Computer Systems Design Lab
Autonomous Snowplow
Project Proposal

SYSC 4805

Team Tickle Me Pink

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

Table of Contents

1. 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. 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

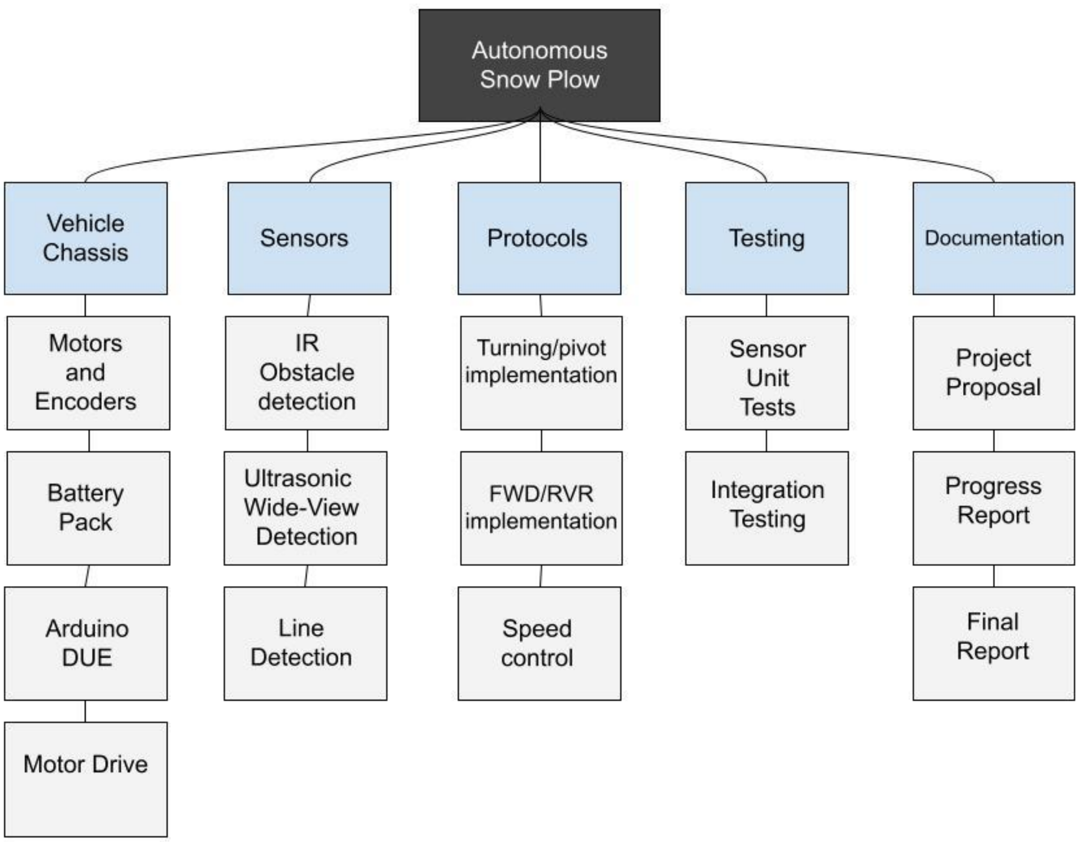


Figure 1: List of Activities Organized by WBS.

2.3 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.

2.3.1 Unit Tests

The robot consists of 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.

2.3.1.1 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.

2.3.1.2 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.

2.3.1.3 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.

2.3.1.4 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.

2.3.2 Integration Tests

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

2.3.2.1 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.

2.3.2.2 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.

2.3.2.3 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.

2.3.2.4 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.

2.3.3 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.

2.3.3.1 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.

2.3.3.2 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.

3. Schedule

3.1 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

3.2 Schedule Network Diagram

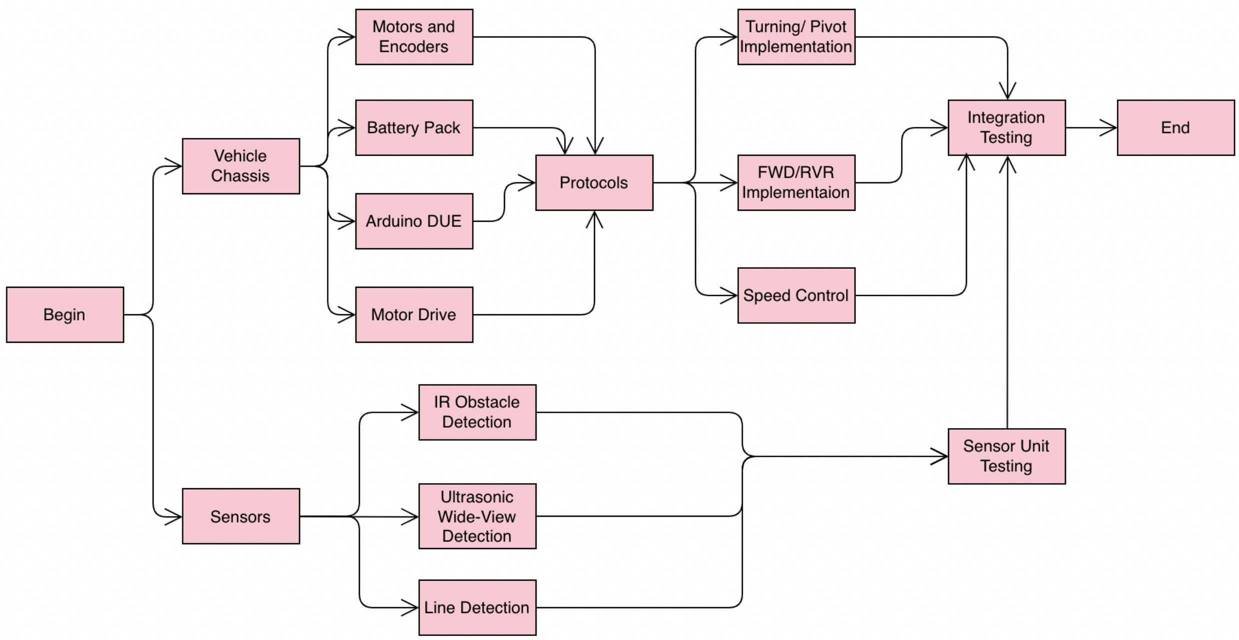


Figure 2: Schedule Network Diagram

3.3 Gantt Chart

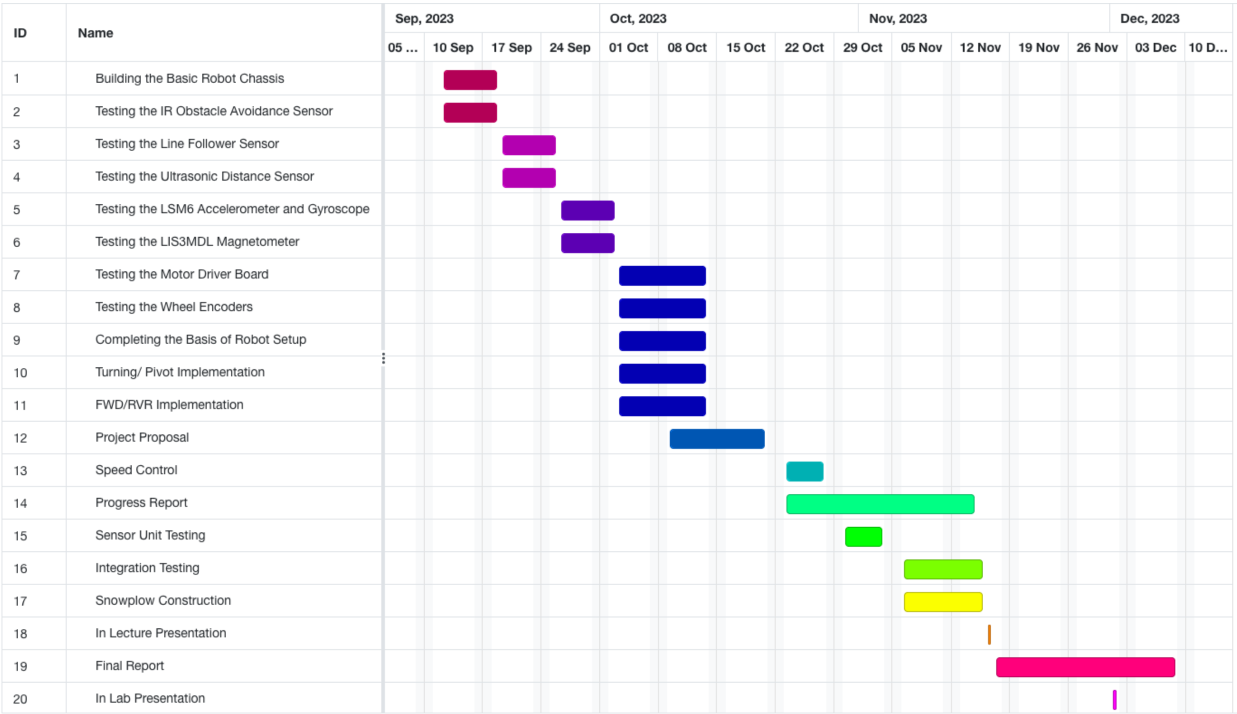


Figure 3: Gantt Chart of Labs with Activities, Milestones and Contingencies

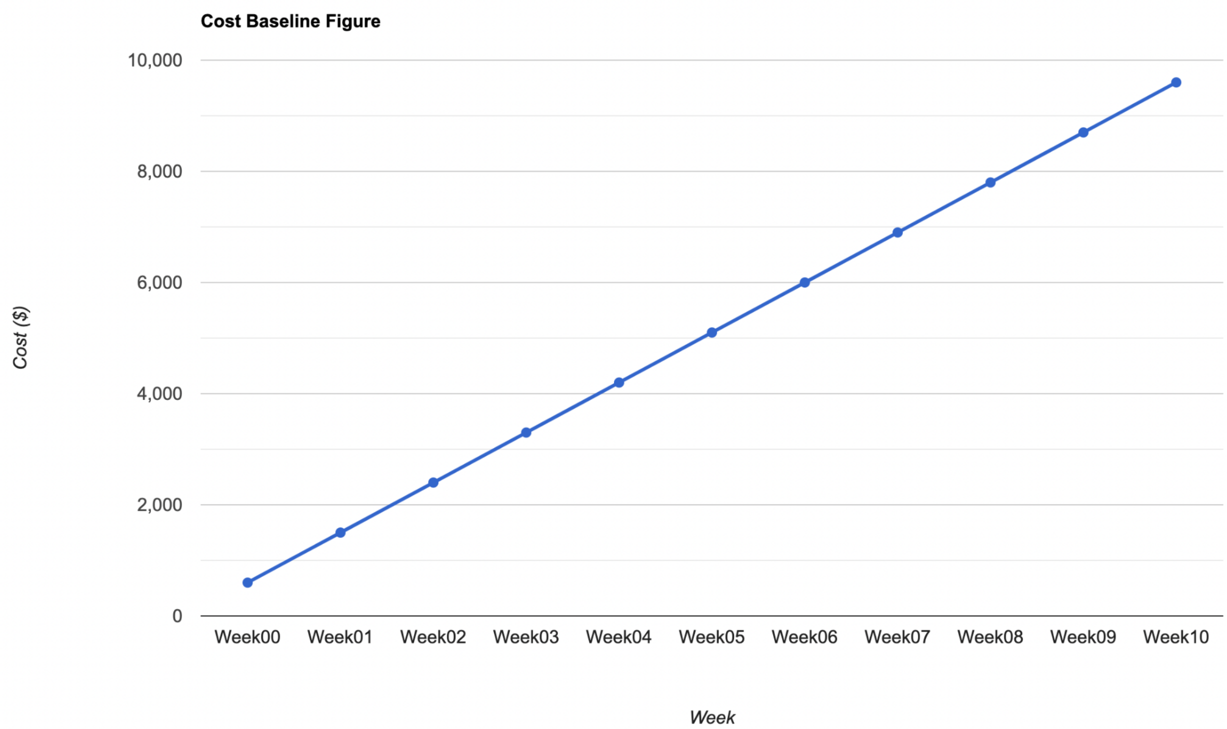
4. Cost

Table 1: Cost Baseline

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

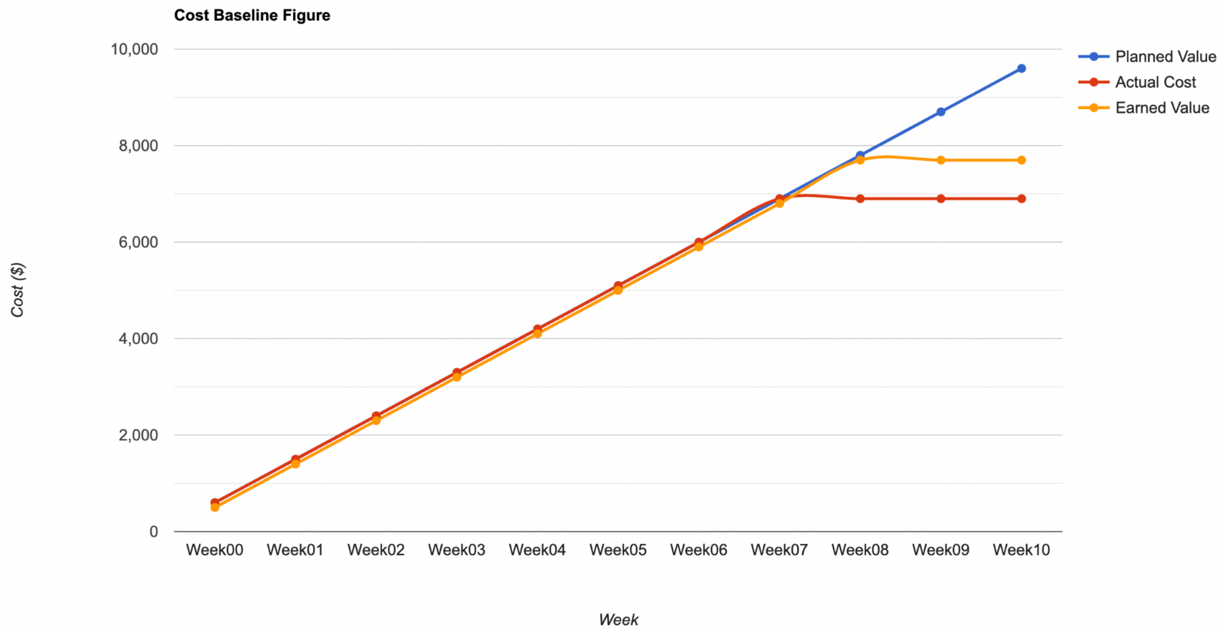
Each developer works on the project 6 hours/ week for 12 weeks. There are three developers in total.

Figure 4: Cost Baseline Figure



Cost Baseline Figure
10,000
8,000
6,000
4,000
2,000
0
Cost ($)
Week00
Week01
Week02
Week03
Week04
Week05
Week06
Week07
Week08
Week09
Week10
Week

Figure 5: Planned Value Analysis figure



5. Human Resources

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