EE175AB Final Report

Semi-Autonomous Earth Rover

EE 175AB Final Report

Department of Electrical Engineering, UC Riverside

Project Team Member(s)	Iraj Shrotri, Nicholas Casillas, Franscisco Lam
Date Submitted	3/17/23
Section Professor	22
Revision	Revision 1.1
URL of Project YouTube Videos, Wiki/Webpage	https://www.youtube.com/playlist?list=PLSsPBfPIpTdGM5SNih e1x88IuJvv46hb4
Permanent Emails of all team members	shrotriiraj@gmail.com nick.casillas.nc@gmail.com francisco21lam@gmail.com

Summary

This report presents our project which aims to provide an automated and intelligent solution for plant monitoring and growth optimization.

Revisions

Version	Description of Version	Author(s)	Date Completed	Approval
Version Number	Information about the revision. This table does not need to be filled in whenever a document is touched, only when the version is being upgraded.	Nicholas Casillas, Iraj Shrotri, Francisco Lam	03/21/23	
0.5	First draft	Nicholas Casillas, Iraj Shrotri, Francisco Lam	02/02/2023	
0.8	List major changes from previous version, reference to the related Engineering Change Notices (ECN) or Engineering Design Notes (EDN).	Nicholas Casillas, Iraj Shrotri, Francisco Lam	03/21/2023	
1.1	Final Draft	Iraj Shrotri	06/13/2023	

Table of Contents

REVISIONS **3**

TABLE OF CONTENTS **4**

1 * EXECUTIVE SUMMARY **6**

2 * INTRODUCTION **7**

2.1 * DESIGN OBJECTIVES AND SYSTEM OVERVIEW 7
2.2 * BACKGROUNDS AND PRIOR ART 7
2.3 * DEVELOPMENT ENVIRONMENT AND TOOLS 8
2.4 * RELATED DOCUMENTS AND SUPPORTING MATERIALS 8
2.5 * DEFINITIONS AND ACRONYMS 8

3 * DESIGN CONSIDERATIONS **9**

3.1 * REALISTIC CONSTRAINTS 9
3.2 SYSTEM ENVIRONMENT AND EXTERNAL INTERFACES 9
3.3 * INDUSTRY STANDARDS 9
3.4 * KNOWLEDGE AND SKILLS 10
3.5 * BUDGET AND COST ANALYSIS 10
3.6 * SAFETY 11
3.7 PERFORMANCE, SECURITY, QUALITY, RELIABILITY, AESTHETICS ETC. 11
3.8 * DOCUMENTATION 12
3.9 RISKS AND VOLATILE AREAS 12

4 * EXPERIMENT DESIGN AND FEASIBILITY STUDY **13**

4.1 * EXPERIMENT DESIGN 13
4.2 * EXPERIMENT RESULTS, DATA ANALYSIS AND FEASIBILITY 16

5 * ARCHITECTURE AND HIGH LEVEL DESIGN **17**

5.1 * SYSTEM ARCHITECTURE AND DESIGN 17
5.2 * HARDWARE ARCHITECTURE 17
5.3 * SOFTWARE ARCHITECTURE (ONLY REQUIRED IF YOUR DESIGN INCLUDES SOFTWARE) 18
5.4 * RATIONALE AND ALTERNATIVES 18

6 DATA STRUCTURES (INCLUDE IF USED) **20**

6.1 INTERNAL SOFTWARE DATA STRUCTURE 20
6.2 GLOBAL DATA STRUCTURE 20
6.3 TEMPORARY DATA STRUCTURE 20
6.4 DATABASE DESCRIPTIONS 20

7 * LOW LEVEL DESIGN **21**

7.1 *Module i (for i = 1 to 7) 21
7.1.1 Processing narrative for Microcontrollers 21
7.1.2 Microcontrollers Interface Description 21
7.1.3 Microcontrollers Processing Details 22
7.2.1 Processing Narrative for Motor Regulation 22
7.2.2 Motor Regulation Interface Description 22
7.2.3 Motor Regulation Processing Details 22
7.3.1 Processing Narrative for User Interface 23
7.3.2 User Interface Description 23
7.3.3 User Interface Processing Details 23
7.4.1 Processing Narrative for Sensor Data 23
7.4.2 Sensor Data Interface Description 23
7.4.3 Sensor Data Processing Details 24
7.5.1 Processing Narrative for 2MP Camera 24
7.5.2 2MP Camera Interface Description 24
7.5.3 2MP Camera Processing Details 24
7.6.1 Processing Narrative LiDAR 25
7.6.2 LiDAR Interface Description 25
7.6.3 LiDAR Processing Details 25
7.7.1 Processing Narrative for Servo Motor 25
7.7.2 Servo Motor Interface Description 25
7.7.3 Servo Motor Processing Details 25

8 * Technical Problem Solving 26

8.1 * Radio Module Problem 26
8.2 * Solving the Radio Module Problem 26
8.3 * 3D Printing Problem 26
8.4 * Solving the 3D Printing Problem 26
8.5 * LiDAR Accuracy Problem 26
8.6 * Solving the LiDAR Accuracy Problem 26
8.7 * pH Sensor Problem 27
8.8 * Solving the pH Sensor Problem 27

9 User Interface Design 28

9.1 Application Control 28
9.2 User Interface Screens 28

10 * Test Plan 29

10.1 * Test Design 29
10.2 * Bug Tracking 30
10.3 * Quality Control 30
10.4 * Identification of critical components 30
10.5 * Items Not Tested by the Experiments 30

11 * Test Report 31

11.1 * Test 1 31
11.2 * Test i (one section for each test i = 2, …., n) 31

12 * Conclusion and Future Work 32

12.1 * Conclusion 32
12.2 Future Work 32
12.3 * Acknowledgement 33

13 * References 34

14 * Appendices 35

1 * Executive Summary

This project is designed to take samples of soil and measure PH data. The concept of the design is a small radio car that can take soil samples and collect data in real time. The inspiration for this project was to build a robot that can take soil samples after learning about the current state of the Riverside Agricultural Park (Ag Park) as it contains high levels of carcinogenic Polychlorinated Biphenyls (PCBs). It is a safe solution to measure the level of PCBs in the area and reduce the amount of labor necessary.

The main goal of the project is to create a rover that can move through a field with either manual or autonomous control using object detection. Once in the field, the rover takes soil and air samples, then reports back to the user at the control station.

The design objectives were focused on creating a rover that could move smoothly over different types of terrain, while also being able to detect objects within five feet to avoid collisions. The rover is able to transmit sensor data back to the control center via radio modules. There are 2 modes: an autonomous mode where the rover could make movement decisions on its own, and a manual mode where the user could navigate via the camera stream.

Key features of the project include a sophisticated 2D LiDAR module that is paired with a 2MP camera for path and object detection. It utilizes motors that provide enough torque to move the rover smoothly through the field, as well as sensors that can accurately capture soil pH and ambient temperature. The radio modules have a range of 100 meters.

The testing results came with the radio not being able to transmit our sensor data. However, the autonomous movement was functional with the lidar movement, and it was able to record sensor data.

One of the major achievements in this project was the successful implementation of object detection using OpenCV and Python. Through this the LiDAR module could measure the distance up to 10 meters in every direction around the rover. These achievements demonstrate the potential of our project to provide scientists with accurate and reliable data on soil health. This is crucial for making informed decisions about crop management and sustainable agriculture practices.

2 * Introduction

2.1 * Design Objectives and System Overview

This project is designed to take samples of soil and measure pH data. The concept of the design is a small radio car that can take soil samples in real time. It has a manual mode and an autonomous mode that utilizes path detection through LiDAR and camera modules. It utilizes radio communication to transmit data and receive instructions as well as a rack and pinion to move the soil sensor down.

This project is meaningful because it reduces the labor necessary for collecting soil data for large areas with a high sampling rate for accuracy. The intended applications are for scientific research and for agricultural purposes.

This relates to electrical engineering through the application of the path detection, the two-way radio communication, the autonomous and manual modes, sensor data usage and the communication between the three areas of development.

The three goals of the project are as follows: to navigate a simple maze through the path detection, to take samples consistently and save data for observation, and too use radio to communicate up to 100 meters.

The 3 areas of functionality consist of the path detection, the sensor data, and the chassis movement. The project should communicate through putty for the sensor data and the computer vision software. The hall motor sensors in the DC motors allowed the rover to measure its speed given the terrain. The system faced problems implementing the radio communication, as the radio modules were inconsistent in their results. The operating environment is the agricultural research site, where the robot will be exposed to varying weather conditions, terrain, and potential hazards like rocks or obstacles in the path. The user environment consists of agricultural researchers or farmers who need to gather accurate soil data to inform their crop management decisions.

List of technical/quantitative design objectives:

- Detects soil pH accuracy within 5%
- Detects humidity and temperature within 10%
- Uses LiDAR range of 10 ft
- Uses Camera resolution is 900x600 to achieve active frame rate of 15 frames per second
- Keeps Total cost less than $500
- Uses Radio transmission range of 100 m
- Transmits 444 kb of data per minute

The individual responsibilities of the group members are as follows. Nicholas Casillasl was responsible for path detection using the LiDAR and the 2 MP camera. Iraj Shrotri was responsible for chassis and movement, including the motor board driver, DC motors with Hall Encoders, and servo motor implementation with the 3D printed rack and pinion. Additionally, Iraj implemented the code to test and read the temperature and humidity sensor, as well as the accelerometer and gyroscope sensor. Francisco Lam was responsible for the rest of the sensors, including the Soil pH and Soil Moisture sensors, as well as the GPS. Radio communication with the nRF24 was a shared responsibility among all team members.

2.2 * Backgrounds and Prior Art

"Smartcore," an autonomous robot developed by a pair of Purdue University College of Engineering graduates, is designed to collect accurate, repeatable soil samples in fields and bring them to the edge of the field for shipment to the lab. SmartCore uses a Bobcat skid steer chassis and navigates fields using boundary algorithms and a variety of obstacle detection sensors. It also is equipped with RTK GPS to ensure that soil samples are taken from the correct spot and can return every season within inches. Smartcore is unique in using a high-speed, self-cleaning hydraulic auger that collects soil to a precise depth.

There is also a paper done by IEEE to report the viability of a similar system that is designed to collect samples and take them back to the lab for analysis. A mechanical soil sampling and storage system based on augers and turntable storage is used in the system. Using a GPS-driven algorithm, the robot navigates autonomously to desired sampling locations. It collects data from the sampling area using sensors connected to an Arduino board. A proof-of-concept demonstrator proved that such a solution can be successfully scaled and deployed, which will aid in more efficient cultivation and sustainable agriculture. This project also uses a rack and pinion as the actuator.

Advantages and disadvantages: The advantages of both Smartcore, and the IEEE systems are that they are heavier duty and can take large amounts of samples and store them which provides much better results and accuracy for a geographical region. Smartcore uses a hydraulic auger that collects soil at a consistent depth which makes for very consistent results. The downside is that they have to take large amounts of soil and store them in their chassis and then take that to the lab where the tests will be done. Our system does this real time in the field which leads to less variability but worse accuracy.

2.3 * Development Environment and Tools

The development environment for this project includes the Arduino IDE for programming the ATmega2560 as well as the ATmega328P chip. PuTTy was used as a digital serial monitor to read all of the sensor results. VSCode was used as an IDE to run Python code for the Computer Vision.

The project requires various libraries which include:

- DHT.h for the DHT11 sensor library
- basicMPU6050.h for the MPU6050 sensor library
- Servo.h for servo control
- YDLidar.h for the LiDAR library
- SoftwareSerial.h, Wire.h for mod bus
- SPI.h, nRF24L01.h, RF24.h for radio communication
- OpenCV for computer vision

2.4 * Related Documents and Supporting Materials

Industry standards involved in this project include the I2C communication protocol for the MPU6050 sensor, the DHT11 sensor library for temperature and humidity measurement, and the Servo.h library for servo control.

Other references and related materials include the datasheets for the sensors used, the Arduino Uno and Arduino Mega boards, and the LiDAR scanner. Additionally, the OpenCV documentation and examples were referenced for computer vision implementation.

The pH sensor came with a reference booklet to explain how it should be used.

2.5 * Definitions and Acronyms

I2C: Inter-Integrated Circuit

LiDAR: Light Detection and Ranging

PCB: Polychlorinated Biphenyls

PID: Proportional-Integral-Derivative

PWM: Pulse Width Modulation

SPI: Serial Peripheral Interface

UART: Universal Asynchronous Receiver-Transmitter

USB: Universal Serial Bus

3 * Design Considerations

3.1 * Realistic Constraints

Power consumption constraints: Since the system requires a 12V power supply to control the motors and use the L298N dual H-bridge motor driver, power consumption is a significant constraint. The system was designed to minimize power consumption to ensure that it can run efficiently without draining the battery too quickly.

Processor speed/memory size constraints: The ATmega2560 chip speed/memory was also an issue considered during the system design, as it only has 256 KB ISP flash memory and 8 KB SRAM. The system was designed to ensure that it can handle the processing requirements of the motors and sensors without experiencing delays or bottlenecks due to inadequate processing power. The speed and memory of the ATmega328P chip was also considered when wanting to perform computer vision on the rover.

Torque constraints: The torque for the motors and wheels were also a significant constraint since the wheels require a high tread for all terrains, which increases general resistance. The system was designed to ensure that it can deliver enough torque to overcome the resistance of the terrain and move effectively.

Weight/size constraints: The weight and size of the system were also considered since there are constraints for the power supply and internals, as well as the LiDAR. The system was designed to minimize weight and size while still providing the necessary functionality.

Space constraints: The space constraints for wiring and sensors are also significant considerations. The system was designed to ensure that all the necessary components can fit within the available space without interfering with each other. The initial design consisted of a 3D printed chassis which was only large enough to accommodate the arduino mega board. Instead a less sophisticated but much more spacious 14" x 8" plastic box was used. The movement components were separated into their own layer in a 3 layer system. Our bottom system consisted of our 12V power supply, and our DC motors with Hall Encoder sensors. The second lair laid a removable plastic sheet to place the Arduino Mega, the H-bridge driver and the sensors with enough space so they are oriented properly. The third layer was the roof of the box which the object detection modules were put on.

Radio frequency constraints: The system was designed to comply with radio frequency constraints, which limit the transmission power and range of the wireless communication between the components. The project used nRF24L01 which operates in the 2.4 GHz band, and is an unlicensed band in the United States. The FCC has established rules for unlicensed devices operating in the 2.4 GHz band to prevent interference with other wireless devices such as Wi-Fi routers, Bluetooth devices, and microwave ovens. The maximum allowed transmission power for devices operating in the 2.4 GHz band in the US is 1 watt (1000 milliwatts), but the nRF24L01 is designed to operate at a lower power level, typically around 7-10 milliwatts.

Budget/time constraints: The system was designed to fit within the available budget which is less than $500 and 2 quarters. The first would consist of research and ordering, with the second quarter reserved for assembling and testing. This required making trade-offs between functionality, performance, and cost to ensure that the system can be developed and deployed within the specified timeline and budget.

3.2 System Environment and External Interfaces

N/A

3.3 * Industry Standards

I2C: This standard was used for communication with the accelerometer and gyroscope sensors in this design. The I2C protocol defines the electrical and communication specifications, including the clock rate and the data transfer format. In this project it was used for sending sensor data from the Accelerometer and Gyroscope module

SPI: This standard was used for communication between the radio rnf24 module and the Arduinos Mega in this design. The SPI protocol defines the electrical and communication specifications, including the clock rate and the data transfer format. In this project, it enabled transmission and receiving of data through the radio module.

UART (RX-TX): This standard was used for communication between the LiDAR and the Arduinos in this design. The UART protocol defines the electrical and communication specifications, including the baud rate and the data transfer format.

Data rate: The design complies with the data rate standard for the Arducam communication, which is specified by the camera's datasheet.

USB: A USB connection was required to read from the Arduino Serial monitor.

Python: Python was used in VSCode to perform object detection.

3.4 * Knowledge and Skills

Nicholas Casilo:

- Engineering Circuit Analysis I and II (EE1A and 1B): Ohm's law, Kirchhoff's laws, nodal and loop analysis, analysis of linear circuits, network theorems, transients in RLC circuits.
- Sensing and Actuation for Embedded Systems (EE 128): Embedded system design, sensor data acquisition, signal processing, control, actuation, sensor and motor interface principles.
- Introduction to Machine Learning and Data Mining (EE 142): Data mining, machine learning, supervised learning, classification, regression, clustering.
- Computer Vision (EE 146): Imaging formation, early vision processing, object representation, recognition techniques.
- Additional skills: Application of SPICE to circuit analysis, application of machine learning algorithms.

Francisco Lam:

- Engineering Circuit Analysis I and II (EE1A and 1B): Ohm's law, Kirchhoff's laws, nodal and loop analysis, analysis of linear circuits, network theorems, transients in RLC circuits.
- Sensing and Actuation for Embedded Systems (EE 128): Embedded system design, sensor data acquisition, signal processing, control, actuation, sensor and motor interface principles.
- Introduction to Embedded Systems (CS 120B): Hardware and software design of digital computing systems, embedded processor programming, custom processor design, interfacing.
- Additional skills: Use of sensors and Modbus for data reading, use of nRF24 and ESP32 for transmitting sensor data to a website.

Iraj Shrotri:

- Engineering Circuit Analysis I and II (EE1A and 1B): Ohm's law, Kirchhoff's laws, nodal and loop analysis, analysis of linear circuits, network theorems, transients in RLC circuits.
- Foundations of Robotics (EE 144, 001): Analysis, control, and programming of robots, robot motion control, sensor data interpretation.
- Sensing and Actuation for Embedded Systems (EE 128): Embedded system design, sensor data acquisition, signal processing, control, actuation, sensor and motor interface principles.
- Introduction to Machine Learning and Data Mining (EE 142): Data mining, machine learning, supervised learning, classification, regression, clustering.
- Logic Design (EE 120A): Boolean algebra, combinational and sequential logic design, state-machine design.
- Additional skills: Use of PID control, state-machine design, and embedded systems.

3.5 * Budget and Cost Analysis

A cost analysis is an essential part of any project, providing insight into the budget and expenses associated with the project. In this particular project, the cost analysis revealed that the most expensive component was the LiDAR, while the other components were relatively affordable. However, the project also experienced some additional costs due to errors, such as having to reorder radio modules due to low range and having to reorder DC motors twice due to missing hall encoders and an incorrect quantity. This cost analysis highlights the importance of careful planning, consideration of component costs, and accounting for potential errors and reordering in any project.

Name	Cost ($)	#	Total
DC Motor w/ Magnetic Hal Encoders	15	4	60
DC Motor Driver L298N	11	1	11
Arduino Mega	25	2	25
Arduino Mega Shield 5 V + GND per pin	10	1	10
nrf24 radio module	7.5	6	45
PH Sensor	25	1	25
Moisture Sensor	10	1	10
Servo Motor 5 1 5	5	1	5
LiDAR 80 1 80	80	1	80
2 MP Camera 25 1 25	25	1	25
Gyroscope/ Accelerometer 11 1 11	11	1	11
GPS 13 1 13	13	1	13

Temp/Humidity Sensor 7 1 7	7	1	7
Joysticks 1 12 12	1	12	12
Buttons/Switches 9 6 9	9	6	9
Battery Pack 8 1 8	8	1	8
Raspberry Pi 50 1 50	50	1	50
Plastic Case 5 1 5	5	1	5

Final : $449 $/person : 149.6666667

3.6 * Safety

The two primary specific safety considerations that were addressed in this project were:

1. The 12 V power supply: The use of a 12 V power supply can be dangerous if not handled properly, as it can cause electric shock or start a fire. To address this safety concern, this design properly isolated and shielded the power supply, and to the rover and ensured that all connections were properly insulated. In addition users were properly warned about the potential dangers of handling the power supply by way of warning labels.
2. Radio frequency interference: Radio frequency interference can be a safety concern if it affects emergency communication systems or other critical systems. To address this concern, this design is coded to use a radio frequency that is licensed for hobbyists to use.

In addition to these specific safety considerations, it was important to consider other potential hazards that may be associated with the design, such as sharp edges, moving parts, or unstable balance. The safety objectives for the rover's operation were as follows:

The rover should not cause injury to users or bystanders during normal operation.

- The rover should not cause damage to property during normal operation.
- The rover should have proper safeguards in place to prevent electrical shock or fire hazards.
- The rover should not interfere with emergency communication systems or other critical systems.

3.7 Performance, Security, Quality, Reliability, Aesthetics etc.

In order to meet the requirements of performance, security, quality, reliability, and aesthetics, several key considerations and processes were implemented. Performance objectives were clearly defined, and a thorough system analysis was conducted to optimize the system's capabilities. Security measures such as encryption and access controls were implemented to protect against unauthorized access. Quality control processes were established to ensure compliance with industry standards, and rigorous testing and inspections were conducted at various stages. Reliability was addressed through redundancy and fault-tolerant design strategies, and ongoing monitoring of reliability metrics. Aesthetics were considered in terms of user experience, visual design, and integration of components for a clean and appealing look. By following these processes and incorporating these considerations, the project was able to successfully meet the requirements of performance, security, quality, reliability, and aesthetics.

3.8 * Documentation

All of the documents were kept in a Google Drive shared among the members. These included a parts sheet that included the prices, a Gantt chart to keep us in time for the deadline, as well as design notes in a notebook which were moved digitally after finalizing the project. Gantt chart- Soil sample bot

3.9 Risks and Volatile Areas

LiDAR Detection: LiDAR detection can be challenging to implement, as it involves accurately measuring distance using laser technology. Some potential risks in this area include inaccurate readings, interference from other objects or light sources, and difficulty in calibrating the system. To mitigate these risks, it may be helpful to perform extensive testing and calibration before deployment and to have a backup system in place in case of failure.

CV/Object Detection: Computer vision and object detection can be prone to false positives, which can result in incorrect data and unreliable performance. To mitigate this risk, it may be helpful to use a combination of techniques such as feature extraction, machine learning, and thresholding to increase the accuracy of the system.

Moisture Sensor: Moisture sensors can be vulnerable to damage or malfunction, which can affect the accuracy of the readings. To mitigate this risk, it may be helpful to use a high-quality capacitance sensor instead of a resistance sensor, which can be more reliable and less prone to failure.

In addition to these specific areas, it's important to have a general risk management plan in place that includes strategies for identifying and addressing potential risks throughout the system design process. This can include techniques such as testing, validation, and verification, as well as contingency planning and ongoing monitoring and evaluation.

4 * Experiment Design and Feasibility Study

4.1 * Experiment Design

This document presents a series of experiments conducted to evaluate the performance of the components of the Earth Rover. Every experiment is designed to assess the functionality and ensure the reliability and accuracy of the critical system elements. The experiments include testing the motor driver with 12 V power supply and hall motor encoders, verifying the accuracy of the sensors, refining the object detection system, testing the LiDAR for obstacle avoidance, assessing the usability of the rack and pinion system, and testing the NRF24 radio module. Through these experiments we aim to gain insight into the performance of the rover and seek to refine its individual systems.

<u>Experiment 1: Testing Motor Driver with 12 V supply and the hall motor encoders</u>

Objective: To verify the reliability of rpm readouts of the motor driver with 12V supply and hall motor encoders.

Setup:

- Motor driver
- 12 V power supply
- Hall motor encoders
- RPM meter
- Multimeter
- Test bench

Procedure:

- Connect the motor driver to the 12V power supply.
- Connect the hall motor encoders to the motor driver.
- Connect the RPM meter to the motor driver.
- Install the motor on the test bench.
- Run the motor at different speeds and record the rpm readouts.
- Verify the rpm readouts using the multimeter.
- Repeat the experiment with different speeds and loads.

Expected Results: The rpm readouts should be accurate and reliable.

Responsibilities: Iraj Shrotri was responsible for the design, set up, and data analysis of this experiment

<u>Experiment 2: Testing the Sensors for Accuracy</u>

Objective: To verify the accuracy of the data collected by the pH sensor, soil capacitor, temperature/humidity sensor, and accelerometer/gyroscope.

Setup:

- pH sensor
- Soil capacitor
- Temperature/humidity sensor
- Accelerometer/gyroscope
- Data acquisition system
- Test bench

Procedure:

1. Connect the sensors to the data acquisition system.
2. Install the sensors on the test bench.
3. Run the sensors and collect the data.
4. Compare the data collected by the sensors with the known values.
5. Repeat the experiment with different known values.

Expected Results: The data collected by the sensors should be accurate and reliable.

Responsibilities: Iraj Shrotri and Francisco Lam were responsible for the design, set up, and data analysis of this experiment

Experiment 3: Testing Object Detection

Objective: To test the object detection system's ability to detect and classify

objects. Setup:

- Object detection system
- Test objects (different sizes and shapes)
- Test environment (indoor/outdoor)

Procedure:

1. Set up the object detection system.
2. Place the test objects in the test environment.
3. Run the object detection system and record the results.
4. Analyze the results to verify the system's accuracy in detecting and classifying the test objects. 5. Repeat the experiment with different test objects and environments.

Expected Results: The object detection system should accurately detect and classify the test objects in different environments.

Responsibilities: Nicholas Caslilas was responsible for the design, set up, and data analysis of this experiment.

Experiment 4: Testing the LiDAR for Obstacle Avoidance

Objective: To test the LiDAR's ability to detect obstacles and avoid them.

Setup:

- LiDAR sensor
- Test environment (indoor/outdoor)
- Test obstacles (different sizes and shapes)

Procedure:

1. Install the LiDAR sensor on the rover.
2. Place the test obstacles in the test environment.
3. Run the rover and record the LiDAR's results.
4. Analyze the results to verify the LiDAR's accuracy in detecting obstacles and avoiding them. 5. Repeat the experiment with different test obstacles and environments.

Expected Results: The LiDAR should accurately detect obstacles and avoid them. Responsibilities: Nicholas Caslilas was responsible for the design, set up, and data analysis of this experiment

Experiment 5: Testing the Rack and Pinion / 3D Prints for Reliability

Objective: To test the reliability of the rack and pinion system and the 3D printed components. Setup:

- Rack and pinion system
- 3D printed components
- Test bench

Procedure:

1. Install the rack and pinion system and 3D printed components on the test bench. 2. Run the system at different speeds and loads.
3. Monitor the system for any signs of wear or failure.
4. Record any issues that arise during the test.
5. Repeat the experiment with different speeds and loads.

Expected Results: The rack and pinion system and 3D printed components should be reliable and free from wear or failure.

Responsibilities: Iraj Shrotri and Francisco Lam were responsible for the design, set up, and data analysis of this experiment

Experiment 6: Testing NRF24 Wireless Communication

Objective: To test the NRF24 wireless communication module's ability to transmit and receive data reliably.

Setup:

- Two NRF24 wireless communication modules
- Microcontroller boards (Arduino, Raspberry Pi, etc.)
- Power supply
- Test environment

Procedure:

1. Connect the NRF24 modules to the microcontroller boards.
2. Program one microcontroller to send data and the other to receive data.
3. Power on the microcontrollers and the NRF24 modules.
4. Transmit data from the sending microcontroller and record any data loss or errors.
5. Repeat the transmission with different data rates and distances between the modules. 6. Analyze the received data to verify its accuracy.
7. Repeat the experiment with different environments (indoor/outdoor) and obstacles.

Expected Results: The NRF24 wireless communication module should reliably transmit and receive data without significant data loss or errors.

Responsibilities: Iraj Shrotri, Francisco Lam and Nicholas Caslilas were all responsible for the design, set up, and data analysis of this experiment

4.2 * Experiment Results, Data Analysis and Feasibility

Based on the results of the first five experiments, the technical design objectives were achieved with the chosen components and sensors. The motor driver, sensors, object detection system, LiDAR, and rack and pinion system were all found to be reliable and accurate. The NRF24 wireless communication module, however, proved to be problematic and was unable to reliably transmit and receive data.

During Experiment 1, the left side of the car initially did not work due to improper wiring, but this issue was resolved. Experiment 2 showed that the sensors worked without issue, but it took some trial and error to read all the data onto the serial monitor by adjusting the baud rate. Experiment 3 was aided by Niclos' expertise and resulted in a faster processing time and improved detection algorithm. Experiment 4 required significant effort to develop a simple algorithm for obstacle avoidance with the LiDAR. Experiment 5 required reprinting 3D prints due to initial printing errors.

While the first five experiments were successful, the team faced challenges with the NRF24 wireless communication module in Experiment 6, which proved to be consistently unreliable. Despite this setback, the experiments helped the team select the best solutions for the design project and identify areas for improvement. Continued testing and refinement will ensure that the system meets the desired technical objectives and is feasible for implementation.

5 * Architecture and High Level Design

5.1 * System Architecture and Design

The system architecture of the Semi-Autonomous Earth Rover is designed to provide a high-level overview of its structural and functional decomposition. It defines the major components of the system and their relationships, serving as a basis for more detailed design work. The decomposition is driven by the specific requirements and objectives of the rover.

At a high level, the system can be divided into the following major components. The path detection system, the chassis, movement, and radio communication, and the sensor system. Each component represents a grouping of related functionalities and has assigned responsibilities. Nicholas Casillas is responsible for the path detection subsystem, including LiDAR and the 2 MP camera. Iraj Shrotri is responsible for the chassis, movement, and radio communication subsystem, which involves the motor board, DC motor with Hall encoders, 3D printing, and the nRF24 wireless communication module. Francisco Lam is responsible for the servo motor and the sensor subsystem, which includes temperature and humidity sensors, accelerometer and gyroscope, soil pH sensor, water pH sensor, and GPS.

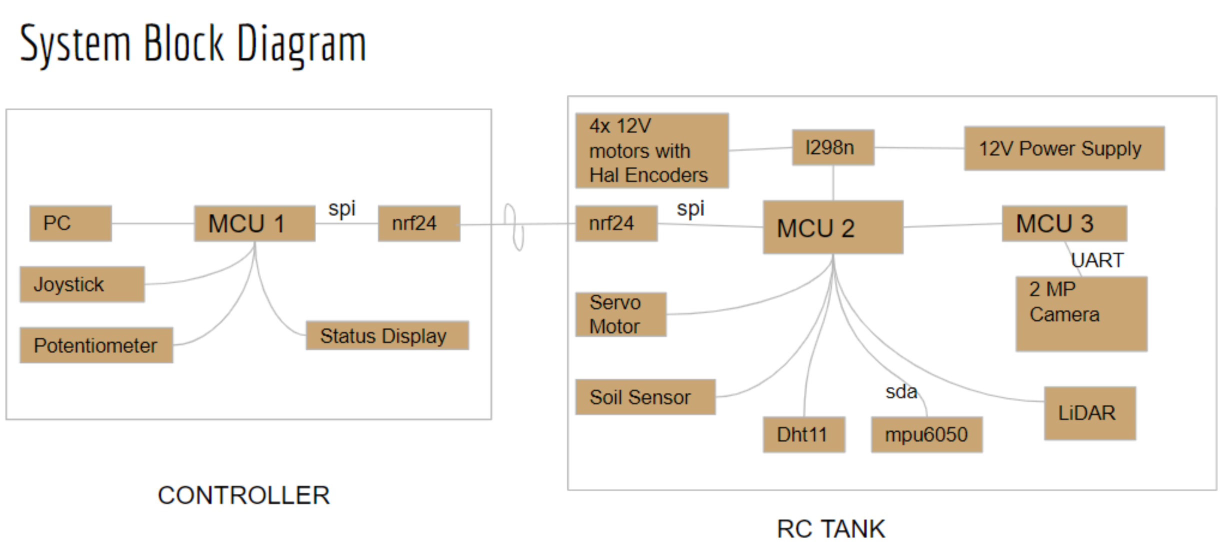

Figure 5.1

5.2 * Hardware Architecture

The hardware architecture of the Semi-Autonomous Earth Rover consists of various components responsible for path detection, chassis movement, radio communication, and sensor functionalities. The following is a breakdown of the hardware modules and their respective responsibilities:

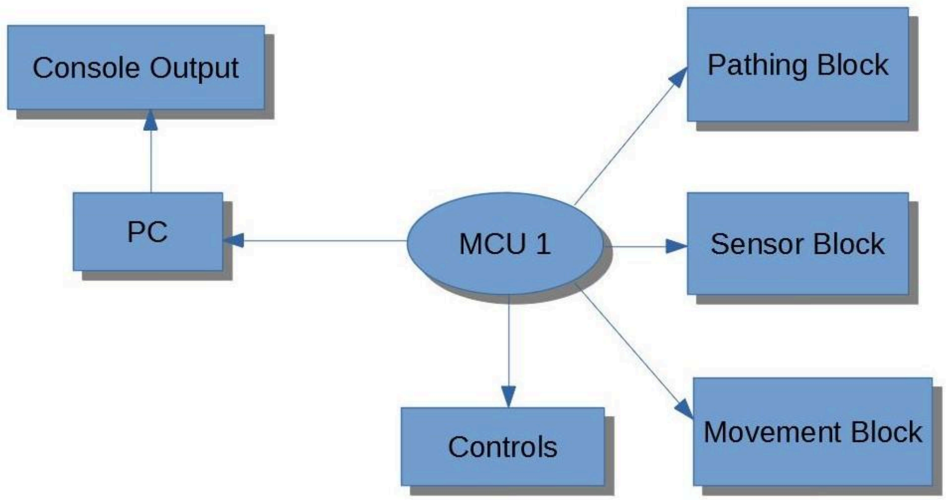

Figure 5.2

Path Detection:

- LiDAR: This module, managed by Nicholas Casillas, is responsible for measuring distances using laser technology, enabling the rover to detect objects and obstacles in its surroundings.
- 2 MP Camera: Also managed by Nicholas Casillas, this camera captures visual data to support computer vision algorithms for object recognition and navigation.

Chassis, Movement, and Radio Communication:

- L298N Motor Board: This module, handled by Iraj Shrotri, provides motor control capabilities for the rover's movement. It enables precise control over the 12V DC motors and facilitates the integration of various motion functionalities.
- 12V DC Motor with Hall Encoders: Managed by Iraj Shrotri, these motors are responsible for propelling the rover and are equipped with Hall encoders for accurate position sensing.
- 3D Printing/Chassis Design: Iraj Shrotri oversees the design and manufacturing of the rover's chassis using 3D printing technology. The chassis provides the structural support and mounting points for all the hardware components.
- nRF24: This wireless communication module, jointly managed by Nicholas Casillas, Iraj Shrotri, and Francisco Lam, enables reliable communication between the rover and external devices. It facilitates data exchange and control commands.

Sensors:

- Temperature and Humidity: Iraj Shrotri is responsible for integrating temperature and humidity sensors into the rover. These sensors measure the environmental conditions and provide valuable data for monitoring and analysis.

- Accelerometer and Gyroscope: Also managed by Iraj Shrotri, these sensors detect changes in acceleration and rotational motion, providing information about the rover's orientation and movement.
- Soil pH: Francisco Lam oversees the integration of the soil pH sensor, which measures the acidity or alkalinity of the soil. This data helps in assessing soil conditions for agricultural or scientific purposes.
- Water pH: Francisco Lam is also responsible for incorporating the water pH sensor, which measures the pH level of water bodies. This information is useful for water quality assessment and environmental monitoring.
- GPS: Managed by Francisco Lam, the GPS module provides precise location and positioning data for the rover. It enables accurate mapping and navigation capabilities.
- Each hardware module is assigned to the respective team member to ensure clear responsibilities and efficient development. By dividing the hardware architecture into these components, the system achieves a modular and scalable design that enables independent development and integration of functionalities.

5.3 * Software Architecture (only required if your design includes software)

Computer Vision:

The computer vision component plays a crucial role in the rover's functionality. It is responsible for processing image data captured by the 2 MP camera and performing various tasks such as object recognition, obstacle detection, and environmental analysis. The Raspberry Pi, which interfaces with the camera module, serves as the main processing unit for this component. It leverages powerful algorithms and libraries to analyze visual data and extract meaningful information. The computer vision algorithms are implemented using programming languages such as Python, which has extensive support for computer vision tasks.

Path Planning and Navigation:

The path planning and navigation component is responsible for determining the optimal path for the rover's movement. It takes inputs from various sensors, including the LiDAR and GPS, and employs predefined algorithms to generate navigation commands. The path planning and navigation algorithms run on the Arduino Mega microcontroller, which provides real-time control and precise movement instructions to the motors. This component ensures that the rover can navigate autonomously, avoiding obstacles and following predefined paths or user-defined commands.

Sensor Data Acquisition:

The sensor data acquisition component collects data from the onboard sensors, including temperature and humidity sensors, accelerometers, gyroscopes, soil pH sensor, water pH sensor, and GPS. The Arduino Mega microcontroller is responsible for acquiring data from these sensors and forwarding it to other software components for further processing and analysis. This component enables the rover to gather important environmental data, which can be used for monitoring, analysis, and decision-making.

User Interface:

The user interface component provides a means for users to interact with the rover, initiate commands, and monitor sensor readings. It offers a user-friendly interface that can be implemented using various methods such as graphical user interfaces (GUIs) or command-line interfaces (CLIs). Users can switch between autonomous and manual modes of operation, send specific commands to the rover, and receive feedback on the rover's status. The user interface component can be developed using programming languages and frameworks suitable for creating interactive interfaces, such as Python with GUI libraries or web-based interfaces.

Communication:

The communication component handles the exchange of data and commands between the rover and external devices or systems. It manages wireless communication between the rover and the remote control interface, enabling bidirectional communication for real-time control and data transmission. The nRF24 module, which supports wireless communication at a frequency of 2.4 GHz, facilitates reliable communication between the rover and the control interface. This component ensures seamless interaction and control of the rover, enabling users to monitor its status, send commands, and receive real-time sensor data.

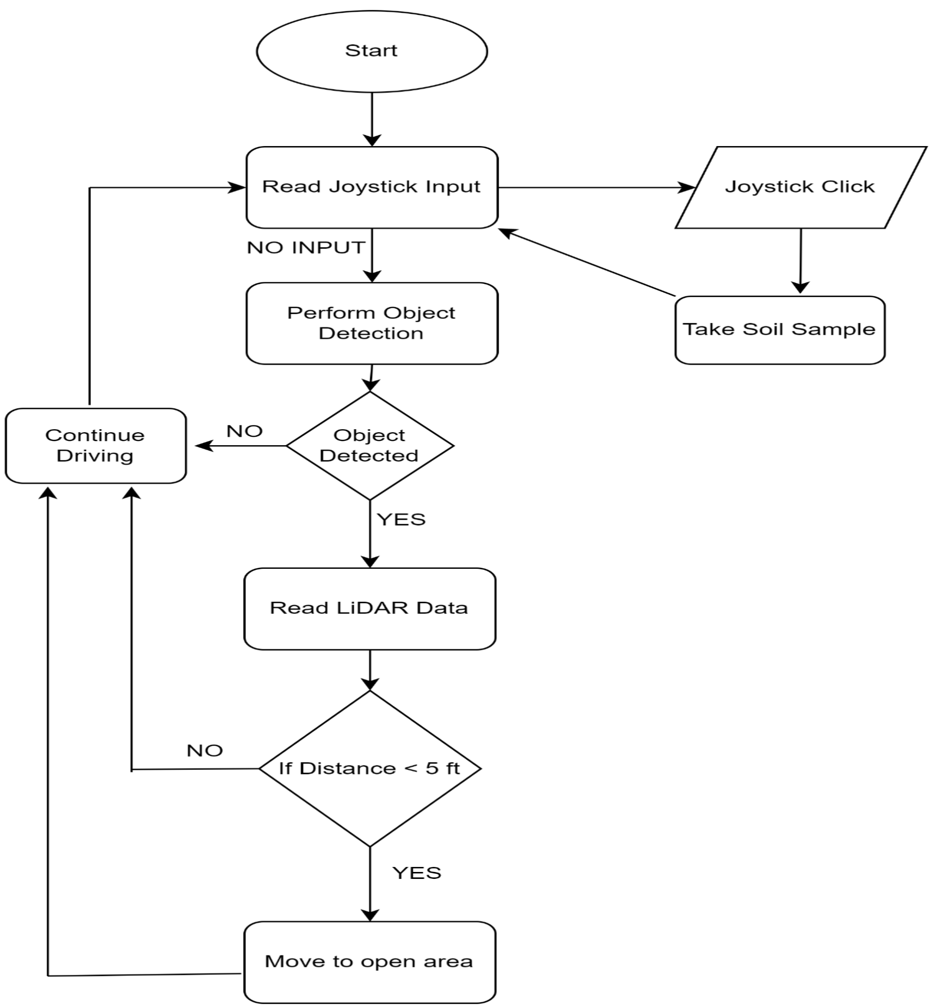

Figure 5.3

5.4 * Rationale and Alternatives

The initial approach was to have 3 separate microcontrollers to handle each part of our system objectives being, path detection, data acquisition, and movement. It would utilize a Raspberry Pi to control the camera and process computer vision. It would calculate the directions, and then communicate them to the Ardunio Mega. Then the arduino mega on the main chassis for the primary tasks of collecting the sensor data as well as giving the directions for movement by the motors. The radio transmitter receives the sensor data as well as the raw data of the LiDAR and camera for manual movement. To switch between autonomous and manual modes, there would be a button. This module operates at a frequency of 2.4 GHz and has a range of up to 100 meters in open space. This module was chosen because it is low-cost, easy to use, and has good range for this project's requirements.

The transmitter also includes a set of buttons or joysticks that are used to send commands to the rover. These inputs are connected to the radio module and are used to send digital signals to the rover to control its movements. This system utilizes simple push buttons to control the movement of the rover, but more advanced controllers could be used if needed. However the radio modules had many technical issues, and stopped working, so the idea for a transmitter was scraped

After considering results from the initial approach, it was decided to simplify the system by using only two microcontrollers instead of three. This allowed for easier communication between the components and reduced the risk of errors or complications. It was ultimately decided to use the Raspberry Pi and Arduino Mega combination for the following reasons: The Raspberry Pi provides the necessary computing power and memory for running computer vision algorithms and processing image data from the camera. It also allows for easy integration with other software components and programming languages.

The Arduino Mega provides the necessary I/O capabilities for collecting sensor data and controlling the motors. Its relatively simple architecture and real-time operating system make it well-suited for these tasks. Other architectures or approaches that were considered included using a single microcontroller to handle all tasks, using a more powerful microcontroller to handle both computing and control tasks, and using wireless communication between components. However, we ultimately decided that the Raspberry Pi and Arduino Mega combination was the best fit for our needs based on factors such as performance, flexibility, and ease of use.

However, the Raspberry Pi that was ordered was lost in transit and never arrived. In the final build an arduino uno was used to separately handle the computer vision aspect and the LiDAR control was moved to the mega as the primary mode of object detection to give instructions to the rover.

6 Data Structures (include if used)

6.1 Internal software data structure

We attempted to use an array to store data for the 8 cardinal directions from the front of the LiDAR at their respective degrees and store that read that data to determine the safe direction

6.2 Global data structure

N/A

6.3 Temporary data structure

N/A

6.4 Database descriptions

N/A

7 * Low Level Design

In this section, the low-level design descriptions that form the building blocks of the Semi-Autonomous Earth Rover are investigated. Each component is dissected into its fundamental units or modules, encompassing both hardware and software elements. These modules are the backbone of the rover's functionality and provide the necessary control and coordination for its various systems. By examining the inner workings of each module, a deeper understanding of how the rover operates can be understood.

Circuit Schematic[Nicholas]

- The left schematic contains all the sensors, motors, and LiDAR connected to the Arduino Mega
- The right schematic shows the ArduCAM OV2640 connected to the Arduino Uno

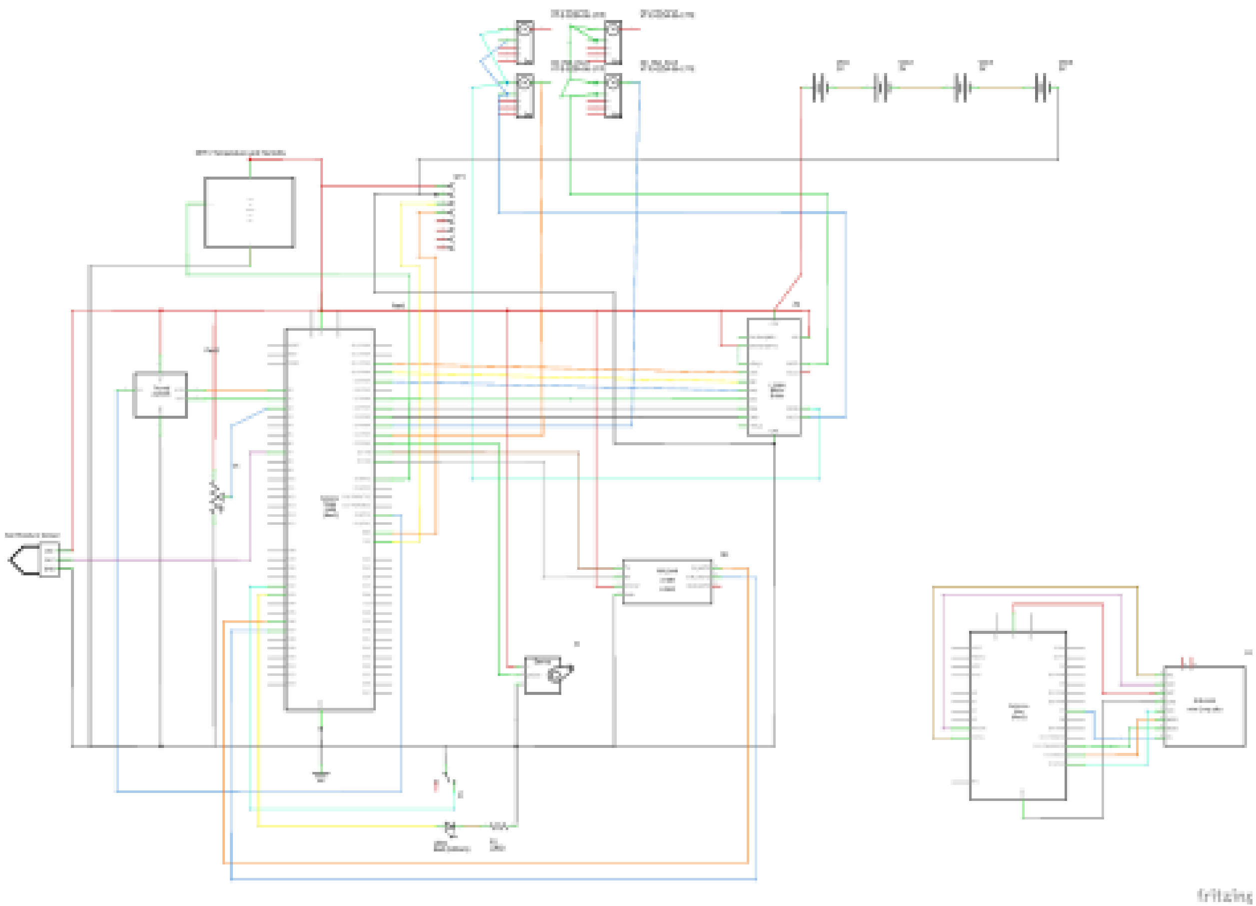

Figure 7.1

Flowchart Posted in 5.3

7.1.1 Processing narrative for Microcontrollers [Nicholas Casillas, Iraj Shrotri, Francisco Lam]

The Microcontrollers module is responsible for controlling and coordinating the overall system functions. In this project, it comprises two microcontrollers, the Arduino Uno and Arduino Mega. The

Arduino Uno is connected to the 2MP camera module, while the Arduino Mega is connected to all the other modules in the system, including the Sensor Data, Watering System, Lighting System, and Environmental Control System.

7.1.2 Microcontrollers Interface description [Nicholas Casillas, Iraj Shrotri, Francisco Lam]

The Arduino Uno interfaces with the 2MP camera module through the I2C interface. On the other hand, the Arduino Mega interfaces with all the other modules through digital and analog pins, I2C, and UART interfaces. The module also interfaces with the main control system, which includes the user interface and decision-making algorithms.

7.1.3 Microcontrollers processing details [Iraj Shrotri]

The Arduino Uno is responsible for capturing images from the 2MP camera module, processing them, and transmitting them to the main control system for display and decision-making. The Arduino Mega, on the other hand, coordinates the overall system functions, including reading sensor data from the Sensor Data module, controlling the wheels, and using the LiDAR.

The microcontrollers are programmed using the Arduino Integrated Development Environment (IDE), which provides a simple and easy-to-use interface for developing and uploading code. The design constraints for the Microcontrollers module include the need for low power consumption, real-time data processing, and compatibility with the main control system. The performance of the module is limited by the processing power and memory of the microcontrollers and the speed and reliability of the interfaces used to communicate with other modules.

7.2.1 Processing narrative for Motor Regulation [Iraj Shrotri]

The motor regulation module is responsible for controlling the direction and speed of the DC motors using feedback from the hal magnetic encoders and a potentiometer. The DC motor driver L298N is used to provide the necessary power to the motors based on the control signals from the module.

The motor regulation module receives input from the path detection and data acquisition modules to determine the direction and speed of the motors based on the current location and obstacle detection. The module then calculates the necessary motor control signals and sends them to the L298N motor driver. The hal magnetic encoders provide feedback to the module to ensure that the motors are rotating at the desired speed and direction. The potentiometer provides manual control over the motor speed.

7.2.2 Motor Regulation interface description [Iraj Shrotri]

The motor regulation module interfaces with the path detection and data acquisition modules to receive input on the current location and obstacles. It interfaces with the L298N motor driver to send motor control signals for speed and direction. The module also interfaces with the hal magnetic encoders and the potentiometer to receive feedback on motor speed and to provide manual control over the motors.

7.2.3 Motor Regulation processing details [Iraj Shrotri]

The motor regulation module is implemented using an Arduino Mega microcontroller board. The module uses pulse width modulation (PWM) to control the speed of the DC motors. The hal magnetic encoders provide a quadrature signal that is used to calculate the speed and direction of the motors. The potentiometer provides an analog voltage signal that is used to adjust the speed of the motors in manual mode.

The module includes a PID (proportional-integral-derivative) controller algorithm to adjust the motor control signals based on the feedback from the hal magnetic encoders. The algorithm calculates the error between the desired motor speed and the actual motor speed, and adjusts the motor control signals based on the error.

The design constraints for the motor regulation module include the maximum voltage and current ratings for the DC motors and the L298N motor driver. The module is also limited by the maximum processing speed and memory capacity of the Arduino Mega microcontroller. Performance issues may arise if the module is unable to process the sensor data and calculate the motor control signals fast enough to maintain accurate control over the motors.

7.3.1 Processing narrative for User Interface [Iraj Shrotri, Nicholas Ca]

The user interface module consists of joysticks, buttons/switches, and LED indicators. The joysticks are used for manual control of the rover's movement, while the buttons/switches are used for mode selection (autonomous/manual) and other functions such as emergency stop. The LED indicators are used to provide feedback on the rover's status and mode.

7.3.2 User Interface description[Iraj Shrotri, Nicholas Casillas]

The joysticks are connected to an Arduino Mega microcontroller and use analog inputs to detect the position of the joystick. The buttons/switches are also connected to the Arduino Mega and use digital inputs to detect their state. The LED indicators are connected to the Arduino Mega and are controlled using digital outputs.

The user interface module communicates with the motor control module and the sensor data acquisition module to provide feedback on the rover's status and to send commands for manual control.

7.3.3 User Interface processing details [Francisco Lam]

The Arduino Mega microcontroller is programmed to read the analog inputs from the joysticks and digital inputs from the buttons/switches. The position of the joysticks is mapped to a range of values that correspond to the desired direction of the rover's movement.

The Arduino Mega also monitors the state of the buttons/switches and sends commands to the motor control module and the sensor data acquisition module based on the button/switch state. For example, pressing the emergency stop button sends a command to stop all motor movement and halt the rover.

The LED indicators are controlled using digital outputs from the Arduino Mega. The LED indicators are used to provide feedback on the rover's status, such as indicating when the rover is in autonomous or manual mode, or when an error has occurred.

Design constraints for the user interface module include the number and placement of buttons/switches and joysticks, as well as the number of LED indicators. The performance of the user interface module is limited by the processing speed of the Arduino Mega microcontroller and the responsiveness of the analog and digital inputs. Limitations may also include the physical size and weight of the user interface components, as well as power consumption requirements.

7.4.1 Processing narrative for Sensor Data [Iraj Shrotri, Francisco Lam]

The Sensor Data module is responsible for collecting data from the DHT11 temperature and humidity sensor, MPU6050 accelerometer and gyroscope, and Soil Moisture Capacitor. The data collected from these sensors is used to monitor the environmental conditions of the plant and make necessary adjustments to ensure optimal growth.

7.4.2 Sensor Data Interface description [Francisco Lam]

The Sensor Data module interfaces with the microcontroller through digital pins for the DHT11 and Soil Moisture Capacitor sensors and through the I2C interface for the MPU6050 sensor. The module also interfaces with the main control system to provide real-time sensor data for monitoring and decision-making.

7.4.3 Sensor Data processing details [Francisco Lam]

The DHT11 sensor is responsible for measuring the temperature and humidity of the plant environment. The data is collected through a digital pin and processed using the DHT library to provide accurate readings.

The MPU6050 sensor is responsible for measuring the acceleration and angular velocity of the plant environment. The data is collected through the I2C interface and processed using the MPU6050 library to provide accurate readings.

The Soil Moisture Capacitor is responsible for measuring the moisture level of the soil in the plant container. The data is collected through a digital pin and processed using a voltage divider circuit to provide accurate readings.

The data collected from the sensors is continuously monitored and processed to ensure the plant is in optimal conditions. The data is also used to make necessary adjustments to the watering schedule, lighting, and environmental control system. The design constraints for the Sensor Data module include the need for accurate and reliable data collection, low power consumption, and compatibility with the main control system. The module's performance is limited by the quality and accuracy of the sensors used and the processing power of the microcontroller.

7.5.1 Processing narrative for 2MP Camera [Nicholas Casillas]

The Arducam 2MP camera module is connected to an Arduino Uno microcontroller for image acquisition and processing. The camera module supports 1600 x 1200 pixel resolution and can capture images at a maximum frame rate of 15 fps. The camera module is controlled through the I2C communication protocol and uses the Arducam library for image capture and processing.

The camera module is used to provide visual feedback for the autonomous navigation of the rover. The camera captures images of the surrounding environment, which are then processed using computer vision algorithms to detect obstacles and determine the direction of movement.

7.5.2 2MP Camera Interface description [Nicholas Casillas]

The Arducam 2MP camera module is connected to the Arduino Uno through a serial interface. The camera module receives power and ground connections from the Arduino Uno and communicates using the I2C communication protocol. The camera module provides image data to the Arduino Uno, which is then processed using computer vision algorithms.

7.5.3 2MP Camera processing details [Nicholas Casillas]

The Arducam 2MP camera module uses a CMOS image sensor and supports various image formats including JPEG and BMP. The camera module is controlled through the Arducam library, which provides functions for image acquisition, configuration, and processing.

The camera module is designed to operate in a range of lighting conditions and includes features such as automatic white balance and exposure control. The camera module also includes an onboard EEPROM for storing configuration settings.

The camera module has certain limitations such as limited resolution and frame rate, which can affect the accuracy and speed of image processing algorithms. The camera module also requires a stable power supply and may be affected by electromagnetic interference from other components in the system.

7.6.1 Processing narrative for LiDAR [Nicholas Casillas]

The LiDAR system in this project involves the use of one YDLiDAR unit to detect the surrounding environment. The LiDAR sensor uses laser beams to scan the environment and return distance measurements, which are then used for mapping and obstacle detection.

7.6.2 LiDAR Interface description [Nicholas Casillas]

The LiDAR module is connected to the main controller, which is an Arduino Mega, via UART. The LiDAR data is processed by the main controller, which then sends instructions to the motor controller to move the rover accordingly.

7.6.3 LiDAR processing details [Nicholas Casillas]

The LiDAR sensor module has a 360-degree range and a 10-meter range radius. The data collected from each sensor is processed by the Arduino Mega to create a 3D map of the surrounding environment. The map is used for obstacle detection and avoidance. The algorithms used for processing the LiDAR data involve converting the raw distance measurements into point cloud data, which is then used for mapping and object detection. The LiDAR system has design constraints related to power consumption, sensor resolution, and sensor range. The performance of the system is affected by environmental factors such as lighting and weather conditions.

7.7.1 Processing narrative for Servo Motor [Iraj Shrotri]

The Servo motor is designed to be attached to a rack and pinion system to operate the soil moisture sensor when a joystick button is pressed. The processing narrative for this module describes the steps involved in controlling the servo motor and how it interacts with the other components in the system.

7.7.2 Servo Motor Interface description [Iraj Shrotri, Francisco Lam]

The Servo Motor module interfaces with other modules in the system through the Arduino Uno. It receives signals from the Joystick/Button module and sends signals to the Soil Moisture Sensor module through the rack and pinion system. The Servo Motor module is designed to work with standard hobby servos, and its inputs and outputs are compatible with the Arduino Uno microcontroller.

7.7.3 Servo Motor processing details [Iraj Shrotri, Francisco Lam]

The Servo Motor module is controlled by a software algorithm that receives signals from the Joystick/Button module and translates them into movements of the servo motor. The algorithm takes into account the position of the rack and pinion system and the current moisture level of the soil as reported by the Soil Moisture Sensor. The Servo Motor module has built-in limitations on the range of movement and speed of the servo motor to prevent damage to the system. The performance of the Servo Motor module is dependent on the accuracy and reliability of the signals received from the other modules in the system.

8 * Technical Problem Solving

Through the development of the Earth Rover, we encountered several technical difficulties. This section documents these challenges and outlines the steps taken to address them effectively.

8.1 * Radio Module Problem [Francisco Lam]

The objective of the project was to have the rover operate fully remote, communicating between user and rover using a nRF24 radio module. Initially, code was developed to allow the radio module to send and receive joystick commands that would move the motors according to the input. However, upon expanding the functionality by incorporating other sensor readings, the radio module stopped working. The receiver completely stopped receiving any data from the transmitter due to a faulty capacitor. Unfortunately, the team was unable to fix it, despite attempts to re-solder a new capacitor. This was an issue that the entire group identified and worked extensively on to try and fix.

8.2 * Solving the Radio Module Problem [Francisco Lam]

Multiple approaches were taken in order to fix the problem. Initially, the first fix suggested that there was too much noise affecting the receiver which prevented it from reading properly. To fix this a shield module for the radio was added that regulated current and acted as a decoupler. This however was unsuccessful. The team ensured that there was a clear line of sight and ran basic tests like sending and receiving "Hello World" to no avail. After further research on the module it was discovered that the capacitors were notoriously faulty on this particular model. An attempt was made to solder new capacitors but it was unsuccessful. After working for many hours on the repairs of the module, the team concluded to scrap it from the system. For now, the rover needs to be wired into a computer in order for the user to receive the sensor data and see the camera data. Other attempts of troubleshooting included changing drivers on our software, trying multiple micro usb cables, and downloading other external nRF24 libraries. In future versions, the team plan to re-incorporate the radio module, or decide on using bluetooth depending on the application of the rover.

8.3 * 3D Printing Problem [Iraj Shrotri]

In the initial plan the team aimed to 3D print a WW1 style tank chassis for our rover. The aim of the project was to have a tank design that could tread through various types of dirt and soils without getting stuck for optimal performance in its autonomous mode. However due to the warping of the larger prints the team concluded it wasn't feasible. Smaller parts like our rack and pinion for the moisture sensor and a prototype chassis were still able to be used in the final assembly.

8.4 * Solving the 3D Printing Problem [Iraj Shrotri]

It was no longer feasible to use the 3D printer for the chassis so it was necessary to quickly come up with a solution. Iraj found a plastic box from Home Depot that had good dimensions to fit all of our components and he was able to retrofit motors to it. To add the durability necessary for the project objectives, the group decided to add high tread tires to allow the rover to operate in uneven conditions.

8.5 * LiDAR Accuracy Problem [Nicholas Casillas]

When testing the LiDAR, there was an accuracy problem in the beginning. When it was connected to the Arduino and running, some of the data wasn't reading properly. It would skip over some of the angles and read inaccurate distance samples.

8.6 * Solving the LiDAR Accuracy Problem [Nicholas Casillas]

When doing research and figuring out what was wrong with the readings, it became obvious that the LiDAR wasn't receiving enough power to properly function. In order to fix this, the LiDAR had to be connected to a power supply. After connecting it to a power supply, and modifying the algorithm, the LiDAR began to be more consistent and accurate in its readings.

8.7 * pH Sensor Problem [Francisco Lam]

During the project's midpoint, an unexpected issue arose with the pH sensor. Initially functioning correctly and providing consistent readings, the sensor suddenly ceased to work, and subsequent tests yielded no values.

8.8 * Solving the pH Sensor Problem [Francisco Lam]

To address the malfunctioning pH sensor, Francisco took the initiative to conduct research and identify potential causes. Some responses pointed to the sensor's lower quality due to its cheaper brand, while another possibility was overheating from excessive power usage. Further investigation and experimentation were necessary to determine the exact cause and implement an appropriate solution.

9 User Interface Design

9.1 Application Control

N/A

9.2 User Interface Screens

N/A

10 * Test Plan

10.1 * Test Design [Iraj Shrotri]

Test Case 1: Sensor Calibration

Objective: To verify that sensor readings are accurate and reliable.

Function tested: Sensor accuracy and reliability

Technical design objective measured: Sensor accuracy

Experiment Setup:

- Connect the sensor to the microcontroller
- Place the sensor in a controlled environment (e.g. water with a known temperature, humidity chamber with known humidity levels, etc.)
- Use a multimeter or reference sensor to measure the actual values of the controlled environment

Experiment Procedure:

- Power up the microcontroller and sensor
- Read the values from the sensor
- Compare the readings with the actual values measured by the multimeter/reference sensor
- Record the difference between the actual and measured values

Expected Results: The difference between the actual and measured values should be within an acceptable range (e.g. +/- 5%)

Responsibility: Sensor calibration test will be conducted by Iraj for DHT11 and MPU6050. Francisco will test Soil Moisture Sensor and PH Sensor

Test Case 2: Obstacle Detection and Avoidance

Objective: To verify that the rover can detect obstacles and avoid them

Function tested: Object detection and avoidance

Technical design objective measured: Accuracy of obstacle detection and avoidance

Experiment Setup:

- Place the rover in a controlled environment with obstacles of varying sizes and shapes
- Use a camera to record the rover's movements

Experiment Procedure:

- Power up the rover and activate obstacle detection sensors
- Direct the rover to move towards the obstacles
- Record the rover's behavior in the presence of the obstacles
- Observe if the rover successfully avoids the obstacles

Expected Results: The rover should detect and successfully avoid the obstacles in its path

Responsibility: Obstacle detection and avoidance test will be conducted by Nicollas

Test Case 3: Wireless Communication

Objective: To verify that the wireless communication between the rover and remote control is reliable

Function tested: Wireless communication

Technical design objective measured: Reliability of wireless communication

Experiment Setup:

- Set up the rover and remote control in a controlled environment with no obstacles or interference

Experiment Procedure:

- Power up the rover and remote control
- Send commands from the remote control to the rover and observe the rover's response
- Record the time it takes for the command to be received and executed by the rover

Expected Results: The wireless communication should be reliable and responsive with no noticeable delay or loss of data

Responsibility: Wireless communication test will be conducted by Iraj. Due to issues with the radio every member participated and attempted to transmit data between two rnf24 modules

10.2 * Bug Tracking [Nicholas Casillas, Iraj Shrotri, Francisco Lam]

A database will be used to track defects found while performing the test cases. All defects will be logged as they are discovered. Defects will be assigned to Nicholas Casillas to fix, and to Francisco Lam to investigate.

10.3 * Quality Control

After completing the test cases, a thorough review will be conducted to ensure the following:

- All test cases were executed.
- All test cases were completed successfully.
- Any deviations from the test cases were properly documented.
- Each step of the test cases will be marked as "Passed" or "Failed."

Test 1: Passed

Test 2: Passed

Test 3: Failed

10.4 * Identification of critical components

During testing, the following components are considered critical and require particular attention:

- Sensor module: The accuracy and reliability of sensor readings are essential for the proper functioning of the system. Any issues with sensor performance need to be addressed.
- Motor driver board: It is crucial to ensure that both sides of the motor driver board are working as intended. Any problems with one side or improper functioning may affect the overall performance of the system.

- LiDAR (Light Detection and Ranging) system: As there have been identified bugs with the LiDAR system, it is necessary to focus on resolving these issues to ensure accurate distance measurements and obstacle detection.
- Rack and pinion mechanism: The rack and pinion system plays a significant role in the rover's movement. It is vital to verify that the mechanism is properly printed, assembled, and functions smoothly to achieve reliable and precise motion control.
- 3D printed components: The integrity and quality of 3D printed parts should be assessed to ensure they meet the required specifications and can withstand the operational demands of the system.
- 10k resistor: Attention should be given to prevent burnout of the 10k resistor. Proper functioning of this component is necessary for the correct operation of the circuit.
- 2MP camera: Special care should be taken to avoid any potential damage or malfunction of the 2MP camera during testing, as it plays a vital role in capturing visual information for the rover.

Responsibility: Nicholas Casillas, Iraj Shrotri, Francisco Lam

10.5 * Items Not Tested by the Experiments

N/A

11 * Test Report

The purpose of the tests is to document the results, compare them with the expected outcomes, and analyze the data. From this information valuable insights into the performance and limitations of the system are determined. This information helps identify areas for improvement and guides corrective actions to enhance the functionality and reliability of our modules and prototype.

11.1 * Test 1

Person(s) performing the experiment: Iraj Shrotri, Francisco Lam

Iteration 1:

1. Results: Failed
2. Comparison with expected results: Obtained garbage values instead of expected results
3. Analysis of test results: The cause of corruption needs to be identified
4. Corrective actions taken: Baud Rate was changed to address the corruption issue

Iteration 2:

1. Results: Passed
2. Comparison with expected results: Obtained expected results
3. Analysis of test results: N/A
4. Corrective actions taken: N/A

11.2 * Test 2

Person(s) performing the experiment: Nicholas Casillas

Iteration 1:

1. Results: Partially successful, ultimately failed
2. Comparison with expected results: LiDAR detected some obstacles but ignored others
3. Analysis of test results: LiDAR accuracy needs improvement
4. Corrective actions taken: Raised the motor frequency of LiDAR and improved the detection algorithm

Iteration 2:

1. Results: More successful, passing rate of 90%
2. Comparison with expected results: LiDAR detected almost all obstacles with a small margin of error
3. Analysis of test results: Behavior adjusted based on LiDAR readings for efficient rover movement
4. Corrective actions taken: N/A

11.3 * Test 3

Person(s) performing the experiment: Iraj Shrotri, Francisco Lam, Nicholas Casillas

Iteration 1:

1. Results: Failed
2. Comparison with expected results: Radio connection was not established
3. Analysis of test results: N/A
4. Corrective actions taken: Attempted to fix the issue but unsuccessful

Iteration 2:

1. Results: Was not connecting
2. Comparison with expected results: Connected to each radio successfully
3. Analysis of test results: N/A
4. Corrective actions taken: Tested using "hello world" code

12 * Conclusion and Future Work

12.1 * Conclusion [Iraj Shrotri, Francisco Lam, Nicholas Casillas]

Our team was tasked with developing a semi-autonomous soil sampling rover that would be able to navigate rough terrain and collect soil samples for analysis. Our overall project goals were to create a rover that was capable of operating independently, was reliable, and could collect accurate and precise data. After completing the project, we believe that we have met the overall project goals and the quantitative technical design objectives. The rover that we developed is capable of navigating rough terrain autonomously and collecting accurate and precise data. We tested the rover in various environments and found that it performed well and met our expectations.

However, during the testing phase, we encountered some technical issues that needed to be addressed. For example, the rover experienced difficulties with the 10K potentiometer and almost burnt out the 2MP camera. These issues were resolved by replacing the faulty components and implementing better temperature control measures. In addition the radio module stopped working 3 weeks before completion forcing us to scrap the transmitter. In addition the raspberry pi we had was faulty and the replacement one was lost in shipping so we replaced them to the best of our abilities.

From this project, we learned a great deal about technical knowledge and skills related to robotics, mechanical and electrical engineering, and software development. We also developed our professional and personal growth by improving our communication, collaboration, and problem-solving skills. Overall, we are proud of the work that we have accomplished, and we believe that our semi-autonomous soil sampling rover has the potential to be a valuable tool for scientific research and exploration. We are confident that our rover will be able to operate reliably in various environments and collect accurate and precise data.

Throughout this project, I have gained valuable technical knowledge and skills, as well as experienced significant professional and personal growth. From a technical standpoint, I have learned a great deal about designing and testing complex systems. The process of developing the rover prototype has given me hands-on experience in working with various components such as sensors, microcontrollers, and wireless communication modules. I have gained a deeper understanding of how these components interact and how to integrate them effectively.

Moreover, I have enhanced my problem-solving abilities through troubleshooting and resolving issues that arose during the project. Dealing with bugs and defects has taught me the importance of meticulous testing and the value of maintaining a bug tracking system. I have also become more proficient in documenting and reviewing test cases, ensuring comprehensive coverage and accurate results.

On a professional level, this project has strengthened my teamwork and collaboration skills. Working alongside my team members, I have learned to communicate effectively, delegate tasks, and coordinate efforts to achieve our common goals. Collaboration has allowed me to leverage the expertise and perspectives of my team members, leading to more innovative solutions and better outcomes.

12.2 Future Work

Expansion and Improvement; discuss the impact of this work and its possible expansion into perhaps a more promising design than what you had started. This is particularly important in order to address the marketability of your design. Or why this project merits another look by perhaps next year's students.

The impact of this work lies in its potential to create a cost-effective and reliable robotic system for agriculture that can be used by small farmers to increase efficiency and productivity. The use of sensors, object detection, and obstacle avoidance systems also helps to ensure that the system operates efficiently and effectively in the field.

There is a possibility for further expansion of this work, particularly in the areas of machine learning and artificial intelligence. These technologies could be integrated into the system to enhance its ability to detect and respond to changing conditions in the field, such as changes in soil moisture or temperature. This could result in even greater efficiency and productivity gains for farmers. Obviously, if we could get the radio transmitter to work, that design allows for remote monitoring and control of the system, which can help reduce labor costs and save time for farmers.

Additionally, the design could be expanded to include other features that are important to farmers, such as automated irrigation or fertilization systems. This would make the system more versatile and customizable to the specific needs of different crops and growing environments.

Overall, this project merits another look by next year's students or researchers because it has the potential to address an important need in the agricultural industry. The use of robotics and automation in agriculture is a growing trend, and this project provides a low-cost and reliable solution that could benefit small farmers in particular. The expansion and improvement of this work could lead to even greater advances in agricultural automation and increase the marketability of the design.

12.3 * Acknowledgement [Iraj Shrotri, Francisco Lam, Nicholas Casillas]

We would like to express our heartfelt gratitude and acknowledge the contributions of the following individuals and organizations who have played a pivotal role in the success of our project:

- DroneBotWorkshop: We extend our appreciation to DroneBotWorkshop for providing the schematic that served as a valuable reference during the development of our project.
- YDLIDAR-SDK: We are grateful to the YDLIDAR-SDK for their software development kit, which enabled us to integrate and utilize the LiDAR sensor effectively.
- ArduCAM: We would like to thank ArduCAM for their library, which facilitated the seamless integration and operation of the 2MP camera module in our project.
- ArduCAM Host App: We express our gratitude to the developers of the ArduCAM host app, as their application allowed us to stream and visualize the camera feed in real-time, enhancing the functionality of our system.
- Adafruit: We are thankful to Adafruit for their DHT library and MPU6050 library, which provided us with the necessary tools and functions for interfacing with the DHT11 sensor and MPU6050 accelerometer.

- Teaching Assistant Cody Simons: We acknowledge our Teaching Assistant Cody Simons for his valuable design insights. His expertise and guidance greatly contributed to the success of our project.
- Professor Roman Chomko: We extend our appreciation to Professor Roman Chomko for his invaluable guidance throughout the project. His expertise and support played a crucial role in our achievements.
- Professor Manglai Zhou: We would like to thank Professor Manglai Zhou for providing assistance with 3D printing, which was essential for the fabrication of certain components in our project.
- Professor Tofigh Heidarzadeh: We express our gratitude to Professor Tofigh Heidarzadeh for his assistance in editing the technical project report. His expertise and attention to detail enhanced the quality of our documentation.
- Fritzing: We would like to acknowledge Fritzing, the schematic designer software, for enabling us to create and visualize our circuit diagrams effectively.

We are deeply grateful for the support and contributions of these individuals and organizations. Their ideas, solutions, guidance, and resources have significantly contributed to the technical knowledge, skill development, and overall growth of our team throughout this project.

13 * References

ArduCAM-M-2MP camera shield, https://www.arducam.com/downloads/shields/ArduCAM_Mini_2MP_Camera_Shield_DS.pdf (accessed Jun. 14, 2023).

Arduino® MEGA 2560 REV3, https://docs.arduino.cc/static/4600a13ae82198c5cffe9847edd090d3/A000067-datasheet.pdf (accessed Jun. 14, 2023).

DHT11 humidity & temperature sensor - mouser electronics, https://www.mouser.com/datasheet/2/758/DHT11-Technical-Data-Sheet-Translated-Version-1143054.pdf (accessed Jun. 14, 2023).

MPU-6000/MPU-6050 9-axis evaluation board user guide - TDK invensense, https://invensense.tdk.com/wp-content/uploads/2015/02/MPU-6000-EV-Board1.pdf (accessed Jun. 14, 2023).

YDLIDAR X4 data sheet,https://www.ydlidar.com/Public/upload/files/2022-06-28/YDLIDAR%20X4%20Data%20sheet%20V1.1(211230).pdf (accessed Jun. 14, 2023).

14 * Appendices

*** Appendix A:** Parts List
DC Motor Wheels
DC Motor Encoders
DC Motor Driver L298N
Arduino Nano
nrf24
PH Sensor
LiDAR
2 MP Camera
Gyroscope/ Accelerometer
GPS
Temp/Humidity Sensor
Joysticks
Buttons/Switches
Battery Pack
Soil Sensor
Stepper Motor + Driver Board
Raspberry Pi

*** Appendix B:** Equipment List
Hot glue gun
Soldering Iron
3D Printer
Dremel Tool
Electrical Tape

*** Appendix C:** Software List (URL to online drive or SVN server, with sharing set to Public. Can omit this appendix if your project didn't involving writing a program)
Arduino IDE
PuTTy
VSCode