Autonomous Snowplow

11.18.2022

Team: Old Gold

L3-Group4

GitHub:https://github.com/SYSC4805-Fall2022/sysc4805_term_project-oldgold_l3g4

Wilson Amoussougbo (101119293)

Tyler Mak (101108389)

TABLE OF CONTENTS

1. Project Charter

1.1.Objective

The following document written by the members of Old Gold is a progress report for the course SYSC4805 and is based on the conception of an Autonomous Snowplow. The goal of the project is to materialize a robot whose task will be to clear off snow in an enclosed area without hitting any fixed or moving obstacles. The project is composed of diverse components and its completion consists of syncing multiple technologies to ensure a functional assembly. The purpose of the following document is to present a project progress report including an updated version of the project proposal as well as UML diagrams depicting the behavior of the overall system.

1.2.Milestone and Final deliverables

This section of the proposal covers the project milestones and final deliverables. The following table lists all the major dates for assignments and team meetings.

Date	Event	Date	Event
September 15th, 2022	Lab 1	November 04th, 2022	-Perimeter Detection -Unit Testing
September 23rd, 2022	Lab 2	November 11th, 2022	-Progress Report -Obstacle Detection -Unit Testing
September 30th, 2022	Lab 3	November 18th, 2022	N/A
October 07th, 2022	Lab 4	November 25th, 2022	-System Testing
October 14th, 2022	Project Proposal	December 02nd, 2022	-Integration Testing -Project Demo -Bug Fixing
October 21st, 2022	-Robot Speed -Unit Testing	December 09th, 2022	-Final Report

Table 1: Project Milestone and Final Deliverable Dates

2. Scope

2.1. Project requirements

Specifying requirements is a crucial part of any project. In this section, all the engineering requirements for this project will be listed.

1. The robot shall not exceed a speed of 30 cm/s while moving
2. The robot shall be a maximum size of 216 by 252 by 150 mm
3. The robot shall be able to detect the perimeter of the testing space
4. The robot shall be able to detect any obstacles before collision
5. Design a plow to attach to the robot for clearing the “snow”
6. Design a route the robot shall take for clearing the “snow”
7. The robot shall clear the “snow” in the testing area within 5 minutes

2.2. Project life-cycle

Every project can be broken down into different phases and these phases make the path the project will take to get from the start to the finish. The following image below shows our project’s work breakdown structure.

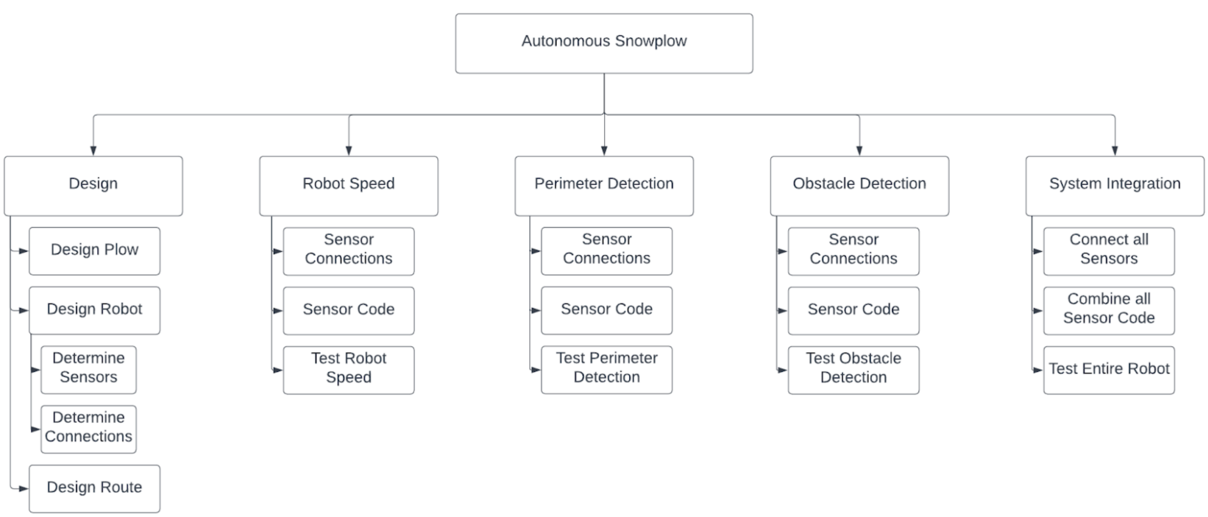

Figure 1: Project Work Breakdown Structure

2.3. Testing

Testing is a fundamental part of our project and a set of testing that covers all the requirements has been put in place. In this section, we will discuss our plan for the unit testing to verify the successful completion of each activity independently from the rest of the project and the integration testing to validate the functionality of the system as a whole.

- UNIT TESTING
  - Design
    1. Take measurement of the robot assembly and verify that it does not exceed the following requirement : width, length, height = {216, 252, 150} mm.
    2. Verify that the plow width and length respectively do not exceed 40mm and 20 mm.
  - Robot Speed
    1. Verify that at maximum speed with minimum friction and no obstacle the speed of the assembly does not exceed 30 cm/s.
    2. Certify that the robot moves at specified speed chosen by the user.
  - Perimeter Detection
    1. Verify that the system generates a signal when sensors detect the closed black path.
    2. Verify that when the signal is generated by the system, the robot changes direction and does not cross the closed black path.
  - Obstacle Detection
    1. Verify that the system generates a signal when sensors detect an obstacle.
    2. Verify that when the signal is generated by the system, the robot changes direction and does not hit obstacles.
- INTEGRATION TESTING
  1. Certify that the robot is able to move in the testing area without hitting any obstacle.
  2. Make sure the plow attached to the robot does not fold or break when clearing off the obstacles.

3. Verify that the plow attached to the robot has a diameter of 42.66mm.

3. Design

3.1. Overall Architecture

The system statechart shows the overall organization of the project with all hardware and software components communicating together. In the following diagram, the major components are the robot assembly, the arduino processor, the motor driver and the collection of sensors. All the modules of the system are mainly connected using their respective GPIO pins except for the Time of Flight sensor which uses I2C communication protocol.

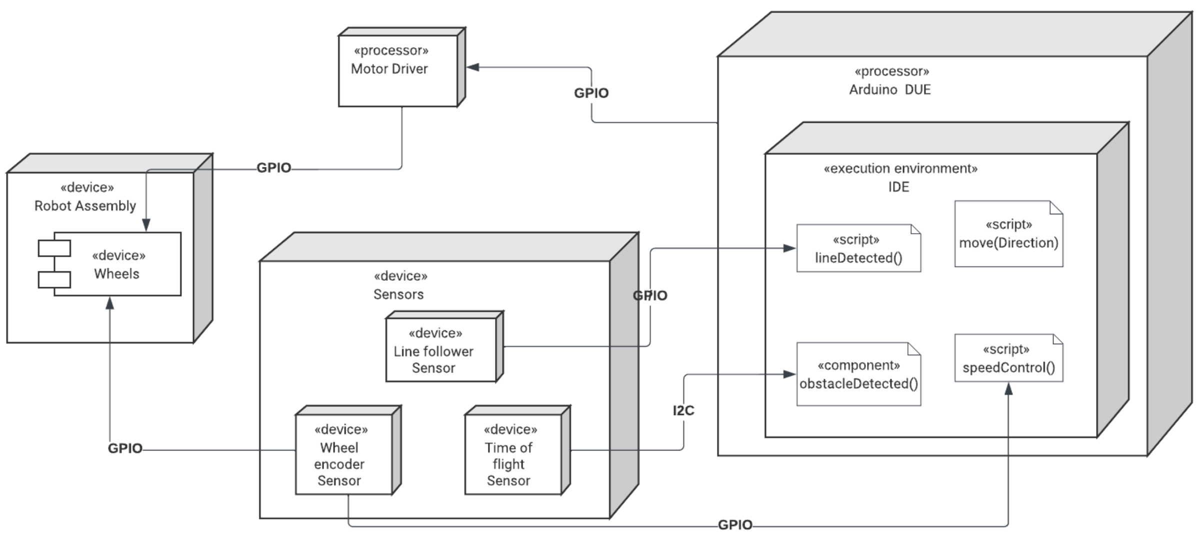

Figure 2: System Architecture

3.2. System Statechart

The statechart or state machine diagram shows the different states of an entity and how it transitions from state to state. It demonstrates how the entity will act depending on the state it is in. The following image below shows our project's statechart diagram for the completed robot.

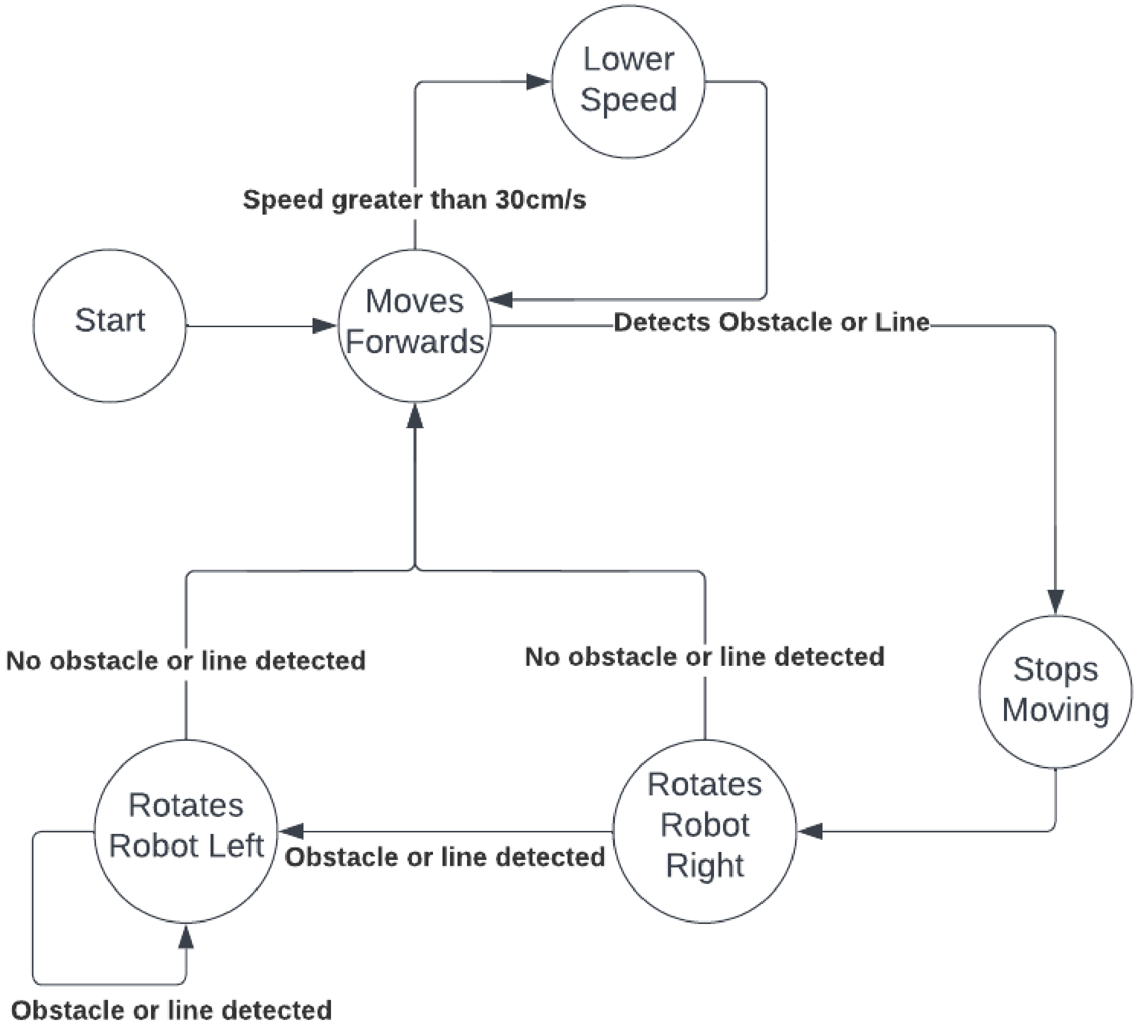

Figure 3: Project Statechart

3.3. System Sequence

The following figure shows the sequence diagram highlighting the major events in the project. In fact, when the user starts the program, the robot assembly starts moving in a loop until an obstacle or the black path line is detected by the respective sensors. At that moment, the robot's speed decreases and it changes direction.

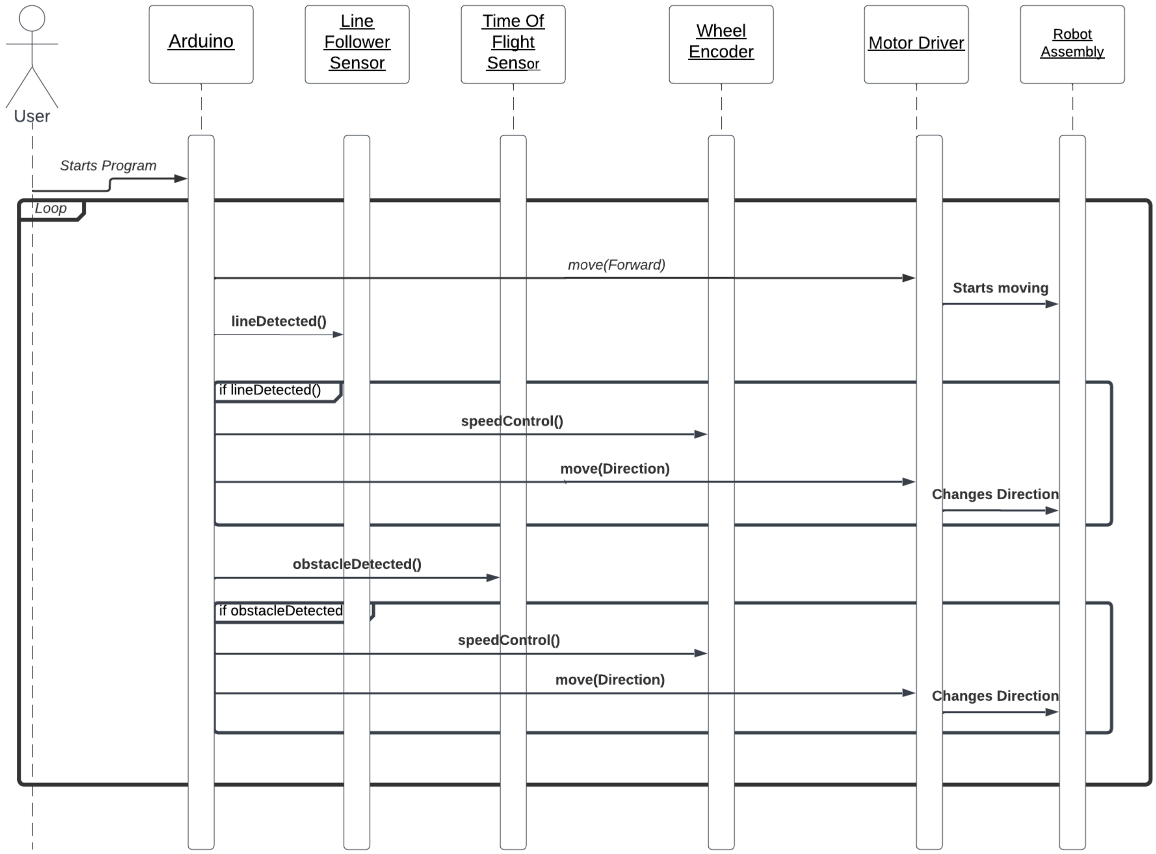

Figure 4: Sequence Diagram

3.4. Watchdog Timer

The watchdog timer is a hardware or software timer that is for detecting and recovering from problems in the code. An example of a problem may be getting stuck in a loop indefinitely. In order to recover, the watchdog timer has a timeout value which if reached resets the main program. This ensures that the system can recover from problems and can reset from a stuck state. For our project we will implement a watchdog timer through software using the Arduino Due's included watchdog timer library. The following code will be used to enable the watchdog timer and set the timeout to be 1 second.

```
void watchdogSetup(void){


}
void setup() {
  watchdogEnable(1000); //enable watchdog timer with timeout of 1s
```

```
}
void loop() {
  watchdogReset();
}
```

Figure 5: Watchdog Timer Code Implementation

4. Schedule

4.1. Schedule network diagram

A schedule network diagram is used to determine the relationships between activities. The following image below describes the sequential and logical relationships between our project activities.

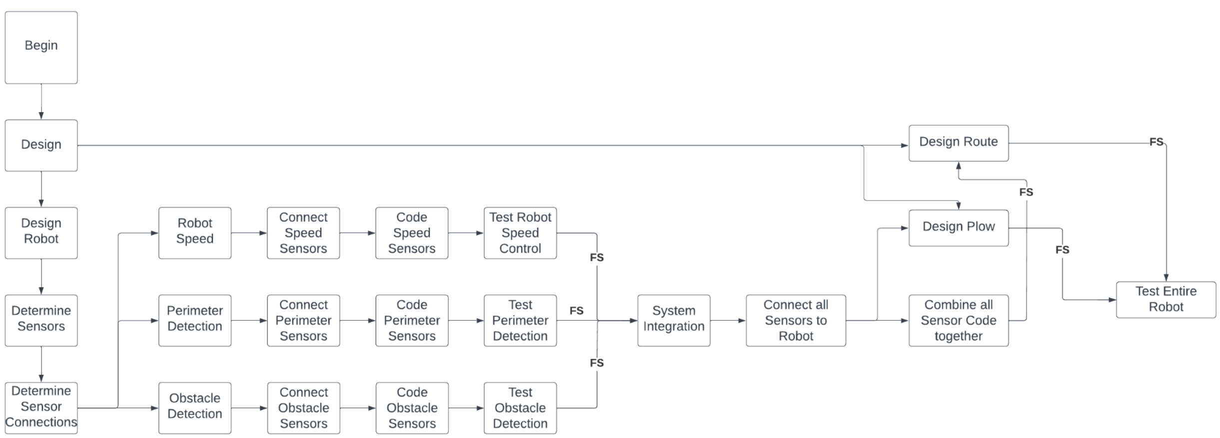

Figure 6: Project Schedule Network Diagram

4.2. Progress

Since the proposal we have made good progress. Looking at the cost baseline diagram below in Section 5.2, we are ahead of schedule and over budget. Currently, we have implemented the line follower sensor and time of flight(ToF) sensor on our robot. We made the necessary connections and attached the sensors to the robot's chassis. The code for the line follower sensor is functioning as intended, data values are being read and we have coded how the robot should respond corresponding to the data values. The testing we have performed gave us mixed results. The robot operated correctly when we physically used black and white paper under the sensor but when testing on the ground, the sensor seemed to have failed to detect the black tape. This issue may be due to the sensor's range. The code for the time of flight sensor is still in progress, data values are being read but we are still working on the response the robot should have.

4.3. Upcoming events

The following weeks, we will need a plan to decide on what tasks we will be working on as well as the amount of time we should spend working on each activity. The following image below is a Gantt chart for the entirety of the project.

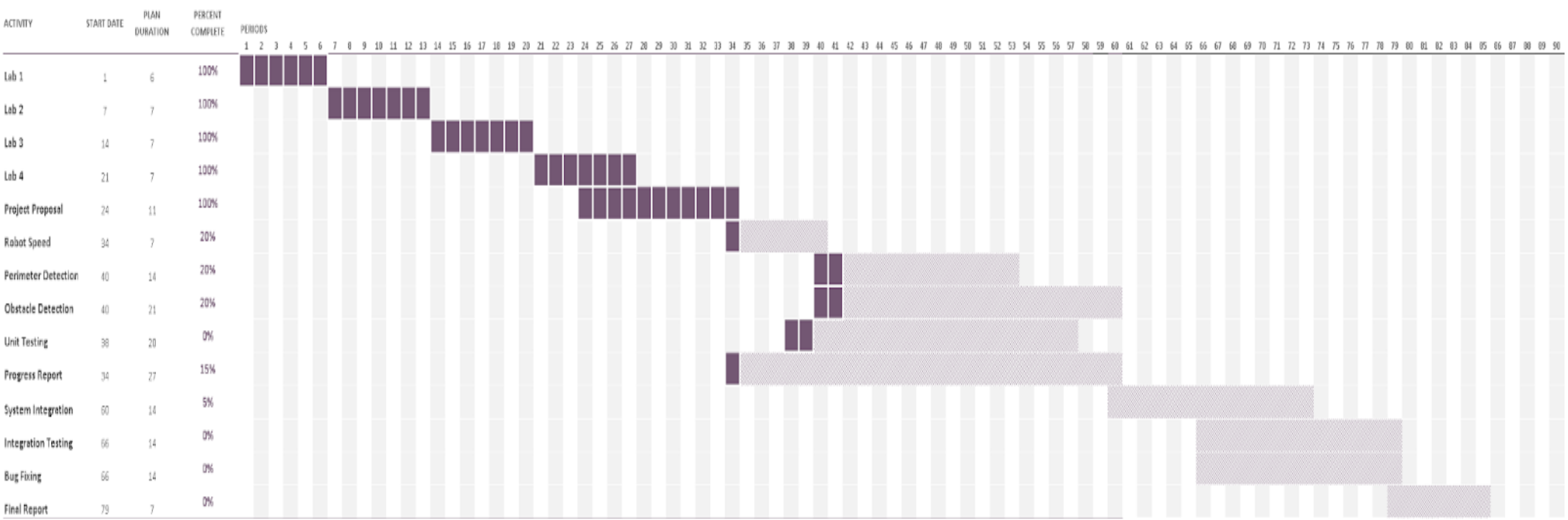

Figure 7: Project Gantt Chart Diagram

This Gantt Chart shows the activities on the left along with the start date, plan duration, and percent completion. The start date and plan duration is measured in periods where one period is equivalent to one day. The start date specifies which period the activity starts and the plan duration specifies the number of periods the activity will take to complete. On the right is a visual representation of the information on the left. The Gantt chart starts from period 1 (September 9th) and increases by 1 every cell. A dark purple cell represents the activity completion rate and a light purple cell represents the plan duration of the activity that is not yet completed.

5. Cost

5.1. Proposal Cost Baseline

The following figure represents a cost baseline diagram of the overall project at the project proposal point of time.

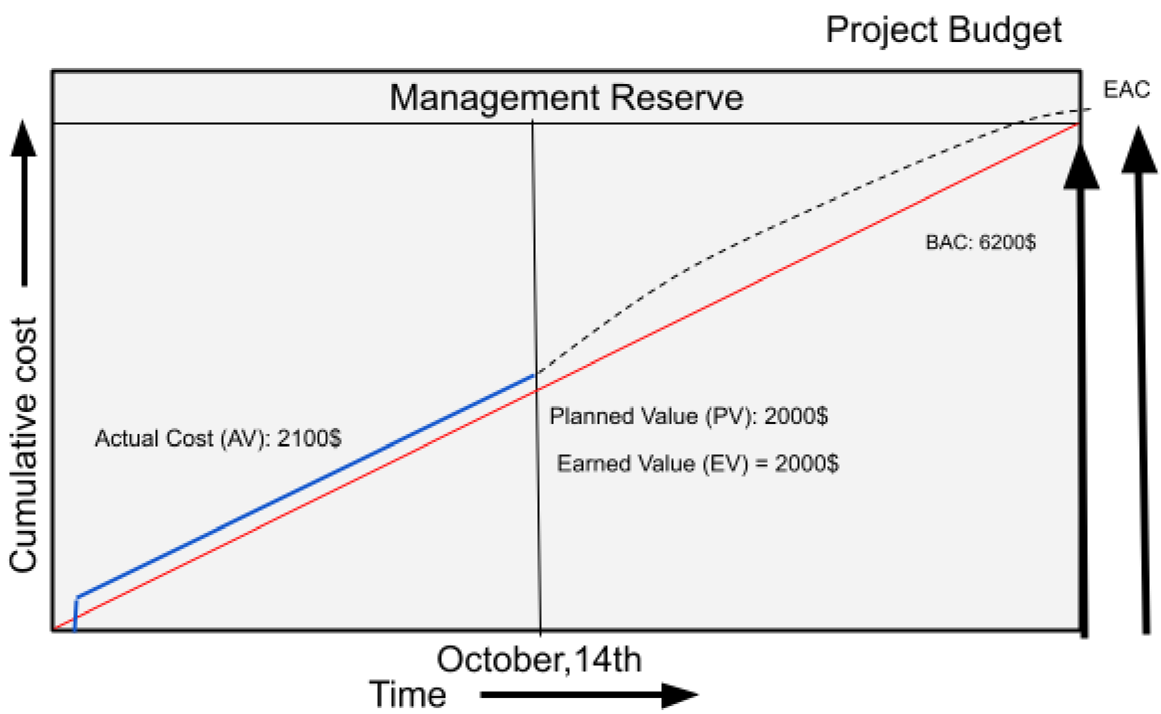

Figure 8: Previous Baseline Cost Diagram

To calculate the Actual Cost we assumed the value of the robot kit to be $100 and added it to our salary provided as $50/hr/developer. Assuming we have worked around 20 hours per person at this point. The Planned Value and Earned Value was calculated similarly to the Actual Cost without considering the $100 robot kit. To calculate the BAC, we assume we will put in 62 hours per person total and multiply it by our salary.

AC = 100 +2 x (20 hrs x 50$/hr) = 2100 $

PV = 2 x (20 hrs x 50$/hr) = 2000 $

EV = 2 x (20 hrs x 50$/hr) = 2000 $

BAC = 124 hrs x 50$/hr = 6200 $

5.2. Progress Report Cost Baseline

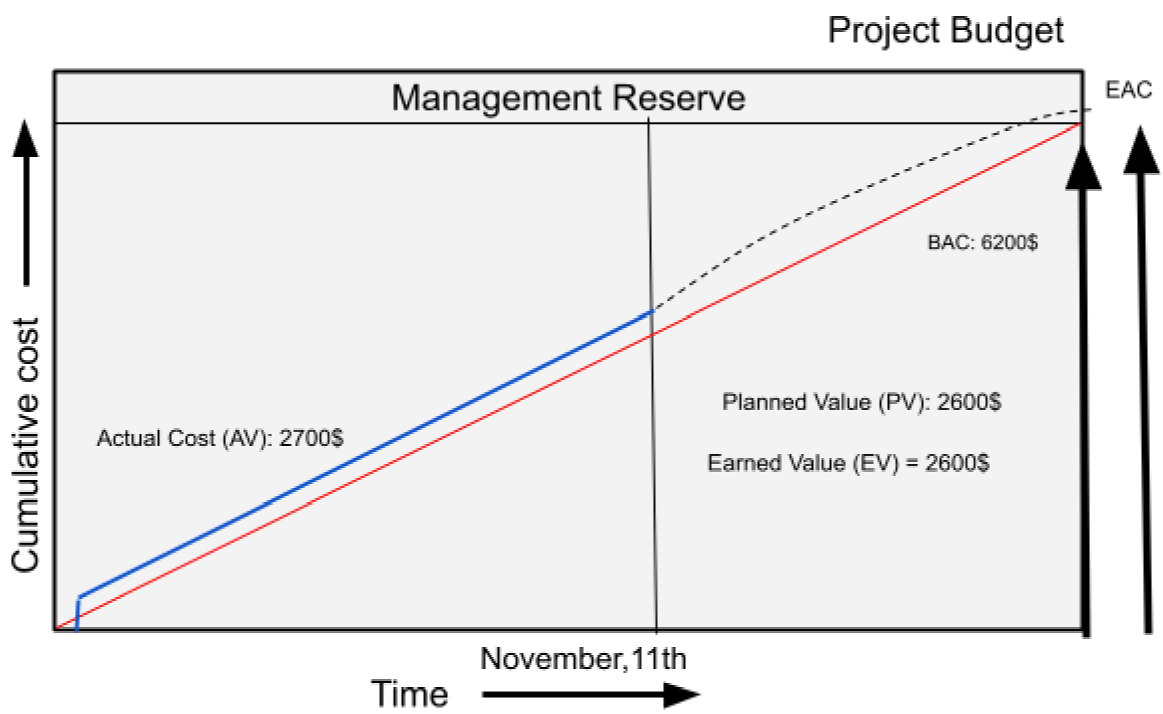

Figure 9: Current Baseline Cost Diagram

To calculate the Actual Cost we assumed the value of the robot kit to be $100 and added it to our salary provided as $50/hr/developer. Assuming we have worked around 26 hours per person at this point. The Planned Value and Earned Value was calculated similarly to the Actual Cost without considering the $100 robot kit. To calculate the BAC, we assume we will put in 62 hours per person total and multiply it by our salary.

AC = 100 +2 x (26 hrs x 50$/hr) = 2700 $

PV = 2 x (26 hrs x 50$/hr) = 2600 $

EV = 2 x (26 hrs x 50$/hr) = 2600 $

BAC = 124 hrs x 50$/hr = 6200 $

5. Human Resources

This section of the proposal will cover the project human resources and the person responsible and an approver for each activity will be assigned.

Activities	Responsable	Approver
Connect Speed Sensors	Wilson	Tyler
Code Speed Sensors	Tyler	Wilson
Test Robot Speed control	Tyler	Wilson
Connect Perimeter Sensors	Wilson	Tyler
Code Perimeter Sensors	Wilson	Tyler
Test Perimeter Detection	Tyler	Wilson
Connect Obstacle Sensors	Tyler	Wilson
Code Obstacle Sensors	Wilson	Tyler
Test Obstacle Detection	Tyler	Wilson
Combine all Sensors to Robot	Wilson	Tyler
Combine all Sensor Code together	Wilson	Tyler
Design Plow	Tyler	Wilson
Design Route	Wilson	Tyler
Test entire Robot	Wilson	Tyler

Table 2: Responsibility Assignment Matrix