School of Electrical Engineering and Computer Science | École de science informatique et de génie électrique

University of Ottawa | Université d'Ottawa

ELG4912 | Winter 2024

Group 5: Final Project Report

ELG 4912

Search and Rescue Remote-Controlled (RC) Car with LiDAR Mapping and Life Detection

Course Professor: **Emil Petriu**
Course TA: **Eslin Karatop**

Name:	Student ID:
Fatmah Bayrli	300159193
Papa Kane	300090159
Walid Rashad	300205109
Julien Kapro	300173066
Moktar Abdillahi-Abdi	300156852
Geoffrey Hooton	300187367

Date: April 10, 2024

School of Electrical Engineering and Computer Science | École de science informatique et de génie électrique

University of Ottawa | Université d'Ottawa

Table of Contents

1. Project Charter

1.1 Charter Introduction

1.1.1 Document Change Control

Revision Number	Date of Issue	Authors	Brief Description of Change
1.0	[2024-02-29]	Fatmah Bayrli, Geoffrey Hooton, Papa Kane, Walid Rashad, Moktar Abdillahi-Abdi, Julien Kapro	Creation of the document.
2.0	[2024-04-08]	Fatmah Bayrli, Geoffrey Hooton, Papa Kane, Walid Rashad, Moktar Abdillahi-Abdi, Julien Kapro	Updating document as required for final report.

1.1.2 Executive Summary

This project was initiated as part of the Capstone course for Electrical Engineering at the University of Ottawa (ELG 4912). Our team plans to complete the design of a remote-controlled (RC) car that can detect life (humans buried in rubble after an earthquake or explosion) in cases where modern sensors would fail to do so. The final deliverable of this project will be submitted to the professor of the course as part of the fulfilment of course requirements. The main parties that will be impacted should this project be successful are organizations that specialize in disaster relief.

1.1.3 Authorization

Geoffrey Hooton

2024-04-10

Project Team Lead

Fatmah Bayrli

2024-04-10

Team Member

Walid Rashad

2024-04-10

Team Member

Papa Kane

2024-04-10

Team Member

Moktar Abdillahi-Abdi

2024-04-10

Team Member

Julien Kapro

2024-04-10

Team Member

1.2 Project Overview

1.2.1 Project Summary

1.2.2 Project Goals, Business Outcomes and Objectives

No.	Goals	Objectives	Business Outcomes
1.	Decrease hazard for first responders	Design an RC car capable of navigating through rubble and tight spaces, eliminating the need for first responders to enter a collapsed building.	Increased safety for first responders
2.	Disaster navigation efficiency	Utilize sensors and mapping technologies to allow the car to navigate through debris in a timely manner.	Faster navigation
3.	Improved search and rescue operations	Equip the car with life detecting technologies to detect people stuck under debris.	Increase survival rates after disastrous situations

1.2.3 Project Scope

1.2.3.1 Scope Definition

Our project encompasses the development of an R/C prototype vehicle intended to function as an innovative instrument for seismic rescue operations. With LiDAR technology, this vehicle can navigate through rubble and is specifically designed to function in situations where standard sensors would struggle. Its ability to detect life beneath the debris is improved with the addition of thermal imaging cameras and vibration-detecting sensors. The car's tracks provide better traction, and its concentration on flexibility to uneven terrain makes it a vital tool for search and rescue teams operating in disaster-affected areas.

1.2.3.2 Boundaries

See sections 2.2.1 and 2.2.2 in the Requirements section.

1.2.4 Milestones

No	Project Milestone	Description	Expected Date
1.	Create project proposal	The team worked together to pick the project topic.	2024-01-26
2.	Research Stage	In-depth research on necessary technologies and existing solutions.	2024-02-2
3.	Design	Selection and assembly of the RC car, sensors, and control station components.	2024-03-18
4.	Implementation	Implementation of hardware, onboard software, control software for hardware, remote control software.	2024-10-21
6.	Testing	Create specific test cases and do simulation	2024-12-5

1.2.5 Deliverables

The table below outlines the project deliverables. A description of each deliverable, an acceptance criterion, and a due date are mentioned:

Project Deliverable 1: Project Proposal
Description: Document outlining the description of the project.
Acceptance Criteria: Include enough general information with only high-level information and a brief plan.
Due Date: Jan 26, 2024
Project Deliverable 2: Midterm Presentation
Description: Presentation outlining the project goal, requirements, constraints, use case, design, budget, risk-management, and business case.

Acceptance Criteria: Follow the presentation requirements outlined on the course Brightspace page.
Due Date: Feb 15, 2024
Project Deliverable 3: Midterm Report
Description: Report outlining the project charter, system requirement specifications, conceptual design, work breakdown structure, schedule with milestones, and estimated budget, hazard identification and risk assessment, safety and risk management plan, and contribution list.
Acceptance Criteria: Follow the report requirements outlined on the course Brightspace page.
Due Date: Feb 29, 2024
Project Deliverable 4: Final Presentation
Description: Presentation outlining the project goals, business case, requirement specifications, detailed design, anticipated risks, proof of concept, test plan, schedule and budget outlook. Like the midterm report, but at the end of the course.
Acceptance Criteria: Follow the report requirements outlined on the course Brightspace page.
Due Date: March 28, 2024
Project Deliverable 5: Final Report
Description: Document outlining the project charter, system requirements specification, detailed design, updated Gantt chart, work breakdown structure, schedule with milestones and estimated budget, updated risk management plan, test plan, proof of concept, detailed contribution list, post-performance analysis, references, and code version control system. Like the midterm report, but at the end of the course.
Acceptance Criteria: Follow the report requirements outlined on the course Brightspace page.
Due Date: April 10, 2024

1.2.6 Project Cost Estimation and Source of Funding

1.2.6.1 Project Cost Estimation

The following budget overview table outlines costs for various components necessary for the project.

Component	Costs
Slamtec RP LiDAR Sensor	$0.00
FLIR Lepton LWIR Camera	$100.21
Geophone SM-24	$87.68
Ultrasonic Sensor	$0.00
Servo Controller	$0.00
ON/OFF Switch	$1.15
Acrylic glass base with 3 pairs of wheels	$0.00
Battery 7.4V	$0.00
Voltage Regulator	$2.69
Servo Motor Driver	$0.00

Ethernet Cable(100ft)	$25.88
Onboard Computer (Raspberry Pi)	$0.00
ADS1115 ADC	$20.32
Software simulation environment	$0.00
Total	**237.93**

The estimated budget for the RC car development project encompasses all critical phases including project initiation, design, and development, and testing and refinement. This comprehensive financial plan is designed to ensure that all necessary resources and materials are adequately funded to achieve project objectives without unnecessary overspending.

The total estimated budget for the RC rescue car development project is $237.93. This budget is meticulously planned to cover all phases of the project, from initial planning and design to comprehensive testing and final refinements. The allocation of funds is carefully considered to balance the need for quality and efficiency with cost-effectiveness. Regular budget reviews and updates will be essential to manage expenses effectively and adapt to any changes in the project scope or requirements. This financial planning aims to ensure the successful completion of the project within the allocated budget, maximizing value while minimizing unnecessary expenditures.

1.2.6.2 Source of Funding

Our project's funding will be jointly contributed by our group. We have agreed to share the financial responsibilities equally, ensuring that all necessary materials and resources for our project are covered.

1.2.7 Dependencies

Our project does not currently have any dependencies but may have some in the future.

Dependency Description	Critical Date	Contact
N/A	N/A	N/A

1.2.8 Project Risks, Assumptions, and Constraints

1.2.8.1 Risks

No.	Risk Description	Probability (High, Medium, Low)	Impact (High, Medium, Low)	Planned Mitigation
1.	Laser Hazards	Low	Low	Avoid looking directly at laser for a prolonged period of time
2.	Battery Incineration	Low	High	Heat Sensor, fire extinguisher, grounding

3.	Treads – Moving Parts	Low	Medium	Shut systems down before handling the device – stay at a safe distance
4.	Electrical Damage – Short circuits	Medium	Low	Test circuit/Heat Sensor
5.	Soldering burns/inhalation	High	Medium	Workspace with proper aeration/gloves if inexperienced + tutorials
6.	Troubleshooting Issues	High	High	Good planning and external support
7.	Conceptual Revisions	Medium	High	Good planning

1.2.8.2 Assumptions

No.	It is assumed that:
1.	All members can shoulder an equal portion of the funding
2.	Workstation is available upon demand
3.	All required materials are available when needed
4.	Documentation and support online should cover conception challenges
5.	Feedback will be available on project updates

1.2.8.3 Constraints

No.	Category	Constraints
1.	Deadlines	- Reports and presentations
2.	Work Location	- Exclusively in the Lab
3.	Legal	- No copyright infringement
4.	Finance	- Taxes and inflation
5.	Program Restrictions	- Project must involve electrical engineering elements

1.3 Project Organization

1.3.1 Project Governance

Decisions for the project will be collectively determined through group consensus. When a decision is required, the team will convene to find a solution, either through a democratic vote or mutual agreement. Additionally, our group leader will act as the primary source for communication with our professor, teaching assistant, and technician, ensuring streamlined and effective coordination.

1.3.2 Project Team Structure

Our project team consists of a team leader and five other group members. All members contribute equally to decisions made for the project, and the team leader will help guide the decisions and mediate in such a way that the team can come to a consensus on decisions.

1.3.3 Roles and Responsibilities

<table>
<tr><th rowspan="2"></th><th colspan="6">Project Role</th></tr>
<tr><th>Team Leader</th><th colspan="5">Group Member</th></tr>
<tr><td>Responsibilities</td><td>• Team management
• Conflict resolution, planning,
• Keeping pace with schedule
• All other responsibilities of a “Group Member”</td><td colspan="5">• Participating in labs
• Communicating progress on assigned work documenting work
• Contributing to the schedule regularly
• Creating test plans for prototypes
• Updating SRS regularly
• Assisting in testing and documenting results
• Updating budget</td></tr>
<tr><td>Assigned to</td><td>Geoffrey Hooton</td><td>Fatmah Bayrli</td><td>Papa Kane</td><td>Walid Rashad</td><td>Julien Kapro</td><td>Moktar Abdillahi-Abdi</td></tr>
</table>

1.3.4 Project Facilities and Resources

All facilities used to create our prototypes will be provided by the University of Ottawa. This includes lab spaces, classrooms, etc. No work involving lab equipment will be done at home. If, for example, soldering is required at any point for the project, the group will use the faculty of engineering’s laboratory soldering stations. Some resources for the project will also be provided by the university. These resources will be discussed with the lab technician prior to use and signed out appropriately. Once the project has finished, these resources will be returned to the University of Ottawa.

1.4 Project References

[1] Market Research Future, “Search and rescue robots market size to grow substantially at 18.2% CAGR from 2020 to 2027 - report by Market Research Future (MRFR),” GlobeNewswire Newsroom, https://www.globenewswire.com/news-release/2021/08/17/2282191/0/en/Search-and-Rescue-Robots-Market-Size-to-Grow-Substantially-at-18-2-CAGR-from-2020-to-2027-Report-by-Market-Research-Future-MRFR.html (accessed Feb. 29, 2024).

[2] “Benewake tf-Luna 8M lidar distance sensor,” RobotShop USA, https://www.robotshop.com/products/benewake-tf-luna-8m-lidar-distance-sensor (accessed Feb. 29, 2024).

[3] “MmWave radar - 24ghz human presence detection sensor (9 meters),” DFRobot, https://www.dfrobot.com/product-2282.html (accessed Feb. 29, 2024).

[4] A. Industries, “Adafruit 16-channel 12-bit PWM/Servo driver - I2C interface,” adafruit industries blog RSS, https://www.adafruit.com/product/815 (accessed Feb. 29, 2024).

[5] Raspberry Pi Foundation, “Teach, learn, and make with the Raspberry Pi Foundation,” Raspberry Pi Foundation, https://www.raspberrypi.org/ (accessed Feb. 29, 2024).

[6] “12V 6AH 6000mah 18650 battery pack BMS with 12.6V 1A charger,” BATTERYINT, https://batteryint.com/products/12-v-4-4-ah-4400-mah-18650-battery-12v-with-bms-lithium-battery-protection-board-12-6v-1a-charger?currency=USD&utm_medium=cpc&utm_source=google&utm_campaign=Google+Shopping&stkn=12d3a63dd342&gad_source=1&gclid=CjwKCAiAivGuBhBEEiwAWiFmYXt1ZEg5YUJm95afA6ORubAPaWBMirT-hbnofU0F9VMQmPlGthItoxoC_0IQAvD_BwE (accessed Feb. 29, 2024).

[7] “DEVMO 4s 14.8v 30A BMS protection PCB cell board Li-Ion Lithium 18650 Battery Balance Module,” Amazon.ca: Electronics, https://www.amazon.ca/DEVMO-Protection-Lithium-Battery-Balance/dp/B07TSRJ15V/ref=sr_1_1_sspa?dib=eyJ2IjoiMSJ9.9dqVeQ9Sz0XfOldfHjf45SMy01YW-w_ro2iaUYN3NhMoGJVKk0rOaEjg_2uEA1QP1Y-YH_ibi_b1cFFNRUwCAJFoJQPG4kYXaXtShuO3rUdV6Yz3pMKxiQl64eyDd1QtdJ_SbFJbKYIT9JXnsED9tfPcGqEcAwyQ0rqbQ0BPd-hlAuvpH_zTb6PTS-7vaIAQG2rnrsmI1fsfujqd0PzTOFNwA8L25LKR5NX935Wz9vhhbXifjBSIIMqQ3C_AKZueenf_oQki2qiUAWPqSLELo1srYPsycjytgmcjF3G4cP4.IcI13GtmSg8Vw5EQeG1i12vjm3hSFu7CK0kxACymmi8&dib_tag=se&hvadid=647846474815&hvdev=c&hvlocphy=9000682&hvnetw=g&hvqmt=e&hvrand=6076484758121903 62&hvtargid=kwd-896339877316&hydadcr=19997_13393203&keywords=12v%2B18650%2Bbms&qid=1709010598&refinements=p_36%3A12035760011&rnid=12035759011&s=electronics&sr=1-1-spons&sp_csd=d2lkZ2V0TmFtZT1zcF9hdGY&psc=1 (accessed Feb. 29, 2024).

[8] “DEVMO 4s 14.8v 30A BMS protection PCB cell board Li-Ion Lithium 18650 Battery Balance Module,” Amazon.ca: Electronics, https://www.amazon.ca/DEVMO-Protection-Lithium-Battery-Balance/dp/B07TSRJ15V/ref=sr_1_1_sspa?dib=eyJ2IjoiMSJ9.9dqVeQ9Sz0XfOldfHjf45SMy01YW-w_ro2iaUYN3NhMoGJVKk0rOaEjg_2uEA1QP1Y-YH_ibi_b1cFFNRUwCAJFoJQPG4kYXaXtShuO3rUdV6Yz3pMKxiQl64eyDd1QtdJ_SbFJbKYIT9JXnsED9tfPcGqEcAwyQ0rqbQ0BPd-hlAuvpH_zTb6PTS-7vaIAQG2rnrsmI1fsfujqd0PzTOFNwA8L25LKR5NX935Wz9vhhbXifjBSIIMqQ3C_AKZueenf_oQki2qiUAWPqSLELo1srYPsycjytgmcjF3G4cP4.IcI13GtmSg8Vw5EQeG1i12vjm3hSFu7CK0kxACymmi8&dib_tag=se&hvadid=647846474815&hvdev=c&hvlocphy=9000682&hvnetw=g&hvqmt=e&hvrand=6076484758121903 62&hvtargid=kwd-896339877316&hydadcr=19997_13393203&keywords=12v%2B18650%2Bbms&qid=1709010598&refinements=p_36%3A12035760011&rnid=12035759011&s=electronics&sr=1-1-spons&sp_csd=d2lkZ2V0TmFtZT1zcF9hdGY&psc=1 (accessed Feb. 29, 2024).

[9] Amazon.com: Customer reviews: DEVMO 4S 14.8v 30A BMS protection ..., https://www.amazon.com/DEVMO-Protection-Lithium-Battery-Balance/product-reviews/B07TSRJ15V (accessed Feb. 29, 2024).

[10] L7812CV STMicroelectronics | Integrated Circuits (ICS) | digikey, https://www.digikey.ca/en/products/detail/stmicroelectronics/L7812CV/585973 (accessed Feb. 29, 2024).

[11] LiDAR Produced Map Using LidarView. Kitware Inc. https://lidarview.kitware.com/ (accessed Feb. 29, 2024).

[12] R. Bain, "Ros rviz Lidar Data," YouTube, https://www.youtube.com/watch?app=desktop&v=jN6E9X0l7xs&ab_channel=RobertBain%29 (accessed Feb. 29, 2024).

1.5 Glossary and Acronyms

Terms, Acronyms, abbreviations	Definitions
RC	Remote Controlled
WBS	Work Breakdown Structure

2. System Requirement Specification

2.1 Introduction

Due to the limits of current sensors, recovering victims trapped beneath collapsed structures following earthquakes is a difficult task. Our project focuses on creating a cutting-edge remote-control prototype car to address this important issue. This vehicle is designed to maneuver through the intricate structures of fallen buildings, where traditional sensors might become unreliable. Our remote-controlled vehicle (RC car) incorporates cutting-edge technologies including LiDAR, vibration-sensing sensors, and thermal imaging cameras to facilitate efficient search and rescue missions in difficult terrain.

2.2 Scope

2.2.1 Will Do:

1. Implement a user interface for remote control on a central computer (laptop).
2. Provide an intuitive remote-control UI with easy steering and acceleration controls.
3. Display a detailed, real-time 3D map of the vehicle's surroundings for navigation.
4. Ensure reliable communication of sensor data to the operator.
5. Detect human movements/vibrations underground.
6. Detect heat signatures underground.
7. Design a rugged and durable car.
8. Enable navigation in constrained environments and over uneven terrain.

9. Utilize tracks for improved traction on irregular surfaces.

2.2.2 Will Not Do:

1. Detailed mapping of the entire collapsed structure (focus on immediate surroundings).
2. Interaction with objects other than detecting signs of life.

2.2.3 Assumptions:

- The car's design will incorporate a remote emergency shut-off mechanism.
- The vehicle will have a physical on/off switch and a start command on the central computer.
- The system will be portable and easily deployed to support ongoing rescue efforts.
- The RC car will be properly grounded to eliminate the possibility of sparking.
- The car’s chassis will include emergency lighting LEDs for enhanced visibility.
- The vehicle’s design will adhere to Canadian safety standards for electrical and material safety.

2.3 Characteristics of the RC Car:

The functional and non-functional features of the radio-controlled car define its capabilities. Its functionality includes a basic remote-control interface that makes steering and acceleration straightforward. While real-time 3D mapping facilitates navigation across difficult terrain, robust conveyance of sensor data back to the operator increases overall efficacy. Among other non-functional characteristics, servo motor control and decreased latency in UI map refreshes provide real-time responsiveness. The system's minimum 3-hour battery life, high accuracy human presence detection, and adherence to safety rules are indicative of its robust design and performance.

2.4 Overall Description:

The challenges faced in earthquake rescue missions are the source of the RC car's general design and objective. Detection of vital signs and advanced sensor technologies for effective navigation through debris are among its capabilities. The car's robust design and adaptability to various terrain types ensure its dependability under demanding circumstances. Strict specifications, including safety features, portability, and compliance with Canadian safety regulations, highlight our commitment to creating an automobile that not only meets the highest performance standards but also prioritizes the safety of rescue personnel and the successful completion of ongoing missions.

2.5 Functional Requirements:

1. **User Interface:** The RC car's functional requirements emphasize an intuitive user interface accessible from a central computer (laptop) for remote control, ensuring ease of operation for steering and acceleration.
2. **Real-time 3D Map:** A significant characteristic is the ability to provide the operator with a detailed, real-time 3D map of the car's surroundings, enhancing navigation through complex terrain.
3. **Reliable Sensor Communication:** The system is characterized by its capability to reliably communicate sensor data to the operator, enhancing the overall effectiveness of the RC car.
4. **Detection Capabilities:** The RC car is designed to detect human movements/vibrations and heat signatures underground, showcasing its adaptability for search and rescue operations.
5. **Durability and Navigational Adaptability:** Ruggedness and adaptability to navigate constrained environments and uneven terrain are essential characteristics, ensuring the RC car's robust performance.

2.6 Non-Functional Requirements

1. **Latency Considerations:** A 3D UI map will update with a latency of no more than 500ms.
2. **Latency Considerations:** The operator can control the car's servo motors with a latency of less than 30ms.
3. **Latency Considerations:** Sensor data will be transmitted with a latency of at most 30ms.
4. **Accuracy in Human Presence Detection:** The system should exhibit high accuracy (at least 95%) in detecting human presence, highlighting its reliability in critical scenarios.
5. **Power and Battery Management:** The RC car's non-functional aspects encompass power-related characteristics, such as being powered by a 7.4V battery ensuring sustained operation.
6. **Power and Battery Management**: The battery life should be at least 3 hours.
7. **Compact Size and Terrain Traction:** The car's size (smaller than 20cm in width and 15cm in height) and the inclusion of tracks contribute to its non-functional characteristics, providing compactness and improved traction on uneven terrain.

2.7 Constraints

1. The car's design will incorporate a remote emergency shut-off mechanism.
2. The vehicle will have a physical on/off switch and a start command on the central computer.
3. The system will be portable and easily deployed.

4. The RC car will be properly grounded to eliminate the possibility of sparking.
5. The car's chassis will include emergency lighting LEDs for enhanced visibility.
6. The vehicle's design will adhere to Canadian safety standards for electrical and material safety.

To summarize, this System Requirements Specification (SRS) provides detailed information about a new remote-controlled prototype automobile designed for earthquake rescue operations. The car is made to maneuver around obstacles in collapsed structures using thermal imaging cameras, vibration-detecting sensors, and LiDAR technology. The functional requirements prioritize dependable sensor data delivery to a central computer, real-time 3D mapping for complex terrain, and an intuitive remote control user interface. Non-functional criteria emphasize tiny but efficient design, low latency, and excellent detection accuracy. Some limitations are emergency shut-off systems, transportability for quick deployment, and compliance with Canadian safety laws. In summary, the SRS establishes a solid foundation for the development of a technologically advanced, agile remote-controlled vehicle which will be a crucial component of efficient and secure earthquake victim rescue efforts.

3. Detailed Design

3.1 Objective

This section will provide a detailed view of the system hardware and software architecture along with component choices and possible alternatives.

3.2 Components Overview

3.2.1 Vehicle Breakdown

This section outlines the components that will be included on the vehicle. This includes the vehicle's key features and control systems.

3.2.1.1 Vehicle Breakdown Summary

Component	Purpose	Proposed Component	Possible Alternatives
LiDAR Sensor	3D mapping of the vehicle's environment.	Slamtec RPLIDAR	Benewake TF-Luna 8m LiDAR Distance Sensor
LWIR Thermal Imaging Camera	Detecting buried heat signatures (life detection).	FLIR Lepton LWIR Micro Thermal Camera Module	-

Ground Penetrating Radar	Detecting underground vibrations (life detection).	SM-24 Geophone	Infineon XENSIV™ 60GHz Radar
Servo Motor Driver	Controlling the servos attached to the car's wheels/tracks.	PCA9685 16-Channel 12-bit PWM Servo Driver	L9110S DC Stepper Motor Driver Board H Bridge 4 channel drive DC motor drive board motor drive
Onboard Computer	Sensor data processing and servo control.	Raspberry Pi 4 Model B	Jetson Nano, VisionFive2
ON/OFF Switch	Physical option for powering on and off vehicle.	- A basic Rocker or slide switch	-
Chassis	Housing sensors, Raspberry Pi, wheels/tracks, and servos.	See 3.2.8	-
Ethernet Cable	Long range serial communication	Generic 100-meter ethernet cable	-

Table 3.2.1.1.1: Vehicle Breakdown Summary

3.2.1.2 LiDAR Sensor

The LiDAR sensor will provide real-time, three-dimensional mapping of the search area. The generated map will be presented to the operator, who will use it to navigate complex terrain. We have chosen Benewake TF-Luna 8m LiDAR Distance Sensor because it provides an accurate and stable detection range of 8m, as well as adapting multiple adjustable configurations and works well in the complex scenarios our project demands.

A possible alternative LiDAR sensor would be the Slamet RPLIDAR, however the high cost makes it a more difficult choice for our project. In future designs, investing in a higher precision LiDAR sensor like the Slamet RPLIDAR could be a strong choice.

3.2.1.3 LWIR (Long-Wave Infrared) Thermal Imaging Camera

For underground life detection, a LWIR thermal imaging camera is preferred to a normal thermal camera because the longer wavelengths leveraged by the camera has a stronger possibility of detecting heat through surfaces. For our application the FLIR Lepton LWIR Micro Thermal Camera Module is a great choice. The FLIR Lepton is a beloved LWIR thermal imaging camera among Raspberry Pi enthusiasts which will make integrating it into our project much simpler.

3.2.1.4 Ground Penetrating Radar (GPR)

Ground penetrating radars are popular choices for detection of underground bodies. There is currently significant research using GPRs to detect human remains in cemetery graves. GPRs can detect both stillness and motion with strong anti-interference abilities. The radar that we decided on is the SM-24, which supports serial input and output control.

A possible alternative to the SM-24 geophone is the Infineon XENSIV™ 60GHz Radar, however similarly to the LiDAR sensor, the high cost makes it difficult to justify for our project. The Infineon radar sensor specializes in power efficiency within a small form factor, and should another, higher-budget iteration of our project come to fruition, it would be a great choice for the GPR.

3.2.1.5 Servo Motor Driver

The PCA9685 servo controller can manage up to 16 servos with precise movement which will easily allow the operator to control the car's tracks for navigation. This component was selected due to its popularity and ease of purchasing online from Adafruit. That said, there are many great options for servo controllers and any other reliable servo controller, like the L9110S DC Stepper Motor Driver, could be used.

3.2.1.6 Raspberry Pi

The Raspberry Pi 4 Model B is a powerful and cost-effective microcontroller that can handle tasks including controlling sensors in real time. The Raspberry Pi 4 is the most popular version in the Raspberry Pi series and offers easily enough computing power for our application. It is widely used by professionals and hobbyists alike and offers plenty of documentation online to help in the project implementation. The Raspberry Pi supports four serial ports, which is perfect for our three primary sensors. A possible alternative for the onboard computer could be the Jetson Nano or VisionFive2, however the accessibility and cost to performance ratio of the Raspberry Pi 4 is unmatched.

3.2.1.7 ON/OFF Switch

An on/off switch is a component that can control the power supply to the vehicle's electronic system. This will serve as an additional level of redundancy while powering on the vehicle to ensure safety.

3.2.1.8 Chassis

A large chassis (pictured in *Figure 3.2.1.8.1*) capable of housing up to six wheels, the Raspberry Pi, six servo controllers, and the vehicle's three sensors has been obtained from the University. This is what our prototype will be built on.

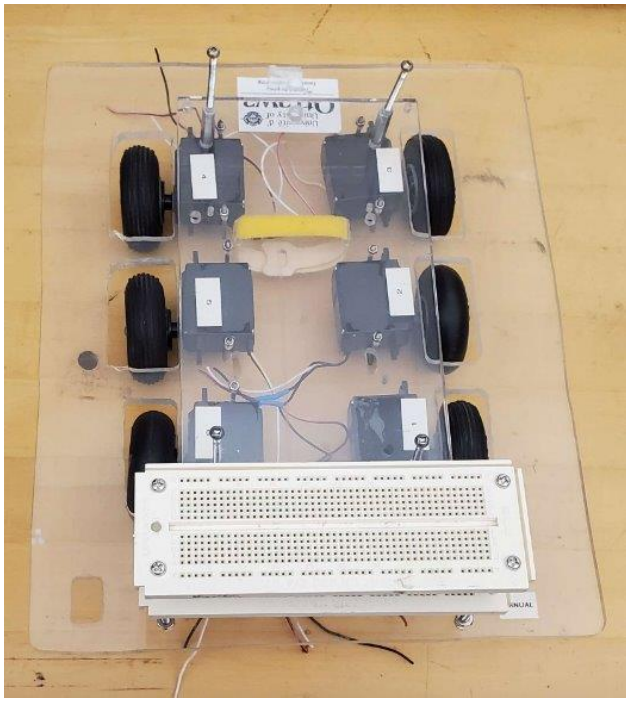


Figure 3.2.1.8.1 Chassis Obtained from the University

3.2.1.9 Ethernet Cable

A 50m ethernet cable will be sufficient to give a long enough range for the RC car, while still permitting communication without loss of data.

3.2.2 Power System Summary

This section outlines the components that will be included in the system's power system. These components are crucial for powering all the car's sensors/computers/motors.

3.2.2.1 Power System Breakdown Summary

Component	Purpose	Proposed Component	Possible Alternatives
Battery 7.4V	Supplying power to the system.	Generic 18650 Li-ion battery Pack	NiMH battery pack
Voltage Regulator	Maintaining consistent voltage to the vehicle's components.	L7812CV voltage regulator	-
Servo Motor Driver	Controlling power supplied to the servo motors	PCA9685 16-channel 12 bits PWM Servo Driver	L9110S DC Stepper Motor Driver Board H Bridge 4 channel drive DC motor drive board motor drive

Table 3.2.2.1.1: Power System Breakdown Summary

3.2.2.2 Battery 7.4V

The rechargeable battery we will use is a generic 18650 Lithium-ion battery with 7.4V. This is a good choice because it offers a high-capacity lithium battery with integrated protection while charging or discharging. This prevents overcharging, extending the battery's lifespan. A viable alternative is the NiMH battery pack which undergoes rigorous testing to ensure quality. The trade-off is that these battery packs are much more expensive.

3.2.2.3 Voltage Regulator

The L7812CV is a fixed-output voltage regulator that can output anywhere from 5V to 18V. In our design, this regulator will help us step down the battery voltage to 5V to power the Raspberry Pi. The L7812CV is very robust and is designed to handle input voltages up to 35V and deliver up to 1.5A of current. It is a popular choice because it is reliable and easy to use. The L7812CV is favored for its straightforward application, availability, and affordability, making it a constant 12V supply.

3.2.2.4 Servo Motor Driver

See 3.2.1.5

3.3 System Architecture

3.3.1 ***Hardware Architecture***

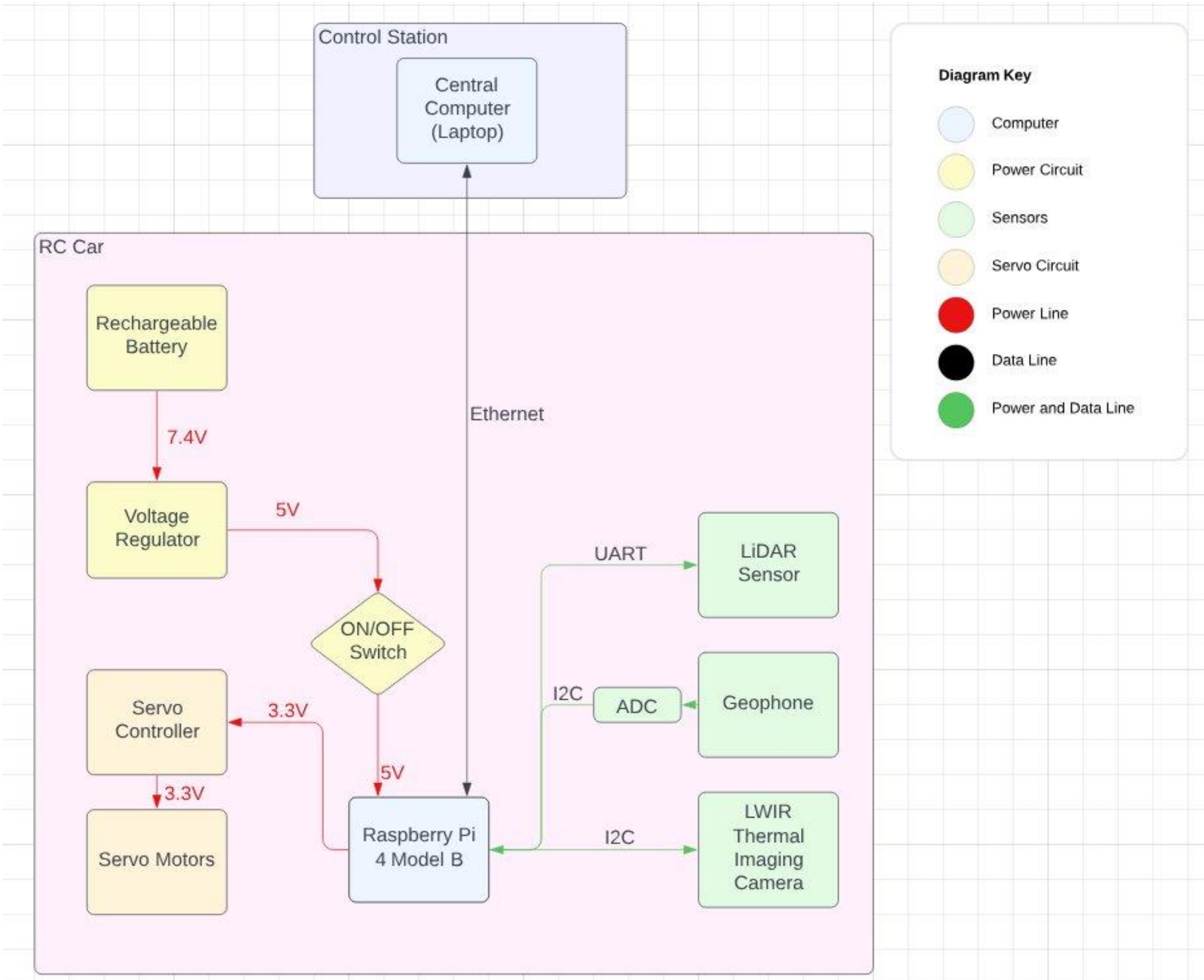


Figure 3.3.1.1 Hardware Architecture

The above block diagram outlines a high-level view of the hardware architecture given the components listed in sections 3.2.1 and 3.2.2. The system will rely on ethernet communication between the central computer (laptop) and Raspberry Pi, and a mixture of I2C and UART between the Raspberry Pi and external sensors. The serial protocols are outlined in *Figure 3.3.1.1* above. The servo motors and sensors will all be powered by the Raspberry Pi. To ensure a long battery life and low power losses, the vehicle will be powered by an onboard 7.4V. The battery will directly power the Raspberry Pi after being stepped down to 5V using a voltage regulator. Previously in the midterm report we had intended to power the central computer and Raspberry Pi microprocessor using a larger centralized battery, however after speaking with the lab technician, concerns were raised about potential power losses when trying to transmit power over a long, wired connection from the control station to the vehicle. Another change suggested by the lab technician was removing the battery management system which was deemed unnecessarily complex for a prototype.

3.3.2 Software Architecture

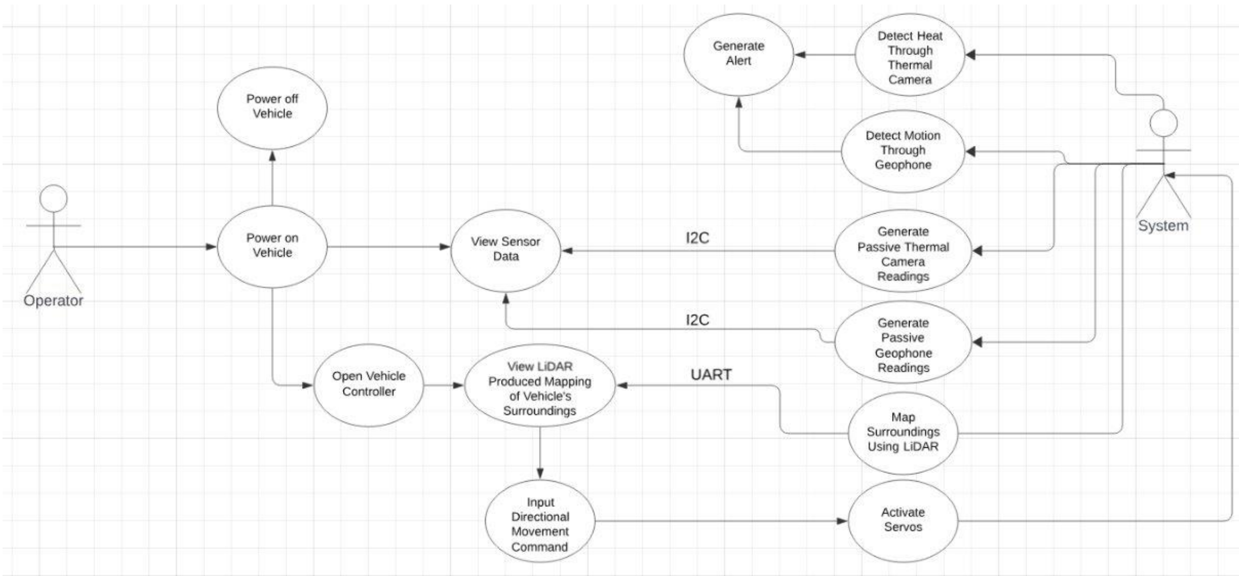


Figure 3.3.2.1 Software Use Case Diagram

The software for our project will be written in Python and will be very basic. The main purpose of the software in our project is to tie together the sensor data into a single application with some additional alert features if the sensor picks up any unusual values. The software should also give a basic user interface for control of the servo motors. The functionalities of the software are displayed in the use case diagram depicted in *Figure 3.3.2.1* above. The operator will log into the central computer (likely to be a laptop), after which they will be able to power on and off the vehicle, view LiDAR/Thermal Camera/Ground Penetrating Radar sensor data and open the vehicle controller. The vehicle controller will present the operator with a 3D map of the vehicle's surroundings generated by LiDAR visualization software (see section 3.4). In this view, the operator can also input directional movement commands, which will send commands to the Raspberry Pi to activate the servo motors. On the system's side, the onboard sensors will be passively generating readings that will be displayed to the operator at their request. If, however, the system detects abnormal readings through its sensors (high temperature/significant movement), the system will generate an alert, which will be presented to the operator. It will be up to the operator's best judgement whether to act on the alerts presented.

3.4 LiDAR Visualization Software

When it comes to available open-source software that can create real-time LiDAR maps of a sensor's surroundings, there are few options. Based on our research, there are two good options for visualization software's that can help turn the LiDAR point cloud data into a map that is viewable by the vehicle's operator. These are LidarView and RVIZ.

3.5.1 LidarView: https://lidarview.kitware.com/

LidarView is an open-source platform built on top of the ParaView visualization engine that can perform real-time visualization and processing of 3D LiDAR data. Since it can produce a real time map of a LiDAR sensor's surroundings based off point cloud data from the sensor, it is a good choice for our application. The operator of the RC car will be able to watch the vehicle in real-time using LidarView, while also moving the car using its controller.

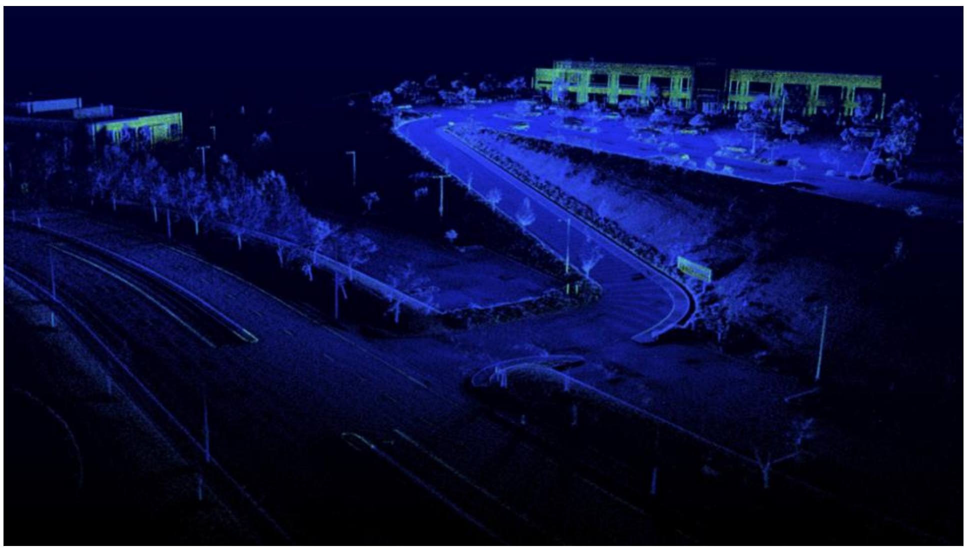


Figure 3.5.1.1 LiDAR Produced Map Using LidarView [11]

3.5.2 RVIZ and ROS: https://github.com/ros-visualization/rviz

RVIZ is a graphical interface that allows real-time visualization of sensor data including LiDAR sensor data. RVIZ operates on the Robot Operating System (ROS) framework, an open-source software used widely in the robotics industry among hobbyists and professionals. Compared to LidarView, RVIZ may be a good option for our project because of its community support and documentation online. RVIZ also comes with ROS, which could be useful for other aspects of the project's implementation that we have not yet planned for.

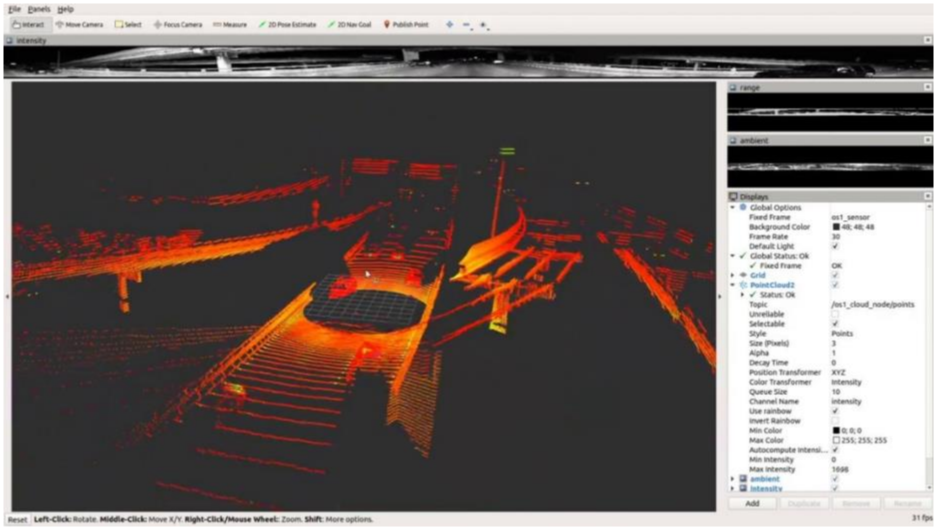


Figure 3.5.2.1 LiDAR Produced Map Using RVIZ [12]

3.6 Safety Considerations in Design

Considering how the primary goal of our RC car is for it to operate in disaster zones, there are many safety aspects of the design that should be taken into consideration. One important consideration is that the final design of the vehicle should be portable and easily deployed. This is important as to not impede ongoing rescue efforts or endanger rescue personnel working in the affected areas. Additionally, to increase safety and visibility in chaotic environments, the car will be equipped with hazard LEDs. In case the vehicle ends up in a compromising position, or is damaged such that its mobility is compromised, the controller on the central computer will have an emergency shutdown button that will halt all power to the vehicle. Furthermore, to avoid accidents and injuries during deployment, the car will have a physical ON/OFF switch as well as a software power-up button. The vehicle will not start unless both have been toggled on. This provides two layers of redundancy and will hopefully reduce the possibility of accidental power-ups. Finally, to ensure a basic level of safety, the vehicle will adhere to all Canadian IEEE and CSA safety standards for electrical and material safety.

4. Proof of Concept

4.1 LIDAR 3D Simulation

The purpose of the lidar simulation is to replicate the functionality of a real-world LIDAR sensor within the controlled environment.

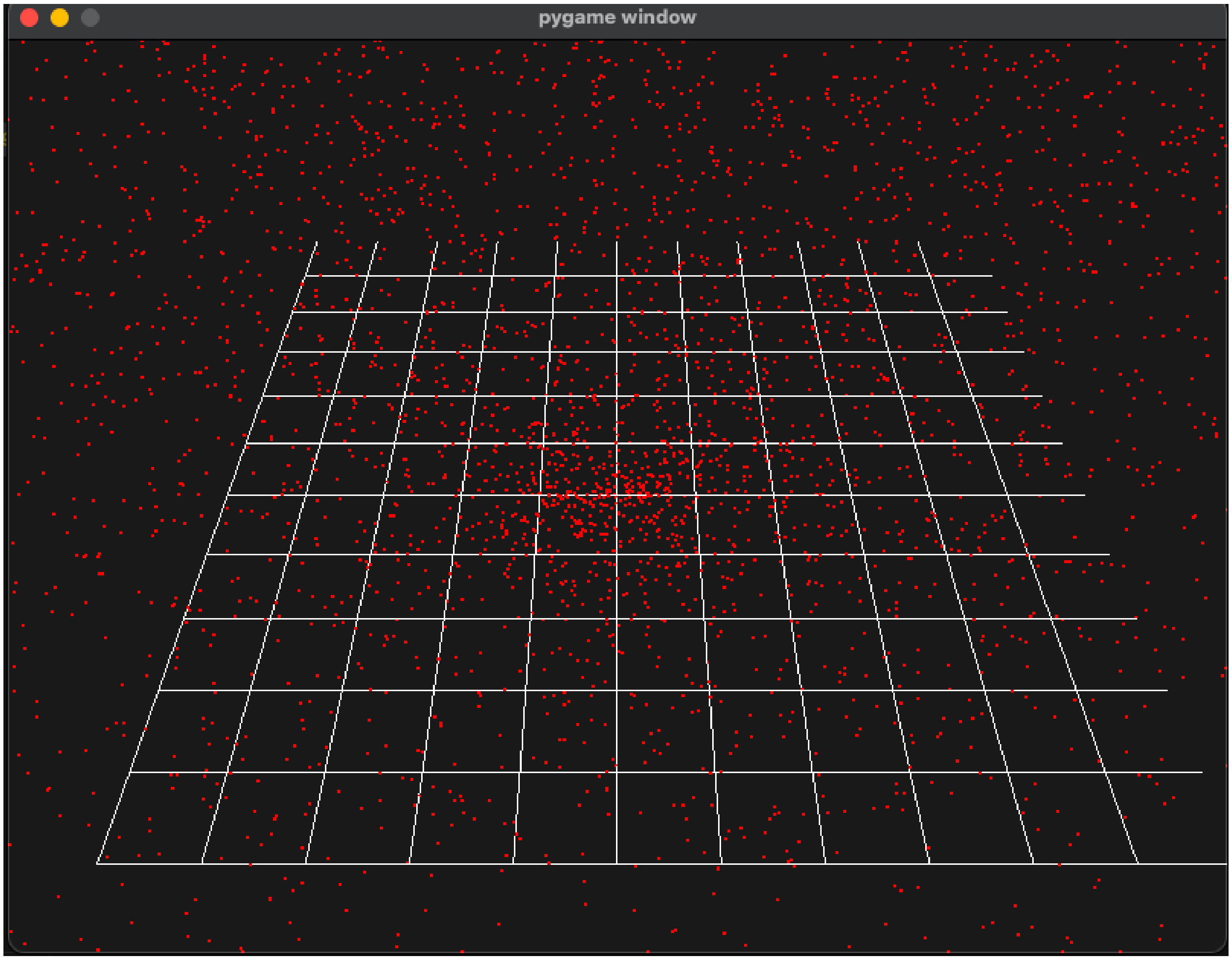


Figure 4.1.1: LiDAR representation1 Mapping Using Python in 3D

The LiDAR simulation presented in figure 4.1.1 is a representation of a 3D scanning process mapped onto a 2D plane, utilizing a graphical library, potentially Pygame, for visualization. The simulation commences with build_environment_3d.py, constructing a virtual landscape dotted with various structures such as buildings and trees. Utilizing laser_sensor_3d.py, a LiDAR sensor model comes into play, projecting laser beams at multifarious angles to capture environmental reflections. As the simulation progresses, each beam's contact point is calculated to determine distances, mirroring the physical behavior of light return times to the sensor. These points are depicted as red dots, whose spatial distribution helps infer physical attributes of the environment. The simulation accounts for the sensor's range, omitting detections beyond its field of view. The resulting data, a dense point cloud, encapsulates the scanned environment's detailed geometries, offering a valuable dataset for post-simulation analysis. Such analysis evaluates the sensor's fidelity and, by comparing with the actual environment setup, aids in refining the model's precision. The simulated scenario can be enhanced by integrating dynamic elements or varying atmospheric conditions, augmenting the simulation's complexity and real-world applicability.

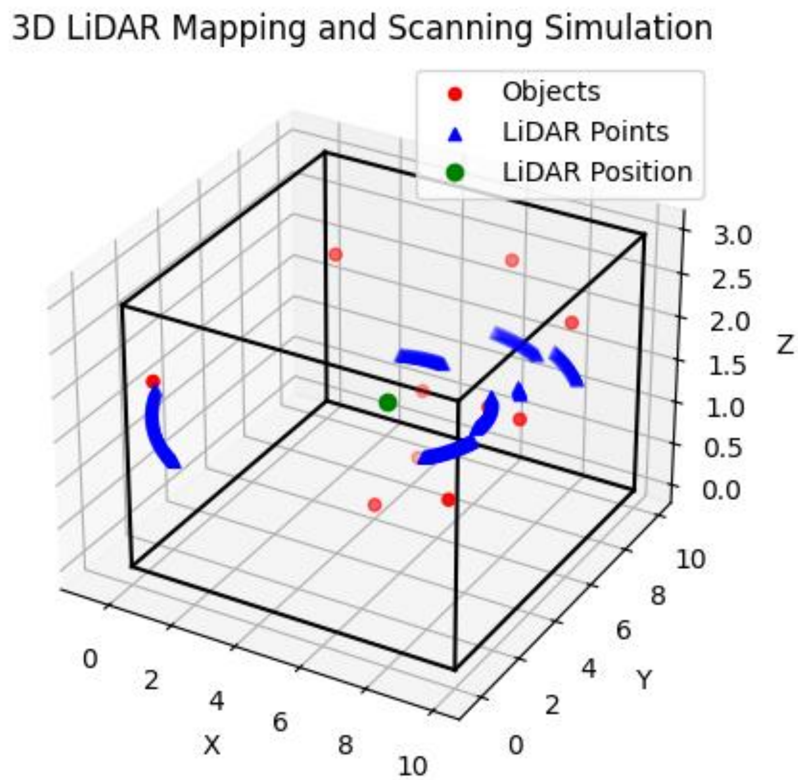


Figure 4.1.2: LiDAR representation2 Mapping Using Python in 3D

The simulation in figure 4.1.2 presents a digital arena for LiDAR exploration, charted along the tridimensional axes of X, Y, and Z. The gridded overlay serves as both a reference and boundary, marking the farthest reaches of LiDAR mapping. The green beacon, the LiDAR sensor, is at the center of the domain, positioned as the omniscient eye. The surrounding red markers represent tangible entities within the space, such as objects with form and volume. Blue triangular indicators, each a precise coordinate, indicate where the sensor's emitted beams intersected with the surfaces of these objects, reflecting a narrative of distance and dimension. These markers collectively form a detailed topographical lexicon, translating physical reality into a digitized understanding of presence, position, and spatial occupancy. The simulation's virtual confines serve as a reference and boundary for LiDAR mapping.

Result read from the 3D simulation.

```
(base) fatmahbayrli@Fatmahs-MacBook-Air Desktop % python LIDAR.py
LiDAR Position:
(5, 5, 1.5)

Objects' Positions (X, Y, Z):
[8.95939045 1.2756719  0.94875895]
[4.74078671 0.92804389 1.20656035]
[4.84472318 4.72902678 0.37429055]
[8.07514009 5.33251596 2.8923884 ]
[8.1481287  4.2851141  1.12411523]
[3.33169793 2.78677502 2.87380453]
[5.44507916 3.66281398 2.39980843]
[4.59370187 8.99188551 2.62871982]
[9.99708744 7.26589024 1.50360463]
[1.16588086 4.35793715 0.98862165]

LiDAR Scan Points (X, Y, Z):
[8.250087 5.       1.5     ]
[8.24958924 5.05687982 1.5       ]
[8.2480961  5.11374222 1.5       ]
[8.24560803 5.17056978 1.5       ]
[8.24212581 5.2273451  1.5       ]
[10.44569065  5.670528    1.5       ]
[10.4331217   5.76573043  1.5       ]
[10.41888853  5.8606983   1.5       ]
[10.40299551  5.95540253  1.5       ]
[10.3854475   6.04981411  1.5       ]
[10.36624988  6.14390413  1.5       ]
[10.34540854  6.23764376  1.5       ]
[10.32292984  6.33100428  1.5       ]
```

Figure 4.1.3: LiDAR Results of Data in 3D

The sensor's multidimensional detection capability is demonstrated in Figure 4.1.3 through distinct object positions with X, Y, and Z coordinates. The LiDAR scan points provide a nuanced depiction of the scanned environment, revealing spatial layout and potential material characteristics. The consistent '1.5' values in the Z-axis indicate a well-calibrated sensor operating at a fixed elevation. The varied intensity readings provide insights into surface reflectivity and distance-related signal attenuation, crucial factors for interpreting the LiDAR data's fidelity. These results demonstrate successful scanning operation, promising practical applications in autonomous navigation systems.

The integration of LIDAR technology into a Python-driven search and rescue RC car requires strategic positioning of the LIDAR unit on a raised platform for a comprehensive view of the vehicle's environment. Python's libraries, including PyLidar, laspy, and pcl-python, will be used for processing complex LIDAR data, enabling the execution of SLAM algorithms for accurate real-time maps. Python visualization tools will be used for 3D mapping, constructing detailed environmental models for the RC car.

Looking ahead, our plan is to develop a system using ROS to create a robust framework. Python's flexibility and modularity make it a strong choice. The goal is to become proficient in ROS packages like mapping or cartographer for SLAM implementation and optimize the synergy between LIDAR hardware and RC car software, laying a solid foundation for the project's success.

4.2 Ultrasonic Sensor Simulation

Ultrasonic sensor simulations are crucial for autonomous obstacle detection and traversal systems. They recreate the real-world function of an ultrasonic sensor, allowing for improved sensor accuracy, calibration of detection algorithms, and optimization of robot reactions to external stimuli. These simulations help to determine distances and potential threats in the real world.

The simulation process involves a robot emitting ultrasonic beams that detect objects and reflect to a sensor. The sensor calculates the distance to each obstacle based on the time lapse of the beam's return. The data is converted into a navigable map, allowing the robot's pathfinding algorithm to adjust its trajectory, avoid collisions, and move towards its intended target.

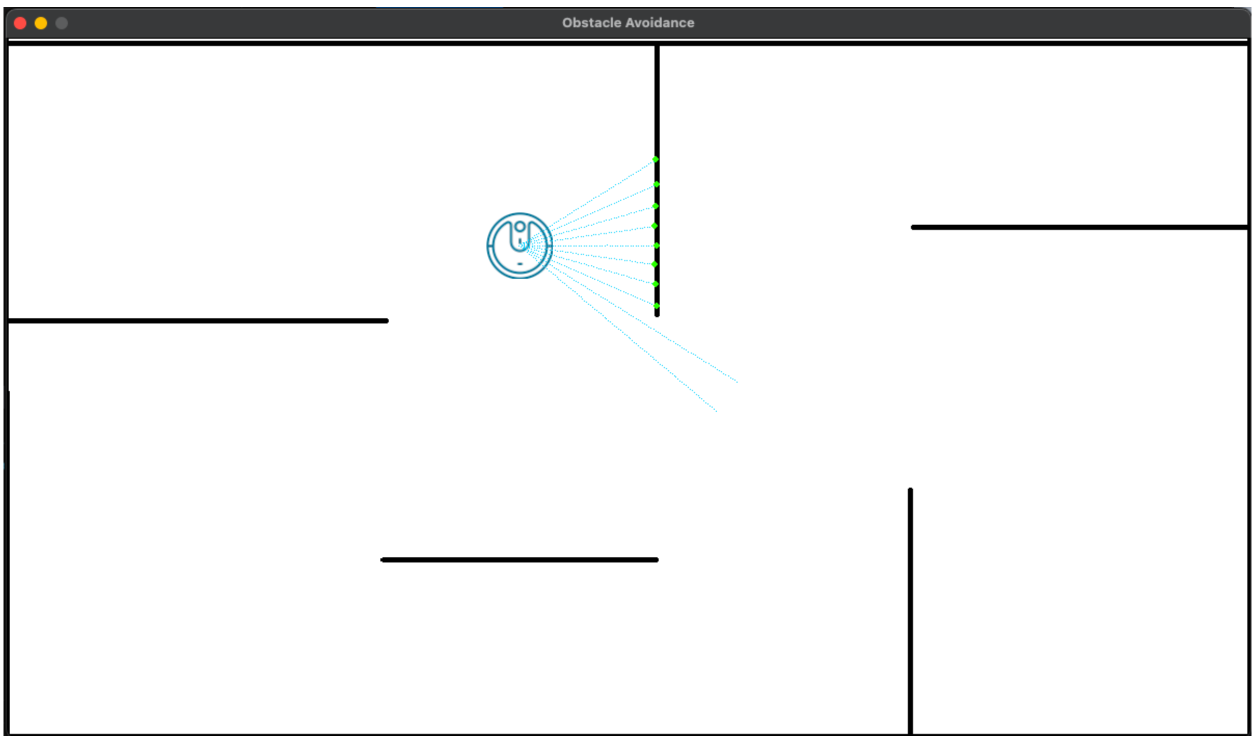


Figure 4.2.1 : Visual representation of the ultrasonic sensor simulation

The ultrasonic sensor, mounted at a lower height on the vehicle, serves as the first line of defense against potential hazards. It sends out sound pulses and calculates the time taken for echoes to bounce back, enabling the sensor to deliver accurate spatial data quickly. This data is crucial for the RC car's swift response, especially in search and rescue sites where every second counts. The sensor should be positioned at the front center of the vehicle, covering the most critical and widest field of view. To enhance its effectiveness, the sensor can be mounted on a servo motor, allowing it to pivot side to side, creating a wider sensory net without requiring multiple sensors. The collected data can be processed to halt the car before impact and inform it to turn away from obstructions, effectively navigating by continuously adapting to detected environmental contours. This agility is essential in dynamic and unpredictable terrains where every second counts and avoiding obstacles is crucial for mission success.

Here you can find the demo :
https://uottawa-my.sharepoint.com/personal/fbayr098_uottawa_ca/Documents/Winter2024/ELG%204912-%20Electrical%20%20Engineering%20Project_/ELG4912/Screen%20Recording%202024-04-08%20at%204.00.27%20PM.mov?csf=1&web=1&e=UfsyVA

4.3 Long Wave Infrared Camera Simulation

4.3.1 Simterm

The primary purpose of the infrared camera in our project is to more easily spot victims that are buried, or partially buried under rubble and may not be easy to spot for a regular camera. This proof of concept uses Simterm which is a simulation software developed by Inframet with the purpose of enabling generation of thermal images in different environments. Figures 4.3.1.1 to 4.3.1.3 depict images of a person in three different environments of varying detection difficulty for an infrared camera.

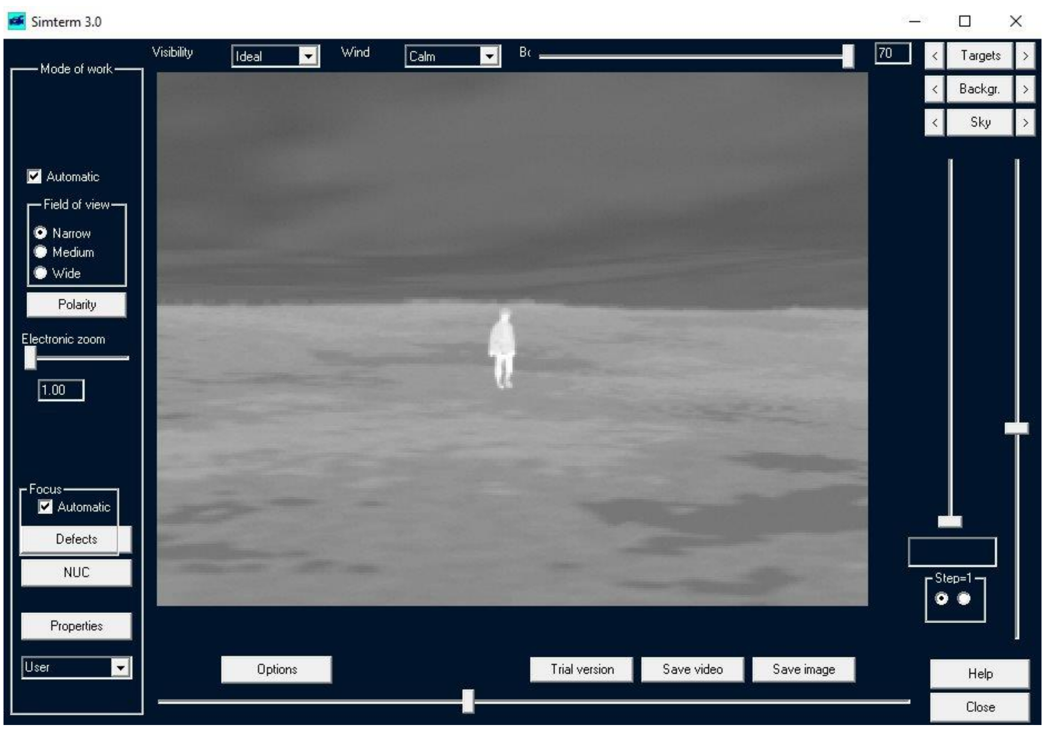


Figure 4.3.1.1 Simterm Image Depicting an Infrared Camera's View of a Person in Ideal Conditions

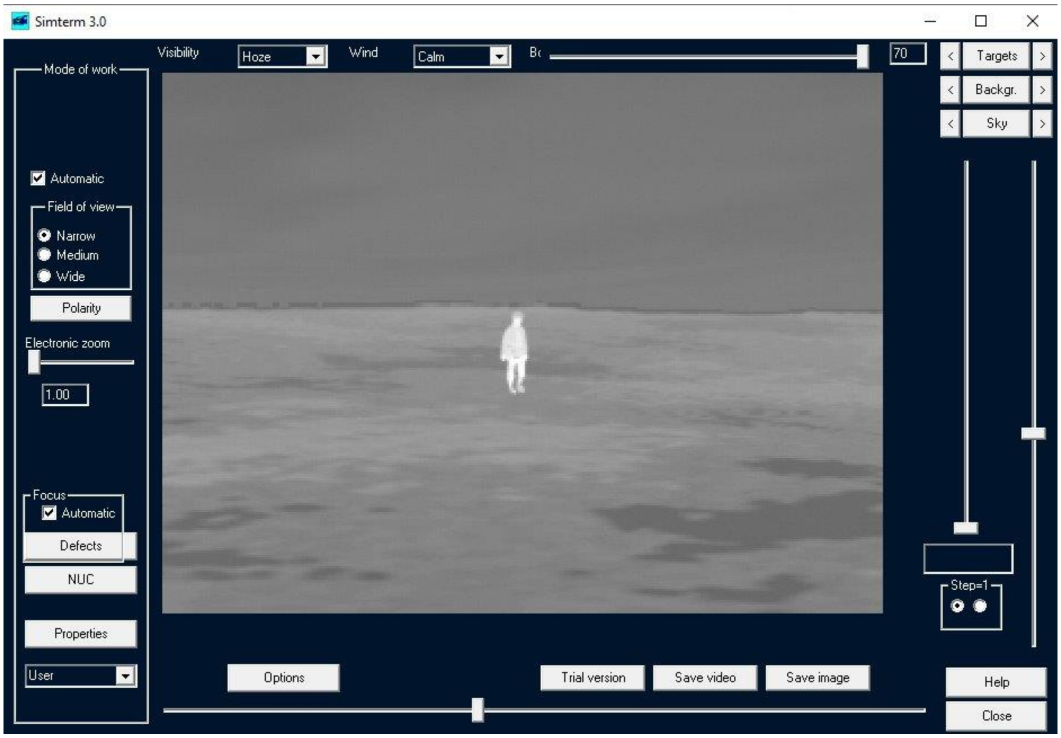


Figure 4.3.1.2 Simterm Image Depicting an Infrared Camera's View of a Person in Hazy Conditions

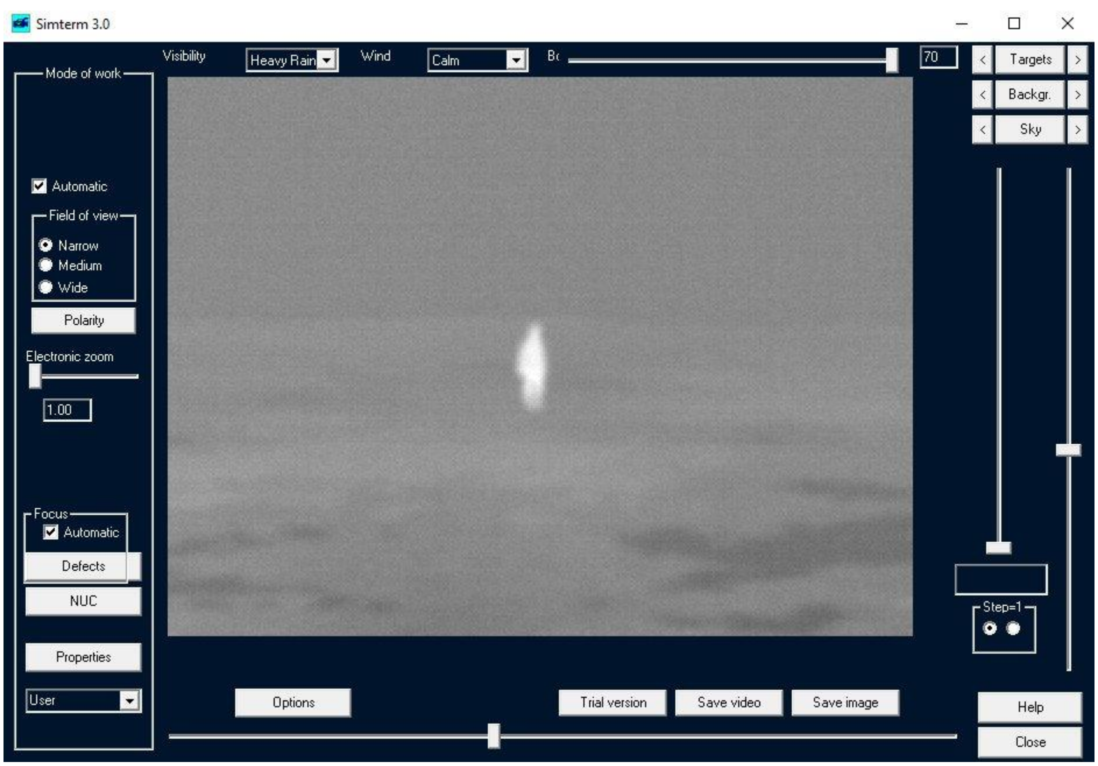


Figure 4.3.1.3 Simterm Image Depicting an Infrared Camera's View of a Person in Heavy Rain

Figure 4.3.1.1 represents how a thermal camera would detect a person in ideal conditions, *Figure 4.3.1.2* represents a thermal camera's view in hazy conditions, and *Figure 4.3.1.3* shows its view in heavy rain. For our application, the conditions in a disaster zone after an earthquake would be closest to haze, as most of the dust hanging in the air would consist of solid particles. *Figure 4.3.1.2* clearly shows that under these sorts of conditions the target is very visible. Even so, a more rigorous option could be the view of a person in heavy rain as water is significantly more difficult for infrared signals to travel through than dust and other solid particles. As *Figure 4.3.1.*3 shows, while it is slightly harder to view the person in heavy rain than in the previous scenarios, it is still relatively easy to distinguish the person from the surroundings. Overall, this simulation provides encouraging results in the way of detecting life using a thermal camera in disaster scenarios.

Simterm and other software by Inframet can be found at:
https://www.inframet.com/computer_simulators.htm

4.3.2 FLIR Lepton Python Image Processing

The second simulation for the thermal camera is much more targeted towards the specific model of camera we intend to include in our final design. This simulation uses python programming to map the data produced by a FLIR Lepton infrared camera to a heatmap where the temperatures are displayed on a spectrum from darker red (for hotter temperatures) to darker blue (for colder temperatures).

The output of the FLIR Lepton camera consists of raw data in the format displayed in the text document of *Figure 4.3.2.1*. Each number corresponds to a temperature in Celsius that is detected by one of the camera's pixels. Since the FLIR Lepton's resolution is 80x60, this means that there are 4800 temperature values that make up each image. The bulk of the image processing python code for this simulation was found online (credit to KhaledSaleh who provided open-source code on GitHub) and then modified since the original code was meant to process two FLIR Lepton images simultaneously at a resolution of 80x120. The code used in this simulation processes data as formatted in *Figure 4.3.2.1* and *Figure 4.3.2.3*, and then produces an image like in *Figure 4.3.2.2* and *Figure 4.3.2.4*.

Since we did not have an actual FLIR Lepton camera, we were not able to produce images based on real camera data. To account for this problem, we wrote another Python script intended to generate a text file as in figures 4.3.2.1 and 4.3.2.3 that include a random set of 4800 numbers in the range of -10 to 50 degrees Celsius that also follow Gaussian distribution. This meant that we could get an image with the full gradient of colour which more accurately represents what a real thermal image will look like.

thermal_image_dataset_1.txt - Notepad

File Edit Format View Help


```
[
  {
    "dataValues": "-10 -10 -10 -10 -10 -10 -10 -10 -9 -9 -9 -9 -9 -9 -9 -
-9 -9 -9 -8 -8 -8 -8 -7 -7 -7 -7 -6 -6 -6 -5 -5 -4 -4 -3 -3 -3 -2 -2 -1 -
8 -9 -9 -9 -9 -9 -9 -8 -8 -8 -8 -7 -7 -7 -6 -6 -5 -5 -4 -3 -3 -2 -1 -1 0
6 15 14 13 12 11 10 8 7 6 5 4 3 2 1 0 -1 -1 -2 -3 -4 -4 -5 -5 -6 -9 -9 -9
16 18 20 21 23 25 26 27 29 30 31 32 33 34 35 35 36 36 36 36 36 35 35 34 3
2 10 9 7 6 5 3 2 1 0 -1 -2 -8 -8 -8 -7 -7 -6 -6 -5 -4 -3 -2 -2 0 1 2 3 5
42 44 45 46 48 48 49 50 50 50 50 50 49 49 48 47 46 44 43 41 40 38 36 34 3
1 0 -1 -8 -8 -7 -7 -7 -6 -5 -5 -4 -3 -2 -1 0 1 2 3 5 6 8 10 11 13 15 17 1
38 39 39 39 39 38 38 37 36 35 34 33 32 31 29 28 26 25 23 22 20 19 17 15 1
6 -5 -5 -4 -3 -2 -2 -1 0 1 2 3 4 6 7 8 9 10 12 13 14 15 17 18 19 20 21 22
4 3 2 1 0 0 -1 -2 -2 -3 -4 -4 -5 -5 -6 -6 -9 -9 -9 -9 -9 -9 -8 -8 -8 -7 -
2 -2 -3 -3 -4 -4 -4 -5 -5 -5 -6 -6 -7 -7 -7 -7 -8 -8 -10 -10 -10 -9 -9 -9
10 -9 -9 -9 -9 -9 -9 -9 -9 -8 -8 -8 -8 -8 -7 -7 -7 -7 -7 -6 -6 -6 -6 -5 -
10 -9 -9 -9 -9 -9 -9 -9 -9 -9 -9 -9 -8 -8 -8 -8 -8 -8 -8 -8 -8 -7 -7 -7 -
  }
]
```


Figure 4.3.2.1 Example Output Data of a FLIR Lepton Infrared Camera for a Spectrum of Temperatures Following Gaussian Distribution

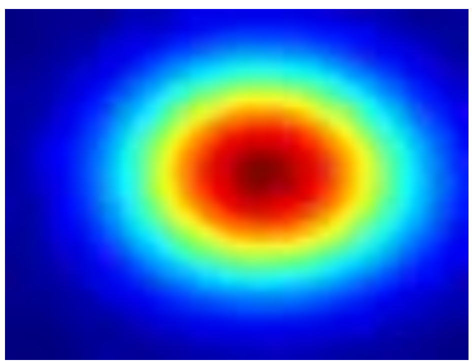
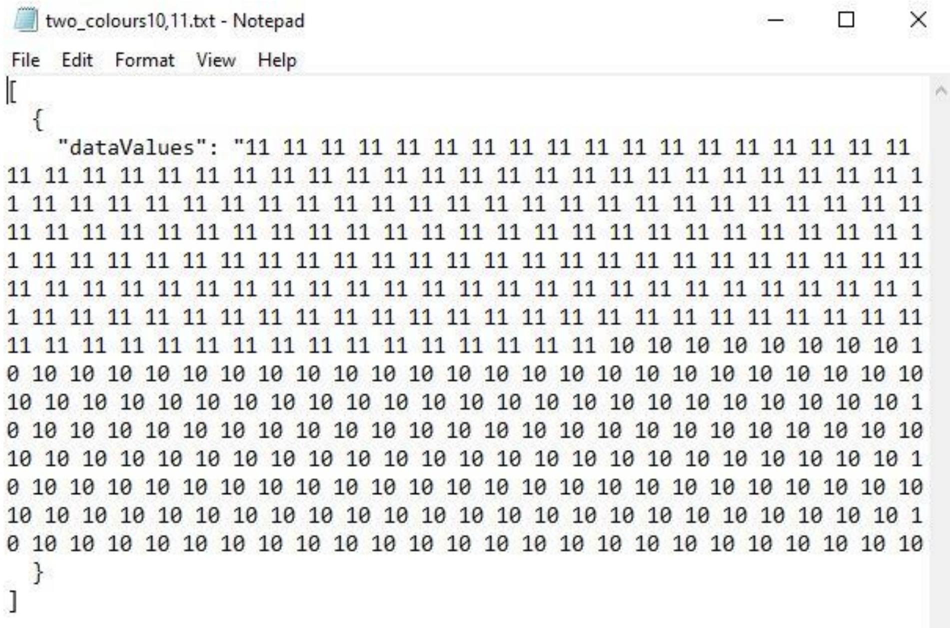


Figure 4.3.2.2 Example Output Thermal Image for a Dataset Including a Spectrum of Temperatures Follwoing Gaussian Distribution

Another feature of the Python code is that it automatically adjusts the colour scaling so the hottest pixel in the image is mapped to the darkest red, while the coldest pixel is mapped to the darkest blue. This ensures the best visibility for the temperatures in between. This feature is depicted in Figures 4.3.2.3 and 4.3.2.4, where the image simply consists of two temperatures one degree apart from each other. As is visible in *Figure 4.3.2.4*, ten degrees Celsius (the colder temperature) is immediately mapped to the darkest blue, while eleven degrees Celsius (the hotter temperature) is mapped to the darkest red.




```
[
  {
    "dataValues": "11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11
11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 1
1 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11
11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 1
1 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11
11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 1
1 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11
11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 10 10 10 10 10 10 10 10 1
0 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10
10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 1
0 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10
10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 1
0 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10
10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 1
0 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10
  }
]
```


Figure 4.3.2.3 Example Output Data of a FLIR Lepton Infrared Camera for Only Two Temperatures

Figure 4.3.2.4 Example Output Thermal Image for a Dataset of Only Two Temperatures

Another feature of the Python code is that it automatically adjusts the colour scaling so the hottest pixel in the image is mapped to the darkest red, while the coldest pixel is mapped to the darkest blue. This ensures the best visibility for the temperatures in between. This feature is depicted in Figures 4.3.2.3 and 4.3.2.4, where the image simply consists of two temperatures one degree apart from each other. As is visible in *Figure 4.3.2.4*, ten degrees Celsius (the colder temperature) is immediately mapped to the darkest blue, while eleven degrees Celsius (the hotter temperature) is mapped to the darkest red.

Overall, implementing the actual long-wave infrared camera after this simulation will be very simple since we will process the outputted camera data using the exact same python script. The sole difference is that we will not need to also run the secondary Python script to generate the data. Credit for the original code for the image processing portion of this simulation goes to KhaledSaleh. The GitHub repository can be found at https://github.com/KhaledSaleh/FLIR-human-detection.

4.4 SM-24 Geophone Simulation

One of the most crucial parts of our design's life detection system is the SM-24 geophone. A geophone is a simple device that allows the detection of underground vibrations. On the inside of a geophone there is a spring-mounted coil that is free to move around a magnet. This is pictured in *Figure 4.4.1*. As vibrations in the earth oscillate the coil around the magnet, changing magnetic fields are produced around the coil and consequently also voltages within the coil. These voltages can be measured and plotted as in *Figure 4.4.3*. This is the primary method of data collection used by a geophone.

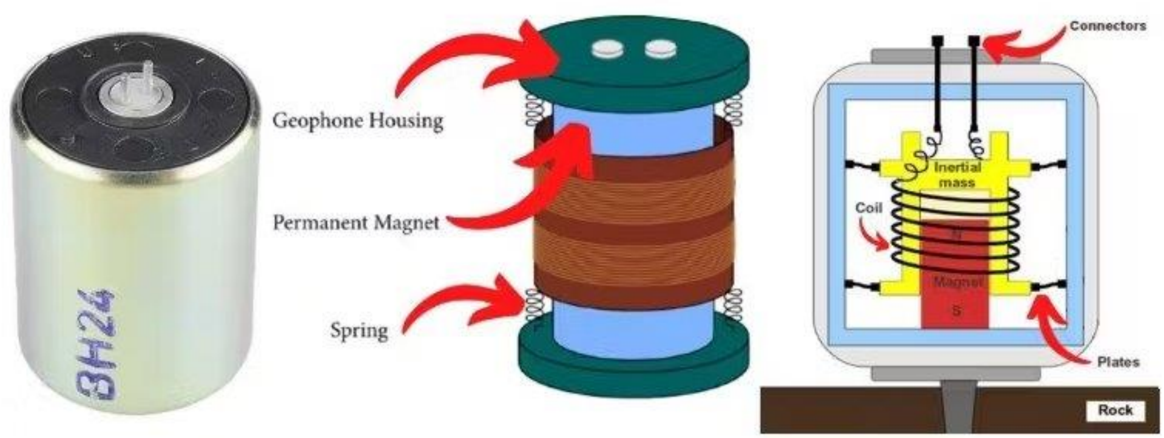


Figure 4.4.1 Interior of a Geophone

This simulation is another simulation tied specifically to the model of geophone we intend to use in our final design. It consists of plotting randomly generated data in the format that the SM-24 geophone would output, over time. For this simulation we were very fortunate to have found a project online that provided open-source Python code for reading data from the SM-24 geophone over a Raspberry Pi's I2C line and then also plotting it. That being said, since we do not have an actual geophone to gather data from, we modified this code to change the input from the I2C line to a very simple function that generates random passive readings between -1 and 1mV with a 0.5% chance of creating a voltage spike. These voltage spikes are meant to simulate movement detected by the geophone. This function is depicted in *Figure 4.4.2*. The result of the simulation is a constantly updating stream of geophone data, a snapshot of which has been included in *Figure 4.4.3*.

```
# Function to generate random voltage values, occasionally simulating spikes
def generate_voltage():
    if random.random() < 0.005:  # 0.5% chance to generate a spike
        return random.uniform(-50, 50)  # Spike voltage between -50mV and +50mV
    else:
        return random.uniform(-1, 1)  # Regular voltage between -1mV and +1mV
```

Figure 4.4.2 Example Output Thermal Image for a Dataset of Only Two Temperatures

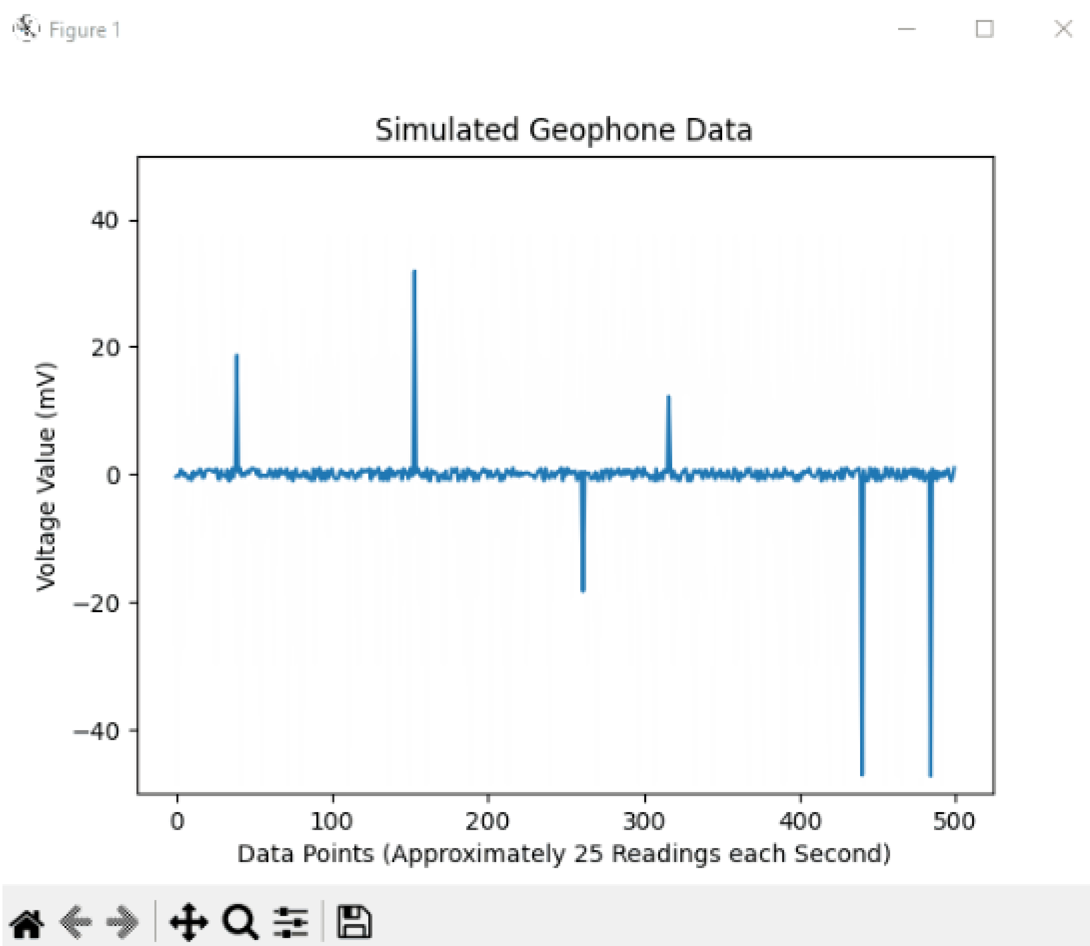


Figure 4.4.3 Example Output Thermal Image for a Dataset of Only Two Temperatures

Next semester, when we begin integrating the actual sensor, this code will be extremely useful for directly reading the geophone's data. The only difference between the code used in this simulation and the code we will eventually implement in our final design is that we will receive the data over the I2C line instead of randomly generating it using the function in *Figure 4.4.2*. Credit for the original code for this simulation goes to Core Electronics. The source code for this project can be found at https://core-electronics.com.au/guides/raspberry-pi/geophone-raspberry-pi/.

4.5 First 3D CAD Model - Prototype

To better visualize an early idea of the prototype's physical design, A CAD modelization of component placement is created using Onshape and can be viewed at the following link:

https://cad.onshape.com/documents/fb32e11b8d84719f547e33a7/w/6dca8c35242ddf8d58317f99/e/9b5a9b049bd657a6c0488e62?renderMode=0&uiState=6615c1c6ccdad5262dccb164

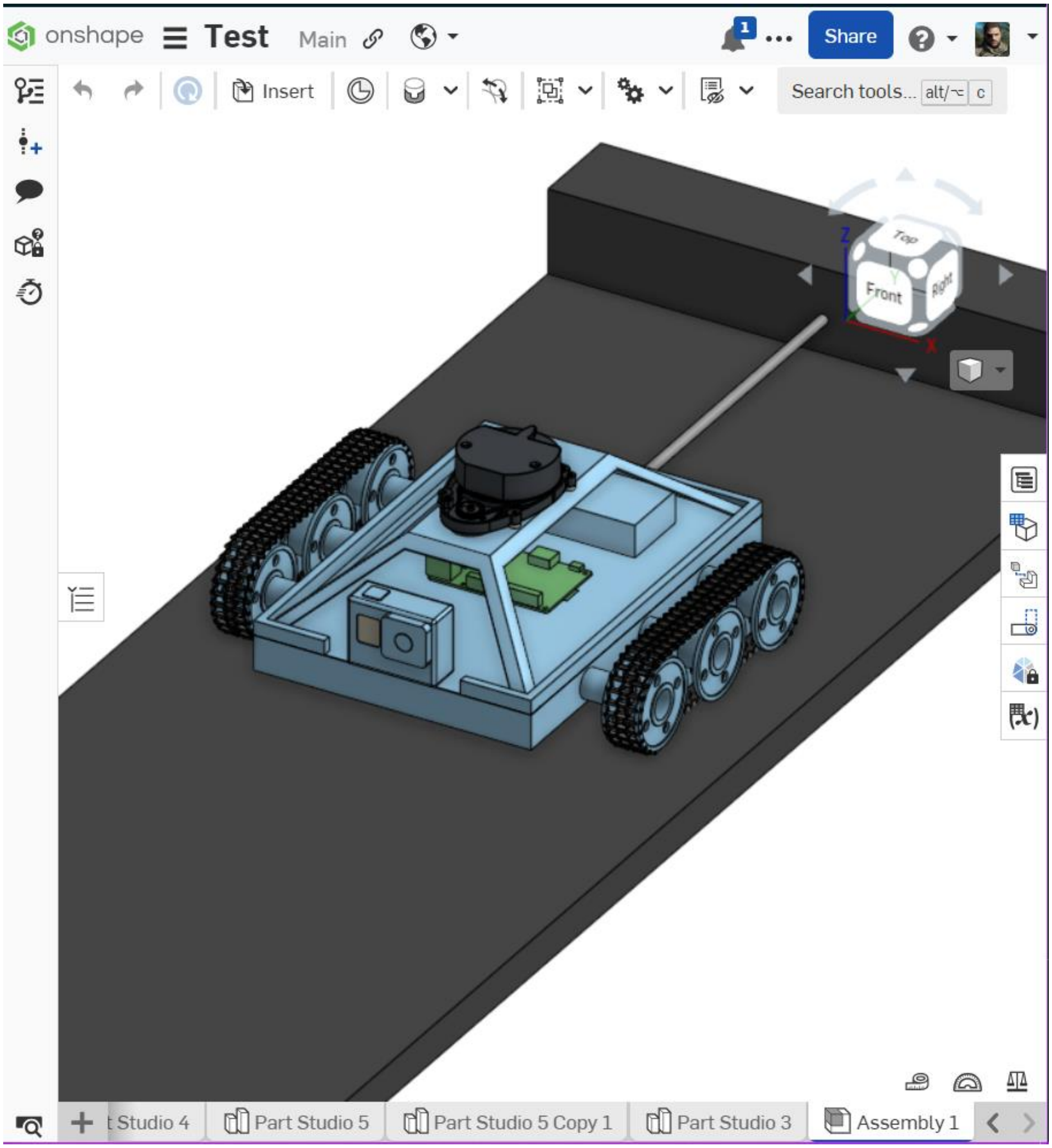


Figure 4.5.1 First 3D CAD Prototype

The model is partially equipped with the components needed and is conceived to accommodate instruments which could obstruct each other.

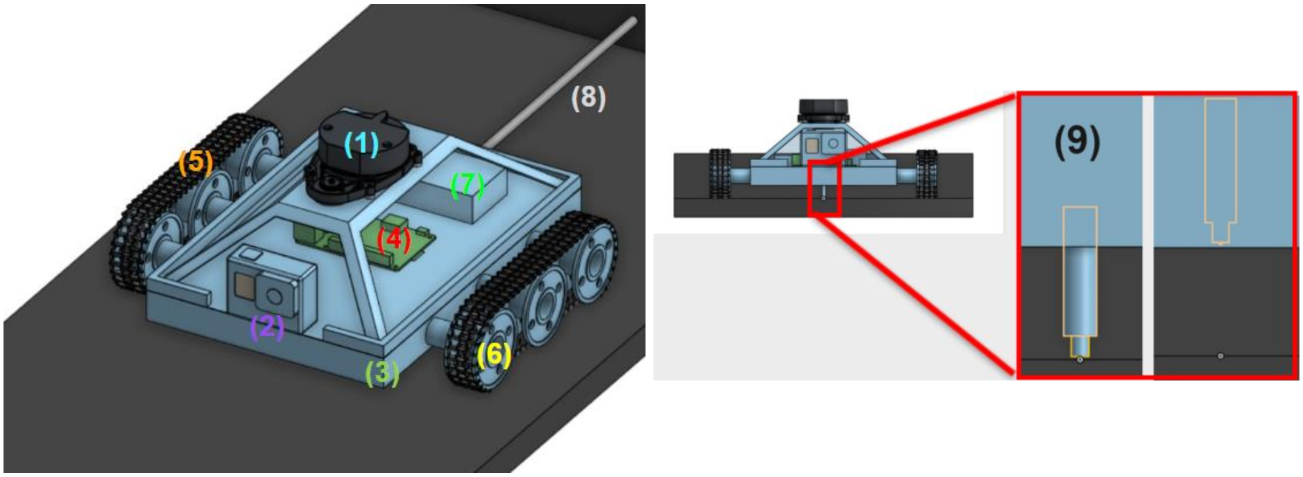


Figure 4.5.2 First 3D CAD Prototype – Components Breakdown

- **(1) – LIDAR**
- **(2) – LWIR Thermal Imaging Camera**
- **(3) – Chassis**
- **(4) – Raspberry Pi**
- **(5) – Tank Tracks**
- **(6) – Wheels**
- **(7) – Power Components**
- **(8) – Ethernet Cable**
- **(9) – Geophone**

The geophone component slides up and down using a sliding mechanism to probe the floor level for vibrations. The chassis mounts the LIDAR on top to offer an unperturbed sweep of the surroundings, while the LWIR camera can scan ahead. The prototype is based on the existing acrylic board in Figure 3.2.1.8.1.

5. Test Plan

Introduction:

Our team is developing a remote-controlled car with cutting-edge sensor technologies in response to the pressing demand for efficient search and rescue operations in catastrophe scenarios. The goal of this research is to solve the difficulties emergency responders encounter when entering and evaluating disaster-affected areas, especially when conventional approaches are insufficient. Our objective is to develop a flexible and dependable instrument that can identify life signs, navigate through intricate settings, and expedite rescue operations by utilizing several sensor technologies such as LiDAR, thermal imaging, and geophone. Our goal is to improve the efficacy of disaster response operations by guaranteeing the robustness, functionality, and integration of our remote-controlled car through meticulous testing and validation.

10. **Functionality Test**:
    - **Case 1: Basic Car Functionality**:
        - **Simulation**:
            - Utilize Python code to simulate car movements in different directions.
            - Verify responsiveness to commands and accuracy of movements.
        - **Hardware Test**:
            - Examine the movement of the remote-controlled car in a safe setting.
            - Analyze how well it responds to instructions.
        - **Expected Result**: The car moves accurately and responsively as commanded.
    - **Case 2: LiDAR Technology Test**:
        - **Simulation**:
            - Use simulated LiDAR data to validate obstacle detection and mapping algorithms.
            - Ensure accuracy in identifying obstacles and generating a 3D environmental map.
        - **Hardware Test**:
            - Perform LiDAR scans in a regulated setting with pre-established barriers.
            - Check the precision of the mapping and obstacle detection.
        - **Expected Result**: LiDAR accurately detects obstacles and generates a comprehensive environmental map.
11. **Sensor Integration Test**:
    - **Case 1: Integration Verification**:
        - **Simulation**:
            - Simulate the integration of LiDAR, vibration sensors, and thermal cameras using Python scripts.
            - Validate seamless data interpretation and decision-making capabilities.
        - **Hardware Test**:
            - Integrate sensors into the remote-controlled car and conduct tests in a controlled environment.
            - Confirm efficient data interpretation and decision-making using a combination of sensor data.
        - **Expected Result**: Sensors work together seamlessly, providing accurate data interpretation and decision-making.

12. **Navigation Test**:

- **Case 1: Route Planning and Obstacle Avoidance**:
  - **Simulation**:
    - Use simulated rubble and obstacles to test the car's navigation algorithms.
    - Evaluate the accuracy of route planning and obstacle avoidance.
    - Use a python script to detect obstacle :

```python
# Parameters
room_size = (10, 10, 3)  # dimensions of the room in meters (width, length, height)
num_objects = 10  # number of objects to place in the room
lidar_position = (5, 5, 1.5)  # position of the LiDAR in the room
num_points = 360  # number of points in one rotation of the LiDAR

# Generate random objects in the room
objects = np.random.rand(num_objects, 3) * room_size

# Room corners (for mapping the room)
corners = np.array([[0, 0, 0], [room_size[0], 0, 0], [room_size[0], room_size[1], 0],
                    [0, 0, room_size[2]], [room_size[0], 0, room_size[2]], [room_size[

# Simulate LiDAR scan
angles = np.linspace(0, 2*np.pi, num_points)
scan_points = []
```

  - **Hardware Test**:
    - Navigate the car through a simulated obstacle course.
    - Assess the accuracy of route planning and obstacle avoidance algorithms.
  - **Expected Result**: The car navigates through obstacles accurately and avoids collisions.

13. **Thermal Imaging Test**:

- **Case 1: Life Sign Detection**:
  - **Simulation**:
    - Simulate thermal imaging data in various environmental conditions.
    - Verify the thermal camera's ability to detect signs of life.
  - **Hardware Test**:
    - Test the thermal imaging camera's performance in detecting heat signatures in controlled environments.
    - Verify the accuracy of temperature readings and heatmaps.
  - **Expected Result**: Thermal imaging camera accurately detects signs of life and provides accurate temperature readings.

14. **Geophone Test**:

- **Case 1: Vibration Detection**:
  - **Simulation**:

- Simulate seismic data using Python scripts to test the geophone's vibration detection capabilities.

```
# Function to generate random voltage values, occasionally simulating spikes
def generate_voltage():
    if random.random() < 0.005:  # 0.5% chance to generate a spike
        return random.uniform(-50, 50)  # Spike voltage between -50mV and +50mV
    else:
        return random.uniform(-1, 1)  # Regular voltage between -1mV and +1mV
```

- Validate sensitivity and reliability in detecting vibrations.

  - **Hardware Test**:
    - Test the geophone's performance in detecting simulated vibrations in various scenarios.
    - Evaluate sensitivity and reliability under different surface materials and noise conditions.
  - **Expected Result**: Geophone accurately detects vibrations, demonstrating sensitivity and reliability.

15. **Robustness Test**:
- **Case 1: Environmental Stress Test**:
  - **Simulation**:
    - Python scripts can be used to simulate harsh environmental conditions in order to evaluate the car's durability.
    - Verify its ability to withstand extreme temperatures, humidity, and vibrations.
  - **Hardware Test**:
    - Subject the remote-controlled car to extreme environmental conditions in a controlled setting.
    - Examine dependability and durability in challenging circumstances.
  - **Expected Result**: The car withstands environmental stressors without malfunctioning.

16. **Range and Connectivity Test**:
- **Case 1: Communication Reliability**:
  - **Simulation**:
    - Simulate remote control range and wireless communication reliability using Python scripts.
    - Validate reliable communication between the operator and the vehicle.
  - **Hardware Test**:
    - Examine the wireless connectivity and range of the remote control in different settings.
    - Analyze how well communication works in various interference scenarios.
  - **Expected Result**: Reliable communication is maintained between the operator and the vehicle under varying conditions.

By conducting these comprehensive tests, we aim to ensure the remote-controlled car's functionality, sensor integration, navigation capabilities, and reliability in real-world scenarios.

Conclusion:

To sum up, our remote-controlled car's development and testing reflect a major advancement in disaster response technology. Through the integration of cutting-edge sensor technology and resilient navigation algorithms, we have developed a flexible and dependable instrument that can effectively tackle the distinct difficulties presented by emergency situations. We have established the functionality, integration, and dependability of our system through extensive testing and validation, providing the framework for its implementation in actual emergency scenarios. We're still dedicated to improving and optimizing going forward in order to save lives and lessen the effects of disasters on impacted areas.

6. WBS, Schedule

6.1 Work Breakdown Structure (WBS)

The updated Work Breakdown Structure (WBS) for the project, which outlines the timeline and key milestones, has been organized into a clear and structured format. This structure is divided into three primary phases:

6.1.1 Project Initiation and Planning:

This initial phase, set to span two weeks, encompasses the foundational activities required to start the project. Key tasks include the official project kickoff, gathering of requirements, sketching of initial designs, and development of the project plan. These steps are critical for setting the project's direction and ensuring all team members are aligned with the project's goals and timelines.

6.1.2 Design and Development Phase:

Over three weeks, this phase focuses on the comprehensive design and preparation for the construction of the RC rescue car. It begins with the completion of detailed designs, including mechanical, electrical, and software components, evidenced by developed CAD models, electrical schematics, and software architecture plans. Following the design completion, the phase continues with the selection and procurement of necessary components, ensuring that each part meets specific performance criteria and budget considerations. The final part of this phase involves the development of control software crucial for the RC car's operation, integrating sensor data for enhanced navigation and life detection capabilities.

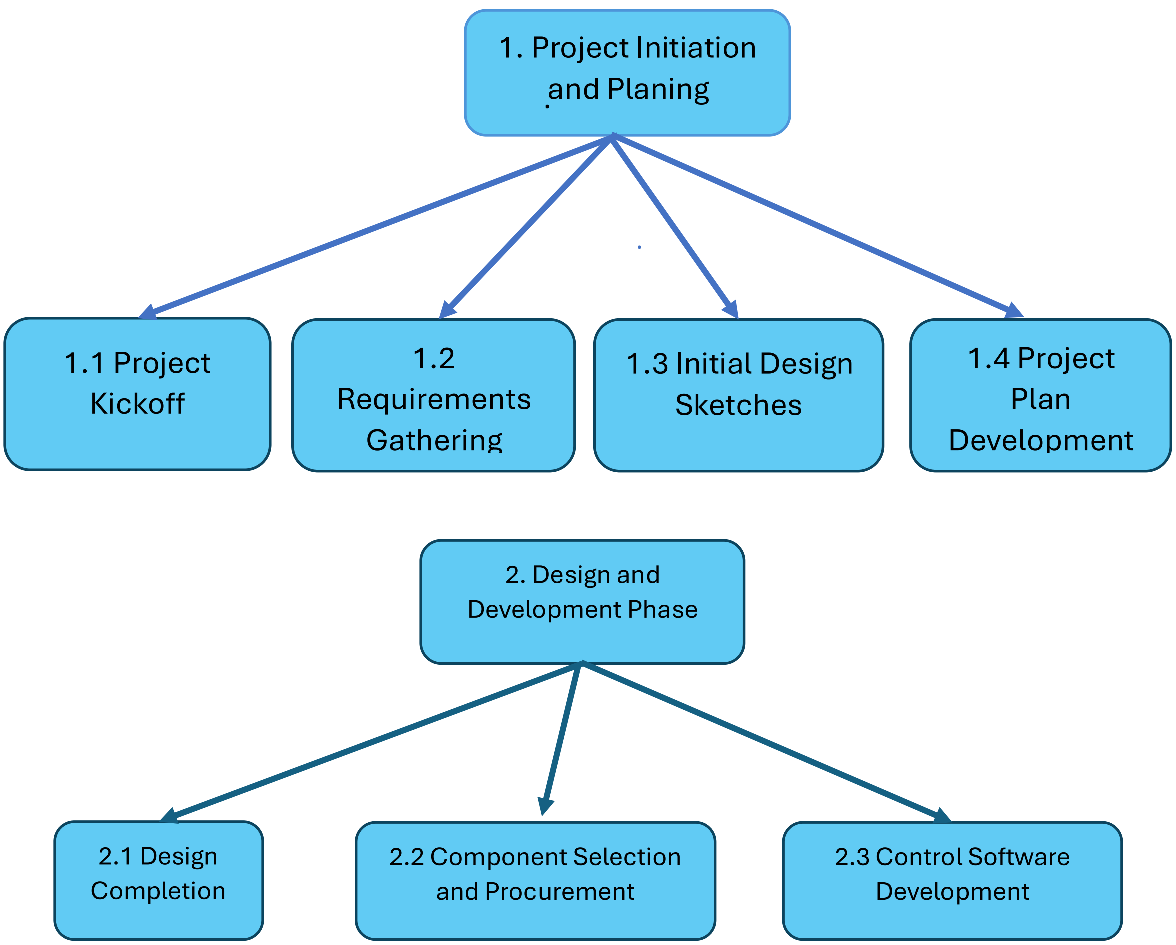
6.1.3 Implementation Phase:

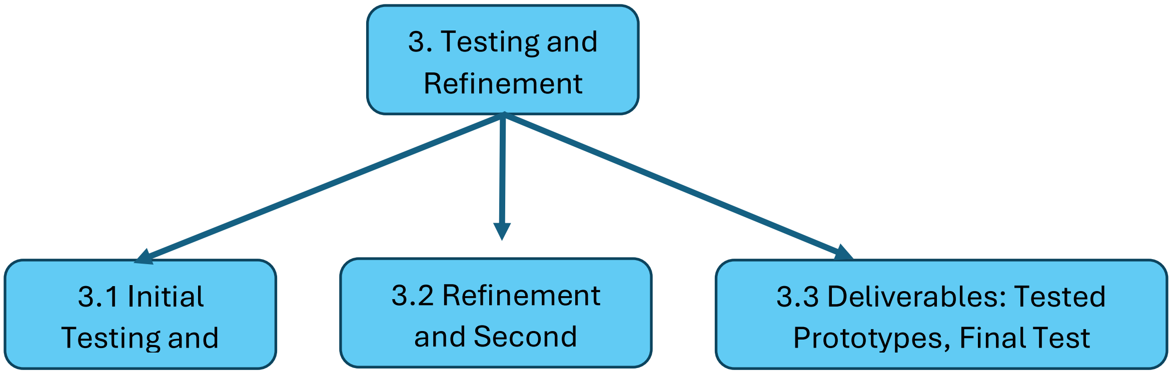
This phase should start around Sep 9, 2024, and be completed by Oct 21. This will be the first stage in which we will start to notice any serious design flaws, if any. We will be utilizing any tools necessary to assemble all the components, build the needed parts, and assemble the prototype.

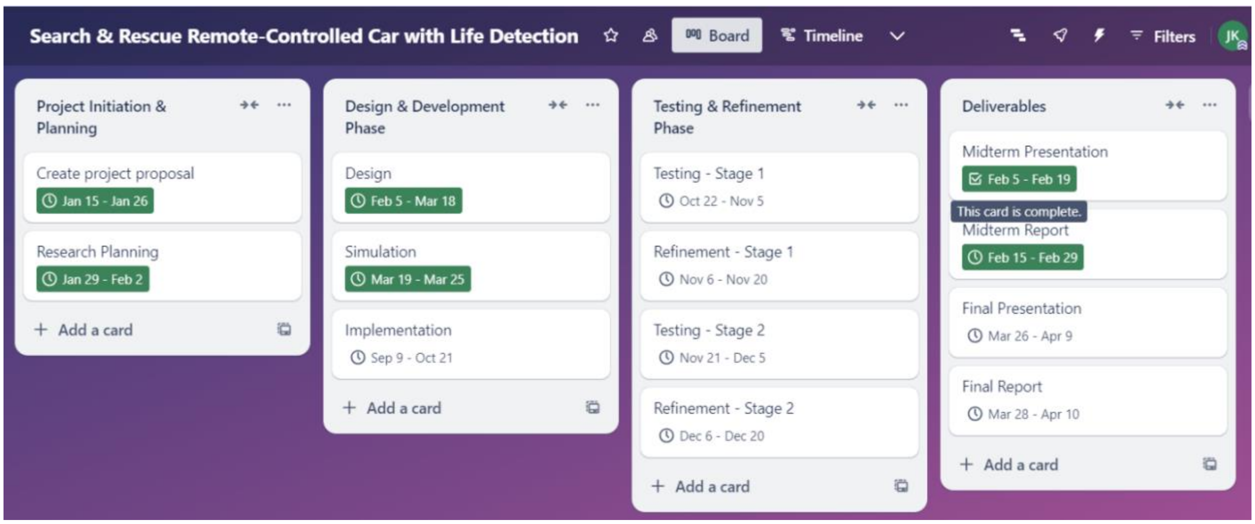
6.1.4 Testing and Refinement Phase:

Although the duration is not specified, this phase is pivotal for evaluating the RC rescue car's functionality and performance. Initial testing assesses navigation, life detection capabilities, and communication systems. Based on the results, the project moves into refinement, where the design undergoes necessary adjustments, and a second prototype is constructed. The phase concludes with detailed testing in simulated disaster environments, allowing for final adjustments before the project's completion.

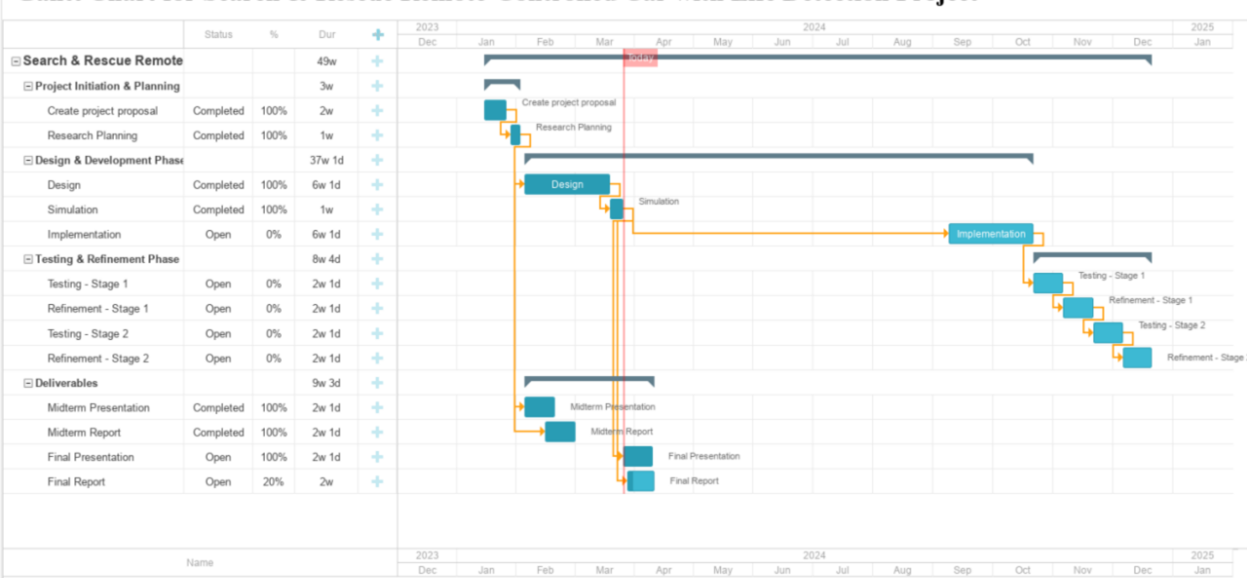
This structured breakdown into distinct phases and tasks facilitates clear project tracking and management, ensuring all objectives are met systematically and efficiently. The WBS document serves as a vital tool for understanding the project's scope and sequence, guiding team members through the project lifecycle from inception to completion.







Gantt Chart for Search & Rescue Remote-Controlled Car with Life Detection Project



	Status	%	Dur
Search & Rescue Remote			49w
Project Initiation & Planning			3w
Create project proposal	Completed	100%	2w
Research Planning	Completed	100%	1w
Design & Development Phase			37w 1d
Design	Completed	100%	6w 1d
Simulation	Completed	100%	1w
Implementation	Open	0%	6w 1d
Testing & Refinement Phase			8w 4d
Testing - Stage 1	Open	0%	2w 1d
Refinement - Stage 1	Open	0%	2w 1d
Testing - Stage 2	Open	0%	2w 1d
Refinement - Stage 2	Open	0%	2w 1d
Deliverables			9w 3d
Midterm Presentation	Completed	100%	2w 1d
Midterm Report	Completed	100%	2w 1d
Final Presentation	Open	100%	2w 1d
Final Report	Open	20%	2w

Group 5 | Exported from Placker.com on Mar 25th 20:28

6.2 Internal Releases:

The internal releases for the project delineate crucial checkpoints and deliverables critical for the development of the remote-controlled (RC) car. Here's a summary of each release:

6.2.1 Concept Design Release:

This initial phase aims to present the foundational designs, including preliminary sketches and basic CAD models, highlighting the layout, track system, sensor placements, and control system concepts. The goal is to obtain feedback on the design's feasibility and identify areas requiring further research or modification.

6.2.2 Detailed Design Release:

In this phase, the project moves forward with the presentation of detailed design documents and refined models. This includes comprehensive schematics and plans for all RC car components. The expected outcome is the approval of these detailed designs and the green light to commence component procurement.

6.2.3 Sensor Integration Design Review:

This release focuses on the integration of essential sensors into the RC car, ensuring optimal functionality and reliability. The objective is to confirm that the design meets operational requirements and to plan for effective data integration.

6.2.4 Modular Design and Repairability Review:

This stage evaluates the RC car's design for modularity and ease of repair, which are vital for its longevity and operational efficiency in field conditions. The expected outcome is an enhanced design that supports easy field repairs and component replacement.

6.2.5 Final Design Approval Release:

The aim here is to obtain final approval on the complete design package, marking the end of the design phase and the beginning of the procurement and assembly phases. This signifies readiness for prototype construction and subsequent testing.

6.2.6 Design Review Release:

This release involves presenting and reviewing the completed preliminary designs, ensuring they are feasible and ready for the next development phase. Feedback obtained here will guide the procurement and prototype development processes.

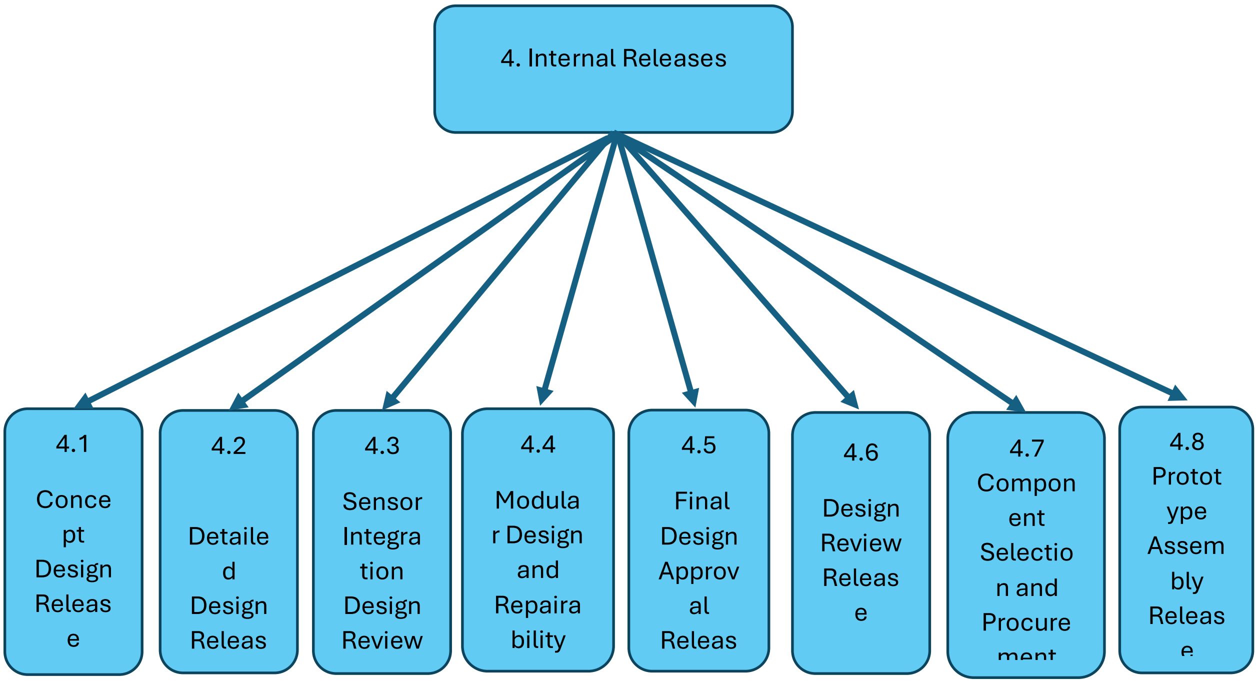
6.2.7 Component Selection and Procurement Release:

The focus is on finalizing and approving the list of necessary components for the prototype, ensuring all parts meet specified requirements and are ready for assembly.

6.2.8 Prototype Assembly Release:

Marks the completion of the initial prototype assembly, showcasing the functional integration of all components. This stage aims to demonstrate basic functionality and identify any potential issues or areas for improvement before moving into functionality testing.

Each release serves as a strategic milestone, ensuring the project progresses in a structured and efficient manner, from conceptual design through to the assembly and testing of the RC car prototype.



The total estimated budget for the RC rescue car development project is $350. This budget is meticulously planned to cover all phases of the project, from initial planning and design to comprehensive testing and final refinements. The allocation of funds is carefully considered to balance the need for quality and efficiency with cost-effectiveness. Regular budget reviews and updates will be essential to manage expenses effectively and adapt to any changes in the project scope or requirements. This financial planning aims to ensure the successful completion of the project within the allocated budget, maximizing value while minimizing unnecessary expenditures.

7. Post-Performance Analysis

The Post-Performance Analysis of our project presents a critical evaluation of the Search and Rescue Remote-Controlled (RC) Car. This analysis is instrumental in assessing the project's success against its defined objectives, technical specifications, and the overarching goal of enhancing efficiency and safety in search and rescue operations. Through this section, we delve into the performance of the RC Car, equipped with state-of-the-art technologies such as LiDAR Mapping, LWIR Thermal Imaging, and SM-24 Geophone sensors, in simulated disaster scenarios. We explore the challenges faced, solutions implemented, and the vehicle's operational efficacy in detecting human life under debris. Furthermore, we compare our innovation with existing solutions, propose future enhancements, and discuss its potential impact on the search and rescue domain. This comprehensive analysis not only highlights our project's contributions to disaster relief efforts but also outlines the pathway for future advancements.

Key Points from the Post-Performance Analysis:

- **Performance Against Objectives:** The report outlines how the RC Car, equipped with LiDAR Mapping, LWIR Thermal Imaging, and SM-24 Geophone sensors, performed in simulated disaster environments. It discusses the extent to which the vehicle met its primary goals of navigating through rubble, creating accurate 3D maps of the surroundings, and detecting signs of human life under debris.

- **Technical Specifications Fulfillment** detailed are how the vehicle adhered to the technical specifications in the design phase. This includes the operational range of the sensors, the durability of the vehicle in various terrains, and its battery life during rescue missions.

- **Challenges and Solutions:** Identified are the significant challenges encountered during the project, such as sensor integration issues, data processing bottlenecks, and mechanical reliability in harsh conditions. The section details the solutions implemented to overcome these challenges, providing valuable insights into the problem-solving processes of the team.

- **Safety and Reliability**: Discussed is the vehicle's safety features, including emergency shut-off mechanisms and compliance with Canadian safety standards. The report evaluates the reliability of the vehicle in performing its intended functions without posing risks to operators or survivors.

- **Comparative Analysis with Existing Solutions:** The report includes a comparative analysis of current search and rescue technologies, highlighting the advantages and unique features of the RC Car. This comparison aims to demonstrate the project's contribution to enhancing search and rescue operations.

- **Future Enhancements:** Suggested are potential improvements and enhancements for future iterations of the project. This may involve advancements in sensor technology, increased automation, and the integration of AI for autonomous navigation and decision-making.

- **Impact on Search and Rescue Operations:** Finally, the analysis reflects on the project's potential impact on real-world search and rescue operations. It considers the practicality of deploying such a vehicle in disaster scenarios, the efficiency it brings to search and rescue teams, and its role in saving lives.

This Post-Performance Analysis provides a comprehensive review of the project's outcomes, offering a balanced perspective on its achievements and areas for growth. It underscores the project's innovative approach to leveraging technology for humanitarian purposes and sets the stage for future developments in the field of search and rescue.

8. Contribution List

Section	*Contributors*						*Total*
	Geoffrey Hooton	*Fatmah Bayrli*	*Papa Kane*	*Moktar Abdillahi-Abdi*	*Walid Rashad*	*Julien Kapro*	
Project Charter	5%	15%	-	5%	5%	70%	100%
System Requirement Specification	10%	-	-	90%	-	-	100%
WBS, Schedule	-	-	85%	-	-	15%	100%
Detailed Design	50%	50%	-	-	-	-	100%
Risk Management	-	-	-	-	100%	-	100%
Budget	-	90%	-	-	-	10%	100%
Test Plan	-	-	-	100%	-	-	100%
Proof of Concept	40%	40%	-	-	20%	-	100%
Post-Performance Analysis	-	-	100%	-	-	-	100%