no more than 5 minutes.
6. The robot should function autonomously.
7. If the robot encounters an obstacle, it should be able to avoid it.
8. The robot should provide enough power to the essential components to operate properly.

2.2 List of Deliverables

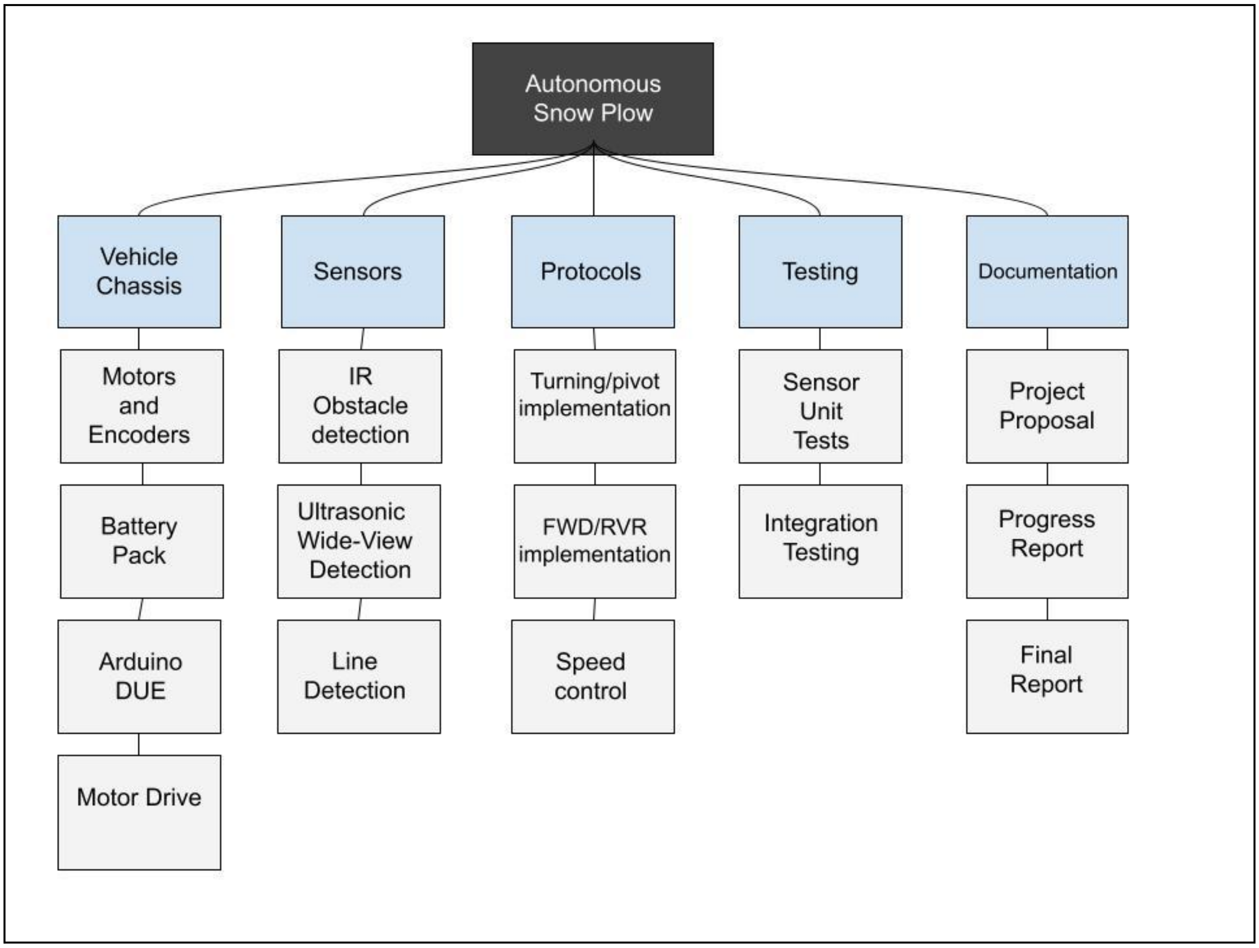

***Figure 1**: List of Activities Organized by WBS.*

3.0 Overall System Architecture

The overall system architecture comprises sensors, the robot's mobility, and the main driving algorithm.

3.1 Sensors

The sensor suite for the robot consists of two ultrasonic sensors mounted on both front sides at a 45° angle, a single infrared (IR) sensor mounted to the front of the robot, and a line follower sensor mounted underneath the front bumper of the robot (see figure below).

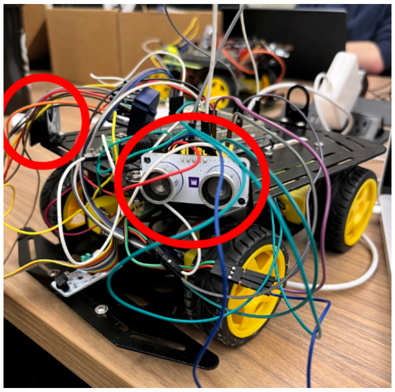

Figure 2: *Placement of Ultrasonic Sensors on Robot Chassis.*

Figure 3: *Placement of IR Obstacle Avoidance Sensor on Robot Chassis.*

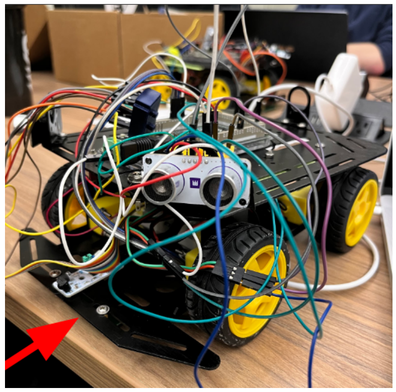

Figure 4: *Placement of Line Follower Sensor on Robot Chassis.*

The purpose of the two ultrasonic sensors is to provide wide-view obstacle detection, provide better detection distance than the IR sensor and can detect obstacles outside the IR sensor's range. Since the IR sensor has minimal range compared to its ultrasonic counterpart, it was chosen as the frontal obstacle detection sensor to detect and prevent head-on collisions. Finally, the line follower mounted beneath the front bumper of the robot is used to detect the plow area's perimeter to prevent the robot from traveling out of bounds.

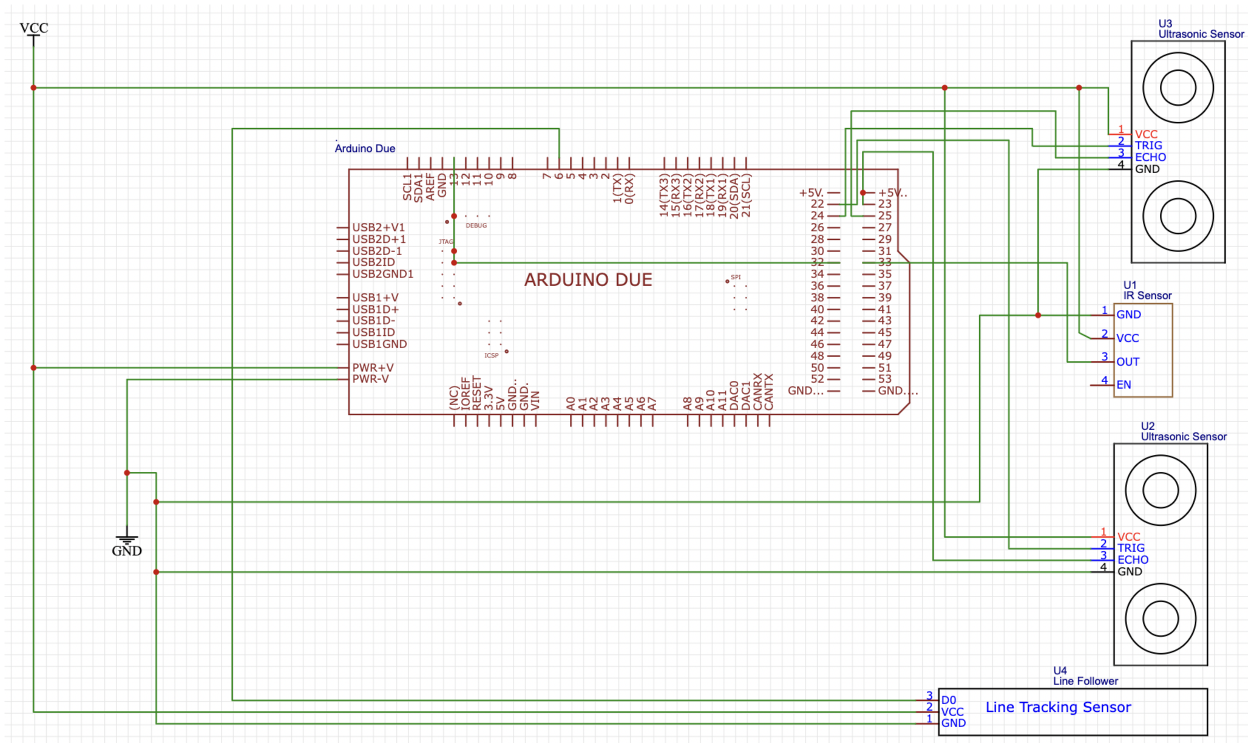

Figure 5: *Arduino and Sensor Pinout Diagram*

3.2 Robot Mobility

The robot has four primary movement states, each performed using its respective function:

- Drive Forward
  - The 'Drive Forward' state powers all four motors forward.
- Drive Backward
  - The 'Drive Backward' state is performed by powering all four motors backward.
- Stop (Idle)
  - The 'Stop (Idle)' state cuts power to all four motors.
- Turn
  - The 'Turn' state is performed by powering 2 motors forward and the other two backward. For example, if the robot wishes to turn right, both right motors rotate backward, and the left motors rotate forward (Vice versa for left turns).

3.3 Main Driving Algorithm

The robot's mobility is defined by the four states described in the previous section, but these states need to be initiated by some event, hence the need for an algorithm to initiate the robot's next movement state. The driving algorithm relies on the sensor output data from all four sensors to react and engage the motors appropriately.

The two ultrasonic sensors have 2 threshold values, ‘DistMin’ and ‘DistStop’. The ‘DistMin’ threshold defines the minimum distance of an obstacle before the robot adjusts its course to avoid it. Likewise, The ‘DisStop’ threshold defines the minimum distance of an obstacle before the robot makes a complete stop, reverses a certain amount, and then turns to bypass said obstacle. Rather than having a threshold value like ‘DistStop’ for the IR sensor, since its range is minimal if the IR sensor detects an obstacle at close proximity, the robot immediately comes to a halt and performs a bypass like previously described to avoid a head-on collision. Lastly, for the robot to remain within the plow area’s perimeter, if a single position on the line follower is high, then the robot should turn opposite to the side on which the perimeter marker was detected. If multiple positions on the line follower are high, then most likely, the robot is facing the perimeter and should stop, reverse to allow for some clearance between the front of the robot and the border, and then turn.

3.4 Statechart of Overall System

With the overall system architecture in mind, the overall operation of the system can be visually described with the help of the following state diagram.

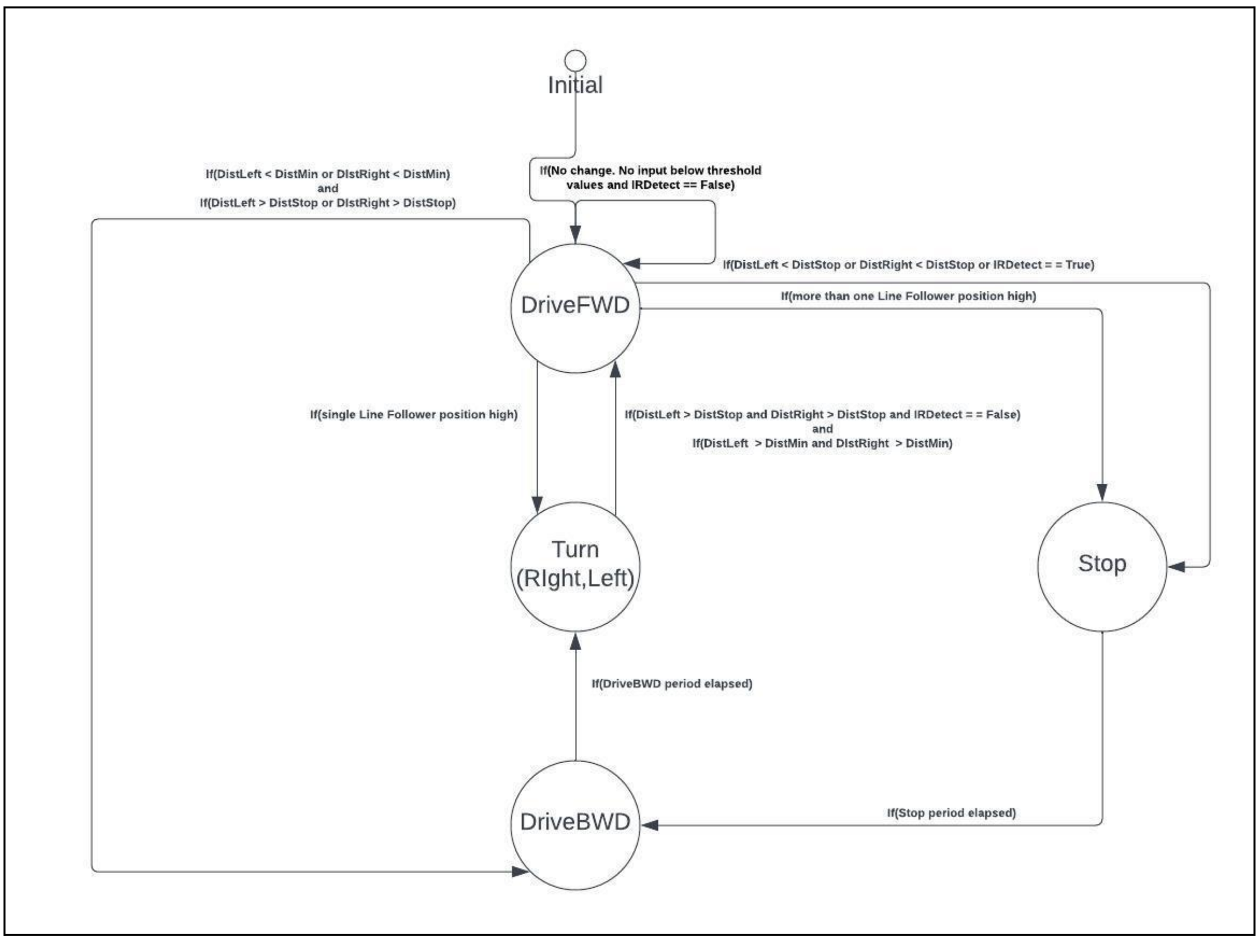

Figure 6: *Statechart of Overall System.*

3.5 Sequence Diagram

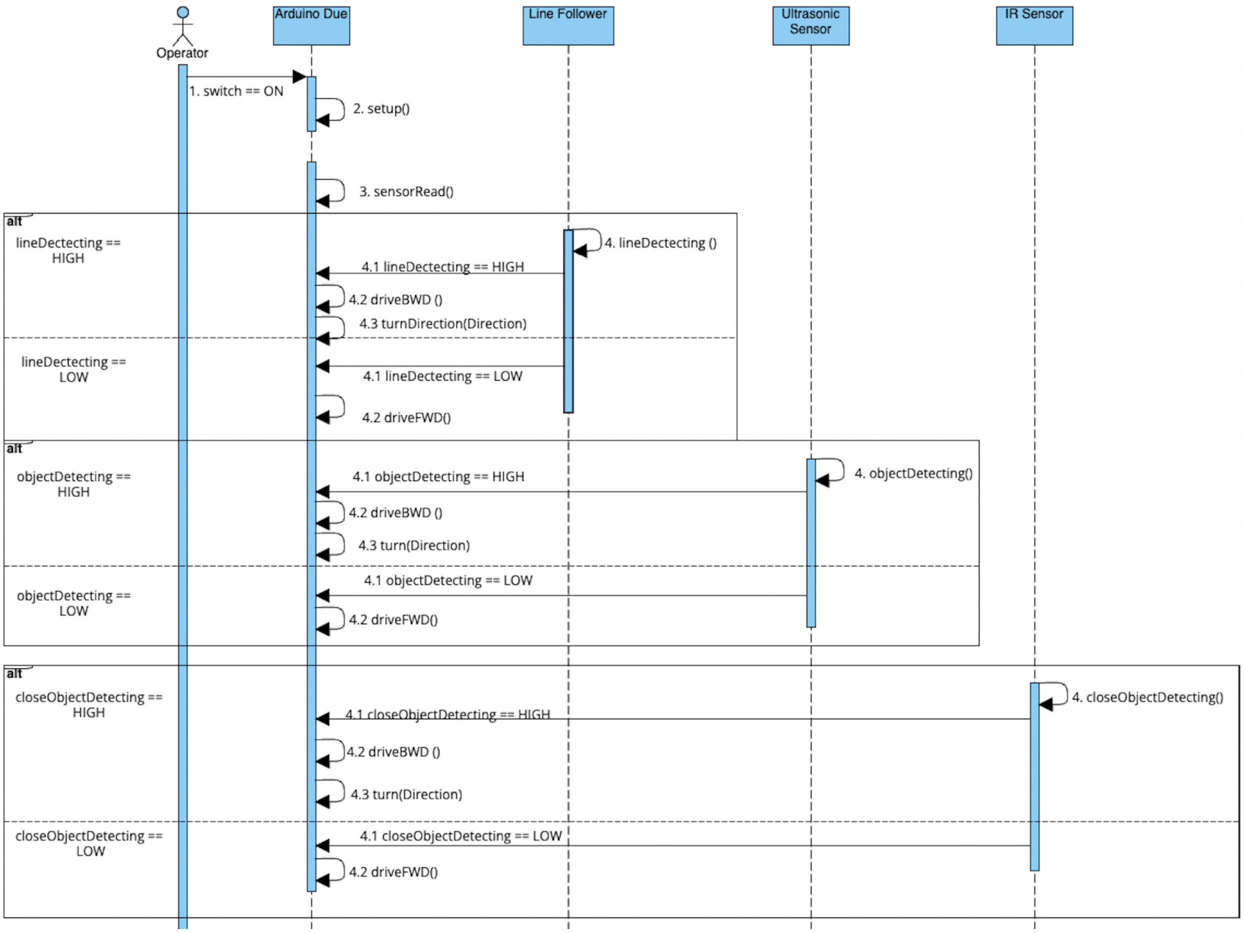

Figure 7: *Autonomous Snowplow Robot Sequence Diagram*

Figure 6 illustrates our system's sequence diagram. Please note that all the numbering of the sensors == 4. This is because the sensors are all running simultaneously, not sequentially.

3.6 Watchdog Timer

Integrating a Watchdog Timer presents a critical layer of reliability in our autonomous snowplow's system architecture. This tool serves as a safeguard against potential system failures or malfunctions. In operation, the Watchdog Timer monitors the system's activity, explicitly watching for signals that indicate normal functioning. If such movements cease due to a software error or a system freeze, the timer will reset the system, ensuring our snowplow robot can recover autonomously from unexpected states. This mechanism is vital for maintaining consistent performance and safety, mainly when human intervention is not feasible. The timer's parameters will be meticulously calibrated to balance responsiveness with false alarms, ensuring that the robot can continue its task with minimal interruptions, thereby enhancing the overall robustness and dependability of our design.

3.7 Power Supply Integration

During our Full System Integration Test, our team identified a crucial modification in the power supply configuration. Initially, the entire system, including motors, Arduino, and sensors, were powered by five AA batteries. However, this setup presented a significant challenge: the batteries could not supply adequate power to all components simultaneously, leading to suboptimal performance and inconsistent performance from the robot.

3.7.1 Implementation Details

Our team’s solution to the unexpected power issue was introducing a dedicated power source for the motors and Arduino. This power source was chosen for its higher current output, ensuring that these components operate optimally. The existing five AA batteries were repurposed exclusively for powering the sensors. This separation not only enhanced the overall system performance but also contributed to the reliability of the sensors by providing a more stable and appropriate power supply.

3.8 Snow Plow Design

We pursued a 225 mm x 150 mm croissant-shaped design for the plow. We concluded that this design would ensure that the plow could collect snow in front of and on the robot's side while keeping the snow within the plow (to be later pushed out of the arena). From system testing, this design worked efficiently and as expected. The blocks remained within the plow and were successfully pushed out of the arena. However, as seen in customer testing - trial 01, we noticed that the robot would fail to reverse enough from the border. This would result in the blocks being scooped back up by the plow's side and entering the arena. Had our robot reversed correctly with the proper amount of power, the original design would still be the most efficient. Unfortunately, we were limited with troubleshooting and decided to remove the sides of the plow, resulting in a straight-edge design.

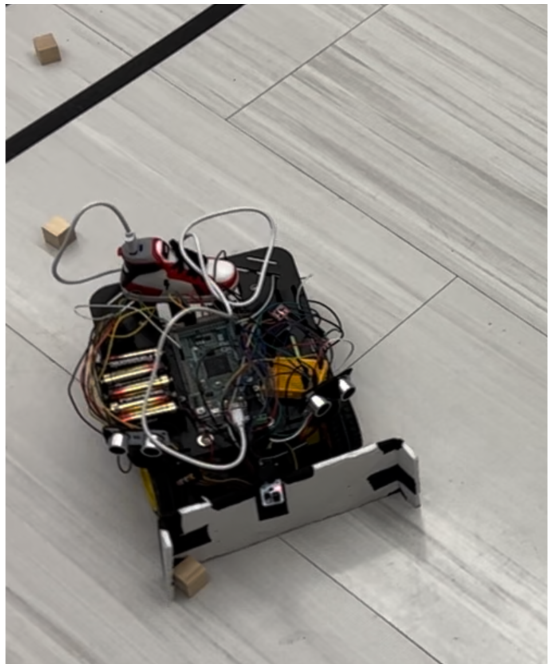

Figure 8: *Final Snow Plow Design*

4.0 Testing Plan

This test plan outlines the procedures and criteria for testing an Arduino robot. There will be three phases to this plan; the first phase will include multiple individual tests to ensure all components are working as intended. Once the individual tests are completed, the next phase commences with integration tests. This will be conducted to verify the compatibility of each component when working in unison. Finally, the last phase will be an end-to-end test where the team will observe the functionality of the completed system against the intended objective.

4.1 Unit Tests

The robot has many components, such as the VMA330 IR Sensor, the Line Follower Sensor, the Ultrasonic Sensor, and the motors. The following unit tests are custom tests to ensure the components function as expected.

4.1.1 VMA330 IR Sensor

Objective: Validate the VMA330 IR sensor's capability to detect obstacles.

Test Steps:

1. Connect the IR sensor to the Arduino.
2. Upload a basic IR detection program to the Arduino.
3. Introduce an obstacle in front of the sensor.
4. Monitor the sensor's output on the Arduino serial monitor.
5. Record results in the test log.

Expected Outcome: The VMA330 IR sensor should detect obstacles and display consistent output values when an obstacle is present.

4.1.2 Line Follower Sensor

Objective: Ensure the line follower sensor can detect and follow a line.

Test Steps:
1. Connect the line follower sensor to the Arduino.
2. Upload a basic line detection program to the Arduino.
3. Create a straight line using white tape or chalk.
4. Place the sensor above the line and move it across.
5. Observe the sensor's output on the Arduino serial monitor.

Expected Outcome: The line follower sensor should consistently detect the line and generate corresponding output signals.

4.1.3 Ultrasonic Distance Sensor

Objective: Verify the sensor's ability to measure distances accurately.

Test Steps:
1. Connect the ultrasonic sensor to the Arduino.
2. Upload a distance measurement program to the Arduino.
3. Position the sensor at known distances from a flat object.
4. Observe the measured distances on the Arduino serial monitor.
5. Compare the sensor's readings with actual measurements.

Expected Outcome: The ultrasonic distance sensor should provide accurate distance measurements with a minimal margin of error.

4.1.4 Motors

Objective: Confirm the motors' operational functionality.

Test Steps:
1. Connect the motors to the Arduino.
2. Activate each motor independently.
3. Observe the motor's rotation, speed, and direction.

Expected Outcome: The motors should rotate smoothly, maintaining consistent speed and responding accurately to directional commands.

4.2 Integration Tests

This integration test plan ensures that the VMA330 IR sensor, line follower sensor, ultrasonic distance sensor, and motors function cohesively when combined in an Arduino robot setup.

4.2.1 IR Sensor & Motors Integration

Objective: Validate that the robot can stop or change direction upon detecting an obstacle with the IR sensor.

Test Steps:

1. Integrate the IR sensor and motors with the Arduino.
2. Upload a program that drives the motors forward until the IR sensor detects an obstacle.
3. Place an obstacle in the robot's path.
4. Observe the robot's response when the obstacle is detected.

Expected Outcome: The robot should stop or change direction upon detecting the obstacle.

4.2.2 Line Follower Sensor & Motors Integration

Objective: Ensure the robot will reverse and turn around when it encounters the marked line.

Test Steps:

1. Integrate the line follower sensor and motors with the Arduino.
2. Upload a program that drives the robot and stops when it encounters a line.
3. Create a mock perimeter using black tape.
4. Place the robot within the perimeter or in a direction where it will cross over the black tape line and activate it.
5. Observe the robot's ability to change directions when it crosses the line.

Expected Outcome: The robot should change direction when encountering the line.

4.2.3 Ultrasonic Distance Sensor & Motors Integration

Objective: Validate that the robot can navigate by maintaining a certain distance from obstacles.

Test Steps:

1. Integrate the ultrasonic sensor and motors with the Arduino.
2. Upload a program that maintains a set distance from obstacles.
3. Place an obstacle in front of the robot.
4. Activate the robot and observe its behavior.

Expected Outcome: The robot should adjust its path to maintain the set distance from the obstacle.

4.2.4 Full System Integration

Objective: Test the combined functionality of all components.

Test Steps:

1. Integrate all sensors and motors with the Arduino.
2. Upload a program that combines the functionalities of obstacle detection, line following, and distance maintenance.
3. Set up a test environment with a marked path, obstacles, and varying distances.
4. Activate the robot and observe its navigation and behavior in the environment.

Expected Outcome: The robot should integrate the functionalities of all sensors and motors seamlessly, effectively navigating the environment and responding to inputs.

4.3 End-To-End Tests

This end-to-end test plan outlines the procedures and criteria for evaluating a snow-clearing robot built based on the provided specifications. The robot aims to efficiently clear "snow" cubes within a confined space while adhering to the design and operational constraints.

4.3.1 Snow Clearing Efficiency

Objective: Ensure the robot can clear the "snow" cubes from the area within the time limit.
Test Steps:

1. Scatter the wooden cubes randomly across the test area.
2. Position the robot at a starting corner inside the perimeter.
3. Activate the robot and start the stopwatch.
4. Observe the robot's snow-clearing operation.
5. Stop the stopwatch once the robot has cleared all cubes or the 5-minute mark is reached.
6. Count the number of cubes remaining inside the perimeter.

Expected Outcome: The robot should clear the test area of "snow" cubes within 5 minutes.

4.3.2 Penalty Check

Objective: Assess potential penalties the robot might incur during operation.
Test Steps:

1. Observe the robot during the snow-clearing test for any violations, such as:
    a. Exceeding size limits.
    b. Hitting obstacles.
    c. Going outside the boundary.
    d. Exceeding maximum speed.
2. Tally the penalties based on the observations.

Expected Outcome: The robot should ideally yield no penalties.

5.0 Control Charts

5.1 Robot Max Speed of 30 cm/s

The robot should not exceed a maximum speed of 30 cm/s, therefore, test were conducted to adjust the robot’s speed parameters to achieve this maximum.

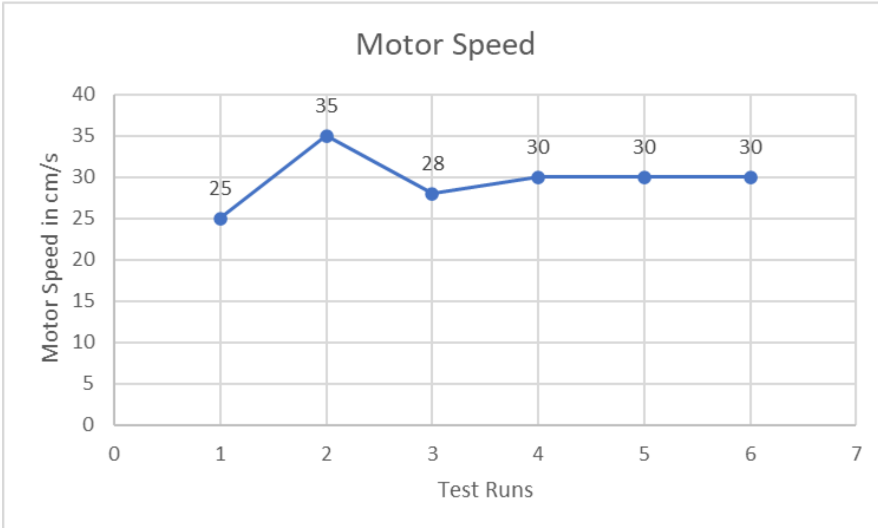

Figure 9: *Robot's max. speed tests. It should not exceed 30cm/s*

5.2 Course Adjustment in the Presence of Obstacles

The ultrasonics serve as wide-view detection and have a minimum threshold value, namely 'DistMin', where if an obstacle falls below this distance the robot should adjust its course in advance to avoid a collision.

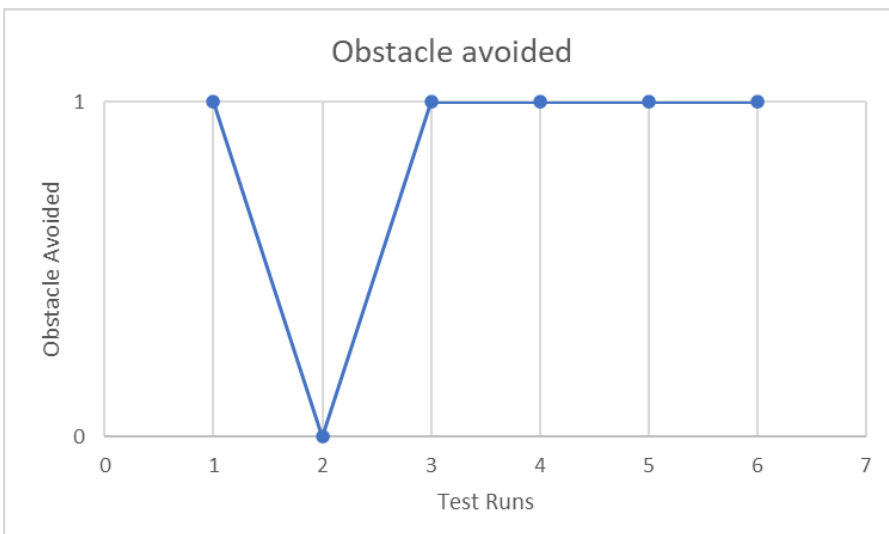

Figure 10: *Robot should adjust course if ultrasonic's detect obstacles below minimum threshold. 1 is obstacle avoided and 0 is failure to avoid.*

5.3 Front Obstacle Detection

The infrared sensor is mounted on the front to detect and avoid head-on collisions. If if the infrared detects an obstacle within about 6 cm of the front of the robot, it performs an evasive maneuver to avoid a collision.

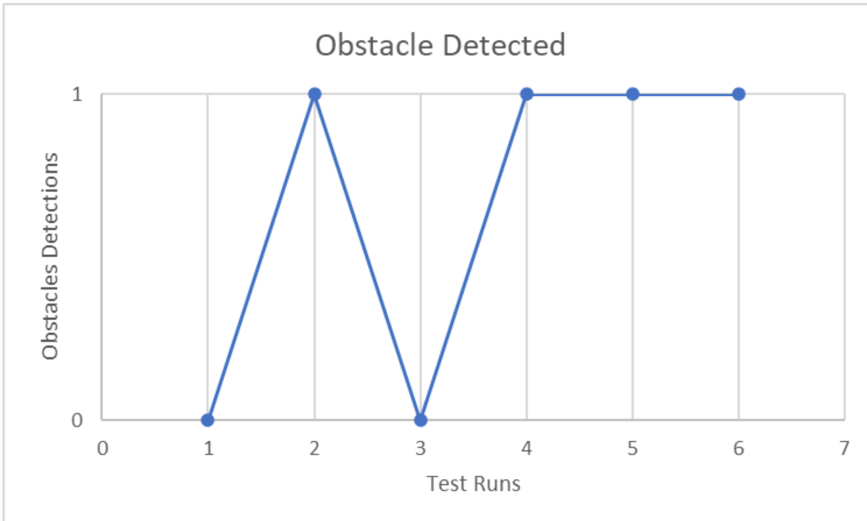

Figure 11: *Robot should detect obstacles directly in front of it, and perform an evasive maneuver. 1 represents a successful detection and 0 is a failure to detect.*

5.4 Perimeter Detection

The line follower is used to detect the perimeter of the plow area so that the robot may avoid crossing it.

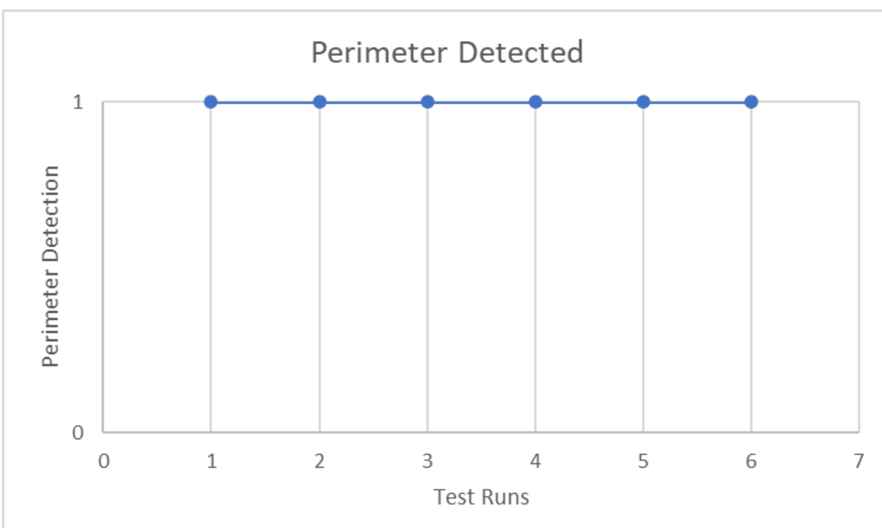

Figure 12: *Robot should detect the black perimeter and avoid crossing it. 1 represents a successful detection and 0 is a failure to detect.*

6.0 Schedule

6.1 List of Activities

- Lab 01
    - Building the Basic Robot Chassis
    - Testing the IR Obstacle Avoidance Sensor
- Lab 02
    - Testing the Line Follower Sensor
    - Testing the Ultrasonic Distance Sensor
- Lab 03
    - Testing the LSM6 Accelerometer and Gyroscope
    - Testing the LIS3MDL Magnetometer
- Lab 04
    - Testing the Motor Driver Board
    - Testing the Wheel Encoders
    - Completing the Basis of Robot Setup
    - FWD/RVR Implementation
    - Turning/ Pivot Implementation
- Project Proposal
- Progress Report
- Testing
    - Sensor Unit Testing
    - Integration Testing
    - Speed Control
- Snowplow Construction
    - Designing and attaching a plow to the robot
- Final Report
- In Lecture Presentation
- In Lab Presentation

6.2 Schedule Network Diagram

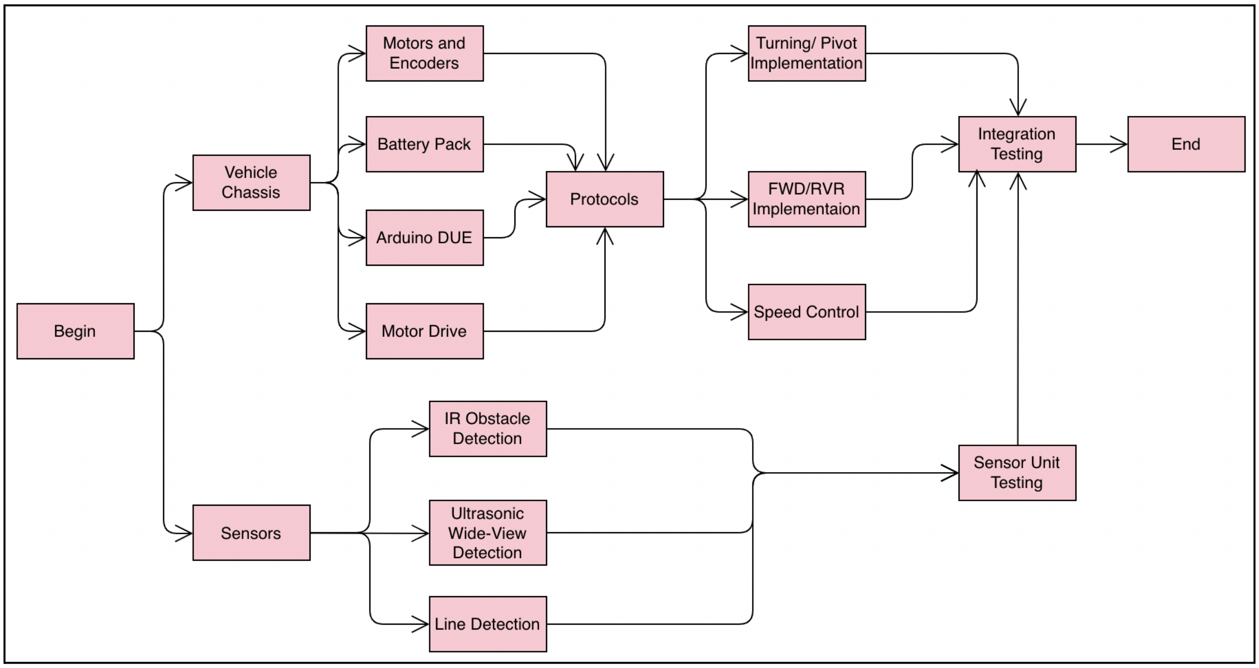

Figure 13: *Schedule Network Diagram*

Figure 8 illustrates our Schedule Network Diagram. This diagram reflects the activities given by the WBS and exhibits the activities performed in their sequential order.

6.3 Gantt Chart

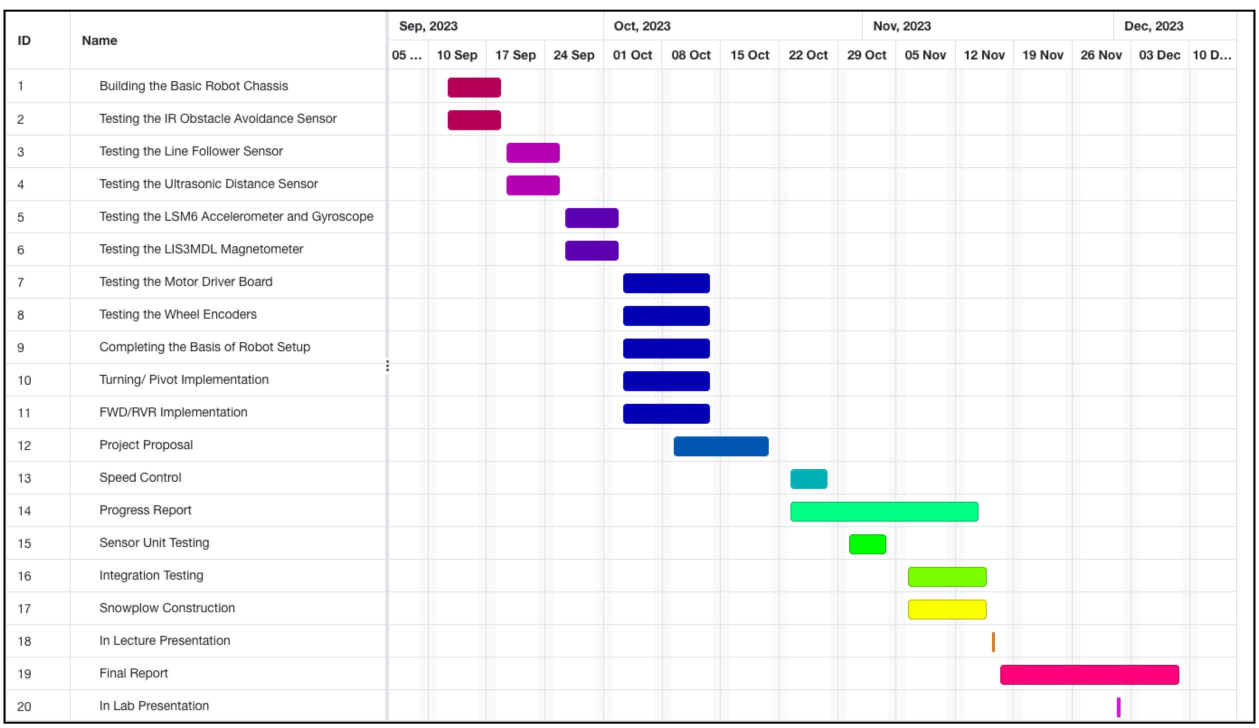

Figure 14: *Gantt Chart of Labs with Activities, Milestones and Contingencies*

Figure 9 shows our Gantt chart– a list of total project activities and their scheduled completion date. Presently, our group is on schedule.

7.0 Cost

The table below outlines the group's cost baseline.

Table 1: Cost Baseline

Item	Cost
Course Kit	$500
Developer ($50/ hour/ developer)	$10 800
Total	$11 300

Each developer works on the project 6 hours/ week for 12 weeks. There are three developers in total. Below is Figure 10, highlighting our predicted cost baseline.

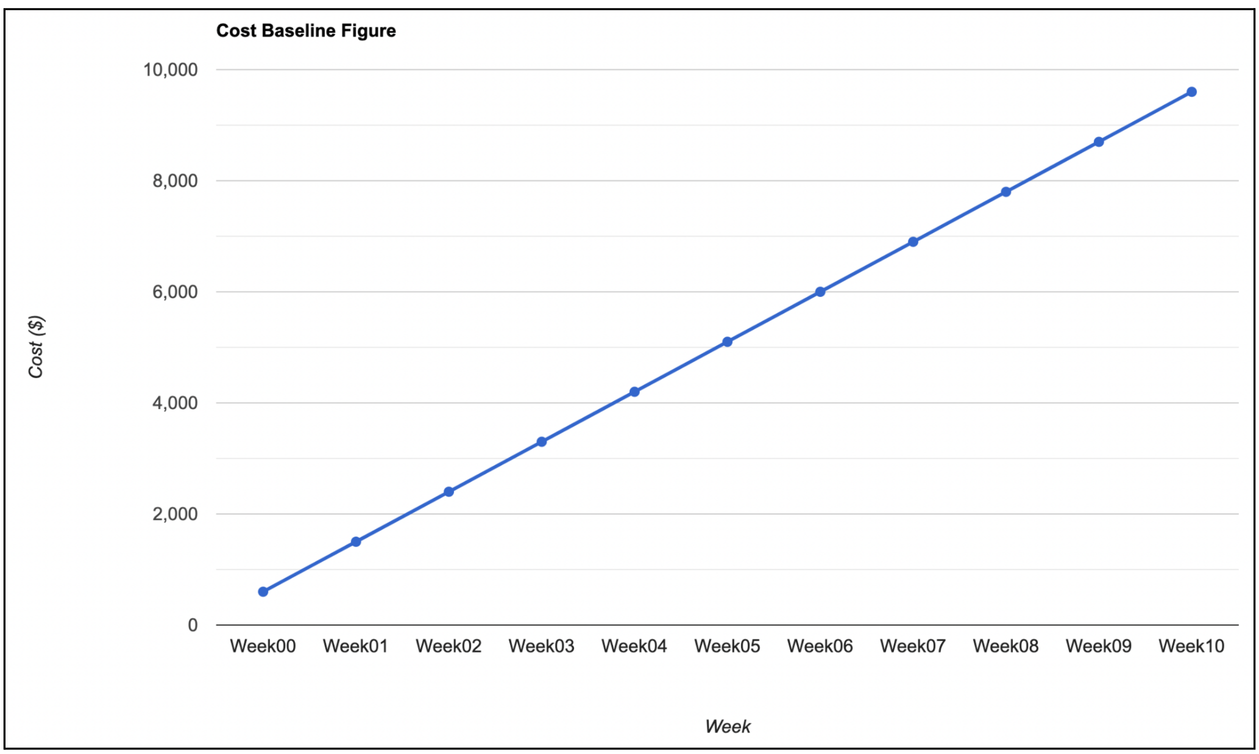

Figure 15*: Cost Baseline Figure*

7.1 Planned Value Analysis Figure

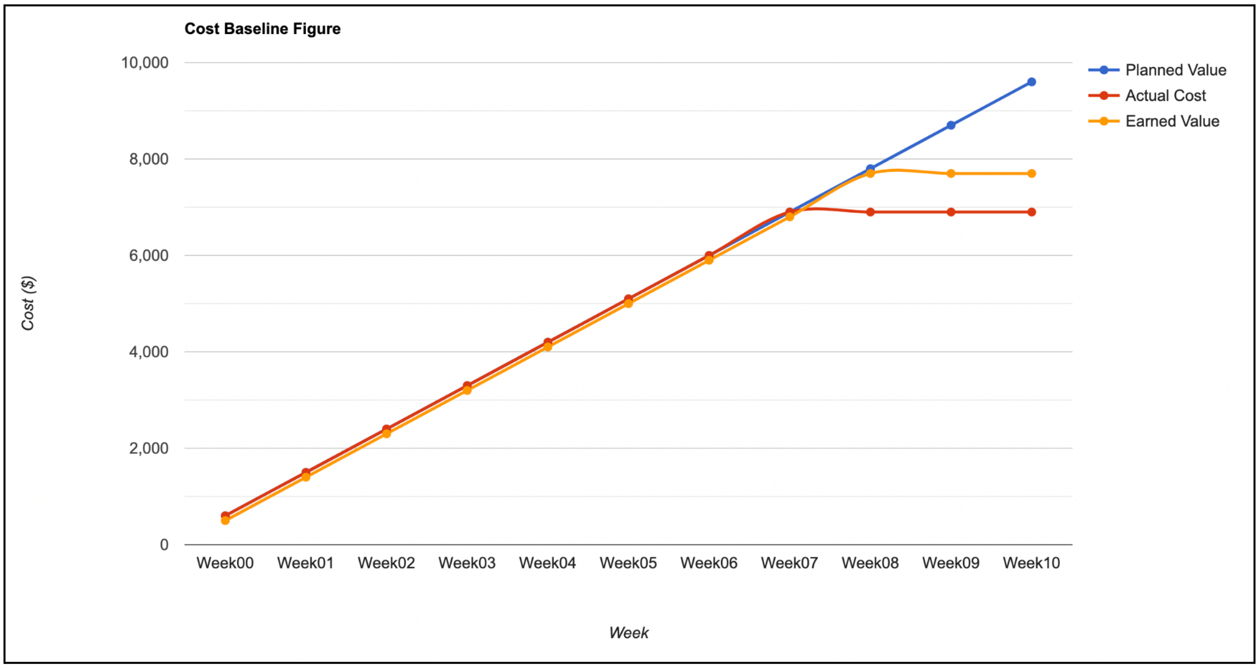

Figure 16: *Planned Value Analysis Figure*

As seen in Figure 11, our group currently has not exceeded the budget and we plan to continue to do so. Planned Value can be described as the authorized budget for all activities. We authorized a budget of $13000. Earned value is the authorized budget for actually completed activities to date. The authorized budget for the actually completed activities to date is approximately $8000. Actual Cost is the actual budget for completed activities to date. Currently, we have only spent approximately $7000, concluding that we are $1000 under budget.

8.0 Human Resources

The following table illustrates the group's responsibilities, as well as the assigned responsible person and approver.

Table 2: Responsibility Assignment Matrix

Responsibility	Responsible	Approver
Activities from Lab 01	Andrew	Lizzy
Activities from Lab 02	Lizzy	Marc
Activities from Lab 03	Marc	Andrew
Activities from Lab 04	Andrew	Lizzy
Project Proposal	Lizzy	Marc

Progress Report	Marc	Andrew
Sensor Unit Testing	Andrew	Lizzy
Integration Testing	Lizzy	Marc
Speed Control	Marc	Andrew
Snowplow Construction	Andrew	Lizzy
Final Report	Lizzy	Marc
Presentations	Marc	Andrew

9.0 Final Testing Results

9.1 System Testing

Table 3: *Results from System Testing during Lab 11*

Trial	Blocks Cleared (total of 10 blocks)	# Times Left Arena	# Times Hit an Obstacle	Duration (minutes)
01	4	1	2	5:02
02	8	1	3	5:04
03	6	0	2	5:00
04	6	1	3	5:00
05	8	0	1	5:00

9.2 Customer Testing

Table 4: *Results from Customer Testing during Lab 12*

Trial	Blocks Cleared	# Times Left Arena	# Times Hit an Obstacle	Duration (minutes)
01	32	1	5	5:00
02	20	11	1	5:00

10.0 Github

Our team, "Tickle Me Pink," has set up a Github repository for our autonomous snowplow robot project. This repository includes all the necessary documentation and code and will be regularly updated until the final demo day. This approach ensures that our progress is transparent and accessible, providing a real-time view of our development journey.