System Design and Development

Team Name Pending

Kevin Tang, Alvin Zou, Ella Zhao

Table of Contents

0. Revision history

Version	Date	Details
1.0	2/1/2024	Document is created
1.1	2/4/2024	Finished draft
1.2	2/25/2024	Requirement matrix, Objective tree and risk management added. Figure caption check
1.3	3/17/2024	Modified document based on the given feedback, including adding a risk likelihood matrix and a larger gantt chart, Also expanded section 7 by adding more details to the test plan.
1.4	4/1/2024	Changed Testing plans for 7.1Requirements Validation
1.5	4/9/2024	Testing Results added

1. Introduction

This System Design and Development Document (SDDD) includes the requirements for our robot, concept of operation, detailed design, and the verification plan.

2. Problem Statement

In an ever increasingly fast paced environment we live in today, people's time is becoming ever more valuable. This is especially true at CMU, where hard working students are in a constant struggle between balancing their studies, social activities, and personal well being. For many students, they will dedicate their meal times to work; thus, it is important that they can collect their meals quickly. However, with an increasing student body size, lines at the dining

locations across campus are seeing longer and longer wait times. Furthermore, students need to travel from their working location to the dining location, which may take a long time. As the time consumption for getting their meal increases, students become less inclined to get food. Therefore, we are proposing a solution to this problem through a robot that can pickup and deliver food orders for students.

3. Motivation

Using a robot that can pickup and deliver food orders for students solves the key problem described above. It frees the student from the need to leave their working station and go to the restaurant to get their meal. Instead, now they can let the robot do the labor of picking up the order and delivering it to them. This saves the students' time and enables them to perform more meaningful work. A key feature of our robot is that it can climb stairs. This is to make sure that the robot can deliver food to the student no matter where they are in the building.

4. System Description

We are building a food delivery robot that specializes in delivering indoors. The robot would need to automatically navigate between locations and climb stairs, given a predetermined, complete map. For our project, we will focus on the stair climbing aspect of the robot. The robot consists of a stair climbing system, a 2-wheel, 2 motor drivetrain with a caster wheel on a rectangular chassis, and a perception system. It will also have an easily accessible container to place food. The stair climbing system contains a frame that can be vertically and horizontally

displaced using pulleys, which mimics the movement of taking steps. The perception system consists of a realsense camera that provides stereo and depth images.

5. Requirements

5.1 Assumption

Since our robot will be interacting with humans, we must make certain assumptions on their behavior to govern how our robot will interact with them. The first assumption that we make is the assumption of honesty and competence of the users. We assume that the cashier will give the robot the correct order, and not unintentionally give the wrong order or purposely try to trick the robot. The food receiver will also take the food off the robot when the robot arrives at the destination. We also assume that when the robot navigates around the building, people passing by will not intentionally try to harm the robot, or purposely cause trouble by blocking its passage or stealing the food. We believe that these assumptions will generally hold at CMU.

5.2 Requirement Specification

*** See Scenarios in Appendix 11.1**

Robot Needs

N1. To deliver food across floors, the robot needs to be able to climb stairs while keeping the food balanced.

N2. To deliver the food, the robot needs to be able to plan a path and navigate from one place to another given a map of the building.

5.3 Robot Requirements

From these robot needs, we determined the following requirements:

*** requirements are prioritized in order**

Functional requirements:

R1. Have the ability to climb stairs that are straight and with consistent step height.

R1.1. Ability to correctly identify stairs.

R1.2. Able to carry a payload of at least 200g

R1.3. Keep payload balanced (+- 10 degrees with respect to the ground) while climbing stairs.

R2. Navigate from a starting location to the stairs, and from the stair exit to the goal location, with distance under 10 meters.

R2.1. Perform path planning given a map of the building.

R2.2. Perform localization given a map of the building.

R3. Ability to switch modes from navigation to stair climbing and vice versa when entering/exiting the stairs.

Nonfunctional requirements:

R4. Total cost is less than $2500

R5. The robot can continuously operate for half an hour after being fully charged

R6. The moving speed of the robot can match human-walking speed (at least 0.6m/s)[2]

R7. The robot can climb the stairs at a speed faster than 15 seconds/ stair

R8. Robot failure rate lower than 1 failure per 100 stairs climbed, on average

R9. Identify stairs with 95% accuracy rate

R10. Robot can successfully navigate from one location to another with a distance <10 meters apart.

5.4 Extensions

If we are able to build our robot that satisfies the above requirements with time to spare, we can work towards satisfying these extensions listed below.

E1. Have the ability to climb stairs that have turns and with different step heights.

E2. Have the ability to detect and avoid obstacles.

E2.1. Have the ability to avoid obstacles when navigating on the floor by re-planning the robot trajectory.

E2.2. Have the ability to avoid obstacles when climbing stairs by stopping and waiting until the obstacle is cleared.

E3. Have the ability to navigate from one location to another that is >100 meters away.

E3.1. Have more robust path planning and localization subject to longer trajectories and potential obstacles.

6. Design

6.1 System Sketches and Scaled drawing

6.1.1 Overall Design:

The robot consists of two main structures:

A pair of symmetrical stair climbing mechanism on the side, made of a piece of 8020 extrusion for both the x-axis rail and structural supports, as well as a piece of acrylic frame to fit the geometry of the stairs;

And a chassis made of two pieces of acrylic plates, with 3d printed parts in between to provide structural support as well as to hold the motors and the z-axis rods, which the stair climbing mechanism rides on.

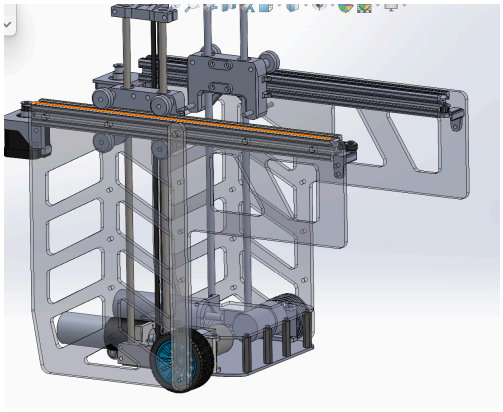


Figure 2.1 CAD model of the Overall Design

6.1.2 Stair Climbing Mechanism:

The mechanism is able to move along the x-axis (horizontal) and z-axis (vertical). The x-axis rail consists of a slide block with rollers riding on a piece of 8020 extrusion. The slide

block is attached to a timing belt, which is then powered by a stepper motor mounted on the side frame to provide relative motion between the slide block and the side frame.

The slide block also contains linear bearings which ride on the linear rods that are attached to the chassis, which guide the z-axis movement. The slide block is attached to another timing belt in z-axis, and this belt is powered by geared DC motors mounted on the chassis, which can pull the chassis up or down relative to the side frame.

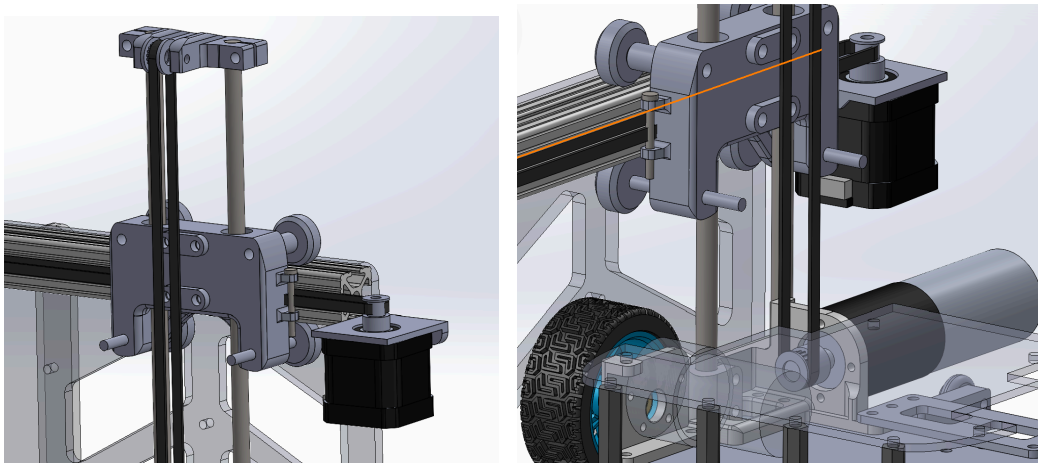


Figure 2.2.1 and 2.2.2 CAD model of theStair Climbing Mechanism

6.1.3 Chassis layout and component placement:

**Since electronics placement is quite flexible, the model only represents the approximate size and does not reflect their exact geometry*

We plan to place the battery and the electronics in between the top and bottom plates of the chassis, the camera on the top plate in the front of the robot, and the food container behind the camera. Since we have a lot of vertical space and do not need to worry about enclosure since the robot will only operate indoors, we may also place additional electronics on the remaining spaces on the top plate if necessary.

Because we will focus on the stair climbing functionality first, the food container is sized to carry only smaller items like hamburgers and sandwiches for now.

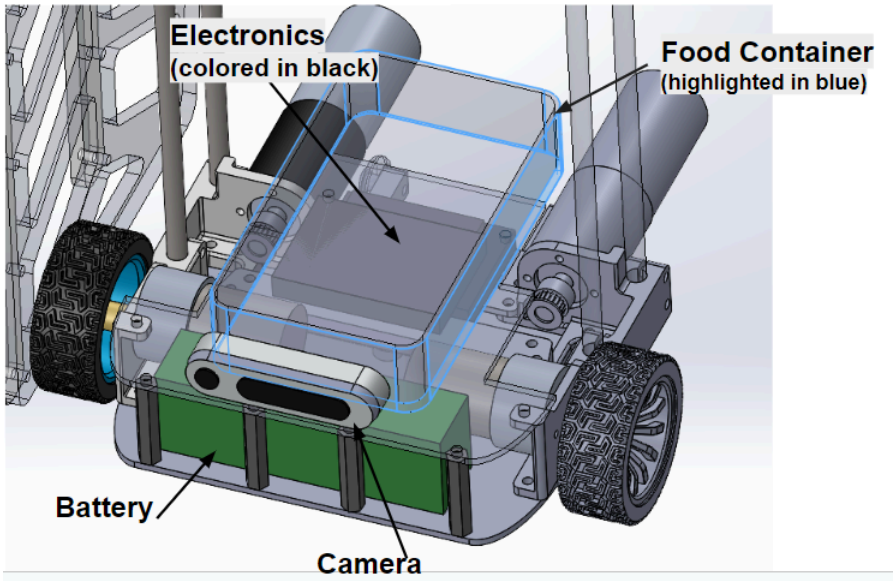


Figure 2.3 CAD model of Chassis Design showing component placement

*** More detailed system sketched and scaled drawings in Appendix 9.1**

6.2 Concept of Operation

The robot has two modes of operation, navigating across the floor and climbing stairs. The sequence of operations that we are interested in will involve the robot moving from a starting location to the edge of the stairs, climbing the stairs, and moving from the exit of the stairs to the goal location. Since the first and third operation is similar, we can analyze them together. Our project focuses on the stair climbing aspect of the robot, and so the task of navigating across the floor will be relatively simple. As such, the distance of the start and goal position from the stairs will be made intentionally short. With a short trajectory, the robot will

not accumulate a lot of drift as it moves from one location to another, which means a simple localization algorithm such as the extended kalman filter will give really good results. Furthermore, we will be assuming that the robot will not encounter any obstacles during its operation. This means that the robot will not need to perform any obstacle avoidance or recalculate its trajectory due to obstacles. To actually perform the navigation, the robot will rely on a two wheel drive system.

Once the robot reaches the bottom of the stairs, it will switch into its stair climbing mode. This transition will be triggered when the camera identifies the stairs and observes that the distance between the robot and the stairs is within some threshold. Within this mode, the robot will first calculate the height of the stairs. This is done through analyzing the stereo and depth images from the camera. Once the height is confirmed, the robot will initiate its climb sequence, which consists of four different steps. First, the robot moves its chassis upwards using its z-axis belt, until it clears the step using the calculated stair height. Second, the robot moves its chassis forwards using its x-axis belt, until it is flush against the step. During these two steps, the robot is supported by the frame. Third, the robot raises its frame using its z-axis belt, until it clears the step. Finally, the robot moves its frame forwards until it is flush against the step. After completing these four steps, the robot will restore its original configuration while successfully climbing one step. The robot then repeats these four steps until it reaches the top of the stairs, at which point it switches back to its navigation mode. Again, this switch is triggered by the camera identifying that the robot has reached the end of the stairs.

6.3 Detailed Scenarios and Sequence Diagrams

In the appendix 9.2.

6.4 Functions and Configuration of each Subsystem

Climbing System: The climbing system enables the robot to climb up the stairs successfully while keeping the payload balanced. It accomplishes this by using an extendable frame that can move up and forwards, along with a retractable chassis.

Chassis: The chassis consists of a two wheel drive system coupled with a third roller wheel for balance. It enables the robot to navigate and adjust its orientation on the ground. The motors also have encoders, providing odometry measurements to aid accurate navigation.

Perception system: The perception system consists of a Realsense RGB-D camera. It is responsible for updating the real-time surroundings to aid navigation and inform the switches between the two modes operation, calculating the stair height to aid stair climbing, and monitoring the robot's status, including tipping over or getting stuck.

6.5 Significant alternatives: Trade studies

Design trade studies: Appendix 9.3

Important component trade studies: Appendix 9.4

6.6 Hardware

Climbing System:

Timing Belts and Pulleys : Powered by motors to provide motion along both x-axis and z-axis

Slide modules: One slide module is attached to the timing belt and connected to the chassis on each side of the robot.

Stepper Motor : Drives the timing belts and pulleys that move along the x-axis

DC motor : Dives the timing belt and pulleys that move along the z-axis

8020 rails: Guide the x-axis movement

Linear rods: Guide the z-axis movement

Chassis:

Geared Motors with Encoder: Two motors are used on the chassis that enables the robot to navigate on the flat floor.

Driving Wheels: Two wheels are utilized on the chassis which enables the robot to navigate on the flat floor.

(passive) Caster Wheel: A caster wheel is attached to the back of the robot.

Perception:

Camera: A camera is attached to the chassis facing forwards, used to complete the perception tasks as mentioned above.

6.6.1 Part List with Fabrication

BOM:

***F – Fabricated parts; P – Purchase parts**

Fabricated Parts Diagram

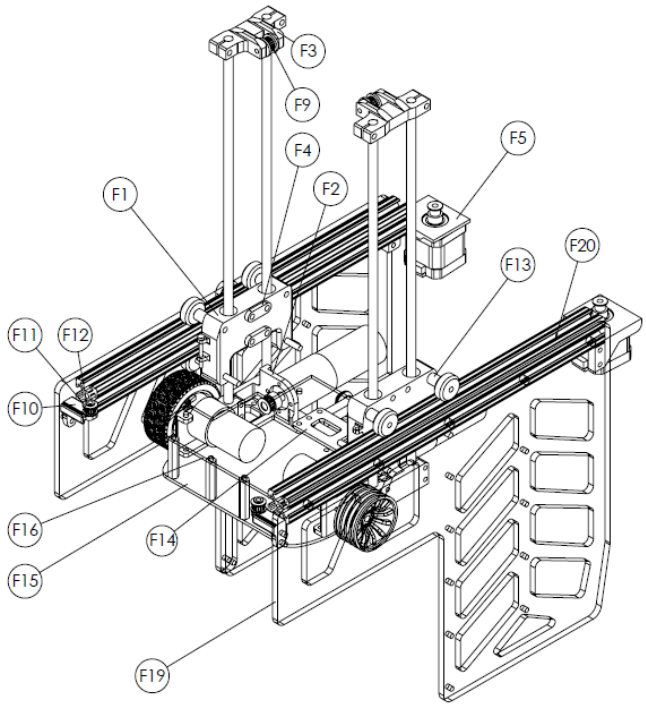


Figure 4.1 Numbered Parts I

Fabricated Parts List:

	PART NAME	QTY.	Fabrication
1	x_axis_slide_block	2	3d print
2	z_axis_motor_holder	2	3d print
3	z_axis_top_pulley_holder	2	3d print
4	Z axis belt clip	4	3d print
5	X axis stepper motor bracket	2	3d print
6	rear_wheel_mount	2	3d print
7	rear_wheel_bushing	2	3d print
8	rear_wheel(passive)	2	3d print
9	2gt-20t PassivePulley	2	3d print
10	passive pulley holder x axis	2	3d print
11	2gt-16t PassivePulley	2	3d print

12	PassivePulleyBushing	2	3d print
13	roller_standoffs	8	3dprint
14	back_reinforce_plate	1	laser cut, 3mm Acrylic plate
15	chassis_bottom_plate	1	laser cut, 3mm Acrylic plate
16	chassis_top_plate	1	laser cut, 3mm Acrylic plate
17	side_frame_reinforcement_bar_1	2	laser cut, 6mm Acrylic plate
18	side_frame_reinforcement_bar_2	2	laser cut, 6mm Acrylic plate
19	Side Frame	2	laser cut, 6mm Acrylic plate
20	8020alumminium extrusion(370mm)	2	Band saw cut

Purchase Parts Diagram:

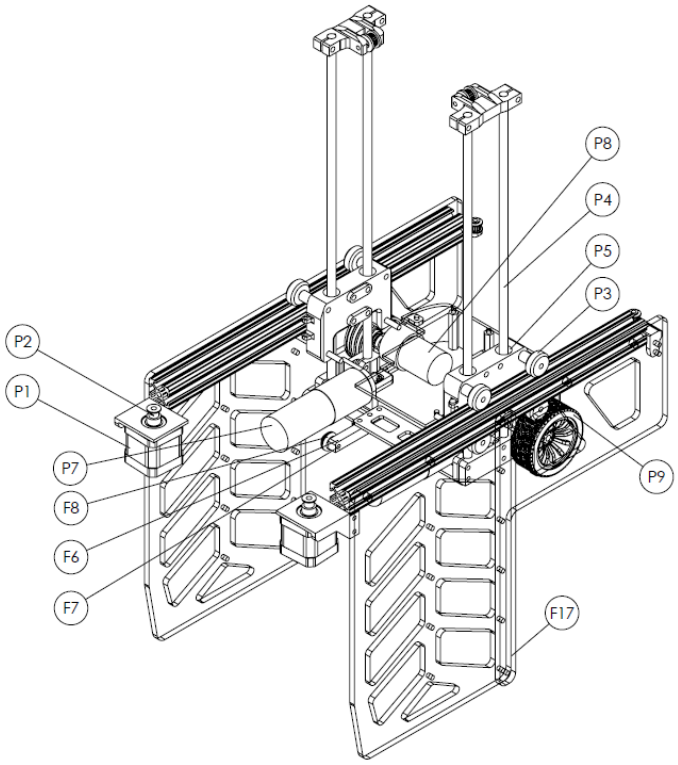


Figure 4.2 Numbered Parts II

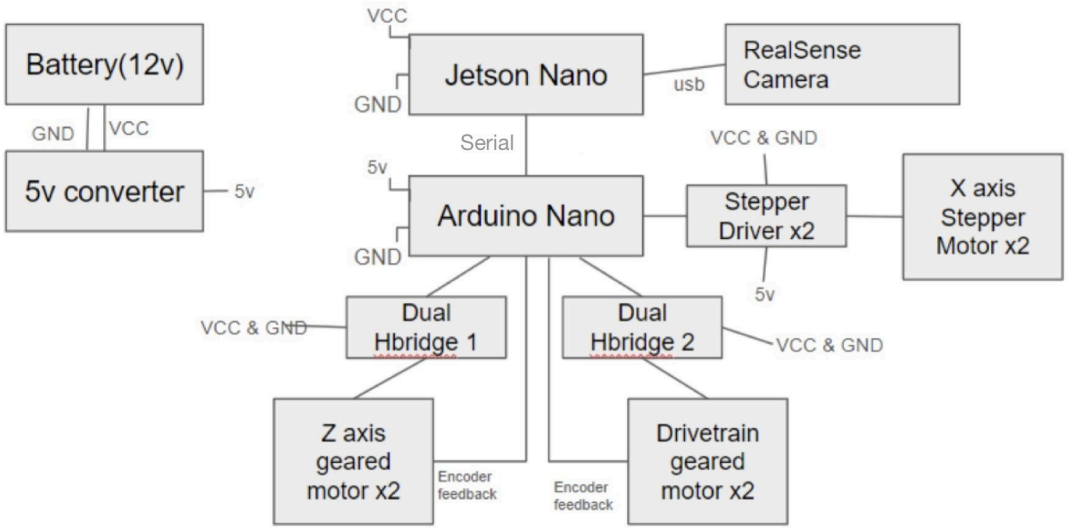
Purchase Parts List

	PART NAME	QTY.	Description
1	NEMA 17 Stepper Motor	2	Powering stair climbing mechanism x axis movement
2	2gt-16t-5mm pulley	2	mounted on stepper motors, pulley for x axis
3	8020_roller_wheel	8	Using rollers on 8020 for x axis structure
4	Steel Rod 8x400mm	4	Steel rods with linear bearings for z axis structure
5	8mm linear bearings	4	
6	2gt-20t-8mm pulley	2	mounted on 555 geared motor, pulley for z axis
7	Planetary gear 555 dc motor w/ encoder	2	Powering stair climbing mechanism z axis movement
8	Spur gear 520 dc motor w/ encoder	2	Drivetrain for navigation on flat floor (non stair environment)
9	65mm Wheel kit	2	
10	7a Dual channel H bridge	2	For controlling dc motors, one for the stair climbing z axis motors, one for drive motors
11	A4988 stepper driver	2	For controlling stepper motor for X axis stair climbing mechanism x axis
12	2gt timing belt	4	Drive belt for stair climbing mechanism (2 belts per axis)
13	Lipo battery (12v 2200mah)	1	Battery for powering the system
14	Arduino	1	Lower level control for drivetrain and stair climbing, taking command from Jetson
15	Jetson Nano	1	Onboard computer, processing computer vision and localization
16	Intel® RealSense Camera 400 Series	1	RGB-D camera for localization and stair identification

***See reference at end of document for links to source of purchase**

6.6.2 Assembly Diagram

6.6.2.1 Wiring Diagram



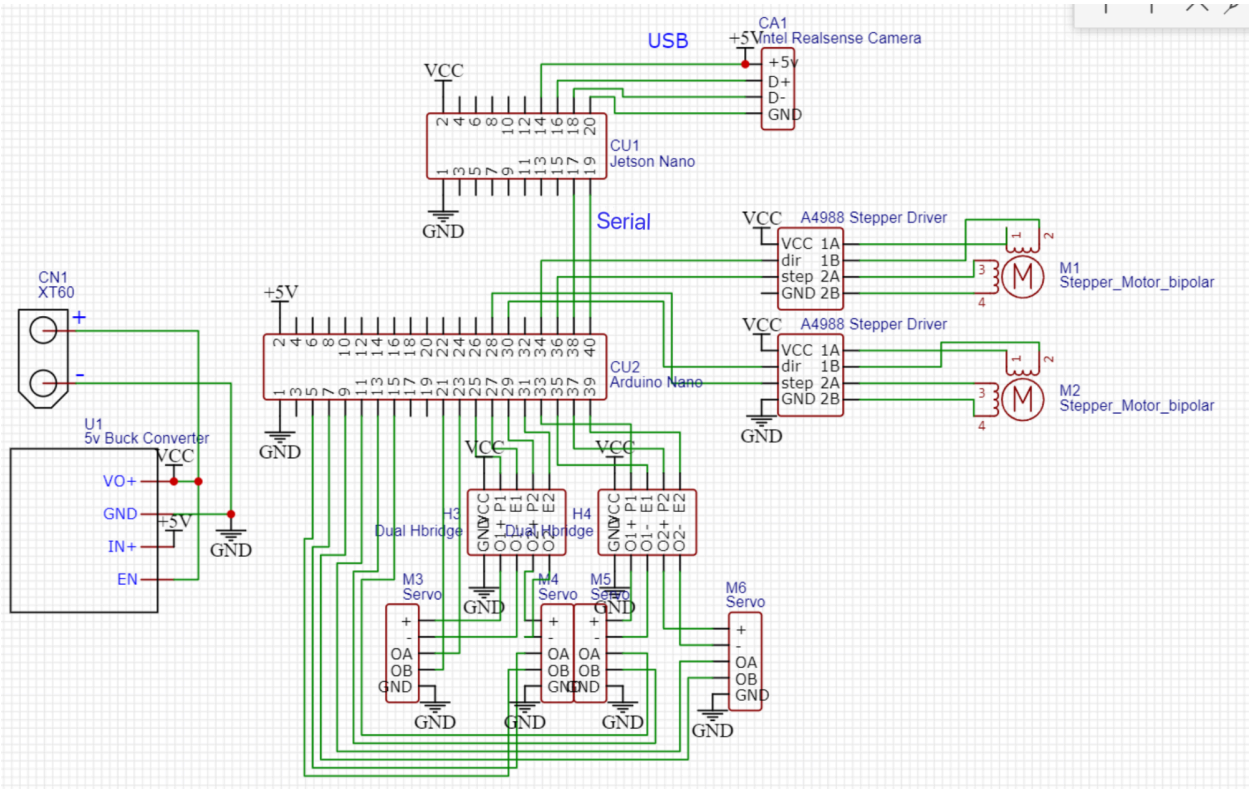


Figure 6.1 Wiring Diagram

6.6.2.2 Detailed Assembly diagram and instructions

***detailed assembly diagrams shown in the Appendix 9.4**

6.7 Software

6.7.1 Software Architecture and Details

The robot has two modes of operation, one for navigation on the ground and one for climbing stairs. As such, the system software is designed in such a way to capture this separation. There is a navigation module responsible for navigating the robot on flat ground. It contains a localization planner and motion planner that sends outputs to the navigation controller, which then commands the chassis motors. There is also a stair climbing module responsible for making the robot climb stairs. It contains a stair climbing planner that sends outputs to the stair climbing controller, which then commands the x-axis and z-axis motors. Finally there is a perception module. This module will be responsible for object detection, pose estimation, and stair height estimation. It then sends these outputs to the respective modules, which are then used by the planners.

Consistent with the idea of two modes of operation, between the navigation module and stair climbing module only one is ever running at a given time. The initiation and termination of the software for each module is controlled by a central computer, which acts as a task manager. The perception module will be running constantly, as the other modules both need its output. It also informs the task manager when to switch between modes of operation. The details of how the task manager interfaces with each module can be found in section 5.

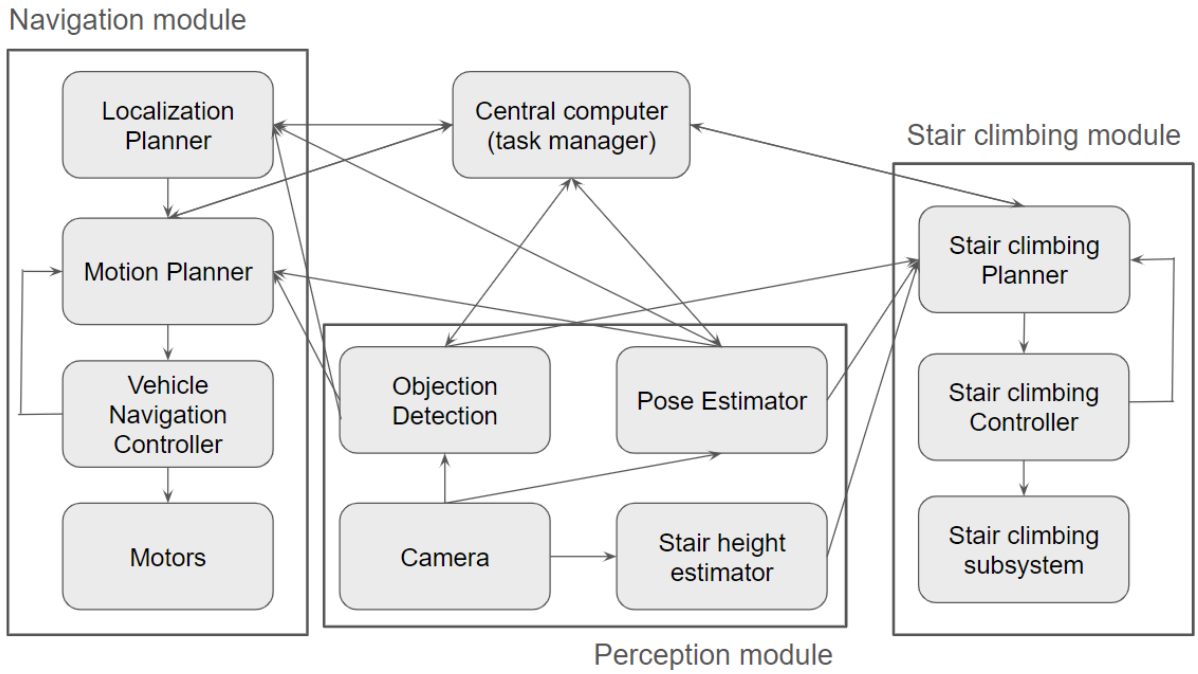


Figure 6.7.1.1 Software Architecture

6.8 Deployment of Software into Hardware

The software will be written in Python using robotics application centered libraries such as PyRobot or Klamp't. The code will include a main function running on the central computer. The submodules will be written as code functions, and they will be called by the main function. The central computer will be a Jetson Nano, which gives it enough compute power to also run the localization planner, motion planner, and stair climbing planner. For the vehicle navigation controller and stair climbing controller we will be using an arduino nano. We will utilize existing Arduino libraries, such as Stepper.h, for interfacing with motor drivers for both stepper motor and DC motors in our subsystems. The arduino nano will use its Serial ports to send and receive data to and from the Jetson Nano.

6.9 System States and Operational Modes

The robot's operation will consist of moving from a starting position to the stairs, climbing the stairs, then moving to an end location. Thus, its operation can be separated into two modes, navigation and stair climbing. In the navigation mode, the robot will be moving across a floor. It will need to sense the environment and plan a path to the stairs. For our project, we will assume that there are no obstacles. In the stair climbing mode, the robot will be ascending/descending stairs. It will need to know where it is on the step and if it clears the step when climbing. For a typical run, the robot will start in its navigation mode as it moves from its starting position to the staircase. Then it will switch to its stair climbing mode and climb the stairs. Finally the robot switches back to its navigation mode and moves to the goal position.

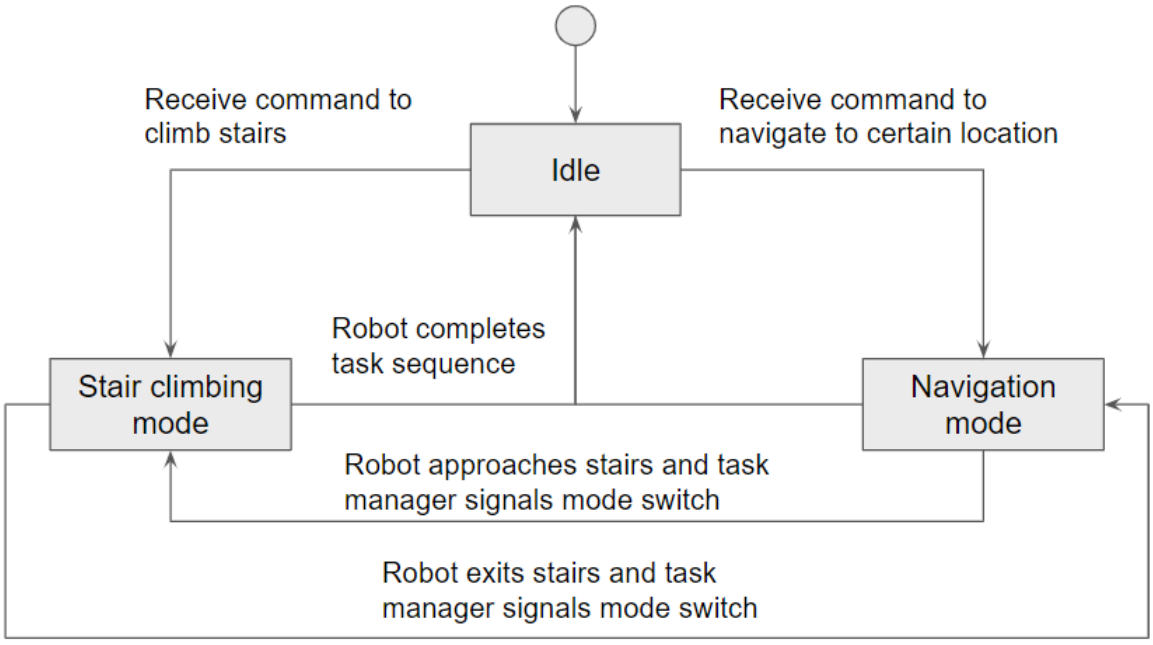


Figure 6.9.1 Overall System Statechart

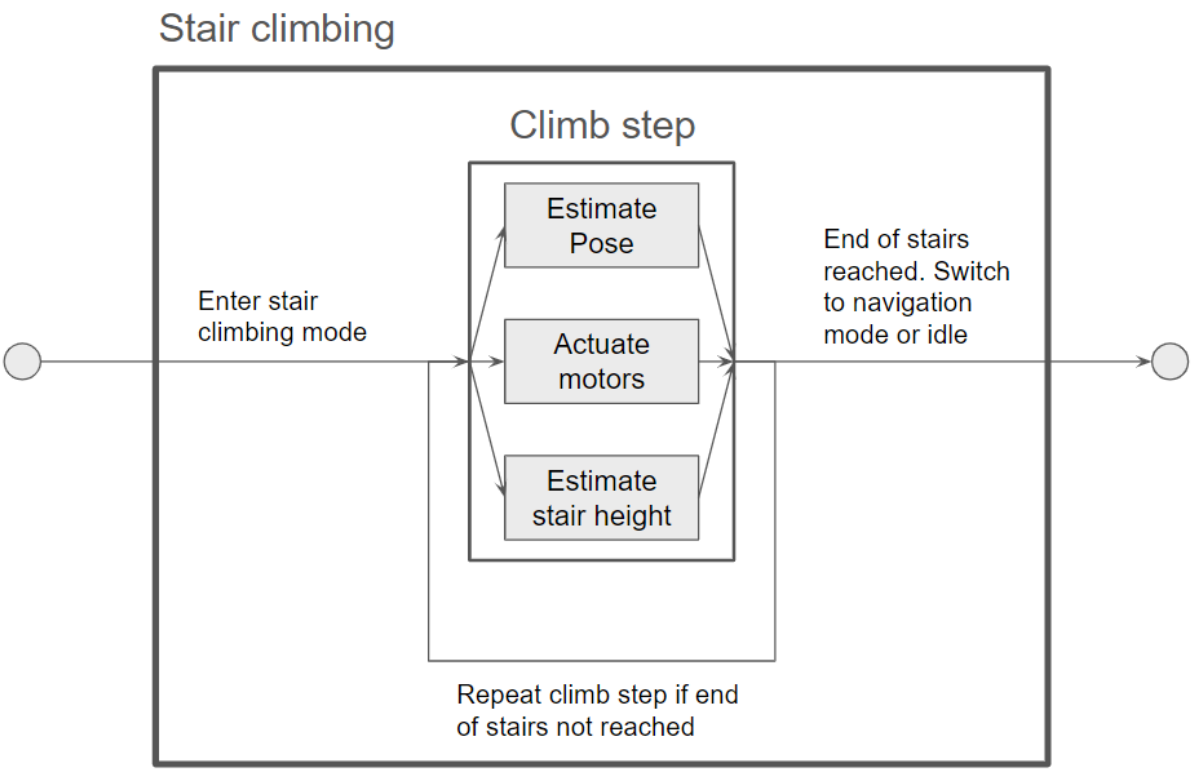


Figure 6.9.2 Stair Climbing Mode Statechart

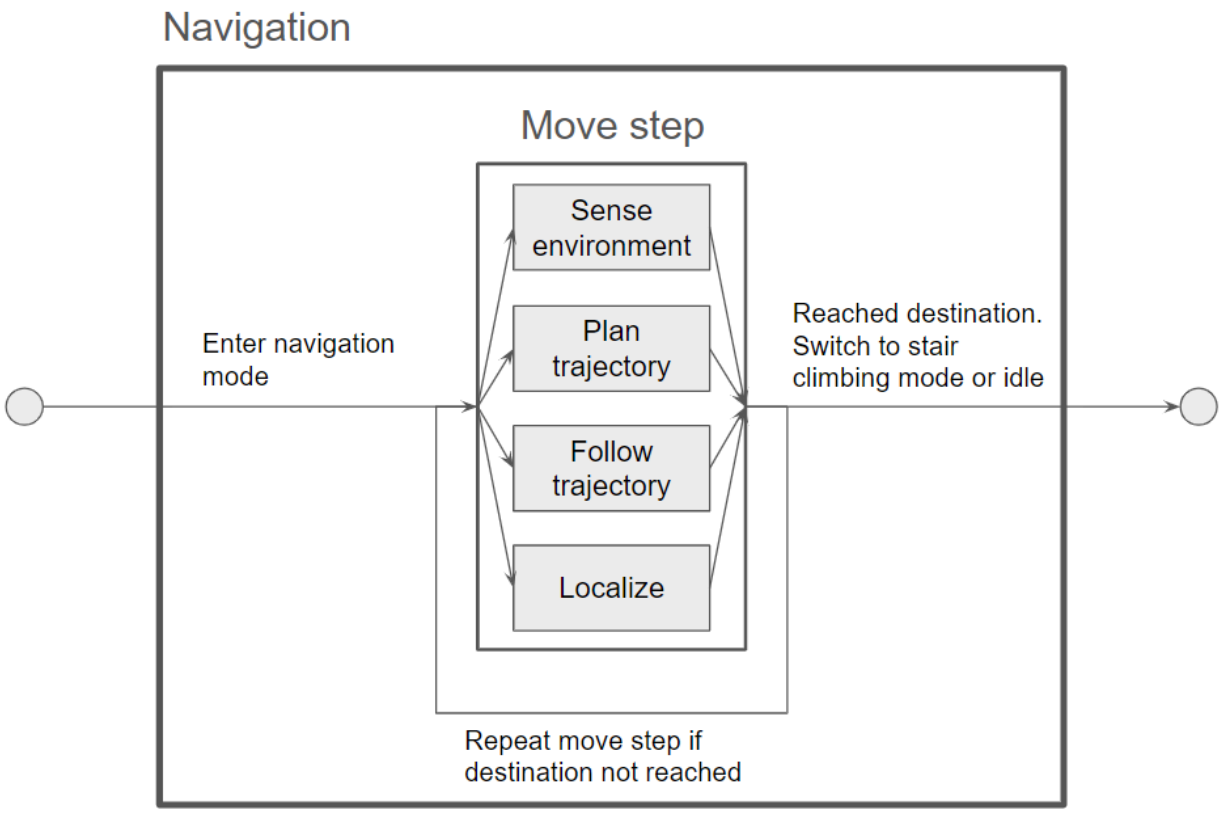


Figure 6.9.3 Navigation Mode Statechart

6.10 Fault Recovery and Degraded modes

Function/Item	**Failure modes**	**Fault Recovery**
Stair Climbing	**Fall off stairs**	**Notify operator and wait for help**
	Stuck on stairs	**Notify operator and wait for help**
Stair Identification	**Identifying stair when it does not exist**	**Robot will drive forwards to attempt to go up against the stair step. If it does not sense contact with the step, stop, report the fault, and attempt to sense the environment again**
	Does not identify stair when a stair is in front of robot	**Motor will detect itself getting stuck when the robot bumps into the stair. Move backward and sense the environment again.**

Navigation in non-stair environment	**Unable to find a path**	**Notify operator and wait for help**
	Localization failure	**Notify operator and wait for help**
	Wheel falls off	**Notify operator and wait for help**
	Motor burned out	**Notify operator and wait for help**

6.11 Operation Plan deployment, preparation and start-up

The operation plan after the robot is assembled is as follows:

1. Test the components with simple unit tests as described in the Verification & Validation document to ensure correct functionality.
2. Select a location of deployment, preferably where the staircase is straight and the surrounding space is flat and without obstacles/occlusions.
3. Create a map of the environment to give to the robot for localization.
4. Before operation:
    a. Check for signs of damage to the robot, observe if there are damaged beams, broken wires, loose cables, etc.
    b. Check if the environment is safe to operate.
5. Deploy the robot. Give the robot a goal position and start running the robot.
6. During operation:
    a. Always ensure that there is an option to remotely terminate the robot.
7. After operation, disable the robot.

6.12 Usage Guidelines and Safety Restrictions

1. When operating the robot, always ensure that there is an option to remotely terminate the robot.
2. Do not place loads that exceed the intended capacity on the robot.
3. Handle the robot with care and avoid touching sharp edges.

7. Verification

7.1 Requirements Validation

These tests are intended to validate that our robot indeed satisfies our requirements.

1. Let the robot climb up 4 steps of stairs. Measure if the robot can do so under 15 seconds per stair. Perform the same test with the robot descending stairs.
    a. Chassis, Stair climbing mechanism, Power & electronics, Perception, Core Software are tested
    b. Purpose & requirements of relevance/significance: [R7]
    c. Data collection and evaluation: use a timer to time the whole process of the robot climbing up 4 steps of stairs and compare the time with 60 seconds
    d. Logistics of testing:

Time	Location	Required equipment and supplies	Test personnel and responsibilities
Mar 19	Ansys C level staircase	Timer / cell phone	Alvin-timing; Kevin- controlling the robot; Ella

e. Potential Post-test actions: Compare the time with 60 seconds. Adjust the speed of the robot if necessary.

2. Test if robot is able to climb 10 steps of stairs consecutively for three times back to back without failure
    a. Chassis, Stair climbing mechanism, Power & electronics, and Core Software are tested
    b. Purpose & requirements of relevance/significance: [R8]
    c. Data collection and evaluation: count the number of stairs it fails to climb up during the process.
    d. Logistics of testing:

Time	Location	Required equipment and supplies	Test personnel and responsibilities
Mar 19	Ansys C level staircase	Eyes n brain/notepad	Alvin; Kevin- controlling the robot; Ella- counting

    e. Post-test actions: Optimize and check the mechanism and code if necessary.

3. Let the robot climb stairs and test if the robot can maintain the payload's balance when raising its chassis up and forwards (i.e the payload does not fall out of the robot)
    a. Chassis, Stair climbing mechanism, Power & electronics, and Core Software are tested
    b. Purpose & requirements of relevance/significance:[R1.3]
    c. Data collection and evaluation: N/A pure observation of the robot's behavior
    d. Logistics of testing:

Time	Location	Required equipment and supplies	Test personnel and responsibilities

Mar 25	Ansys C level staircase	Eyes, notepad	Alvin; Kevin- controlling the robot; Ella

e. Post-test actions: check whether any failure occurred during the process. Adjust the center of mass and the mechanism if necessary.

4. Let the robot drive in a straight line. Measure if the robot can achieve maximum speed of at least 0.6m/s.
    a. Chassis, Perception, Power & electronics, and Core Software are tested
    b. Purpose & requirements of relevance/significance: [R2, R6, R10]
    c. Data collection and evaluation: use a timer to time the process of the robot moving from its starting position to the goal position. Measure the distance of the path between the two positions with a tape measure. Calculate the speed by dividing distance by time and compare the speed with 0.6m/s.
    d. Logistics of testing:

Time	Location	Required equipment and supplies	Test personnel and responsibilities
Mar 19	Techspark	Tape measure, timer/cell phone, notepad	Alvin- timing; Kevin- control the robot; Ella-measure distance, evaluate the result

    e. Post-test actions: check whether the speed of the robot is equal or larger than 0.6m/s. If not, adjust the power for the driving motors.
5. Let the robot operate for 15 minutes with a fully charged battery, and measure if the remaining battery level is greater than 50%.
    a. Power & electronics is tested
    b. Purpose & requirements of relevance/significance: [R5]

c. Data collection and evaluation: set a timer for 15 mins, let the fully charged robot operate and take down the battery level after operation.

d. Logistics of testing:

Time	Location	Required equipment and supplies	Test personnel and responsibilities
Mar 19	Techspark	Timer / cell phone	Alvin-timing; Kevin- starting the robot; Ella

e. Post-test actions: Compare the battery level after operation with 50%. Choose an alternative type of battery if necessary.

6. Let the robot perform the transition between navigation and stair climbing. Position the robot 3-10 meters away from the staircase and let it navigate to the stairs then climb them. Test if the robot can navigate to the stairs, orient itself correctly, and start climbing.

   a. Core Software is tested

   b. Purpose & requirements of relevance/significance: [R2, R3, R10]

   c. Data collection and evaluation: measure the starting point of the robot to be 10 meters away from the staircase. No data collected. Observe and record whether the robot can find the stair and start climbing.

   d. Logistics of testing:

Time	Location	Required equipment and supplies	Test personnel and responsibilities
Mar 28	Ansys C level staircase	Tape measure, notepad	Alvin-measuring; Kevin-controlling the robot; Ella

   e. Post-test actions: Recall whether the robot can identify the stair and start climbing successfully. Optimize stair detection algorithm, odometry and mechanism is necessary.

7. Let the robot perform the transition between stair climbing and navigation. Position the robot at the last step of the stairs and give it a goal position within 10 meters of the staircase. Test if the robot can climb the last step, orient itself, and navigate to the goal position.
    a. Chassis, Stair climbing mechanism, Power & electronics, and Core Software are tested
    b. Purpose & requirements of relevance/significance: [R2, R3, R10]
    c. Data collection and evaluation: measure the goal point to be 10 meters away from the robot. No data collected. Observe and record whether the robot can reach the goal position successfully
    d. Logistics of testing:

Time	Location	Required equipment and supplies	Test personnel and responsibilities
Mar 28	Ansys C level staircase	Tape measure, notepad	Alvin-measuring; Kevin-controlling the robot; Ella

    e. Post-test actions: Recall whether the robot can identify the end of the stair and start navigation successfully. Optimize stair detection algorithm, odometry and mechanism is necessary.
8. Let the robot climb stairs with a payload greater than 250g. Test if the robot is able to climb stairs while keeping balance and with a speed that satisfies R7. [R1.2, R1.3]
    a. Chassis, Stair climbing mechanism, Power & electronics, Perception, Core Software are tested
    b. Purpose & requirements of relevance/significance: [R1.2, R1.3, R7]

c. Data collection and evaluation: use a timer to time the whole process of the robot climbing up 4 steps of stairs for 10 times with varying weights and compare the time with 60 seconds

d. Logistics of testing:

Time	Location	Required equipment and supplies	Test personnel and responsibilities
Mar 21	Ansys C level staircase	Timer / cell phone	Alvin-timing; Kevin- starting the robot; Ella

e. Post-test actions: Compare the time with 60 seconds. Adjust the speed of the robot if necessary.

9. Set the robot in front of different objects(stairs, wall, people, furniture) and test if the robot can correctly identify stairs. Repeat this test 10 times and measure if the accuracy is greater than 95%.

a. Core Software is tested

b. Purpose & requirements of relevance/significance: [R1.1, R9]

c. Data collection and evaluation: record whether the robot failed to identify stairs or identified other objects as stairs. Repeated 50 times with different objects

d. Logistics of testing:

Time	Location	Required equipment and supplies	Test personnel and responsibilities
Mar 28	Ansys C level staircase	notepad	Alvin-running code; Kevin-; Ella-taking notes

e. Post-test actions: divide the number of mistakes by 50 and compare the result with 95%. Improve the software if necessary.

10. Check the bill of materials and verify that the cost of buying/fabricating all the parts for the robot is less than $2500. [R4]

7.2 Functional Verification

These tests are intended to verify that the components and subsystems of our robot function as expected.

1. Give the stepper motor of the stair climbing system a command to turn a specific amount. Measure if the stepper motor for stair climbing mechanism can accurately turn with +- 6 degrees accuracy.
    a. Purpose & significance: Verifies that step motor is functional and accurate
    b. Stepper motors with be tested
    c. Data collection and evaluation: mount a pointer to the output shaft of stepper motor, give it a random rotation command between 10 and 1000 degrees, and use the pointer to measure the difference between commanded and actual rotation.
    d. Logistics of testing:

Time	Location	Required equipment and supplies	Test personnel and responsibilities
Mar 21	Techspark	Stepper motor, stepper driver, microcontroller	Kevin

    e. Post-test actions: adjust code or substitute the motor if necessary

2. Give the chassis motor a command to turn a specific amount. Measure if the motor can accurately turn the commanded amount, with +- 30 degrees accuracy.
    a. Purpose & significance: Verifies that chassis motor is functional and accurate
    b. Chassis motors will be tested
    c. Data collection and evaluation:mount a pointer to the output shaft of the motor, give it a random rotation command between 10 and 1000 degrees, and use the pointer to measure the difference between commanded and actual rotation.
    d. Logistics of testing:

Time	Location	Required equipment and supplies	Test personnel and responsibilities
Feb ..	Techspark	Chassis motor, motor driver, microcontroller	Kevin

    e. Post-test actions: adjust code or substitute the motor if necessary
3. Place the camera in front of stairs or non-stair objects and test whether the camera can identify stairs at a distance between 0.2 to 1.0 meter.
    a. Purpose & significance: Verify camera can detect stairs
    b. Camera and core software will be tested
    c. Data collection and evaluation: measure the distance between the camera and the staircase. Observe/ take notes of whether the stairs are identified
    d. Logistics of testing:

Time	Location	Required equipment and supplies	Test personnel and responsibilities
Mar 19	Ansys C-level staircase	Tape measure	Alvin, Ella

e. Post-test actions: improve the code if necessary

4. Load up to 250g of weights on the robot and test whether the motor has enough torque to raise the chassis.
    a. Purpose & significance: Stress test to verify the robot can climb stairs under different conditions.
    b. Stair climbing mechanism will be tested
    c. Data collection and evaluation: pure observation
    d. Logistics of testing:

Time	Location	Required equipment and supplies	Test personnel and responsibilities
Mar 21	Ansys C-level staircase	eyes	Kevin, Alvin, Ella

    e. Post-test actions: improve the mechanism if necessary
5. Load up to 250g of weights on the robot and test whether the chassis has enough force to move the chassis horizontally.
    a. Purpose & significance: Stress test to verify the robot can navigate under different conditions.
    b. Chassis will be tested
    c. Data collection and evaluation: pure observation
    d. Logistics of testing:

Time	Location	Required equipment and supplies	Test personnel and responsibilities
Feb ..	Techspark	eyes	Kevin, Alvin, Ella

    e. Post-test actions: improve the mechanism and adjust motor power if necessary

6. After integrating the chassis with the stair climbing frame, test if the robot can still navigate as before, and test if the robot can still climb stairs as before.
    a. Purpose & significance: Verify that integrating subsystems does not impact overall functionality.
    b. Chassis, Stair climbing mechanism, Power & electronics will be tested
    c. Data collection and evaluation: pure observation
    d. Logistics of testing:

Time	Location	Required equipment and supplies	Test personnel and responsibilities
Feb ..	Ansys C-level staircase	eyes	Kevin, Alvin, Ella

    e. Post-test actions: improve the mechanism, code or electronics if necessary

7.3 Performance Verification

These tests are intended to verify that the performance of the subsystems are satisfactory for our robot to function properly.

1. Let the stair climbing mechanism travel at the fastest speed in vertical direction. Measure if it is greater than 7.5cm/s using a timer.
    a. Stair climbing mechanism, Power & electronics, Perception, Core Software are tested
    b. Purpose & significance: To verify the performance of the stair climbing mechanism

c. Data collection and evaluation: time the process of the stair climbing mechanism moving up at the highest speed (another trial down). Measure the distance of movement.

d. Logistics of testing:

Time	Location	Required equipment and supplies	Test personnel and responsibilities
Mar 21	Techspark C level staircase	timer / cell phone, tape measure, notepad	Alvin-measuring; Kevin-running code; Ella- taking notes

e. Post-test actions: divide the distance (in cm) by time (in second) and compare the result with 7.5cm/s. Optimize the mechanism or substitute the motors if necessary.

2. Let the stair climbing mechanism travel at the fastest speed in horizontal direction. Measure if it is greater than 7.5cm/s using a timer.

a. Stair climbing mechanism, Power & electronics, Perception, Core Software are tested

b. Purpose & significance: To verify the performance of the stair climbing mechanism

c. Data collection and evaluation: time the process of the stair climbing mechanism moving forward (another trial backward). Measure the distance of movement.

d. Logistics of testing:

Time	Location	Required equipment and supplies	Test personnel and responsibilities
Mar 21	Techspark C level staircase	timer / cell phone, tape measure, notepad	Alvin-measuring; Kevin-running code; Ella- taking notes

e. Post-test actions: divide the distance (in cm) by time (in second) and compare the result with 7.5cm/s. Optimize the mechanism or substitute the motors if necessary.

3. Run the localization loop on the controller, and measure if it can run faster or equal to 50Hz per loop using the timer and loop counter within the controller.

7.4 Failure Modes

RV = Requirement Validation, FV = Functional verification, PV = Performance verification

Function/Item	Failure mode	Failure effects	Failure causes	Test to detect failure	SEV	OCC	DET	RPN
Stair Climbing	Fall off stairs	Robot is broken	Losing balance, did not climb high enough.	RV3	10	5	1	50
	Stuck on stairs	Unable to complete tasks	Battery ran out, Mechanism defect	RV2, FV4	5	4	1	20
Stair Identification	Identifying stair when it does not exist	Attempt to climb when on flat ground, cause robot to tip forward	Computer vision software inaccuracy, sensor defect	FV3	7	2	3	42
	Does not identify stair when a stair is in front of robot	May need human interaction to unstuck robot	Computer vision software inaccuracy, sensor defect	FV3	7	1	3	21
Navigation	Unable to find a path	Unable to complete tasks	Goal position beyond Workspace, Path Planning Algorithm failure	RV4	4	5	2	40
	Localization failure	Robot don't know where it is at, inaccurate position leads to drift and robot might get stuck or need reset	Inaccurate sensor readings, software failure	RV4, RV6, RV7	4	6	3	72
Drivetrain	Wheel falls off	Robot is unable to move	Loose shaft due to wear	RV4, FV2	5	3	1	15
	Motor burned out	Robot bis broken	Get stuck while motor continue to move	RV4, FV2, FV4, FV5	8	1	4	32

We identified some failure modes for the different subsystems of the robot. We assigned severity scores based on if the failure will damage the robot and/or affect the robot's operation. We

assigned occurrence scores based on the likelihood of the failure's occurrence. And we assigned detectability scores based on whether the cause of the failure can be easily detected and identified when the robot malfunctions.The priority of each failure is calculated by multiplying the three scores together, with higher scores having a higher priority.

Based on the FMEA analysis, we identified two failures that have high priorities. The first is the robot falling off the stairs. This failure is the most severe out of all the failures we have identified, and it is largely because the robot will likely be heavily damaged if it occurs. The frequency at which this failure occurs can be non-negligible if the robot system is not designed carefully and robustly tested. Thus, we will be putting a lot of effort in ensuring that the robot system can climb stairs without falling. The second failure we want to highlight is failure with localization. This failure occurs when the robot localizes incorrectly with an inaccurate position estimate. This will lead to the robot veering off course, which potentially will require the operator to reset the robot.

*See fault tree diagrams of selected failure modes in appendix

8. Development Plan

8.1 WBS

1.0 Stair climbing robot that can deliver food

1.1 Stair climbing mechanism (14h)

1.1.1 Design (5h)

1.1.1.1 X-axis mechanism

1.1.1.1.1 Structure

1.1.1.1.1.1 Side plate (Completed)

1.1.1.1.1.2 Rail (Completed)

1.1.1.1.1.3 Sliding block, pulley and belt (Completed)

1.1.1.1.2 Controls

1.1.1.1.2.1 Stepper Control (1.5h) - Kevin

1.1.1.2 Z-axis mechanism

1.1.1.2.1 Structure

1.1.1.2.1.1 Supporting structure (Completed)

1.1.1.2.1.2 Pulley and belt (Completed)

1.1.1.2.2 Controls

1.1.1.2.2.1 Servo Control (Completed)

1.1.1.2.2.2 Current limiting and stall protection (3.5h) - Kevin

1.1.2 Build (One side is mostly complete, two sides needed) (5h)

1.1.2.1 z-axis mechanism

1.1.1.2.1.1 Supporting structure (In progress, 1h) - Ella

1.1.1.2.1.2 Pulley and belt (In progress, 1.5h) - Ella

1.1.2.2 x-axis mechanism

1.1.1.2.2.1 Side plate (In progress, 0.5h remaining) - Ella

1.1.1.2.2.2 Rail (In progress, 0.5h remaining) -Ella

1.1.1.2.2.3 Sliding block, pulley and belt (1.5h) -Ella

1.1.3 Test (4h)

1.1.3.1 Subsystem testing (1h)

1.1.3.1.1 x-axis mechanism testing (0.5h) - Ella

1.1.3.1.2 z-axis mechanism testing (0.5h) - Ella

1.1.3.2 Full mechanism testing (3h)

1.1.3.2.1 Test if the mechanism can climb up and descend the stairs without payload (0.5h) - Ella

1.1.3.2.2 Test if the mechanism can climb up and descend one step within 15s (0.5h) - Ella

1.1.3.2.3 Test the failure rate of the mechanism keep climbing up and descend the stairs (1h) - Ella

1.1.3.2.4 Test the if the robot can climb up and descend the stairs with payloads of at least 200g (0.5h) - Ella

1.1.3.2.5 Test the percentage of battery used to keep climbing and descending stairs in 15 mins (0.5h) - Ella

1.2 Chassis and Locomotion (20.5)

1.2.1 Design (11h)

1.2.1.1 Structural

1.2.1.1.1 Side Chassis (Completed)

1.2.1.1.2 Top and bottom chassis plate (In progress, 1h remaining) - Ella

1.2.1.1.3 Food container (1h) - Ella

1.2.1.1.4 Other miscellaneous structures (Completed)

1.2.1.2 Electronics

1.2.1.2.1 Power Distribution (Completed)

1.2.1.2.2 Voltage sensing and battery management (1h) - Kevin

1.2.2.2.3 Jetson - Arduino communication (3h) - Alvin

1.2.1.3 Locomotion Controls

1.2.1.3.1 Drive Motor Controls (Completed)

1.2.1.3.2 Odometry (Completed)

1.2.1.3.3 Stall detection and protection (0.5h) - Kevin

1.2.2 Build (8.5h)

1.2.2.1 Acquire Parts (0.5h)

1.2.2.2 Structural

1.2.2.2.1 Side Chassis (In progress, 0.5h remaining) - Ella

1.2.2.2.2 Top and bottom chassis plate (1h) - Ella

1.2.2.2.3 Food container (0.5h) - Ella

1.2.1.1.4 Other miscellaneous structures (1h) - Ella

1.2.2.2.5 Final assembling and fitting (2h) - Ella

1.2.2.3 Electronics

1.2.2.3.1 Power Distribution (2h) - Kevin

1.2.2.3.2 Voltage sensing and battery management (1h) - Kevin

1.2.2.3.3 Jetson - Arduino communication (3h) - Alvin

1.2.2.4 Locomotion Controls

1.2.2.4.1 Drive Motor Controls (1h) - Kevin

1.2.2.4.2 Odometry (3h) - Kevin

1.2.2.4.3 Stall detection and protection (1.5h) - Kevin

1.2.3 Test (3.5h)

1.2.3.1 Structural

1.2.3.1.1 Sanity test for structural integrity (0.5h) - Ella

1.2.3.2 Electronics

1.2.3.2.1 Voltage measurement error < 0.2v (0.5h) - Kevin

1.2.3.3 Locomotion Controls

1.2.3.3.1 Test odometry error < 2% when traveling linearly, < 4% when turning in place (0.5h) - Kevin

1.2.3.3.2 Test maximum travel speed > 0.6 m/s (0.5h) - Ella

1.2.3.3.3 Test that the robot has a range of at least 1080m (30 minutes) when traveling at the speed of 0.6m/s, at full battery charge (1h) - Ella

1.2.3.3.4 Test able to detect motor stalling / robot hits obstacle (0.5h) - Kevin

<u>1.3 Perception</u> (33h)

1.3.1 Design (30h)

1.3.1.1 Stair detection (10h, Alvin)

1.3.1.2 Localization (20h, Alvin)

1.3.2 Build

1.3.2.1 Stair detection

1.3.2.2 Localization

1.3.3 Test (3h)

1.3.3.1 Test if camera can detect stairs with >95% accuracy (1h, Alvin)

1.3.3.2 Place the camera in front of stairs or non-stair objects and test whether the camera can identify stairs at a distance between 0.2 to 1.0 meter. (1h, Alvin)

1.3.3.3 Test if camera can capture the world correctly and provide clean data for the robot to localize. (1h, Alvin)

1.4 Integration

1.4.1 Integration between chassis and stair climbing mechanism (20h, Alvin, Ella, Kevin)

1.4.2 Integration between chassis and perception (40h, Alvin, Ella, Kevin)

1.4.3 Integration between staring climbing mechanism and perception (40h, Alvin, Ella, Kevin)

1.4.4 Whole robot testing (5h, Alvin, Ella, Kevin)

1.5 Management (1h/person/week + 0.5h/everyone/week)

1.5.1 Manage work

1.5.1.1 Work delegation

1.5.1.2 Work journal

1.5.2 Manage finances

1.5.2.1 Budget tracker

1.5.3 Manage schedule

1.5.3.1 Gantt chart

1.5.3.2 Work timeline

1.5.4 Manage Risk

1.5.4.1 Risk assessment

8.2 Schedule

Feb 11: Build one side of stair climbing mechanism, program Jetson to Arduino communication

Feb 18: Build chassis, complete electronics wiring, program locomotion controls and stair detection program

Feb 25: Integrate chassis and stair climbing mechanism, program odometry and localization. Finish unit testing subsystems (all subsystem complete)

March 17: Complete integration and full robot testing

***See appendix 11.7 for Gantt chart**

9. Requirements Management Plan

9.1 Objective Tree

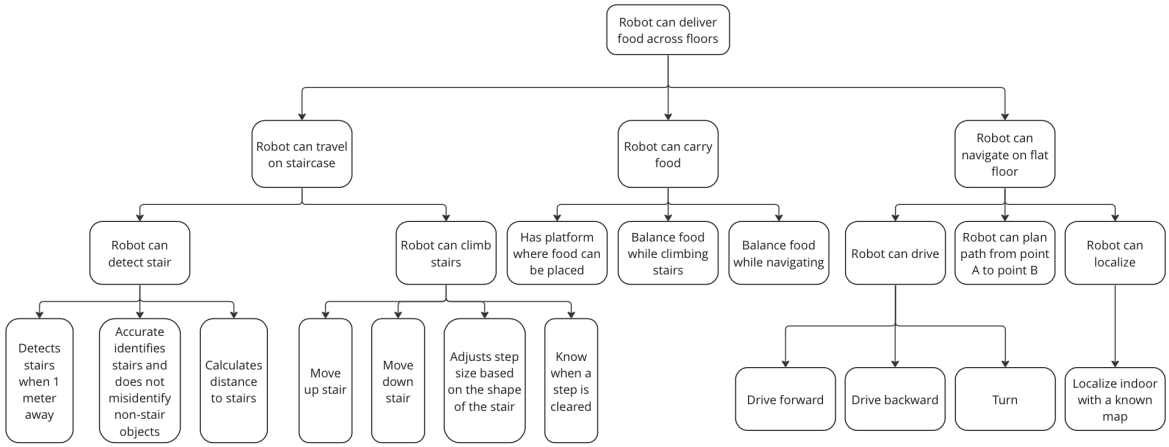


Figure 9.1.1 Objective Tree

9.2 Requirement Matrix

ID	Requirement description	Testing Method	Chassis	Stair climbing mechanism	Power & electronics	Perception	Core software
R1.1	Ability to correctly identify stairs.	Set the robot in front of different objects(stairs, wall, people, furniture) and test if the robot can correctly identify stairs. Repeat this test 100 times and measure if the accuracy is greater than 95%.			X	X	X
R1.2	Able to carry a payload of at least 200g	Let the robot climb stairs with payloads of varying weight(150g, 200g, 250g). Test if the robot is able to climb stairs while keeping balance and with a speed that satisfies R7.	X	X	X		
R1.3	Keep payload balanced (+- 10 degrees with respect to the ground) while climbing stairs.	Let the robot climb 100 steps of stairs and measure if the robot can maintain the payload's balance when raising its chassis up and forwards(i.e the payload does not fall out of the robot. [R1.3]	X	X	X		X
R2	Navigate from a starting location to the stairs, and from the stair exit to the goal location, with distance under 10 meters.	Let the robot move autonomously from one location to another goal location on the same floor that is within 10 meters. Repeat this test 10 times with different location pairs and measure if the robot can move with an average speed of at least 0.6m/s.	X		X	X	X
R2.1	Perform path planning given a map of the building.						X
R2.2	Perform localization given a map of the building.		X		X	X	X
R3	Ability to switch modes from navigation to stair climbing and vice versa when entering/exiting the stairs.	Let the robot perform the transition between stair climbing and navigation. Position the robot at the last step of the stairs and give it a goal position within 10 meters of the staircase. Test if the robot can climb the last step, orient itself, and navigate to the goal position.					X
R5	The robot can continuously operate fo	Let the robot operate for 15 minutes with a fully charged battery, and measure if the remaining battery level is greater than 50%.			X		
R6	The moving speed of the robot can ma	Let the robot perform the transition between navigation and stair climbing. Position the robot 10 meters away from the staircase and let it navigate to the stairs then climb them. Test if the robot can navigate to the stairs, orient itself correctly, and start climbing.	X		X		X
R7	The robot can climb the stairs at a speed faster than 15 seconds/ stair	Let the robot climb up 4 steps of stairs. Repeat this test 10 times and measure if the robot can do so under 60 seconds on average. Perform the same test with the robot descending stairs. [R7]	X	X	X	X	X
R8	Robot failure rate lower than 1 failure per 100 stairs climbed, on average	Let the robot climb 500 steps of stairs, and measure if the average failure rate is less than one 1 fail per 100 stairs traveled. [R8]	X	X	X		X
R9	Identify stairs with 95% accuracy rate	Set the robot in front of different objects(stairs, wall, people, furniture) and test if the robot can correctly identify stairs. Repeat this test 100 times and measure if the accuracy is greater than 95%.				X	
R10	Robot can successfully navigate from one location to another with distance <10 meters apart.	Let the robot perform the transition between navigation and stair climbing. Position the robot 10 meters away from the staircase and let it navigate to the stairs then climb them. Test if the robot can navigate to the stairs, orient itself correctly, and start climbing.	X		X	X	X
R4	Total cost is less than $2500		X	X	X	X	

9.2.1 Requirement Matrix

10. Risk Management Plan

ID	Risk Title	Risk Owner	Date Submitted	Date Updated
1	Mechanical Wear and Tear	Ella	2/22/2024	2/22/2024
Description				
Continuously conducting tests might cause mechanical wear to some parts, especially the 3D printed ones.				

Consequences	Risk Type	Risk Level
The robot's movement might be no longer accurate and might cause localization failure.	Mechanical	25%
Risk Reduction Plan	**Expected Outcome**	**Likelihood & consequence**
Consider fatigue during design stage, prototype conservatively for mechanical wear, and check key components before running	The robot functions as expected during tests and get repaired in time when needed	2,3

ID	Risk Title	Risk Owner	Date Submitted	Date Updated
2	Center of gravity wrongly estimated	Ella	2/22/2024	2/22/2024
Description				
The estimation of the chassis including the electronic components and the stair climbing mechanism are not accurate. Thus the robot would not balance when moving on stairs.				
Consequences			**Risk Type**	**Risk Level**
The robot tips over on the stairs			Mechanical	50%
Risk Reduction Plan			**Expected Outcome**	**Likelihood & consequence**
Conduct careful mass estimation for each part. Prepare to adjust the positions of components or add weights to balance			The robot can balance itself while climbing the stairs	3, 3

ID	Risk Title	Risk Owner	Date Submitted	Date Updated
3	Stair Climbing Mechanism Shaky	Ella	2/22/2024	2/22/2024
Description				
Shaky Mechanism that causes rocky motion when stair climbing				
Consequences			**Risk Type**	**Risk Level**
Food may spill, or robot may fall off stair and get damaged			Mechanical	50%
Risk Reduction Plan			**Expected Outcome**	**Likelihood & consequence**
Add connection between both sides of the stair climbing mechanism. Structure check before running.			The stair climbing mechanism should not shake while	3, 3

	climbing	

ID	Risk Title	Risk Owner	Date Submitted	Date Updated
4	Z axis motion went over limit	Kevin	2/22/2024	2/22/2024
Description				
Wrong servo command or wrong feedback causes Z axis slide block to hit chassis or upper endstops				
Consequences			**Risk Type**	**Risk Level**
Damage caused to the slide block or chassis, motor and controller may overheat if not stopped quickly			Electrical	40%
Risk Reduction Plan			**Expected Outcome**	**Likelihood & consequence**
Firmware includes protection for commands above the limit, fuse and current limit to prevent overheat			No commands above limit should be given, and motor feedback should always be accurate	2, 4

ID	Risk Title	Risk Owner	Date Submitted	Date Updated
5	Camera sensor giving incorrect readings	Alvin	2/22/2024	2/22/2024
Description				
The camera sensor gives incorrect readings due to incorrect calibration or sensor malfunction.				
Consequences			**Risk Type**	**Risk Level**
The robot will be unable to localize correctly, or be unable to detect stairs successfully.			Mechanical	15%
Risk Reduction Plan			**Expected Outcome**	**Likelihood & consequence**
Before making the robot climb stairs, perform a basic camera test to check that it is functional.			Errors associated with the camera will be detected before causing the robot any harm	2, 2

ID	Risk Title	Risk Owner	Date Submitted	Date Updated
6	Stepper loss step	Kevin	2/22/2024	2/22/2024

Description		
external impact or commands above capability cause stepper to loss step		
Consequences	**Risk Type**	**Risk Level**
Robot no longer accurately knows the X axis location of stair climbing mechanism	Electrical	40%
Risk Reduction Plan	**Expected Outcome**	**Likelihood & consequence**
Prevent collision and have a safety margin in command	Stepper never loss step	4,2

ID	Risk Title	Risk Owner	Date Submitted	Date Updated
7	Robot does not clear step	Alvin	2/22/2024	2/22/2024
Description				
The robot does not clear the full height of the step when climbing the stairs.				
Consequences			**Risk Type**	**Risk Level**
The robot's frame will hit the step and cause the motor to stall			Software	30%
Risk Reduction Plan			**Expected Outcome**	**Likelihood & consequence**
Integrate stall protection into the motors.			The motors will stop in time to not damage the robot. The robot can then properly recover	3, 3

ID	Risk Title	Risk Owner	Date Submitted	Date Updated
8	Wire get loose and disconnect	Kevin	2/22/2024	2/22/2024
Description				
Loose wire causes robot to stop working or short circuiting				
Consequences			**Risk Type**	**Risk Level**
Robot stop working, may damage electrical components			Electrical	50%
Risk Reduction Plan			**Expected Outcome**	**Likelihood & consequence**
Use a printed pcb board to reduce wiring			No wire get disconnected	3,4

ID	Risk Title	Risk Owner	Date Submitted	Date Updated
9	Robot tips over	Ella	2/22/2024	2/22/2024
Description				
The robot tips over due to mechanical failure, control system failure, sensor failure or possible software error				
Consequences			**Risk Type**	**Risk Level**
The hardware components might get damaged. The operator or bystanders might be injured. The robot will fail to finish the task			Mechanical & Electrical & software	50%
Risk Reduction Plan			**Expected Outcome**	**Likelihood & consequence**
Conduct inspections before each testing. Make sure there is at least one operator around while testing the robot.			The robot does not tip over. No functional impairment occurs	3,4

ID	Risk Title	Risk Owner	Date Submitted	Date Updated
10	Broken communication between Jetson and Arduino	Alvin	2/22/2024	2/22/2024
Description				
The communication between the Jetson and Arduino becomes corrupted.				
Consequences			**Risk Type**	**Risk Level**
The robot is unable to operate anymore			Software/electrical	20%
Risk Reduction Plan			**Expected Outcome**	**Likelihood & consequence**
Ensure communication is working properly before runs			Errors can be identified before causing damage to the robot	1, 5

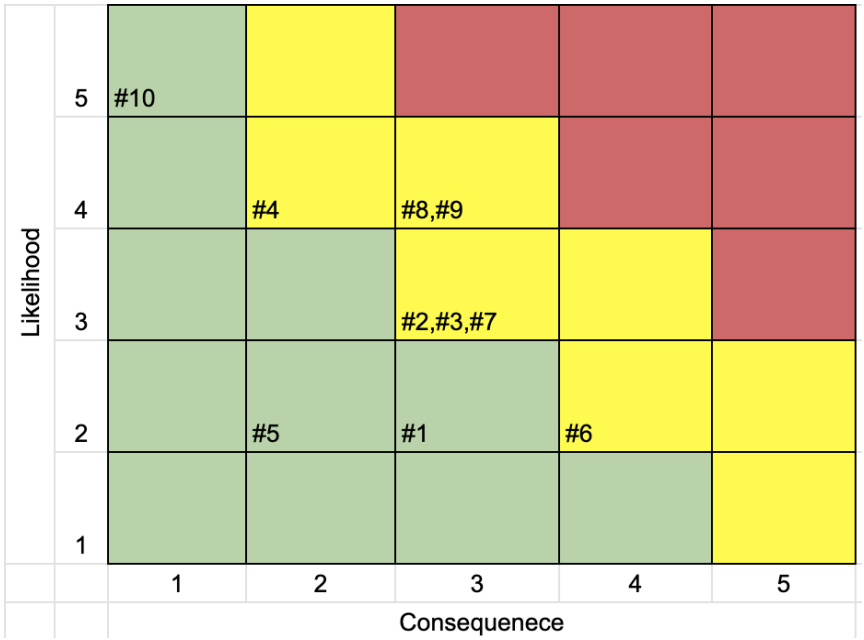


Figure 10.1 Likelihood-consequence matrices

11. Result of Testing

Requirement Validation Results

T7.1.1:

Let the robot climb up 4 steps of stairs. Measure if the robot can do so under 15 seconds per stair. Perform the same test with the robot descending stairs.

Result: The robot is able to achieve 8.8 seconds per step going up, and 14 seconds per step going down, thus satisfying the requirements.

T7.1.2:

Test if robot is able to climb 10 steps of stairs consecutively for three times back to back without failure

Result: After slight adjustment and fixation of the electronic components within the chassis, the robot is able to climb 10 steps of stairs consecutively.

T7.1.3:

Let the robot climb stairs and test if the robot can maintain the payload's balance when raising its chassis up and forwards (i.e the payload does not fall out of the robot)

Result: The robot is able to maintain the balance of payload when climbing.

T7.1.4:

Let the robot drive in a straight line. Measure if the robot can achieve maximum speed of at least 0.6m/s.

Result: The robot is able to achieve maximum speed of 1.05m/s, greater than 0.6m/s

T7.1.5:

Let the robot operate for 15 minutes with a fully charged battery, and measure if the remaining battery level is greater than 50%.

Result: The robot has more than 65% battery after 15 minutes of testing operations.

T7.1.6:

Let the robot perform the transition between navigation and stair climbing. Position the robot 3-10 meters away from the staircase and let it navigate to the stairs then climb them. Test if the robot can navigate to the stairs, orient itself correctly, and start climbing.

Result: Although the robot has errors when moving forward and turning, the robot is able to correct these errors and reach the staircase. The robot is also able to successfully start climbing the stairs once it orients itself.

T7.1.7:

Let the robot perform the transition between stair climbing and navigation. Position the robot at the last step of the stairs and give it a goal position within 10 meters of the staircase. Test if the robot can climb the last step, orient itself, and navigate to the goal position.

Result: The robot is able to navigate to its goal position after climbing up the stairs

T7.1.8:

Let the robot climb stairs with a payload greater than 250g. Test if the robot is able to climb stairs while keeping balance and with a speed that satisfies R7.

Result: The robot is able to climb stairs with a payload up to 800g without noticeable affect to its speed.

T7.1.9:

Set the robot in front of different objects(stairs, wall, people, furniture) and test if the robot can correctly identify stairs. Repeat this test 10 times and measure if the accuracy is greater than 95%.

Result: The robot is generally able to differentiate stairs from non-stairs objects. Namely it can detect stairs with >95% accuracy if placed in front of the stairs. However the accuracy goes down when it is placed at a large angle from the center. The robot is also able to correctly identify non stair objects.

Functional Verification Results

T7.2.1:

Give the stepper motor of the stair climbing system a command to turn a specific amount. Measure if the stepper motor for stair climbing mechanism can accurately turn with +- 6 degrees accuracy.

Result: The stepper motor is able to accurately turn within +- 1.8 degrees, which satisfies the requirements.

T7.2.2:

Give the chassis motor a command to turn a specific amount. Measure if the motor can accurately turn the commanded amount, with +- 30 degrees accuracy.

Result: The chassis motors are able to turn the desired commands under the 30 degree tolerance. However, its accuracy decreases when given a command to turn/move forward slightly. This is mainly due to the fact that the wheels need to overcome the friction of the floor to start moving, but when the desired turning amount is small there is sometimes not enough torque.

T7.2.3:

Place the camera in front of stairs or non-stair objects and test whether the camera can identify stairs at a distance between 0.2 to 1.0 meter.

Result: The camera is able to identify stairs at this range

T7.2.4:

Load up to 250g of weights on the robot and test whether the motor has enough torque to raise the chassis.

Result: The motors have enough torque to lift the chassis

T7.2.5:

Load up to 250g of weights on the robot and test whether the chassis has enough force to move the robot horizontally.

Result: The drivetrain has enough force to move the robot

T7.2.6:

After integrating the chassis with the stair climbing frame, test if the robot can still navigate as before, and test if the robot can still climb stairs as before.

Result: The robot is able to navigate and climb stairs after the chassis and frame is integrated.

Performance Verification Results

T7.3.1:

Let the stair climbing mechanism travel at the fastest speed in vertical direction. Measure if it is greater than 7.5cm/s using a timer.

Result: The stair climbing mechanism is able to travel at more than 10 cm/s in the vertical direction.

T7.3.2:

Let the stair climbing mechanism travel at the fastest speed in horizontal direction. Measure if it is greater than 7.5cm/s using a timer.

Result: The stair climbing mechanism is able to travel more than 18 cm/s horizontally.

T7.3.3:

Run the localization loop on the controller, and measure if it can run faster or equal to 50Hz per loop using the timer and loop counter within the controller.

Result: We changed implementation of navigation controls to correct error by replanning, rather than using a continuous localization loop. Therefore the requirement no longer applies.

12. Appendix

11.1 Use Case and Scenarios

Scenario 1: Navigating on the ground

Summary: The robot is tasked to travel from point A to point B on the same floor

Dependency: Successful path planning from A to B, successful localization

Actors: Motors, Camera

Precondition: The robot is on a flat floor at point A and has a destination on the same floor to go to

Description: The robot generates a feasible path from its position to its destination using a path planner. It then drives along the planned path while performing localization to continuously correct any deviation from the path(by replanning the path) until it reaches the destination.

Postcondition: The robot is on a flat floor and is at the destination, point B

Scenario 2: Climb stairs

Summary: The robot plans to go upstairs. It does so by detecting the stairs and then initiating a stair climbing sequence.

Dependency: Successfully identify stairs; Climbing up the stairs; Keep balance

Actors: Stairs, payload, camera, motors

Preconditions: Robot stops in front of an object and identifies it as stairs.

Alternatives: If the object in front of the robot is not identified as stairs, then deal it as an obstacle.

Postcondition: Robot keeps climbing the stairs until it reaches the top.

Scenario 3: Exit the stairs

Summary: The robot has climbed the last step and needs to exit.

Dependency: Define the end of the stairs; Keep Balance

Actors: Camera, stairs, motors, payload

Preconditions: Robot has identified that it has climbed the last step.

Description: The robot climbs the last step and the camera recognizes that there is no more stairs ahead. The robot switches modes from stair climbing to floor navigation, and drives away from the stairs.

Alternatives: If there are still stairs detected, continue climbing the stairs.

Postcondition: Robot has exited the stairs and is ready to perform floor navigation.

Scenario 4: Descend stairs

Summary: The robot plans to go downstairs. It does so by detecting the stairs and then initiating a stair climbing sequence.

Dependency: Detect downward stairs, descend stairs, keep balance

Actors: Stairs, camera, motors, payload

Preconditions: The robot is at the edge of the stairs and ready to go down.

Description: The robot detects the descending staircase. The robot orients itself properly and starts descending the stairs while keeping balance.

Postcondition: Robot keeps descending the stairs until it reaches the bottom.

Scenario 5: Getting stuck on the staircase

Summary: The robot is unable to move upward or downward on stairs.

Dependency: Working electronics system (e.g. not failing because out of battery)

Actors: Stairs, camera, motors, operator

Preconditions: Robot on stair, unable to climb due to mechanical failure / physical limitation / other unforeseeable circumstances

Description: The robot is stuck on the stairs. It detects this through observing no motion even though it is trying to climb stairs. The robot responds by stopping its operation, and sends a message to the operator signaling that it is stuck.

Alternatives: Signal help is needed with flashing lights / sound.

Postcondition: The robot stops operation and waits for help.

Scenario 6: Tipping over

Summary: The robot tips over on one side.

Dependency: Balance

Actors: operator, camera, food

Preconditions: The robot is tipped over.

Description: The robot will detect that it has tipped over by examining the images from the camera. The robot does not have any self adjusting mechanism. Thus, after tipping over, it will need to wait for help from the operator to readjust it. It will send a signal to the operator and stop operation.

Alternatives: Have a self adjusting mechanism that can automatically readjust the robot.

Postcondition: The robot stops operation and waits for help.

Scenario 7: Getting stuck on the ground

Summary: The robot becomes stuck and unable to move on the ground, it responds by calling for help

Dependency: Detect being stuck

Actors:motors, operator

Preconditions: The robot is stuck and unable to move on flat floor

Description: Robot detects that it is stuck either through the motor encoder detecting that the wheels are not moving, or from the localization not detecting expected movement(wheels spinning but robot not moving). Robot will send a signal to the operator and stop operation.

Alternatives: Detect being stuck by measuring drive train current (too low / too high for given rpm), send help with flashing lights/speaker.

Postcondition: The robot stops operation and waits for help.

Scenario 8: Power Validation

Summary: The robot checks its battery level to ensure it has enough power.

Dependency: Battery level sensing

Actors: Battery, operator

Preconditions: The robot is not in active operation.

Description: The robot checks its battery level. And make sure it has enough battery to perform at least another sequence of navigating from its current position to the staircase, climbing the staircase, and then navigating to the destination. If the robot does not have enough battery, it will notify the operator about its battery level and stop operation.

Postcondition: The robot is ready to perform tasks if power level is enough. If not, it will stop operation and signal that charging is needed.

Scenario 9: Charging

Summary: The robot's battery gets recharged

Dependency: low battery

Actors: Battery, operator

Preconditions: The robot is low on battery and needs to recharge.

Description: The operator removes the battery from the robot and charges the battery. After finishing charging, the operator will attach the battery back to the robot.

Alternatives: The robot charges itself automatically through a charging port.

Postcondition: The robot is charged and can perform tasks again.

Scenario 10: Receive Command from Operator

Summary: The robot receives a command to perform task or stop operation from operator

Dependency: Wifi communication

Actors: Onboard computer, Wifi, Operator's computer

Preconditions: The Onboard computer is running and can communicate with operator's computer with Wifi

Description: Using SSH via Wifi, the Operator can send commands to the robot. Commands include navigation and stair climbing commands, such as telling the robot to navigate to a goal position from its current position. The Operator can also send a stop command that will make the robot stop operation.

Alternatives: If the robot does not receive any command, it will stay idle

Postcondition: The robot is performing the commanded task.

11.1 System Sketches and Scaled drawing

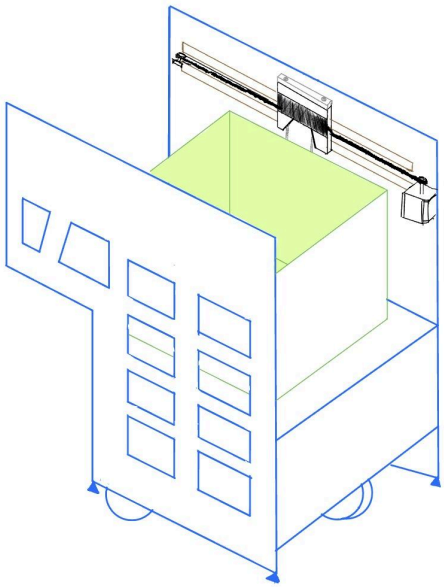


Figure 11.1 Sketch of whole system

11.1.1 Sketch of Stair Climbing Mechanism in Operation

11.1.1.1. Robot comes in contact with stairs

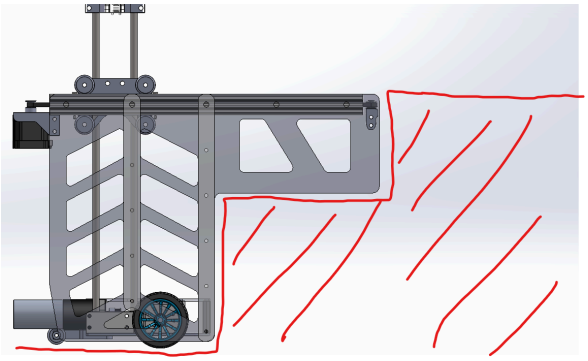


Figure 11.1.1 Sketch of stair climbing mechanism ready to climb up

11.1.1.2. Chassis move up and forward onto next step, while the side frame supports the robot

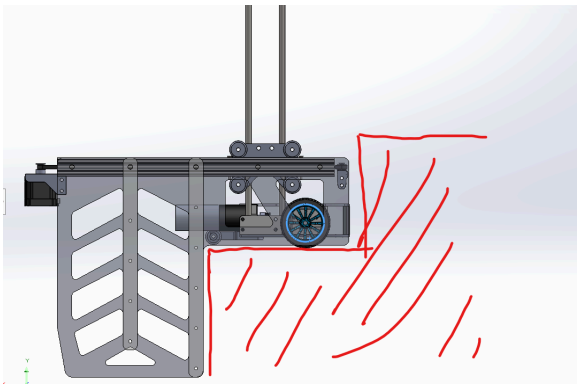


Figure 11.1.2 Sketch of stair climbing mechanism ready to climbing

11.1.1.3. Side frame moves up and forward to the next step, while the wheels on the chassis support the robot

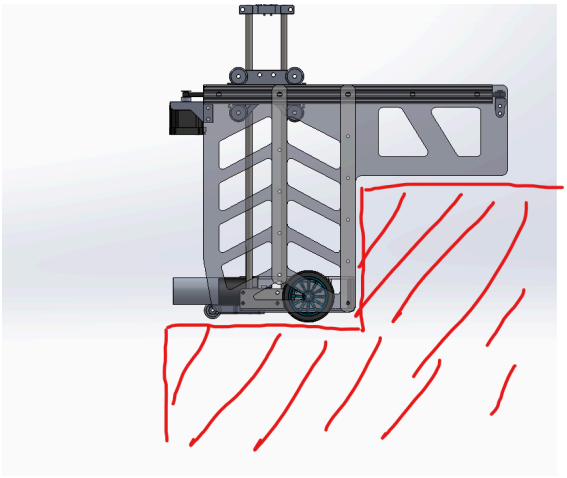


Figure 11.1.3 Sketch of stair climbing mechanism climbed up one staircase

11.1.2 Mechanism Detailed Sketch:

We are using acrylic as the material for the side frames. This is because we will be cutting multiple holes within the frame, and acrylic can be laser cutted, making it easy to fabricate. The reason we are cutting holes is to make the frame weigh less, so that the robot will be able to climb stairs more easily and satisfy our R6 and R7 requirements. The philosophy of keeping everything as light as possible extends to the rest of our design, as we try to reduce the use of heavy parts such as 8020s.

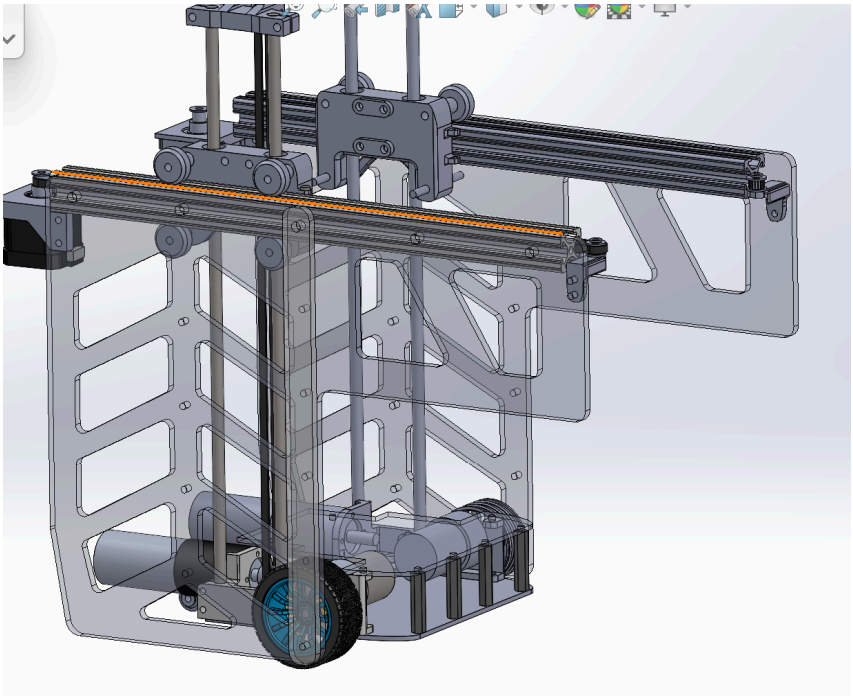


Figure 11.1.2.1 Isometric view of mechanism detailed sketch

Both x-axis and z-axis are driven by timing belts and pulleys. X-axis rides on a 8020 aluminum extrusion using rollers, while the z-axis rides on a pair of linear rods using linear bearings. Originally we planned on using a threaded rod for the z-axis, but after creating a prototype we realized it was too slow. Thus we changed it to a belt design to ensure it can satisfy our R7 requirement.

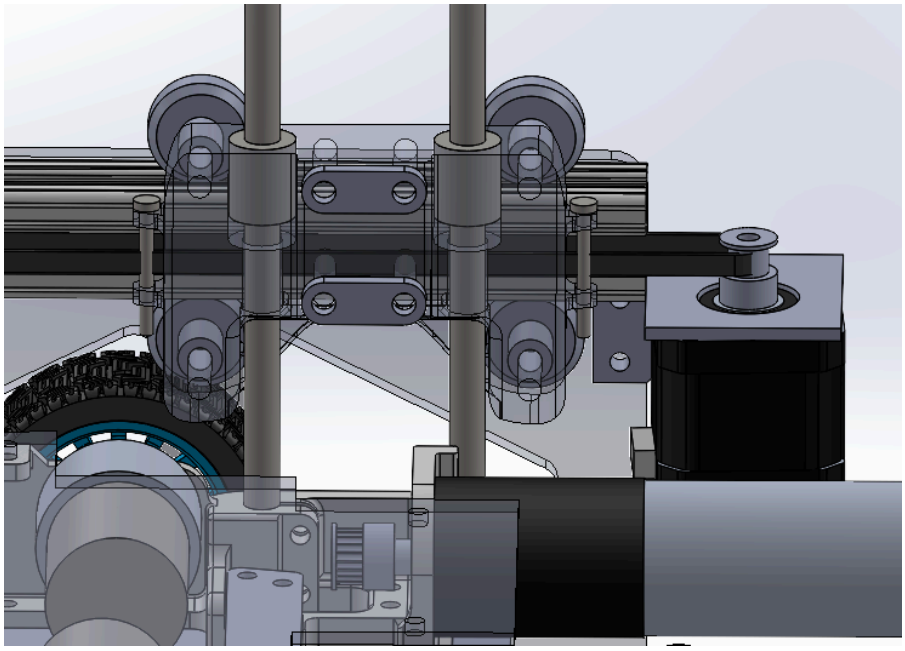


Figure 11.1.12.2 Detailed Sketch of the belt and pulley system

11.2 Detailed Scenarios and Sequence Diagrams

11.2.1 Climbing stairs: The robot is oriented at the bottom of the stairs and is ready to start climbing. The task manager first sends a signal to the perception system to detect the stair height. Then it sends this information to the stair climbing system and initiates the stair climbing sub sequence. Once the robot reaches the top of the stairs, the perception system will detect this and send a signal to the task manager, which in turn signals the stair climbing system to terminate stair climbing. The task manager then initiates the stair exiting sequence.

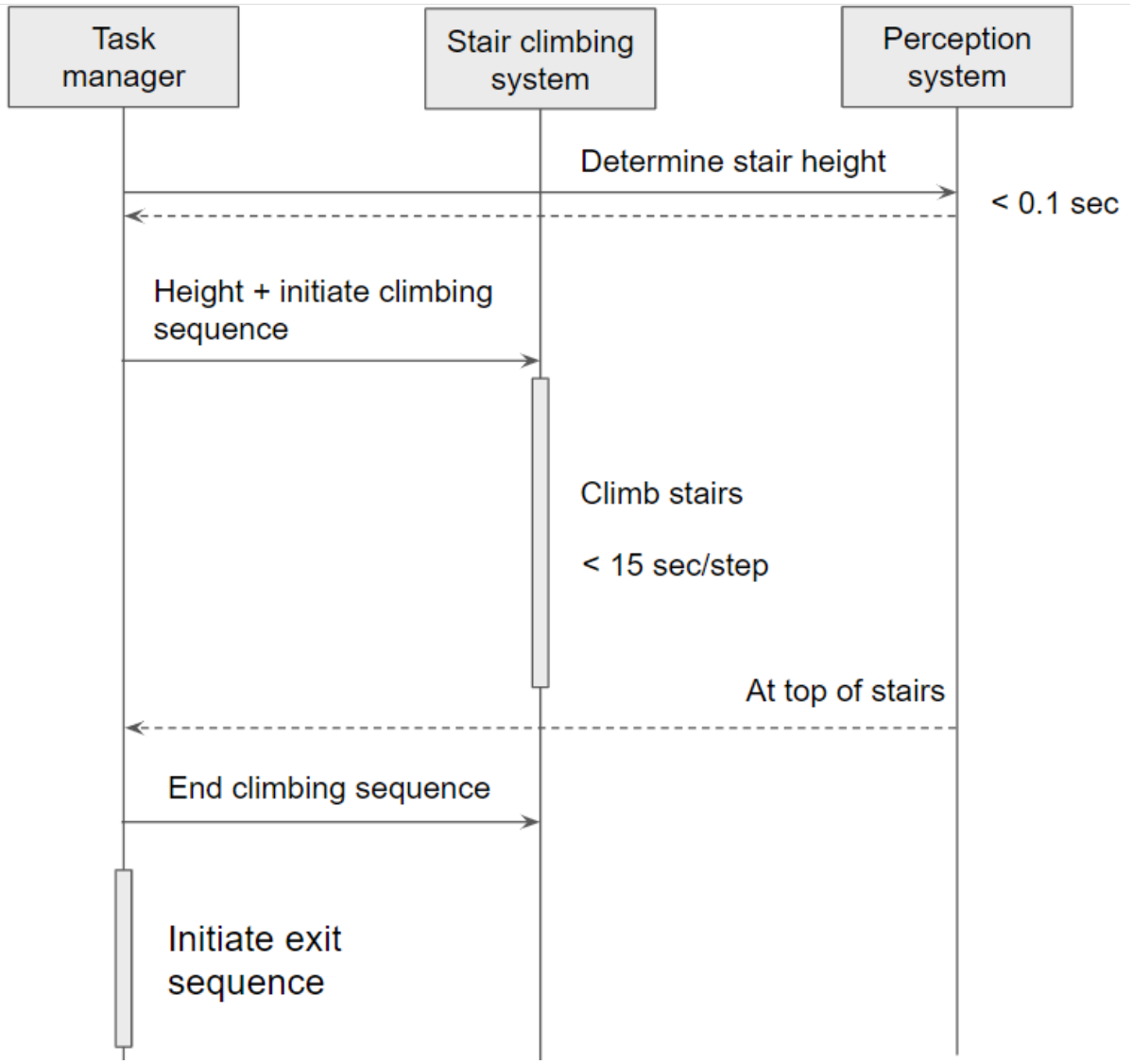


Figure 11.2.1.1 Sequence Diagram for Climbing stairs

11.2.1.1 Stair climbing sub sequence: the stair climbing system includes a low level controller that commands the x-axis and z-axis motors.

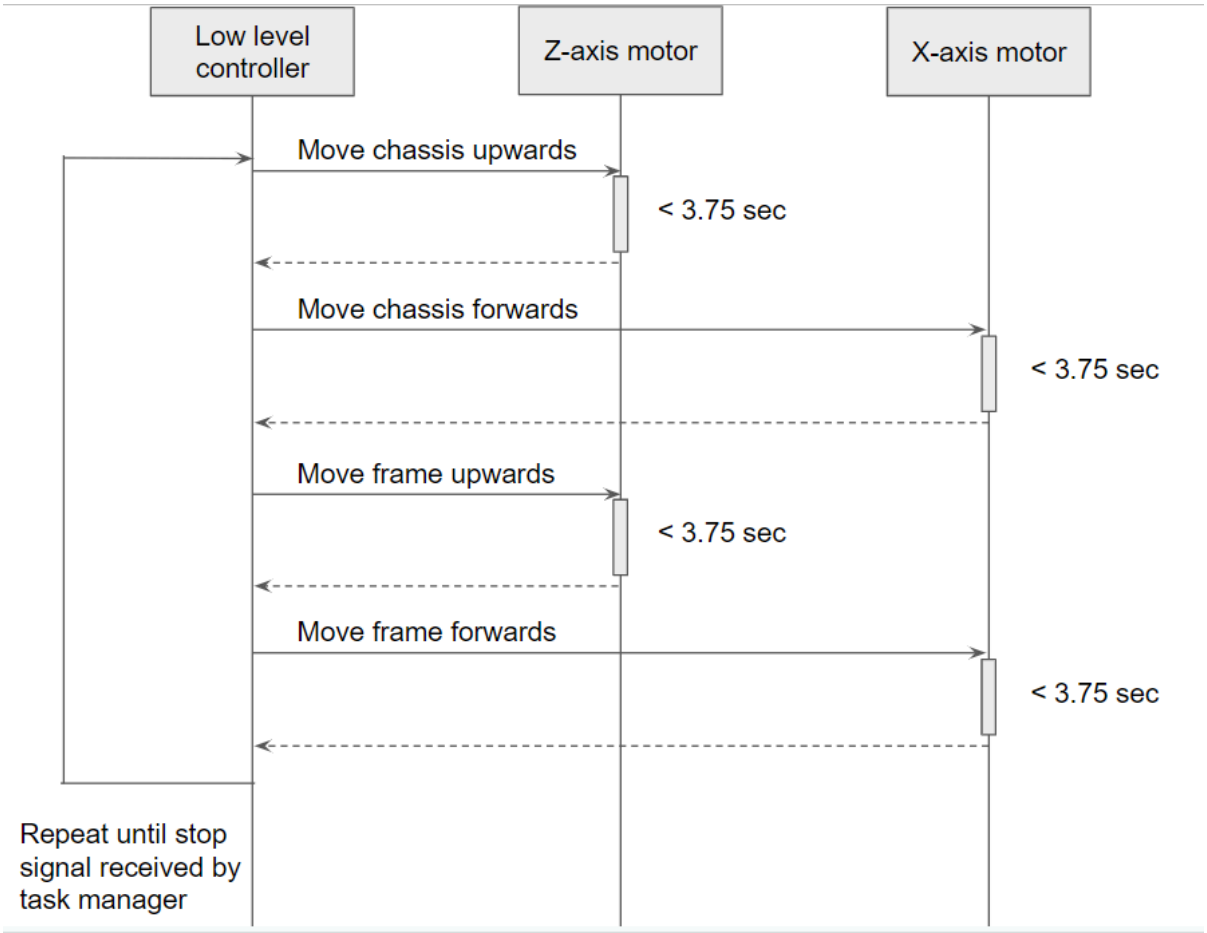


Figure 11.2.1.2 Sequence Diagram for Climbing stairs

11.2.2 Exit stairs: This step serves as the transition between the stair climbing mode to the navigation mode. It is facilitated by the task manager, and illustrates an important architectural design: the two systems are completely separate from each other, on the hardware level and software level. This comes from the fact that there is no situation when both systems need to be engaged simultaneously.

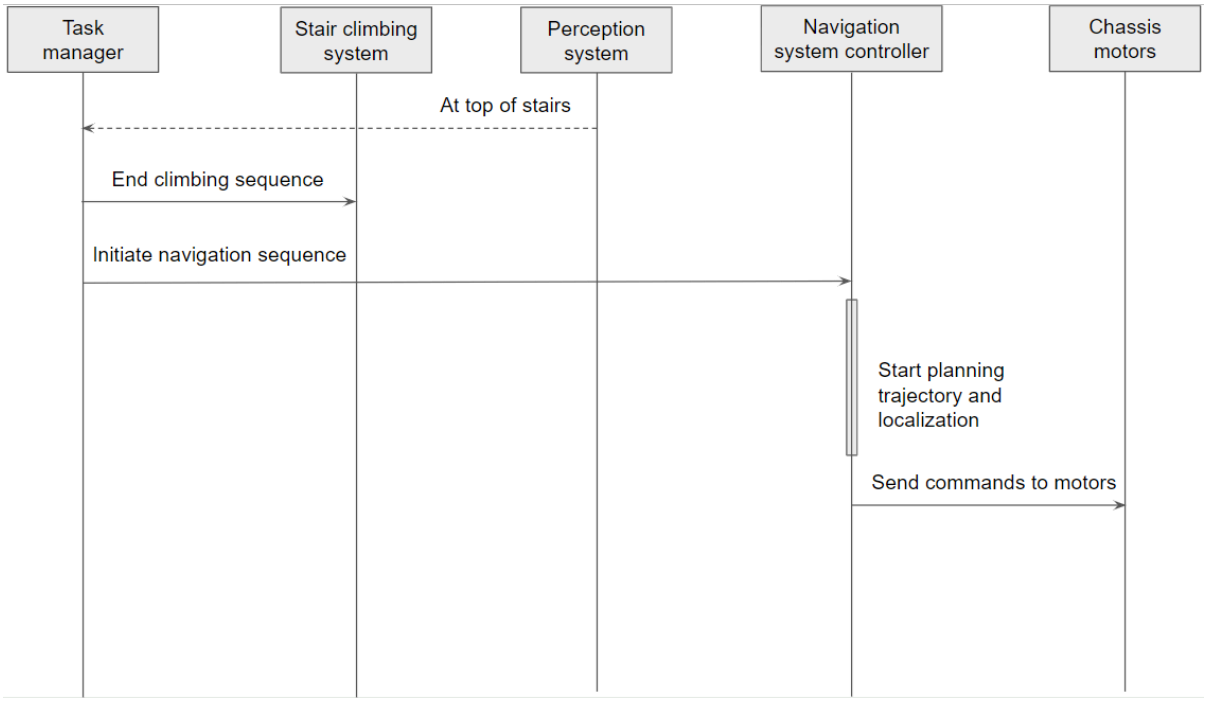


Figure 11.2.2.1 Sequence Diagram for Exit stairs

11.2.3 Navigate on the same floor: The robot is tasked from point A to point B on the same floor. The task manager first initiates the navigation sequence by sending a signal to the navigation system. The perception system will be continuously sending images to the motion planner and localization planner. The localization planner uses the images to estimate the current pose of the robot, and sends the pose to the motion planner. Based on the received images and robot pose, the motion planner will plan the trajectory from the current robot pose to the goal, updating each timestep to account for drift and other errors. The motion planner will also send the trajectory to the controller, which in turn calculates the corresponding motion commands to follow the trajectory, and sends them to the motors. Once the goal is reached, the perception system will send a signal to the task manager, which will stop the robot operations.

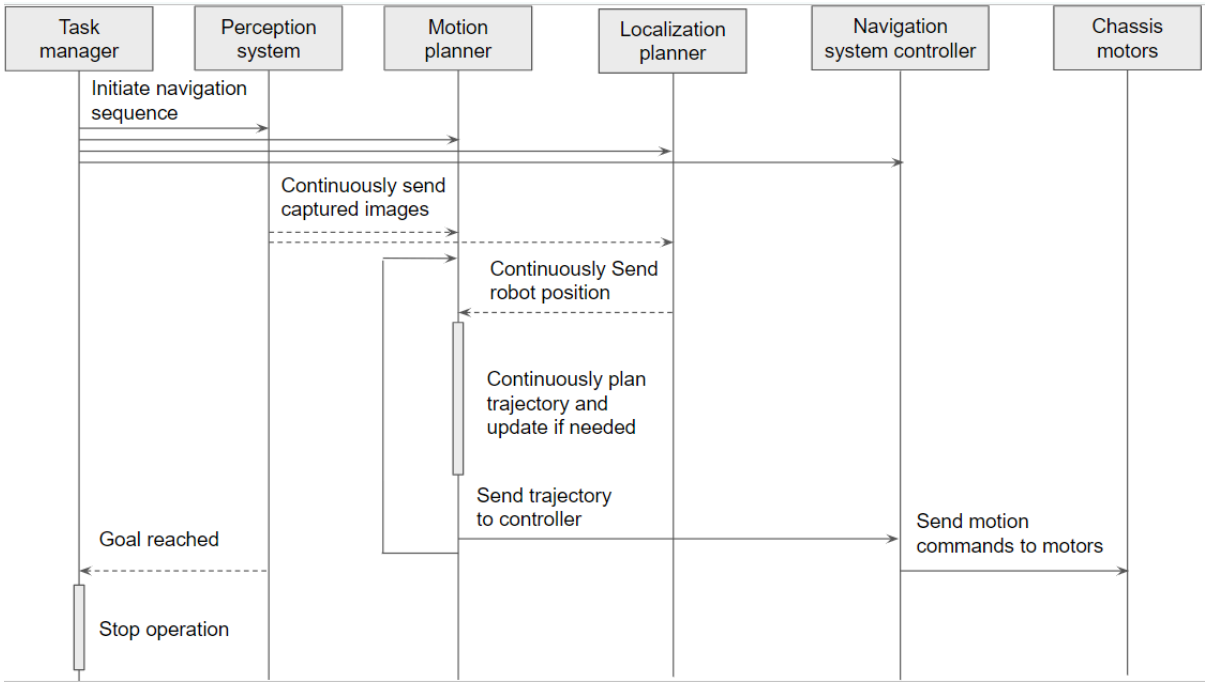


Figure 11.2.3.1 Sequence Diagram for Navigation on the same floor

11.2.4 Tip over: The robot tips over on one side. When the camera detects any abnormal rotations, it will send a signal to the task manager. The task manager will analyze the state of the robot by reading the encoders of motors and the images, and determine if tip over occurred. If tip over has occurred, the task manager will send a signal to the navigation system to stop operation, then it will send a signal to the operator notifying them.

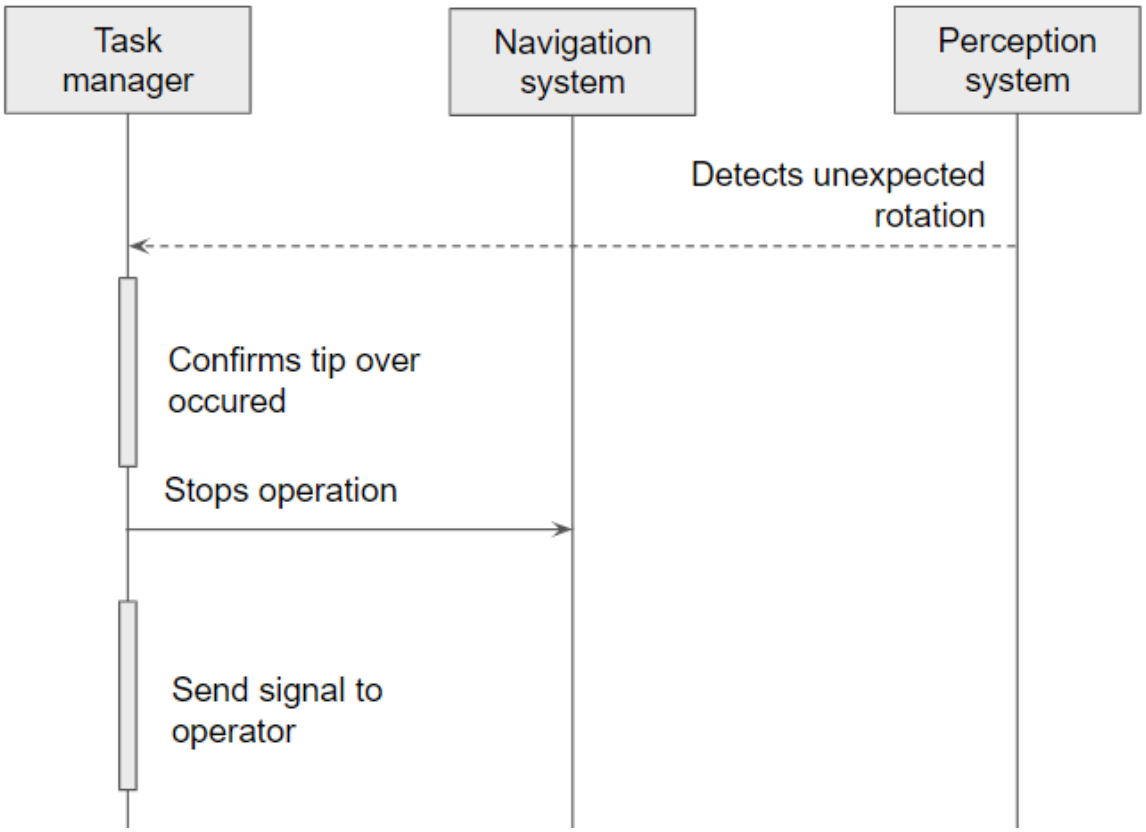


Figure 11.2.4.1 Sequence Diagram for Tip over

11.3 Trade Studies of System Alternatives

11.3.1 Robot overall mechanism:

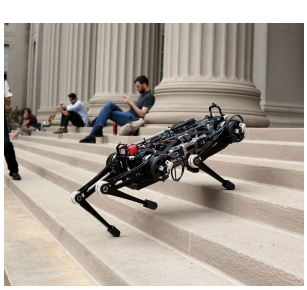
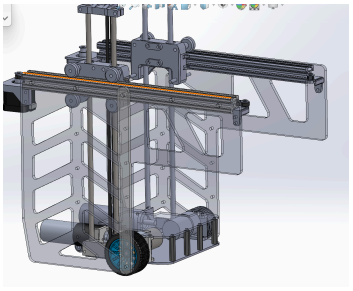
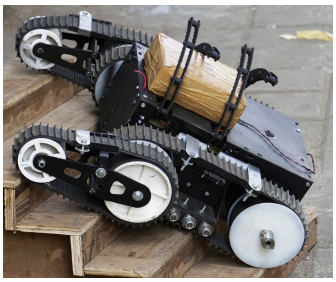
This trade study assists in selecting the type of robot design we will build. The robot design should be one that is capable of both climbing stairs and traveling on flat ground.

	Importance Weight	Legged Robot (Integrated*)	Robot with Dedicated Climbing Module	Tread Drive Robot (Integrated*)
Speed-Climbing	**3**	**4**	**2**	**5**
Speed-Flat Floor	**3**	**2**	**5**	**5**

Navigation				
Stability-Climbing	**4**	**3**	**5**	**1**
Consistency-Climbing	**4**	**4**	**5**	**2**
Robustness-Overall	**3**	**3**	**4**	**5**
Cost	**-1**	**5**	**3**	**2**
Manufacturability	**-3**	**4**	**2**	**2**
Total Score		**38**	**64**	**49**

***Integrated: mechanism will be used for both stair climbing and navigation on flat floors.**
**** Sketches of robot overall concepts in Appendix**

When performing the trade study, some factors that we put the most weighting on was the ability of the robot to climb stairs reliably and consistently. This is because our project focuses on stair climbing, and also because faults related to stair climbing(e.g losing balance and falling over) will likely cause serious damage to the robot, which is undesired. Based on the result, we decided to construct a robot equipped with a specialized climbing module to effectively navigate both stairs and flat surfaces. Using such a module also has an advantage over the other designs where the robot's orientation will be level to the ground at all times, making balancing the food easier.

Overall concept selection: Legged, dedicated climbing mechanism, tread

11.3.2 Climbing subsystem Design

*** Sketches of subsystem concepts below**

After deciding to build a dedicated climbing system, we then conducted a trade study to select the most suitable mechanism we should use to drive the climbing subsystem

	Importanc	**Gear Ring**	**Screw +**	**Timing**	**Legs**	**Wheel-base**

	e Weight		Timing Belt	Belt only		d mechanism
Speed	**3**	**4**	**1**	**4**	**3**	**4**
Stability	**5**	**4**	**5**	**5**	**1**	**2**
Consistency	**4**	**5**	**5**	**5**	**2**	**2**
Robustness	**3**	**3**	**5**	**4**	**3**	**2**
Cost	**-1**	**3**	**2**	**2**	**4**	**2**
Manufacturability	**-3**	**4**	**2**	**2**	**4**	**2**
Total Score		**46**	**55**	**61**	**15**	**28**

Just like what we did for the previous trade study, we placed a large weighting on stability and consistency. This effectively ruled out using legs or a wheel based mechanism. When comparing the remaining options, the gear ring was eliminated due to the difficulties associated with manufacturing the parts. Finally, timing belts have the additional boost in speed, as validated in our prototype, making it our top choice. Based on the result, we decided to use the timing belt for both the x-axis and z-axis for the stair climbing subsystem.

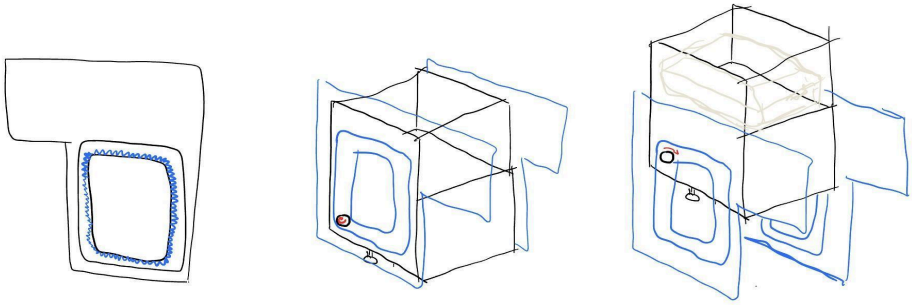
11.3.3 Navigation Subsystem

We need a navigation subsystem so the robot can travel on flat ground. We conducted a trade study to compare three different types of drivetrains.

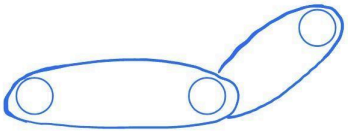
	Importance Weight	Four-wheel drive	Two-wheel drive with caster wheel	Two-wheel drive (self-balancing)	Tread drive	Ballbot
Speed	**4**	**5**	**5**	**4**	**3**	**3**
Stability	**3**	**5**	**5**	**2**	**5**	**1**
Reliability	**5**	**5**	**4**	**2**	**4**	**2**
Cost	**-2**	**5**	**4**	**4**	**5**	**4**
Weight	**-3**	**5**	**4**	**3**	**4**	**2**

Manufacturability	**-3**	**3**	**3**	**4**	**4**	**5**
Maneuverability	**5**	**3**	**4**	**2**	**3**	**2**
Total		**41**	**46**	**13**	**28**	**6**

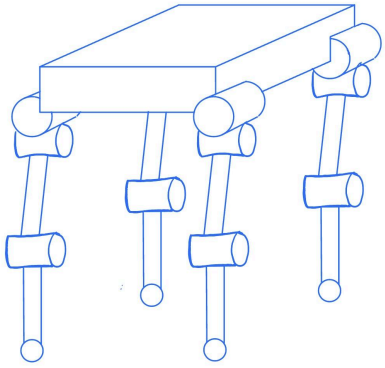
When conducting this trade study, we realized that there wasn't much of a difference in terms of performance between a four wheel drive and a two wheel drive with caster wheel. Since our requirements for navigation performance are relatively relaxed, the main factor that differentiated the two was cost and the weight of the mechanism. Based on our need and the result from our trade study, we decided to build a two wheel drive with caster wheel chassis for our robot. .It offers a good balance between stability and maneuverability.



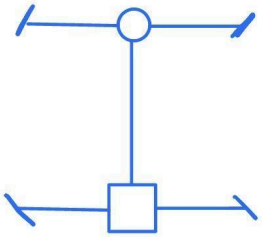
Climbing subsystem concept: Gear Ring



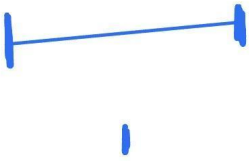
Climbing & Navigation subsystem concept: Tread



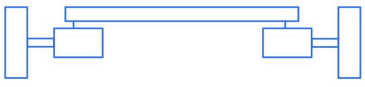
Climbing & Navigation subsystem concept: Legs



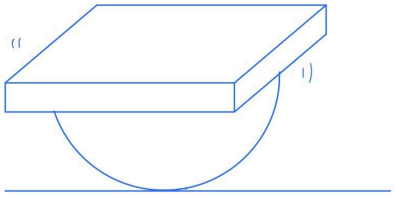
Navigation subsystem concept: Four-Wheel Drive



Navigation subsystem concept: Three-Wheel Drive



Navigation subsystem concept: Two-Wheel Drive



Navigation subsystem concept: Ballbot

11.4 Trade studies of components

** For trade studies, we have separate charts for the specifications of each component and scoring. The specifications are from the retailer's website, and the scoring is a normalized representation of that specification, in comparison to the alternatives. The highlighted component is what we select.*

11.4.1 Stair climbing Z-axis* Motor

This is the motor that will drive the z-axis movement of the stair climbing mechanism. When considering various options, our criteria were set mainly based on the following aspects: The torque of the motor, which determines the maximum weight our robot can carry; its RPM, which determines the speed of climbing; being structurally strong since a significant amount of force and torque will be exerted on the shaft and gearbox; its mass, since we want to make the robot as light as possible to aid stair climbing; and the cost of the motor. We only considered the z-axis motor in our trade study because the x-axis requires much less force, which means a stepper motor is enough. This was also validated in our prototype.

Spec

	Weight	Nm17 Stepper	555 Planetary Geared servo(1:27)	545 Spur Geared servo (1:30)	DJI M2006 Brushless servo
Mass(g)	**3**	**255**	**480**	**290**	**90**
Cost(usd)	**4**	**8**	**25**	**12**	**80 (include proprietary controller)**
Torque(NM)	**5**	**0.4**	**2.5**	**0.5**	**1**
RPM	**3**	**540**	**440**	**380**	**500**
Gearbox Strength(estimated rating out of 5)	**2**	**5(no gearbox)**	**5**	**3**	**4**

Rating

	Weight	Nm17 Stepper	555 Planetary Geared servo(1:27)	545 Spur Geared servo (1:30)	DJI M2006 Brushless servo
Mass	**3**	**3**	**1**	**3**	**5**
Cost	**-4**	**1**	**2**	**1**	**5**
Torque	**5**	**1**	**5**	**2**	**3**
RPM	**3**	**5**	**4**	**3**	**5**
Gearbox Strength	**2**	**5**	**5**	**3**	**4**
Total Score		**35**	**42**	**30**	**33**

Based on the results, the 555 Planetary Geared servo motor with 1:27 gear ratio is the most suitable option with the lightest weight, less cost, the highest torque, high RPM and the strongest gearbox system among competitors.

11.4.2 Camera

The camera will be used to recognize features of the environment to conduct localization and correct for the odometry error during navigation. It will also be used to detect stairs and calculate stair height. 3D information such as depth sensing or stereo vision will be helpful for both of these tasks. Besides 3d information, we will also consider its resolution, FPS, size, and cost as the most essential criterias.

Spec

	Weight	**Intel® RealSenseTM Camera 400 Series**	**IMX219-83 Stereo Camera**	**BetaFPV C02 2.1mm 1200TVL**
Resolution	**2**	**1920x1080**	**3280x2464**	**1920x1080**
Cost	**-4**	**112.36***	**56.99**	**14.99**
FPS	**1**	**90**	**N/A**	**N/A**
Camera type	**5**	**RGB-D**	**Stereo**	**RGB**
Physical Dimension	**2**	**84.00mm x 10.00mm x 5.00mm**	**24x85mm**	**13x10x11mm**

***can get it for free:)**

Rating

	Weight	**Intel® RealSenseTM Camera 400 Series**	**IMX219-83 Stereo Camera**	**BetaFPV C02 2.1mm 1200TVL**

Resolution	**2**	**3**	**5**	**3**
Cost	**-4**	**0**	**4**	**2**
FPS	**1**	**5**	**0**	**0**
Camera type	**5**	**5**	**3**	**1**
Physical Dimension	**2**	**3**	**2**	**5**
Total Score		**42**	**13**	**13**

***We scored Intel camera Cost as 0 since we can get this for free.**

Based on the results, the Intel RealSenseTM Camera 400, a RGB-D camera, seems to be the best fit. Its resolution is high enough as we need. Besides, it has high FPS that can update the surroundings fast enough.

11.4.3 Computer

We are using a central computer to do the computationally intensive tasks like stair detection and localization. Localization algorithms will be handled by the CPU, while computer vision tasks will need the GPU. Ram and weight will be less important since our program will likely not use much ram, and the weight of the computer is low compared to other parts. However, its cost is quite important since it might be the most costly component of the robot.

Spec

	Weight	**Jetson Nano**	**Intel Nuc 11**	**Jetson TX2**
Mass(g)	**2**	**77**	**520**	**314**
Cost(usd)	**-7**	**149**	**559**	**479**

CPU	**4**	**Quad-Core ARM® Cortex®-A57**	**Intel i7-1165g**	**Dual-Core NVIDIA Denver 2 64-Bit CPU + Quad-Core ARM® Cortex®-A57 MPCore**
GPU	**4**	**NVIDIA Maxwell architecture with 128 core**	**Intel IGPU***	**256-core NVIDIA Pascal™**
Ram	**2**	**4**	**32**	**8**

*** (intel IGPU likely won't work with our software)**

Rating

	Weight	**Jetson Nano**	**Intel Nuc 11**	**Jetson TX2**
Mass	**2**	**5**	**1**	**3**
Cost	**-7**	**1**	**5**	**4**
CPU	**4**	**2**	**5**	**4**
GPU	**4**	**3**	**1**	**5**
Ram	**2**	**1**	**5**	**3**
Total Score		**25**	**1**	**20**

Based on the result, we will choose the Jetson Nano as our central computer due to its lightweight and good CPU and GPU for handling the tasks, such as localization. Although the Jetson TX2 has more compute power than its Nano counterpart, we believe it is excessive for the tasks that our robot will perform.

11.5 Failure modes and fault trees

11.5.1 Failure Mode I: Fall off stairs

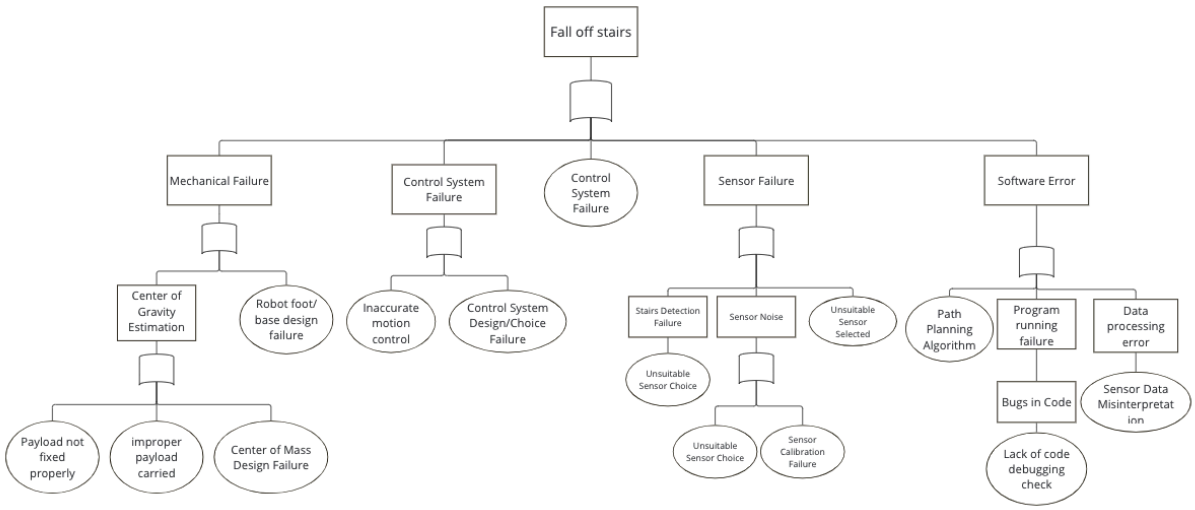


Figure 9.5.1 Fault tree for Failure ModeI, Fall off stairs

11.5.2 Failure Mode II: Localization Failure

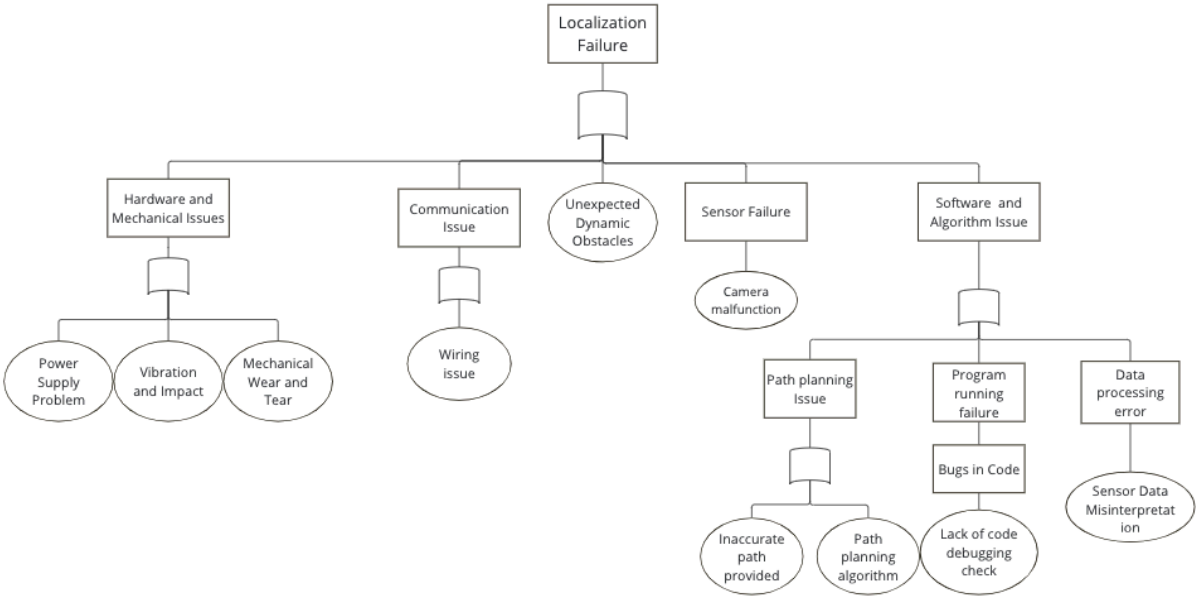


Figure 9.5.2 Fault tree for Failure ModeI, Localization Failure

11.6 Detailed assembly Diagram and Instructions

**** We use numbers in parentheses to denote multiple copies of the same part being used in the step.***

<u>Step1</u>: Mount Z-axis motor holder(2), m4 standoffs(4), rear_reinforcement_plate, rear_wheel_mount(2) to chassis bottom plate using screws. Then mount 520 dc motors(2) and 555 dc motors(2) to Z axis motor holder(2) with screws

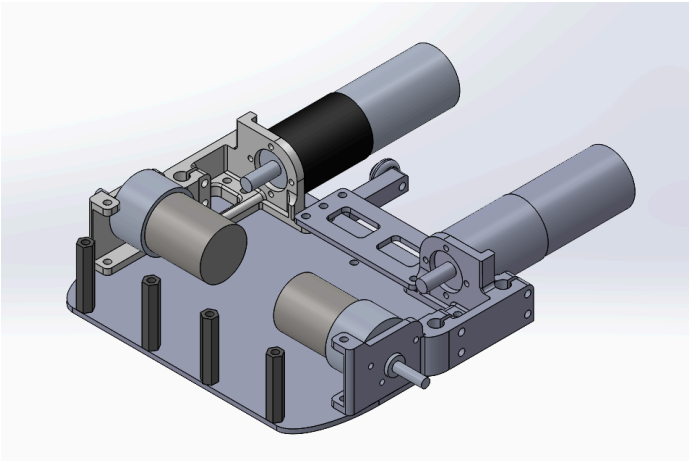


Figure 9.6.1 Chassis Assembly Diagram I

<u>Step2</u>: Attach 65mm wheels(2) to 520 dc motors(2), and 2gt-20t-8mm pulleys(2) to 555 dc motors(2), and tighten the set screws. Then, attach rear wheel bushings(2) and rear wheels (2) to rear wheel supports(2) with screws.

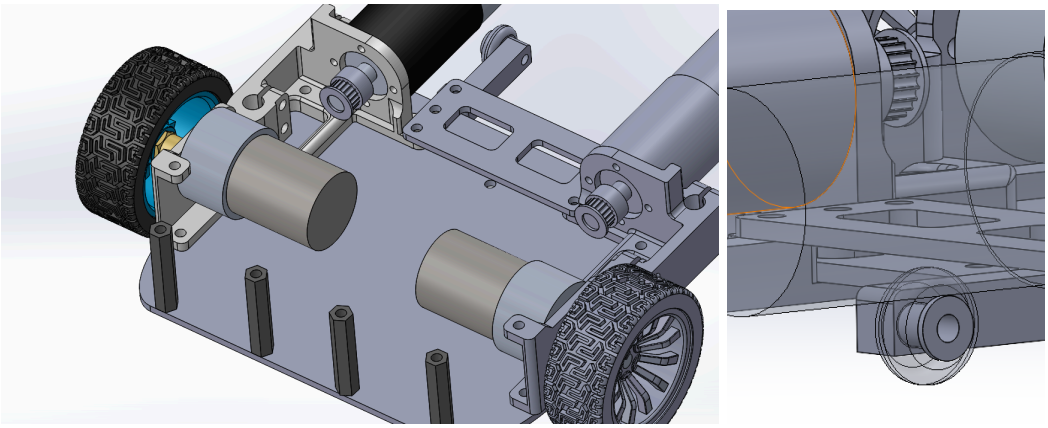


Figure 9.6.2 Chassis Assembly Diagram II

<u>Step3</u>: Insert 8mm linear bearings (2) into the slide block. Then mount roller standoffs(4) and 8020 rollers (4) to slide block with screws. Attach 2gt-20t-passive pulley on the Zaxis top pulley holder with a screw.

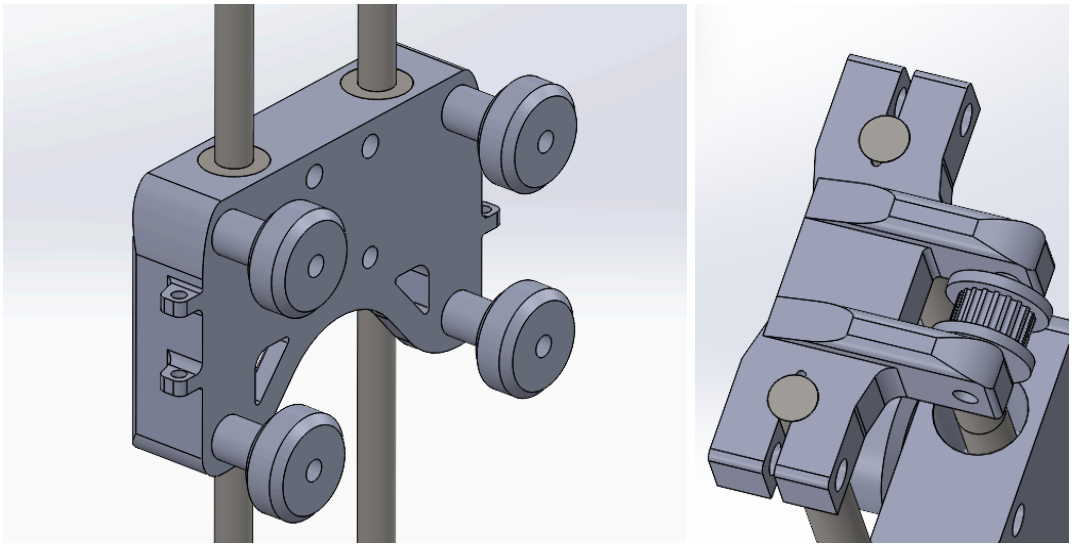


Figure 9.6.3 Slide Block and Z-axis Assembly Diagram I

<u>Step4</u>: Clamp 8x400 steel rods(2) onto Z axis motor holder with screws, then slide in the assembled slide block, and clamp the assembled Z axis top pulley holder at the end of the rods with screws.

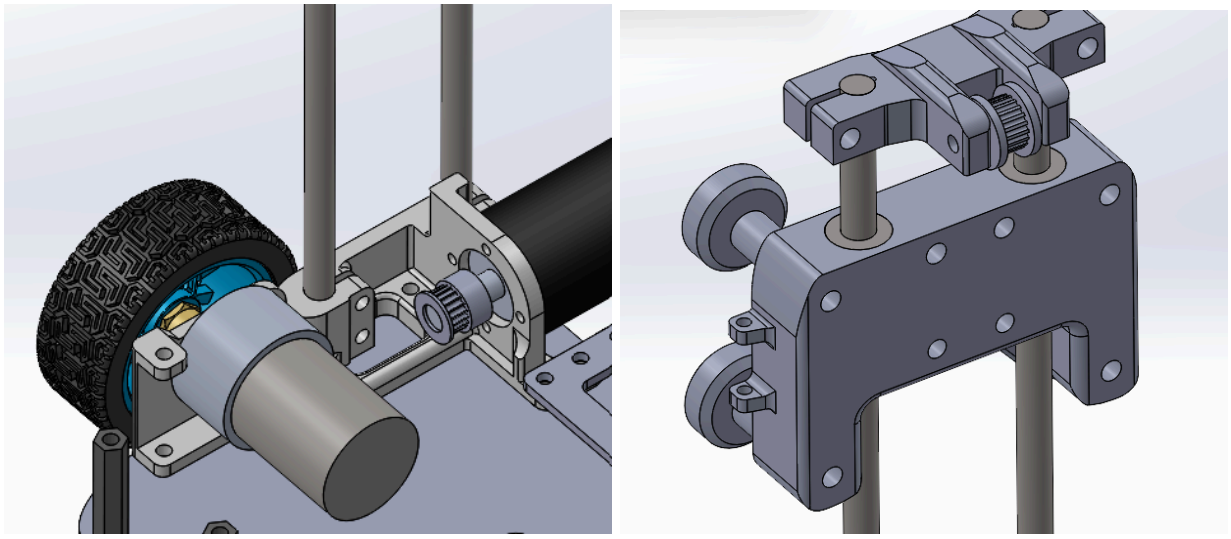


Figure 9.6.4 Z-axis System Assembly Diagram II

<u>Step5</u>: Attach 2gt-16t-5mm pulley onto NEMA17 stepper motor with set screw. Attach the motor to the stepper bracket, and attach the stepper bracket to the side frame with screws. Attach the side frame to the 8020 aluminum extrusion, and slide the extrusion in between the rollers.

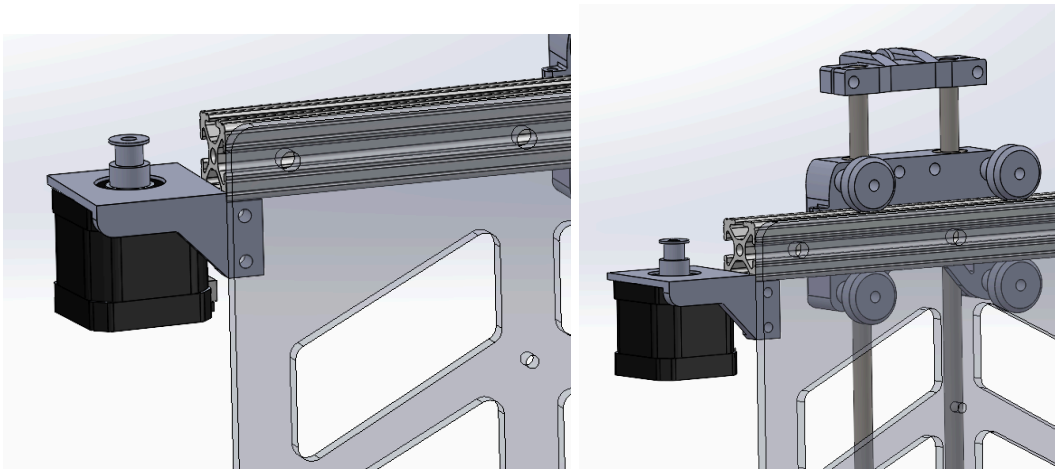


Figure 9.6.5 X-axis System Assembly Diagram I

Step6:Attach the horizontal passive pulley holder to the side frame with screws, then attach the 2gt-16t passive pulley and passive pulley bushing onto the holder. Mount side frame reinforcement bar 1 and 2 on to the side frame with screws

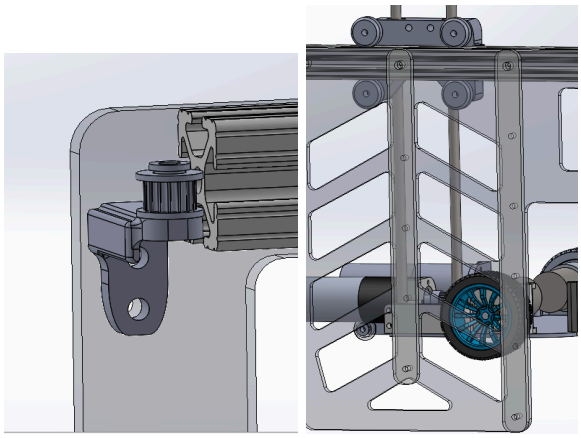


Figure 9.6.6 X-axis System Assembly Diagram II

Step7: Attach m3x30 screws(2) on both sides of the slide block, then attach the X axis and Z axis belts. Adjust belt length as needed. Tie the horizontal belt onto the m3x30 screws, and clamp the Z axis belt on the slide block with Z axis belt clips(2).

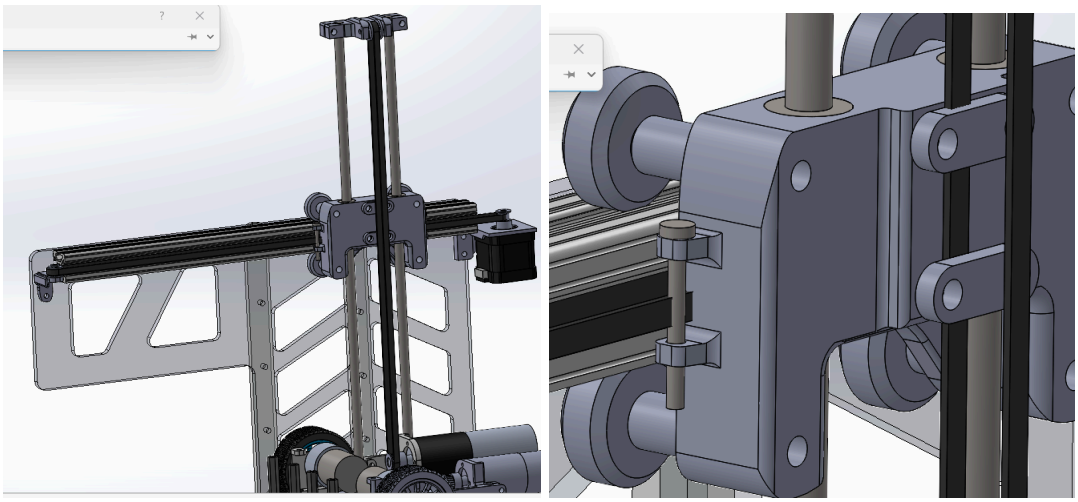


Figure 9.6.7 Stair Climbing Subsystem Z n X-axis Integration Assembly Diagram

<u>Step8</u>: Repeat steps 3-7 for the other side of the robot. Then attach the chassis top plate to the Z axis motor holder and the m4 standoffs. Connect the wirings and place the electronics either on the chassis top plate, or between the top and bottom plates

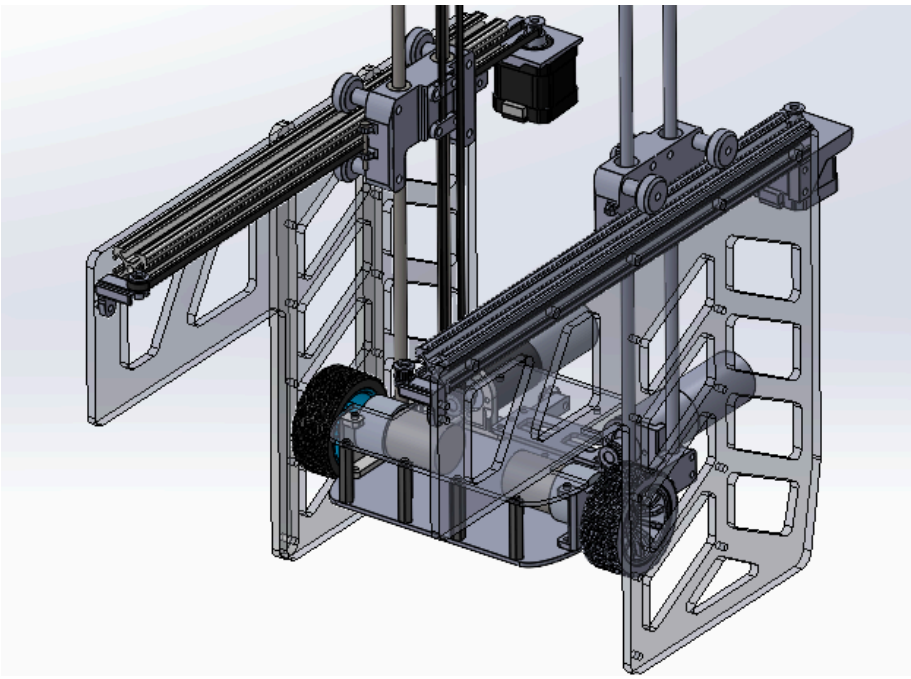
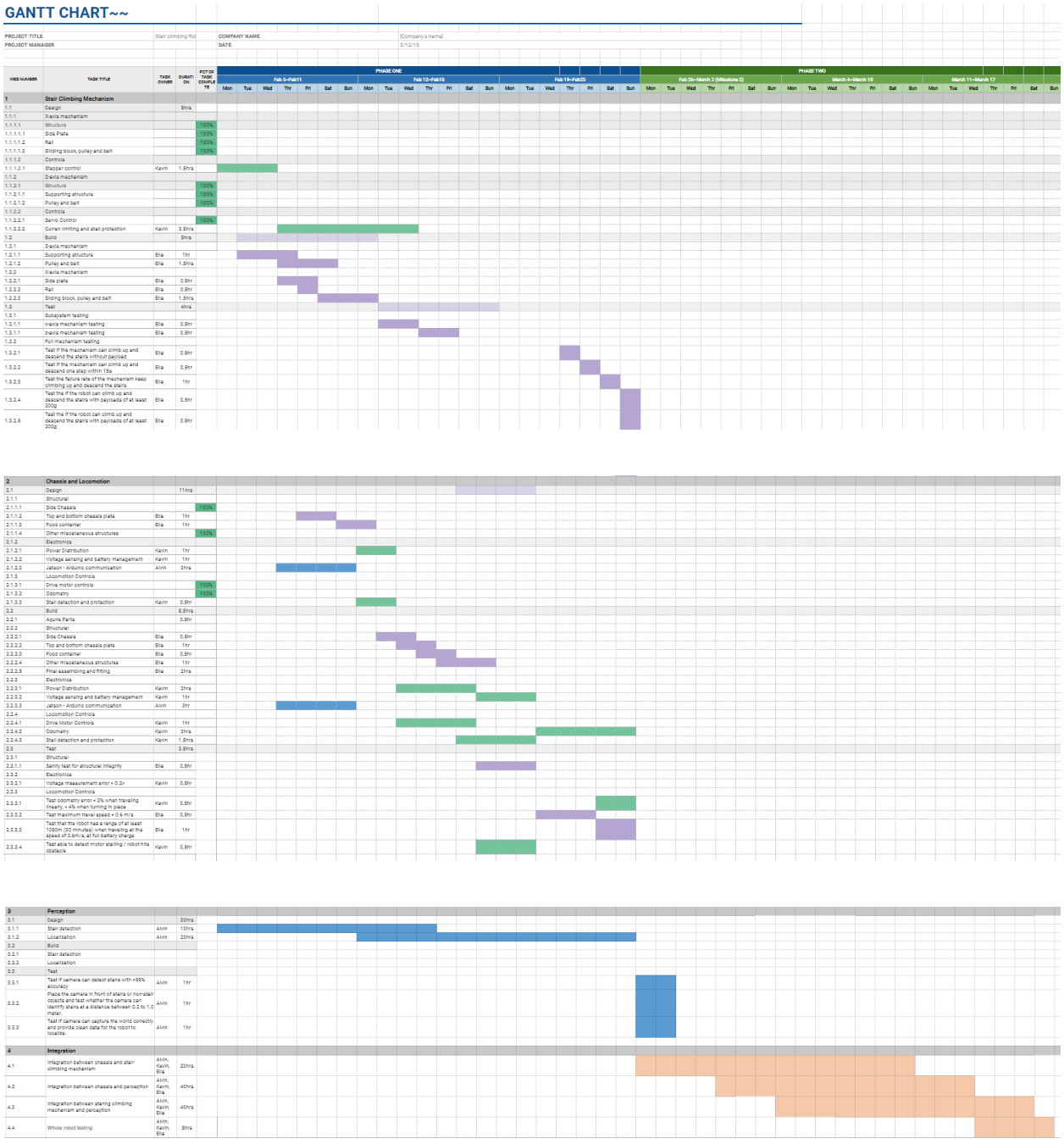
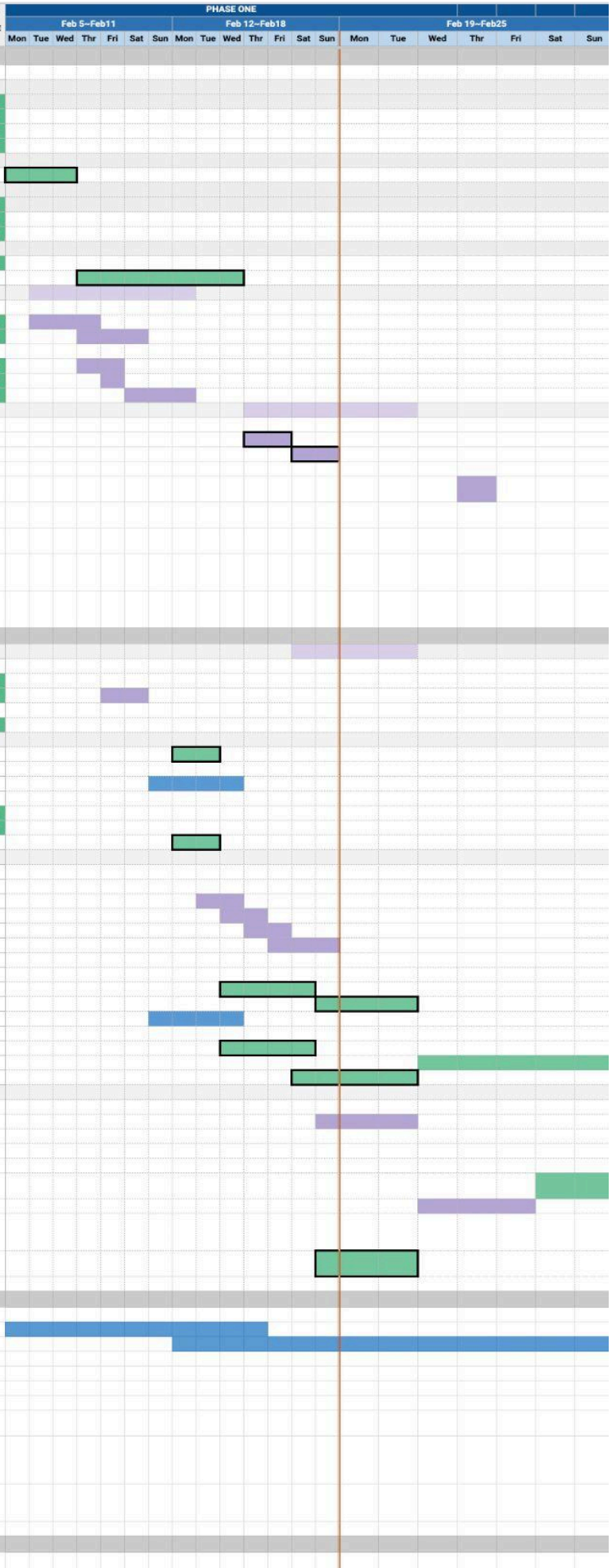


Figure 9.6.8 Stair Climbing Subsystem and Chassis Integration Assembly Diagram

11.7 Gantt Chart



WBS NUMBER	TASK TITLE	TASK OWNER	DURATION	PCT OF TASK COMPLETE
1	**Stair Climbing Mechanism**			
1.1	Design		5hrs	
1.1.1	X-axis mechanism			
1.1.1.1	Structure			100%
1.1.1.1.1	Side Plate			100%
1.1.1.1.2	Rail			100%
1.1.1.1.3	Siliding block, pulley and belt			100%
1.1.1.2	Controls			
1.1.1.2.1	Stepper control	Kevin	1.5hrs	
1.1.2	Z-axis mechanism			
1.1.2.1	Structure			100%
1.1.2.1.1	Supporting structure			100%
1.1.2.1.2	Pulley and belt			100%
1.1.2.2	Controls			
1.1.2.2.1	Servo Control			100%
1.1.2.2.2	Curren limiting and stall protection	Kevin	3.5hrs	
1.2	Build		5hrs	
1.2.1	Z-axis mechanism			
1.2.1.1	Supporting structure	Ella	1hr	100%
1.2.1.2	Pulley and belt	Ella	1.5hrs	100%
1.2.2	X-axis mechanism			
1.2.2.1	Side plate	Ella	0.5hr	100%
1.2.2.2	Rail	Ella	0.5hr	100%
1.2.2.3	Sliding block, pulley and belt	Ella	1.5hrs	100%
1.3	Test		4hrs	
1.3.1	Subsystem testing			
1.3.1.1	x-axis mechanism testing	Ella	0.5hr	
1.3.1.1	z-axis mechanism testing	Ella	0.5hr	
1.3.2	Full mechanism testing			
1.3.2.1	Test if the mechanism can climb up and descend the stairs without payload	Ella	0.5hr	
1.3.2.2	Test if the mechanism can climb up and descend one step within 15s	Ella	0.5hr	
1.3.2.3	Test the failure rate of the mechanism keep climbing up and descend the stairs	Ella	1hr	
1.3.2.4	Test the if the robot can climb up and descend the stairs with payloads of at least 200g	Ella	0.5hr	
1.3.2.5	Test the if the robot can climb up and descend the stairs with payloads of at least 200g	Ella	0.5hr	
2	**Chassis and Locomotion**			
2.1	Design		11hrs	
2.1.1	Structural			
2.1.1.1	Side Chassis			100%
2.1.1.2	Top and bottom chassis plate	Ella	1hr	100%
2.1.1.3	Food container	Ella	1hr	
2.1.1.4	Other miscellaneous structures			100%
2.1.2	Electronics			
2.1.2.1	Power Distribution	Kevin	1hr	
2.1.2.2	Voltage sensing and battery management	Kevin	1hr	
2.1.2.3	Jetson - Arduino communication	Alvin	3hrs	
2.1.3	Locomotion Controls			
2.1.3.1	Drive motor controls			100%
2.1.3.2	Odometry			100%
2.1.3.3	Stall detection and protection	Kevin	0.5hr	
2.2	Build		8.5hrs	
2.2.1	Aquire Parts		0.5hr	
2.2.2	Structural			
2.2.2.1	Side Chassis	Ella	0.5hr	
2.2.2.2	Top and bottom chassis plate	Ella	1hr	
2.2.2.3	Food container	Ella	0.5hr	
2.2.2.4	Other miscellaneous structures	Ella	1hr	
2.2.2.5	Final assembling and fitting	Ella	2hrs	
2.2.3	Electronics			
2.2.3.1	Power Distribution	Kevin	2hrs	
2.2.3.2	Voltage sensing and battery management	Kevin	1hr	
2.2.3.3	Jetson - Arduino communication	Alvin	3hr	
2.2.4	Locomotion Controls			
2.2.4.1	Drive Motor Controls	Kevin	1hr	
2.2.4.2	Odometry	Kevin	3hrs	
2.2.4.3	Stall detection and protection	Kevin	1.5hrs	
2.3	Test		3.5hrs	
2.3.1	Structural			
2.3.1.1	Sanity test for structural integrity	Ella	0.5hr	
2.3.2	Electronics			
2.3.2.1	Voltage measurement error < 0.2v	Kevin	0.5hr	
2.3.3	Locomotion Controls			
2.3.3.1	Test odometry error < 2% when traveling linearly, < 4% when turning in place	Kevin	0.5hr	
2.3.3.2	Test maximum travel speed > 0.6 m/s	Ella	0.5hr	
2.3.3.3	Test that the robot has a range of at least 1080m (30 minutes) when traveling at the speed of 0.6m/s, at full battery charge	Ella	1hr	
2.3.3.4	Test able to detect motor stalling / robot hits obstacle	Kevin	0.5hr	
3	**Perception**			
3.1	Design		30hrs	
3.1.1	Stair detection	Alvin	10hrs	
3.1.2	Localization	Alvin	20hrs	
3.2	Build			
3.2.1	Stair detection			
3.2.2	Localization			
3.3	Test			
3.3.1	Test if camera can detect stairs with >95% accuracy	Alvin	1hr	
3.3.2	Place the camera in front of stairs or non-stair objects and test whether the camera can identify stairs at a distance between 0.2 to 1.0 meter.	Alvin	1hr	
3.3.3	Test if camera can capture the world correctly and provide clean data for the robot to localize.	Alvin	1hr	
4	**Integration**			
	Integration between chassis and stair	Alvin,		



WBS NUMBER	TASK TITLE	TASK OWNER	DURATION	PCT OF TASK COMPLETE	PHASE TWO																		
					Feb 26--March 3 (Milestone 2)							March 4--March 10							March 11--March 17						
					Mon	Tue	Wed	Thr	Fri	Sat	Sun	Mon	Tue	Wed	Thr	Fri	Sat	Sun	Mon	Tue	Wed	Thr	Fri	Sat	Sun
1	**Stair Climbing Mechanism**																								
1.1	Design		5hrs																						
1.1.1	X-axis mechanism																								
1.1.1.1	**Structure**			100%																					
1.1.1.1.1	Side Plate			100%																					
1.1.1.1.2	Rail			100%																					
1.1.1.1.3	Siliding block, pulley and belt			100%																					
1.1.1.2	Controls																								
1.1.1.2.1	Stepper control	Kevin	1.5hrs	100%																					
1.1.2	Z-axis mechanism																								
1.1.2.1	**Structure**			100%																					
1.1.2.1.1	Supporting structure			100%																					
1.1.2.1.2	Pulley and belt			100%																					
1.1.2.2	Controls																								
1.1.2.2.1	Servo Control			100%																					
1.1.2.2.2	Curren limiting and stall protection	Kevin	3.5hrs	100%																					
1.2	Build		5hrs																						
1.2.1	Z-axis mechanism			100%																					
1.2.1.1	Supporting structure	Ella	1hr	100%																					
1.2.1.2	Pulley and belt	Ella	1.5hrs	100%																					
1.2.2	X-axis mechanism			100%																					
1.2.2.1	Side plate	Ella	0.5hr	100%																					
1.2.2.2	Rail	Ella	0.5hr	100%																					
1.2.2.3	Sliding block, pulley and belt	Ella	1.5hrs	100%																					
1.3	Test		4hrs																						
1.3.1	Subsystem testing			100%																					
1.3.1.1	x-axis mechanism testing	Ella	0.5hr	100%																					
1.3.1.1	z-axis mechanism testing	Ella	0.5hr	100%																					
1.3.2	Full mechanism testing			100%																					
1.3.2.1	Test if the mechanism can climb up and descend the stairs without payload	Ella	0.5hr	100%																					
1.3.2.2	Test if the mechanism can climb up and descend one step within 15s	Ella	0.5hr	100%																					
1.3.2.3	Test the failure rate of the mechanism keep climbing up and descend the stairs	Ella	1hr																						
1.3.2.4	Test the if the robot can climb up and descend the stairs with payloads of at least 200g	Ella	0.5hr																						
1.3.2.5	Test the if the robot can climb up and descend the stairs with payloads of at least 200g	Ella	0.5hr																						
2	**Chassis and Locomotion**																								
2.1	Design		11hrs																						
2.1.1	Structural																								
2.1.1.1	Side Chassis			100%																					
2.1.1.2	Top and bottom chassis plate	Ella	1hr	100%																					
2.1.1.3	Food container	Ella	1hr																						
2.1.1.4	Other miscellaneous structures			100%																					
2.1.2	Electronics																								
2.1.2.1	Power Distribution	Kevin	1hr	100%																					
2.1.2.2	Voltage sensing and battery management	Kevin	1hr	100%																					
2.1.2.3	Jetson - Arduino communication	Alvin	3hrs	100%																					
2.1.3	Locomotion Controls																								
2.1.3.1	Drive motor controls			100%																					
2.1.3.2	Odometry			100%																					
2.1.3.3	Stall detection and protection	Kevin	0.5hr																						
2.2	Build		8.5hrs																						
2.2.1	Aquire Parts		0.5hr	100%																					
2.2.2	Structural			100%																					
2.2.2.1	Side Chassis	Ella	0.5hr	100%																					
2.2.2.2	Top and bottom chassis plate	Ella	1hr	100%																					
2.2.2.3	Food container	Ella	0.5hr	100%																					
2.2.2.4	Other miscellaneous structures	Ella	1hr	100%																					
2.2.2.5	Final assembling and fitting	Ella	2hrs	100%																					
2.2.3	Electronics			100%																					
2.2.3.1	Power Distribution	Kevin	2hrs	100%																					
2.2.3.2	Voltage sensing and battery management	Kevin	1hr	100%																					
2.2.3.3	Jetson - Arduino communication	Alvin	3hr																						
2.2.4	Locomotion Controls																								
2.2.4.1	Drive Motor Controls	Kevin	1hr	100%																					
2.2.4.2	Odometry	Kevin	3hrs	100%																					
2.2.4.3	Stall detection and protection	Kevin	1.5hrs	100%																					
2.3	Test		3.5hrs																						
2.3.1	Structural																								
2.3.1.1	Sanity test for structural integrity	Ella	0.5hr																						
2.3.2	Electronics																								
2.3.2.1	Voltage measurement error < 0.2v	Kevin	0.5hr	100%																					
2.3.3	Locomotion Controls			100%																					
2.3.3.1	Test odometry error < 2% when traveling linearly, < 4% when turning in place	Kevin	0.5hr	100%																					
2.3.3.2	Test maximum travel speed > 0.6 m/s	Ella	0.5hr	100%																					
2.3.3.3	Test that the robot has a range of at least 1080m (30 minutes) when traveling at the speed of 0.6m/s, at full battery charge	Ella	1hr																						
2.3.3.4	Test able to detect motor stalling / robot hits obstacle	Kevin	0.5hr	100%																					
3	**Perception**																								
3.1	Design		30hrs																						
3.1.1	Stair detection	Alvin	10hrs																						
3.1.2	Localization	Alvin	20hrs																						
3.2	Build																								
3.2.1	Stair detection																								
3.2.2	Localization																								
3.3	Test																								
3.3.1	Test if camera can detect stairs with >95% accuracy	Alvin	1hr																						
3.3.2	Place the camera in front of stairs or non-stair objects and test whether the camera can identify stairs at a distance between 0.2 to 1.0 meter.	Alvin	1hr																						
3.3.3	Test if camera can capture the world correctly and provide clean data for the robot to localize.	Alvin	1hr																						
4	**Integration**																								
4.1	Integration between chassis and stair climbing mechanism	Alvin, Kevin, Ella	20hrs																						

WBS NUMBER	TASK TITLE	TASK OWNER	DURATION	PCT OF TASK COMPLETE	PHASE THREE												
					March 18~March 24(Milestone 3)							March 25~March 31					
					Mon	Tue	Wed	Thr	Fri	Sat	Sun	Mon	Tue	Wed	Thr	Fri	Sat	Sun
1	**Stair Climbing Mechanism**																	
1.1	Design		5hrs															
1.1.1	X-axis mechanism																	
1.1.1.1	**Structure**			100%														
1.1.1.1.1	Side Plate			100%														
1.1.1.1.2	Rail			100%														
1.1.1.1.3	Siliding block, pulley and belt			100%														
1.1.1.2	Controls																	
1.1.1.2.1	Stepper control	Kevin	1.5hrs	100%														
1.1.2	Z-axis mechanism																	
1.1.2.1	**Structure**			100%														
1.1.2.1.1	Supporting structure			100%														
1.1.2.1.2	Pulley and belt			100%														
1.1.2.2	Controls																	
1.1.2.2.1	Servo Control			100%														
1.1.2.2.2	Curren limiting and stall protection	Kevin	3.5hrs	100%														
1.2	Build		5hrs															
1.2.1	Z-axis mechanism			100%														
1.2.1.1	Supporting structure	Ella	1hr	100%														
1.2.1.2	Pulley and belt	Ella	1.5hrs	100%														
1.2.2	X-axis mechanism			100%														
1.2.2.1	Side plate	Ella	0.5hr	100%														
1.2.2.2	Rail	Ella	0.5hr	100%														
1.2.2.3	Sliding block, pulley and belt	Ella	1.5hrs	100%														
1.3	Test		4hrs															
1.3.1	Subsystem testing			100%														
1.3.1.1	x-axis mechanism testing	Ella	0.5hr	100%														
1.3.1.1	z-axis mechanism testing	Ella	0.5hr	100%														
1.3.2	Full mechanism testing			100%														
1.3.2.1	Test if the mechanism can climb up and descend the stairs without payload	Ella	0.5hr	100%														
1.3.2.2	Test if the mechanism can climb up and descend one step within 15s	Ella	0.5hr	100%														
1.3.2.3	Test the failure rate of the mechanism keep climbing up and descend the stairs	Ella	1hr															
1.3.2.4	Test the if the robot can climb up and descend the stairs with payloads of at least 200g	Ella	0.5hr															
1.3.2.5	Test the if the robot can climb up and descend the stairs with payloads of at least 200g	Ella	0.5hr															
2	**Chassis and Locomotion**																	
2.1	Design		11hrs															
2.1.1	Structural																	
2.1.1.1	Side Chassis			100%														
2.1.1.2	Top and bottom chassis plate	Ella	1hr	100%														
2.1.1.3	Food container	Ella	1hr															
2.1.1.4	Other miscellaneous structures			100%														
2.1.2	Electronics																	
2.1.2.1	Power Distribution	Kevin	1hr	100%														
2.1.2.2	Voltage sensing and battery management	Kevin	1hr	100%														
2.1.2.3	Jetson - Arduino communication	Alvin	3hrs	100%														
2.1.3	Locomotion Controls																	
2.1.3.1	Drive motor controls			100%														
2.1.3.2	Odometry			100%														
2.1.3.3	Stall detection and protection	Kevin	0.5hr															
2.2	Build		8.5hrs															
2.2.1	Aquire Parts		0.5hr	100%														
2.2.2	Structural			100%														
2.2.2.1	Side Chassis	Ella	0.5hr	100%														
2.2.2.2	Top and bottom chassis plate	Ella	1hr	100%														
2.2.2.3	Food container	Ella	0.5hr	100%														
2.2.2.4	Other miscellaneous structures	Ella	1hr	100%														
2.2.2.5	Final assembling and fitting	Ella	2hrs	100%														
2.2.3	Electronics			100%														
2.2.3.1	Power Distribution	Kevin	2hrs	100%														
2.2.3.2	Voltage sensing and battery management	Kevin	1hr	100%														
2.2.3.3	Jetson - Arduino communication	Alvin	3hr															
2.2.4	Locomotion Controls																	
2.2.4.1	Drive Motor Controls	Kevin	1hr	100%														
2.2.4.2	Odometry	Kevin	3hrs	100%														
2.2.4.3	Stall detection and protection	Kevin	1.5hrs	100%														
2.3	Test		3.5hrs															
2.3.1	Structural																	
2.3.1.1	Sanity test for structural integrity	Ella	0.5hr															
2.3.2	Electronics																	
2.3.2.1	Voltage measurement error < 0.2v	Kevin	0.5hr	100%														
2.3.3	Locomotion Controls			100%														
2.3.3.1	Test odometry error < 2% when traveling linearly, < 4% when turning in place	Kevin	0.5hr	100%														
2.3.3.2	Test maximum travel speed > 0.6 m/s	Ella	0.5hr	100%														
2.3.3.3	Test that the robot has a range of at least 1080m (30 minutes) when traveling at the speed of 0.6m/s, at full battery charge	Ella	1hr															
2.3.3.4	Test able to detect motor stalling / robot hits obstacle	Kevin	0.5hr	100%														
3	**Perception**																	
3.1	Design		30hrs															
3.1.1	Stair detection	Alvin	10hrs															
3.1.2	Localization	Alvin	20hrs															
3.2	Build																	
3.2.1	Stair detection																	
3.2.2	Localization																	
3.3	Test																	
3.3.1	Test if camera can detect stairs with >95% accuracy	Alvin	1hr															
3.3.2	Place the camera in front of stairs or non-stair objects and test whether the camera can identify stairs at a distance between 0.2 to 1.0 meter.	Alvin	1hr															
3.3.3	Test if camera can capture the world correctly and provide clean data for the robot to localize.	Alvin	1hr															
4	**Integration**																	
4.1	Integration between chassis and stair climbing mechanism	Alvin, Kevin, Ella	20hrs															
4.2	Integration between chassis and perception	Alvin, Kevin, Ella	40hrs															
4.3	Integration between staring climbing mechanism and perception	Alvin, Kevin,	40hrs															