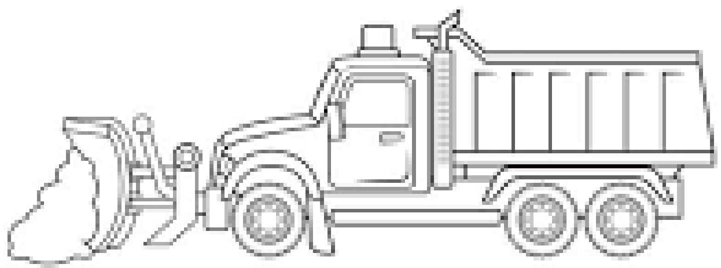
SYSC4805 | Computer Systems Design Lab

# Zaffre L3-5

## Autonomous Snowplow Final Report



Abdalla Abdelhadi - 101142768
Jakob Delaney - 101201314
Chris Nguyen - 101167689

December 8th, 2023

# Table of Contents

# 1.0 Proposal

Refer to the Appendix for revised Project Proposal.

## 1.1 Changes from Proposal

Some changes made from our project proposal in the final implementations is removing the Time of Flight (ToF) sensors from the system. When using the ToF sensors in the system it could never work in concurrence with the motors or other sensors. As a group decision, we decided to replace the ToF sensor's functionality with the Ultrasonic sensors which includes changing the position of the original placement in the proposed design. The Ultrasonic sensors and Line follower were able to operate as a whole with the motors running albeit with some difficulty which was to be expected.

## 1.2 Contributions

Abdalla worked on the motors and ToF sensors handler code, Chris worked on the Line Follower handler code and snow plower attachment, Jakob worked on the Ultrasonic sensors. As a group, everyone worked on the reports. When the removal changes happened , Abdalla worked on changing the code's loop with the removal of the ToF sensors, Chris updated the circuitry and snow plower attachment, and Jakob worked on debugging the issues with the robot and code.

*Table 1: Table displaying the activities and the person that completed them*

| Milestone | Activity | Completed By |
|---|---|---|
| Building Snow Plow Hardware | Attaching line follower to robot | Chris |
| | Attaching ultrasonic sensor to robot | Abdalla |
| | Attaching time of flight sensor to robot | Jakob |
| | Connecting the motors to the Ardunio | Jakob |
| Handler Code for Sensors and Motors | Handler code for line follower | Chris |
| | Handler code for ultrasonic sensor | Abdalla |
| | Handler code for time of flight sensor | Jakob |

| | | |
|---|---|---|
| | Handler code for motors | Abdalla |
| Object Avoidance | Detect and avoid stationary objects | Abdalla |
| Line Avoidance | Detect and avoid line | Chris |
| Finishing Snowplow | Build and attach snow plow | Chris |
| | Watchdog timer and end-end testing | Jakob |
| Progress Report | Finish progress report | All |
| Final Report | Finish final report | All |
| Presentation | Finish and practice presentation | All |

# 2.0 Control Charts

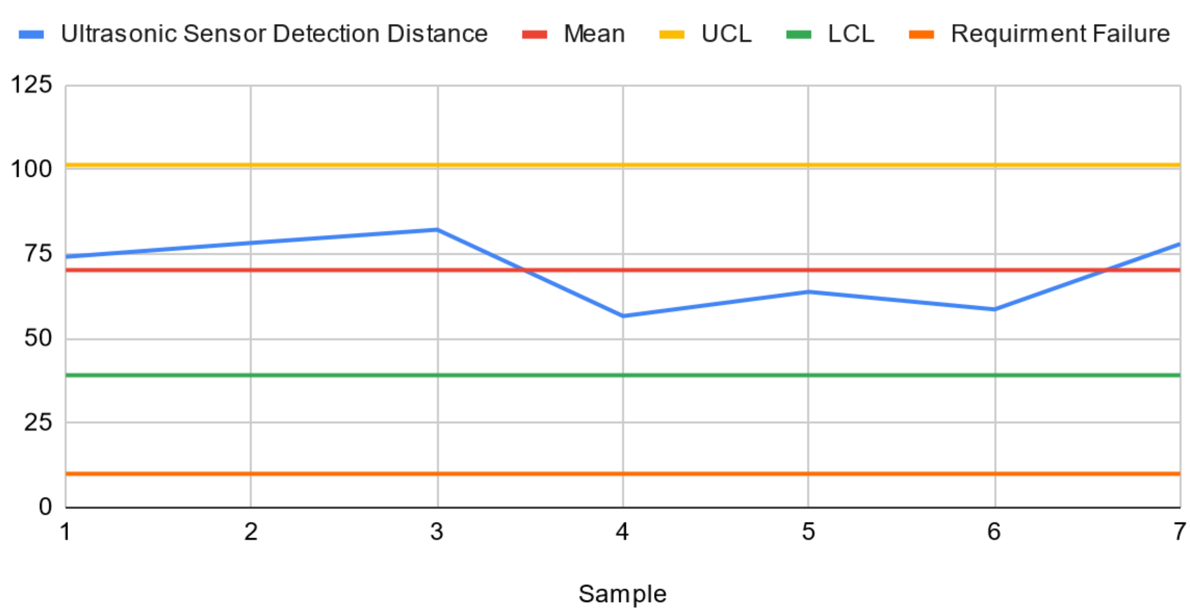



Figure 1: Control Chart Performance of the Ultrasonic Sensor

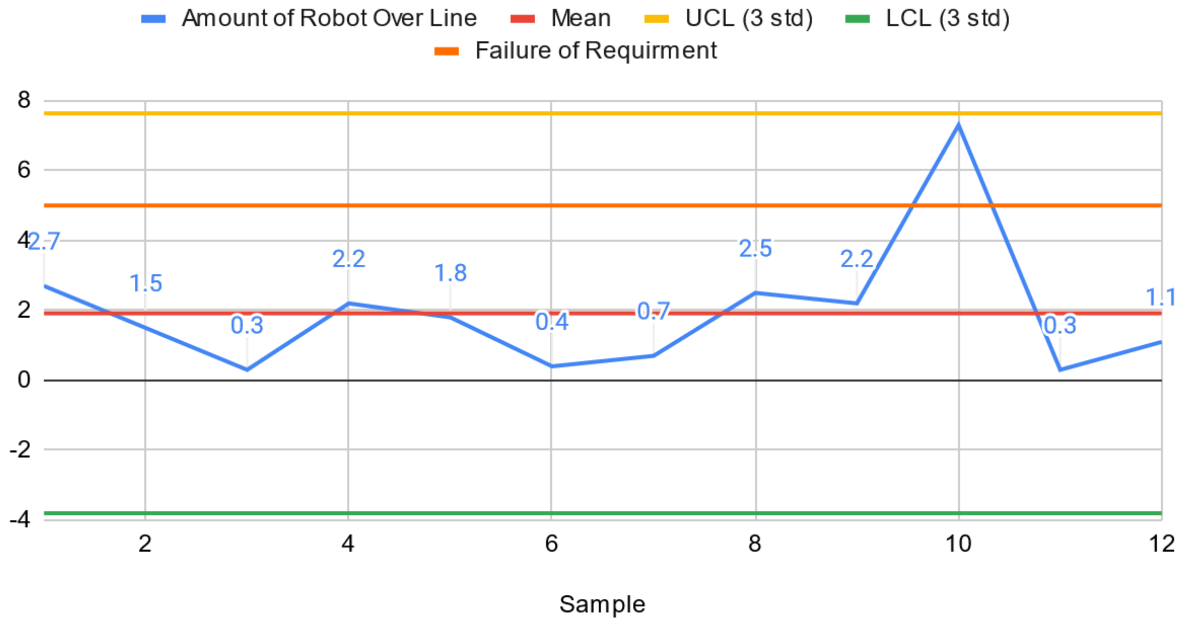



Figure 2: Control Chart Performance of the Line Follower

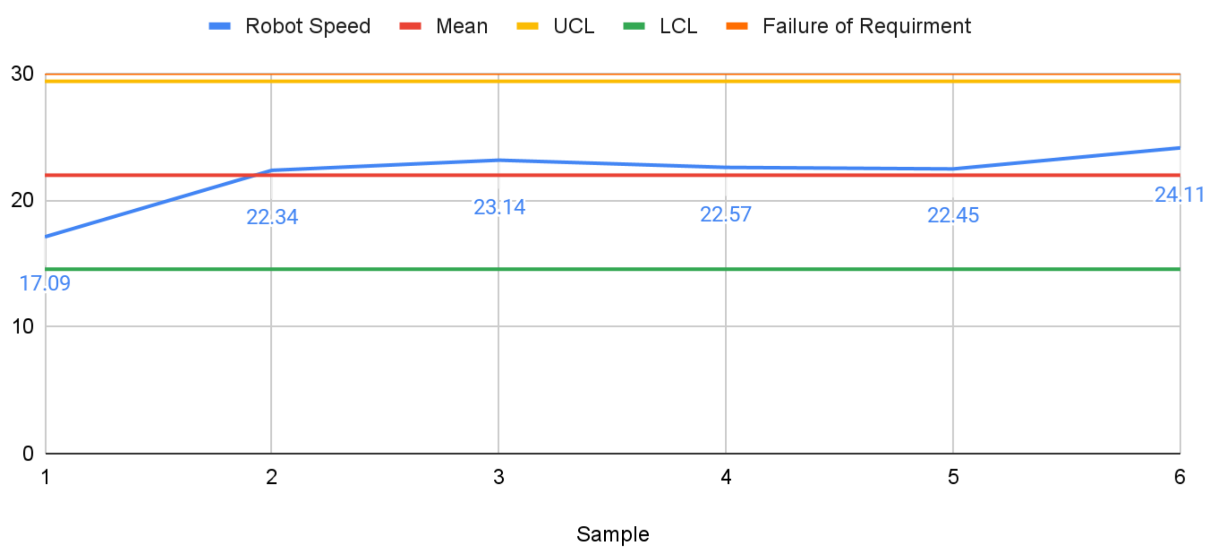



Figure 3: Control Chart for Speed of the Robot

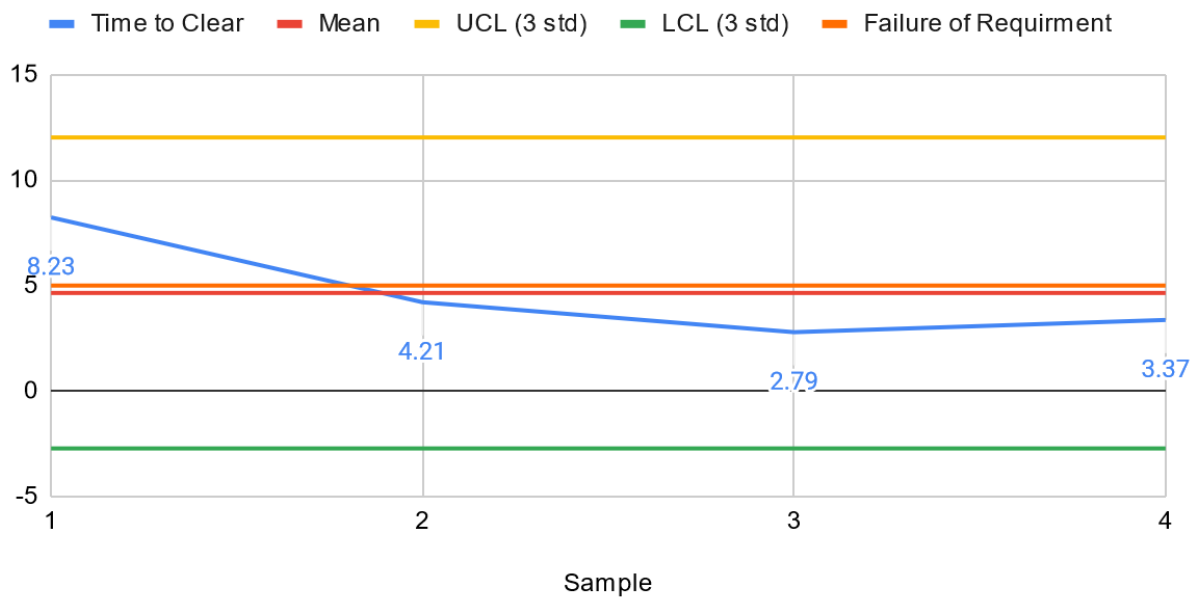



Figure 4: Control Chart of Time to Clear 75% of Blocks in Testing Area

# 3.0 Testing Results

## 3.1 System Testing

During Lab 11, testing results weren't as consistent as we had hoped. A lot of testing results were failing due to inconsistent power output as the batteries were draining a lot quicker than we had anticipated. This is mainly due to the motors causing the drainage quicker. Here are the results of the various tests that occurred:

**Unit Testing:**

- Time of Flight (ToF): Testing the ToF sensors independently were successful and they operated as intended. It was able to detect good measurements for objects in front of it.
- Ultrasonic sensor: Testing the Ultrasonic sensors independently were successful and they operated as intended. It was able to detect good measurements for objects on the side.
- Line Follower: Testing the Line Follower Sensor independently were successful and they operated as intended. It was able to detect when the robot was over a black line.
- Motors: Testing the motors independently were successful and they operated as intended. It was able to turn both directions, stop, and move forwards.

**Integrated Testing:**

- All Sensors Combined: When testing all the sensors together, they all were successful. All of the sensors operated as intended.

- <u>ToF with Motors:</u> When testing the Ultrasonic sensors' functionality alongside the motors they failed to work. This is because everytime the ToF would try to read a value it would time out.
- <u>Ultrasonic with Motors:</u> When testing the Ultrasonic sensors' functionality alongside the motors they worked successfully and produced the intended functionality.
- <u>Line Follower and Motors:</u> When testing the Line Follower sensors' functionality alongside the motors they worked successfully and produced the intended functionality.

**End-End Testing:**

- When combining all the sensors and motors together the ultrasonics and the line follower were working correctly, however, the ToF sensors still failed to work. As a result of this failure we decided to remove them and put the ultrasonics at the front as our main obstacle detection sensors.

## 3.2 Customer Testing

In Lab 12, the demo day, our robot didn't function fully as intended. To begin with, in the first demo run, the plow started to move forward but failed to turn when it detected a line. This was due to the battery being empty. After replacing the plow's batteries, the plow was working as intended and it was able to detect the line and stay inside. It was able to detect and avoid a few obstacles, but not the majority of them. This is mainly due to a major flaw in the Ultrasonic sensors, known as the invincible wall. If the ultrasonic sensor was facing the obstacle at an angle then the sound waves wouldn't rebound back to the sensors and the ultrasonic wouldn't detect an obstacle. As a result the plow hit obstacles. Another major flaw in our robot was the snowplow design. The snowplow would occasionally not capture the snow and the snow would slide under it causing it to hit the Line Follower sensor. This caused the Line Follower handler to click in and make the robot turn indefinitely. As a result, one of our team members would have to pick up the robot and fix the line follower. Apart from these flaws the robot was able to stay inside the arena and push blocks out.

In our second demo run, all the problems that we faced in the first demo were still there except for the dead batteries. The robot still hit obstacles occasionally and the snow blocks would slide under the plow and hit the line follower. However, in this run we managed to push more blocks out since we didn't have to switch the robot batteries.

Overall, our customer testing was a failure because of our poor design for the plow and our robots inability to avoid obstacles.

# 4.0 Working Github Code

To view the working code on the GitHub repository please refer to this link: [https://github.com/Jadyin/ZaffreL3-5](https://github.com/Jadyin/ZaffreL3-5).

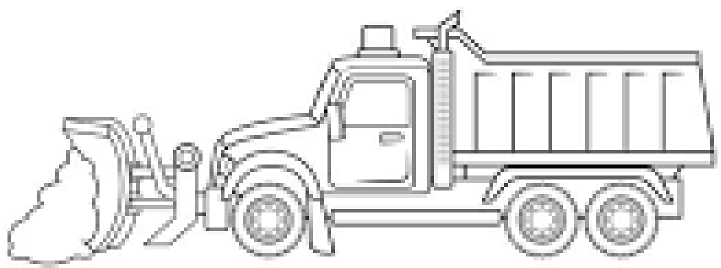
# 5.0 Appendix

Bellow is the revised project proposal:

**SYSC4805 |**
**Computer Systems Design Lab**

# Zaffre L3-5

## Autonomous Snowplow Proposal




Abdalla Abdelhadi - 101142768
Jakob Delaney - 101201314
Chris Nguyen - 101167689


October 20th, 2023

# Table of Contents

# 1.0 Project Charter

## 1.1 Overall Objectives

The overall objective for our group *Zaffre* is to design an autonomous snow-plower robot that clears snow in an enclosed area while being able to avoid obstacles/interference. Some design considerations that are able to accomplish this goal is the use of sensors, and motors. Multiple sensors with each function specifically related to one task. To be able to avoid interference and natural obstacles in an environment, sensors capable of detecting objects are required. Our group will be using the time of flight sensors to detect objects ahead of head. Also, our group will be using an ultrasonic distance sensor to detect any boxes on the side, this will help when the robot is turning so it doesn't collide with an object from the side. Another task is to ensure our robot won't leave the enclosed area. To accomplish this, our group will be using the VMA330 line follower module which will keep the robot inside the arena. A likely example would be when the sensor detects a black line the robot should do a complete 180 or close to that angle which will without doubt keep the robot inside. Our final goal will be designing a "snow-plow" capable of withstanding the weight of the snow and removing it. Everything considered, our group will need to follow the specifications of the size of the robot allowed as well as the functionality outlined by the rules in the project document.

## 1.2 Overall Deliverables

The milestone deliverables are listed as follows: **Oct 30 - Nov 4,** completing physical robot hardware. In this deliverable the team will securely attach the motors and all the sensors required to the robot and ensure their correct functionality. **Nov 5 - Nov 14**, completing handler codes for all motors. In this deliverable create handler code to interact with all sensors and the motor. **Nov 14th** progress report due, which is a maximum of 20 pages containing the updated proposal document addressing all raised concerns, the overall architecture of the system, a statechart of the system as well as a sequence diagram, an updated planned value analysis figure, a watchdog timer demo, and updated Github code repository with each member having 3 recorded committed activities. **Nov 15 - 23,** completing the object avoidance algorithm. Using the handlers created, write an algorithm that will detect and avoid obstacles. **Nov 19 - Nov 27** completing the line avoidance algorithm. Using the handlers created, write an algorithm that will detect and avoid the border of the arena which is a black line. **Nov 28 - Dec 3** should be used for final testing, i.e watchdog timer test and snowplow test.

The final deliverables are as follows: Lecture 12 - **Nov 30** is an in-lecture presentation (L3-Group 5). Lab 12 (**Dec 5**) is the testing day where the robot is demonstrated. **Dec 8** - the final report is due and will consist of: the final proposal document - with all comments and contributions of each member, Control charts with at least 5 runs per requirement, Results of testing from both lab 11 and 12, and a complete commented Github repository.

# 2.0 Scope

## 2.1 List Of Requirements

**Requirement 1 Snow:**

Snow shall be represented using wooden cubes with dimensions of 20mm each.

The robot shall be able to apply a force that will move the snow cubes in the desired direction to remove the snow cubes from the testing area.

**Requirement 2 Obstacle and Cube Detection:**

The robot shall be able to identify and differentiate between obstacles and snow cubes using the various modules provided.

The robot shall use the ultrasonic sensor module to detect obstacles beside it while it is turning. The robot can confirm if an ultrasonic detection is correct by turning to face the object and viewing it with its front mounted modules.

The robot shall use the VL53L1X as the primary detection module for obstacles to be avoided. Once the module detects an object that is 10cm away, the handler will cause a software interrupt. This will trigger the system to stop the robot from moving and turn until the object is avoided. Once the object has been avoided the robot will continue normal operation.

**Requirement 3 Obstacle Response:**

The robot during operation shall not apply any force to any obstacle unless it is the wooden boxes representing the snow.

The robot shall detect objects that are 10cm away (not wooden boxes representing snow) then the object avoidance algorithm will kick in and turn the robot until the obstacle isn't detected anymore. Then the robot will continue moving forward.

The robot shall not have its operation hindered or broken due to any obstacle.

**Requirement 4 Testing Space:**

The total area of the testing space shall be 6 square meters ($6m^2$).

The robot shall begin its operation from one of the corners inside the perimeter of the testing space.

The robot's wheels (or wheel) shall not leave the testing area by 5cm.

The robot shall use Line Follower Sensor to detect if it has encountered an edge of the testing space. During plowing of snow cubes, when the robot encounters an edge of the testing space, that shall be confirmation that a successful plow has been completed. The robot should stop moving in the direction it currently is and instead turn back into the testing space. During normal movement if the robot encounters an edge of the testing space it has the option to move along it. However, it must not cross the edge as it could result in the robots' wheels exiting the testing area by more than 5cm.

**Requirement 5 Robot Size:**

The robot's dimensions, including width, length, and height, shall not exceed 226mm, 262mm, and 150mm respectively. The robot chassis is already at the size of 176mm (width), 232mm (length), and 79.5mm (height).

The plow attachment, if used, shall not extend in width, length, and height by more than 50mm, 30mm and 70.5mm respectively.

**Requirement 6 Robot Speed:**

The robot shall not exceed the maximum allowable speed of 30 cm/s at any point during snow clearing.

The robot shall control its wheels using the Motor Driver Board which will be given specific instructions from the onboard Arduino.

**Requirement 7 Time:**

The robot shall complete snow clearing operations within a maximum allowable time of 5 minutes.

**Requirement 8 Operation:**

The robot shall be fully autonomous once it has been activated in the target area. Operators of the robot shall not touch, adjust, or reset the robot for any reason.

## 2.2 List of Deliverables

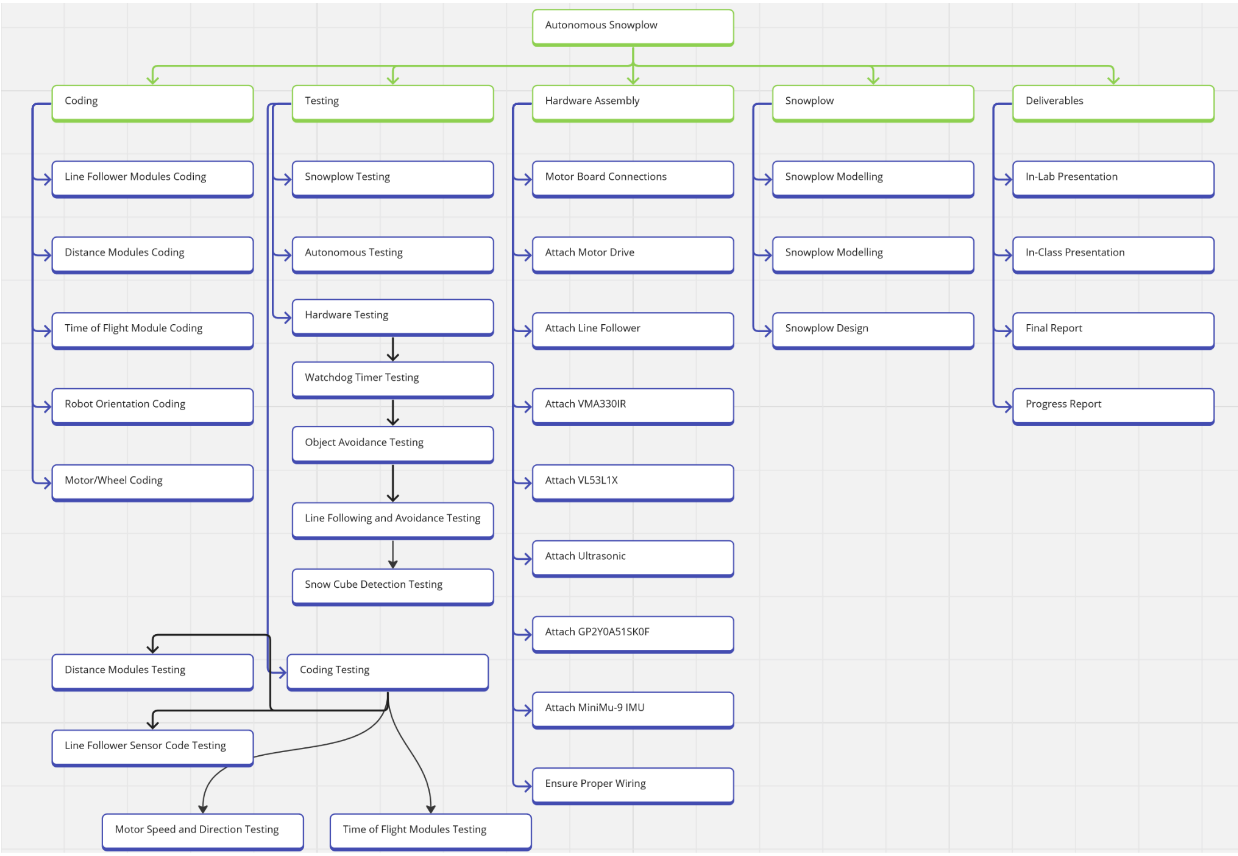



*Figure 1: WBS figure displaying list of deliverables based on the requirements*

## 2.3 Testing Plan

### 1. Snowplow Testing

- **Test 1: Snow Cube Removal**
  - **Objective:** Verify that the robot effectively removes snow cubes from the practice zone.
  - **Test Procedure:** Place robot in a practice zone with clearly defined borders and limited obstacles. Place a small amount of snow cubes in the borders. Observe and measure time taken to clear snow cubes from the practice zone as well as the force that the robot can apply on the snow cubes.
  - **Success Criteria:** The robot must clear 80% or more of the snow cubes within the specified time limit and must be able to apply a force strong enough to push cubes at a speed of at least (don't know snow cube weight currently, no clue how fast should be able to push).

### 2. Autonomous Testing

- **Test 2: Autonomous Operation**
  - **Objective:** Ensure that the robot operates autonomously without human intervention.
  - **Test Procedure:** Place the robot within a practice zone and monitor its performance and correct operation.
  - **Success Criteria:** The robot must be able to navigate obstacles and identify snow cubes without any human intervention.

**3. Hardware Testing**

- **Test 3: Watchdog Timer Testing**
  - **Objective:** Verify that the watchdog timer on the Arduino functions correctly.
  - **Test Procedure:** Intentionally disrupt the robot's/Arduino's operations and assess if the watchdog timer successfully restarts.
  - **Success Criteria:** The robot should recover and continue operation without manual intervention.
- **Test 4: Object Avoidance Testing**
  - **Objective:** Ensure the robot avoids obstacles as required.
  - **Test Procedure:** Place the robot within a practice zone with multiple non-moving obstacles and evaluate its ability to detect and avoid them. Then while the robot is avoiding non-moving obstacles ensure that the robot can also asset and deal with moving obstacles.
  - **Success Criteria:** The robot must successfully detect and avoid obstacles without leaving the practice area or getting stuck.
- **Test 5: Line Following and Avoidance Testing**
  - **Objective:** Verify that the robot can follow the border of a practice area, avoid leaving the practice area and detect edges of the practice area when not following.
  - **Test Procedure:** Create a practice area with black lines. Have the robot tested on following the lines, detecting edges, and staying within the practice area. Evaluate the robot's behavior.
  - **Success Criteria:** The robot must follow lines and avoid leaving the practice area.

**4. Coding Testing**

- **Test 7: Motor Speed and Direction Testing**
  - **Objective:** Verify that the robot's motors respond correctly to movement and speed commands.
  - **Test Procedure:** Issue motor speed and movement awards and assess the robot's ability to track its own speed.
  - **Success Criteria:** The robot must be able to control its speed, direction.
- **Test 8: Distance Modules Testing**

- **Objective:** Confirm the accuracy and reliability of distance measurement modules.
- **Test Procedure:** Use snow cubes, obstacles and objects at different distances and assess the modules' accuracy.
- **Success Criteria:** The distance modules must be able to provide accurate information to the Arduino about the distances and placements of objects using the specific design algorithms.

- **Test 9: Line Follower Sensor Code Testing**
    - **Objective:** Verify that the line follower sensor code functions as intended.
    - **Test Procedure:** Use printed black lines to ensure that the line follower module is sensing lines as intended.
    - **Success Criteria:** The sensor should be able to accurately determine the difference between a black line and the rest of the group and provide that information to the Arduino.

**5. Time and Operation Testing**

- **Test 10: Time Testing**
    - **Objective:** Ensure the robot completes snow clearing within the specified time limit.
    - **Test Procedure:** Place and activate the robot in a specified practice area with determined snow cubes and obstacles. Measure the time it takes to clear the testing area of all snow cubes.
    - **Success Criteria:** The robot must finish snow clearing within 5 minutes.
- **Test 11: Operation Testing**
    - **Objective:** Confirm that the robot operates without human intervention.
    - **Test Procedure:** Monitor the robot's operation and assess if it can function autonomously.
    - **Success Criteria:** The robot must perform all tasks without human intervention.

# Schedule

## 3.1 List Of Activities

Here is a list of activities based on the deliverables:

**Building Snowplow Hardware Activities:**

- Attaching line follower to robot
  - Connect the line follower to the arduino and ensure correct response. Then attach the line follower to the robot and make sure it is attached firmly so it won't fall during movement.
- Attaching ultrasonic sensor to robot
  - Connect the ultrasonic sensor to the arduino and ensure correct response. Then attach the ultrasonic sensor to the robot and make sure it is attached firmly so it won't fall during movement. Two ultrasonic devices will be connected to the sides of the robot, one on the right side and one on the left side.
- Attaching time of flight sensor to robot
  - Connect the time of flight sensor to the arduino and ensure correct response. Then attach the time of flight sensor to the robot and make sure it is attached firmly so it won't fall during movement. Two time of flight sensors will be connected to the front of the robot, one on the left and one on the right.
- Connecting the motors to the Ardunio
  - Connect the motors to the arduino and ensure a maximum speed of 30 cm/s. Also make sure to allow for the right wheels and left wheel to be controlled independently by connecting them to two different pins.

**Handler Code for Sensors and Motors Activities:**

- Handler code for line follower
  - Write a handler code for the line follower module that will read the value and then return whether the reading of the module indicates it is over a black line or not.
- Handler code for ultrasonic sensor
  - Write a handler code for the ultrasonic sensors that will read the sensor's values and return if it detects an object and specifies which direction it has detected it from.
- Handler code for time of flight sensor
  - Write a handler code for the time of flight sensors that will read the sensor's values and return if it detects an object and specifies which direction it has detected it from.
- Handler code for motors
  - Write handler code for the motor that will take an input, right or left, and turn the motor into that direction.

**Object Avoidance Activities:**

- Detecting and avoiding stationary objects
  - Develop an algorithm that will use the time of flight, ultrasonic and motor's

handlers to detect and avoid obstacles.

**Line Avoidance Activities:**

- Detecting and avoiding line
    - Develop an algorithm that will use the line follower, ultrasonic and motor's handlers to detect and avoid obstacles.

**Finishing Snowplow Activities:**

- Build and attach snow plow
    - Build the snow plow that matches the requirements and attach to the robot firmly so it won't fall.
- Watchdog timer and end-end testing
    - Use a watchdog timer in the code. Also, test the whole robot on a test arena to ensure the predicted functionality is correct.

**Progress Report Activities:**

- Finish progress report
    - Write and complete the progress report.

**Presentation Activities:**

- Finish final report
    - Write and complete the final report.

**Final Report Activities:**

- Finish and practice presentation
    - Create a presentation and practice presenting it.

## 3.2 Schedule Network Diagram

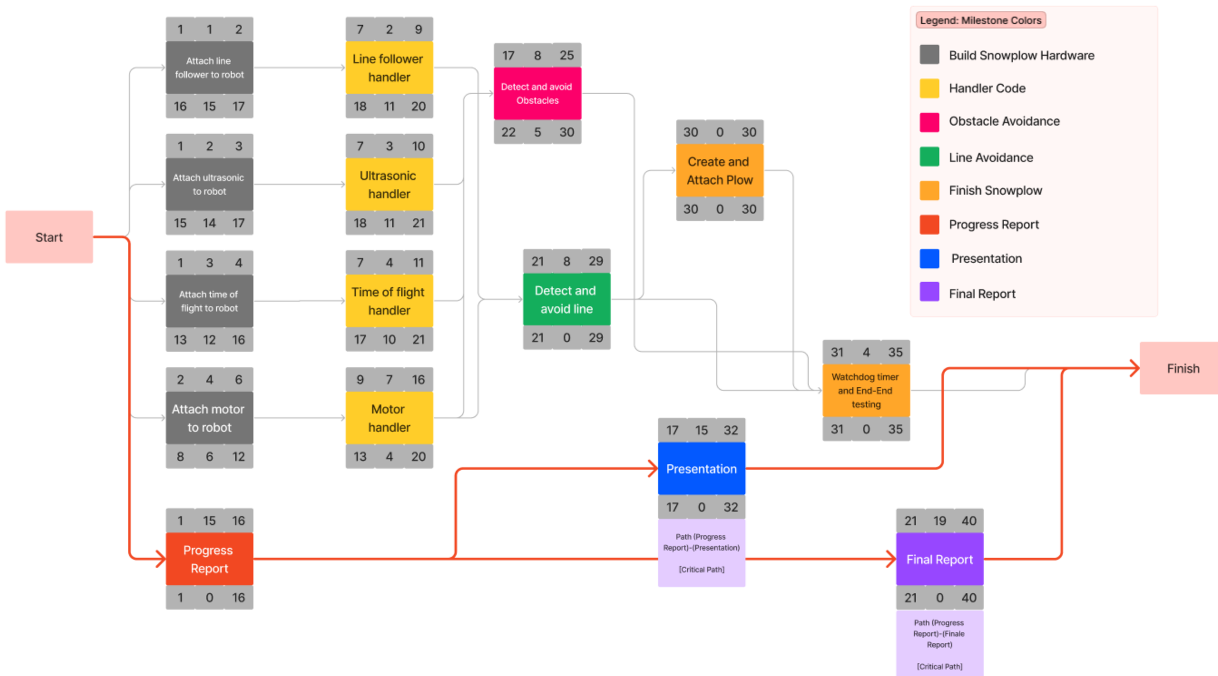



*Figure 2: Schedule Network Diagram displaying the scheduled activities and their dependencies. NOTE, all the relationships are finish-to-start.*

## 3.3 Gantt Chart

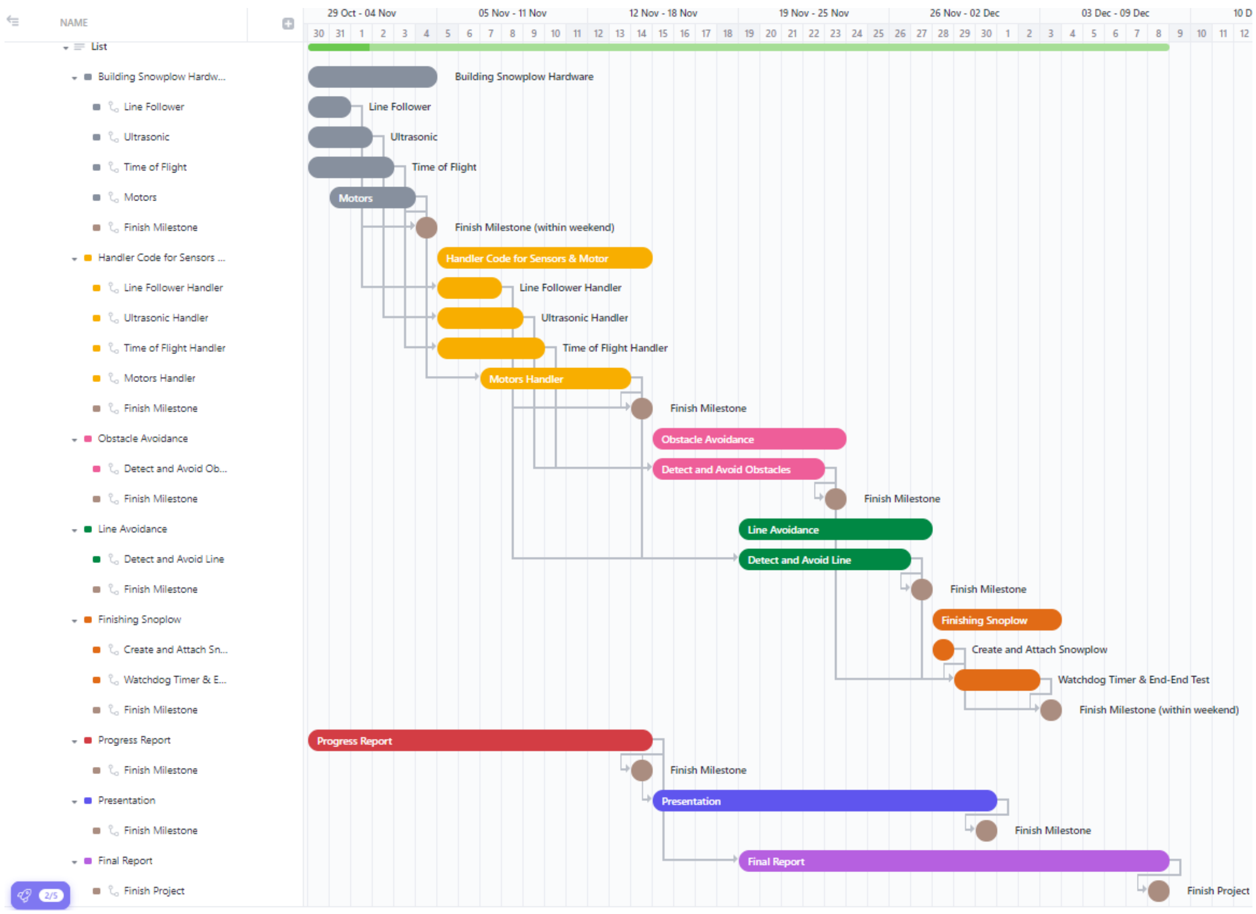



*Figure 3: Gantt Chart displaying the timeline in which the Zaffre team will follow to complete the milestones and activities.*

It is assumed that all project milestones will be completed during the course's allocated lab hours. With the exception of the progress report, final report, and presentation, the team may work on these milestones during lab hours and on their own time. Regarding other milestones planned for completion during the lab hours, it is anticipated that the responsible group or individual will be given the task of transporting the lab equipment home in the event of any problems or setbacks that could potentially prevent the fulfillment of a given milestone or task. This ensures the successful conclusion of the milestone or task during their non-class hours prior to the following lab session and that the team upholds the established timeline provided in *Figure 3*.

# 4.0 Cost

## 4.1 Cost Baseline Figure

Based of the project description in which Labs 7-11 (total of 5 labs) is used for the development of the robot, the cost would be estimated to be ($50 per developer/hour $\cdot\ 3(members)\ \cdot\ 4(hours\ per\ lab)\ \cdot\ 5(working\ labs)\ +\ 500(cost\ of\ kit\ )$ around $3500. These values were calculated using the provided assumption of the project document description. A cost baseline figure is provided estimating the cost over time of the project period. These are initial estimated values and don't actually reflect the real cost of the project but will be updated as the project progresses. (This will be updated later on.)

| | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| 1 | # of Members | # Hrs/lab | # of labs | Cost of kit ($) | Member/hr ($) | Total cost ($) |
| 2 | 3 | 4 | 5 | 500 | 50 | 3500 |

*Figure 4: Estimated budget at completion based off project document description*

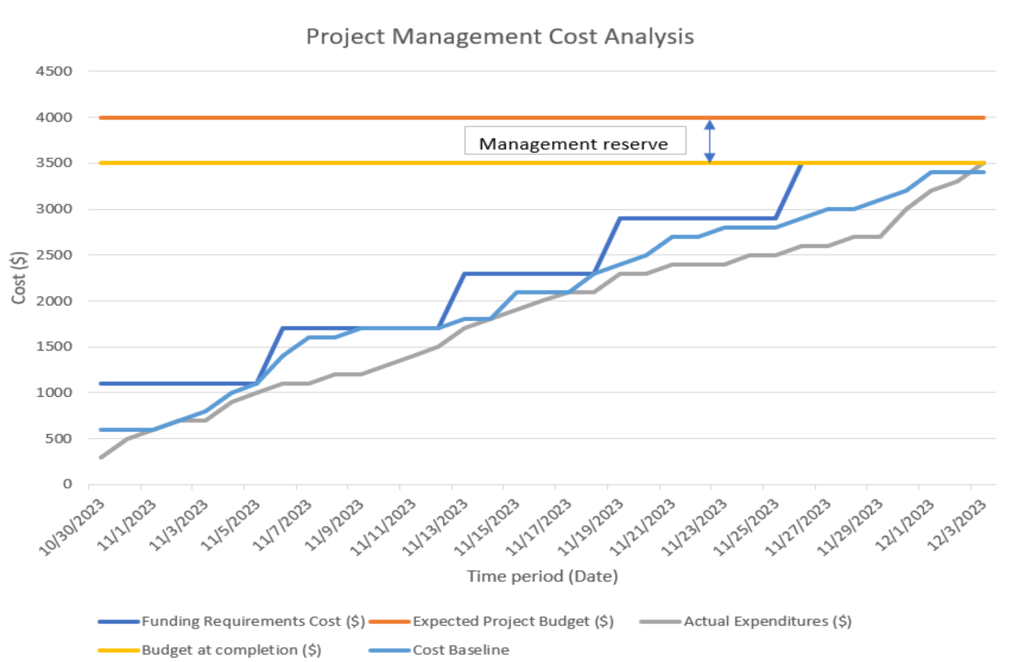



*Figure 4: Cost baseline figure for the project management analysis*

# 5.0 Human Resources

## 5.1 Assignment Matrix

*Table 1: Responsibility Matrix displaying the person who is responsible and the approver for this project's milestones and activities.*

| Milestone | Activity | Responsible | Approver |
|---|---|---|---|
| Building Snow Plow Hardware | Attaching line follower to robot | Chris | Jakob |
| | Attaching ultrasonic sensor to robot | Abdalla | Jakob |
| | Attaching time of flight sensor to robot | Jakob | Abdalla |
| | Connecting the motors to the Ardunio | Jakob | Chris |
| Handler Code for Sensors and Motors | Handler code for line follower | Chris | Abdalla |
| | Handler code for ultrasonic sensor | Abdalla | Jakob |
| | Handler code for time of flight sensor | Jakob | Abdalla |
| | Handler code for motors | Abdalla | Chris |
| Object Avoidance | Detect and avoid stationary objects | Abdalla | Chris |
| Line Avoidance | Detect and avoid line | Chris | Abdalla |
| Finishing Snowplow | Build and attach snow plow | Chris | Jakob |
| | Watchdog timer and end-end testing | Jakob | Chris |
| Progress Report | Finish progress report | All | All |
| Final Report | Finish final report | All | All |

| | | | |
|---|---|---|---|
| Presentation | Finish and practice presentation | All | All |