Final Project Presentation

Search and Rescue Remote-Controlled Car with Life Detection

Group 5

Fatmah Bayrli – 300159193

Geoffrey Hooton – 300187367

Papa Kane – 300090159

Walid Rashad – 300205109

Julien Kapro – 300173066

Moktar Abdillahi-Abdi – 300156852

Project Goal

Objective: The project goal is to design and develop a remote-controlled (RC) car for search and rescue missions.

Technology: Equipped with LiDAR mapping and life detection capabilities (thermal camera, seismic sensor)

Purpose: To efficiently navigate through rubble and tight spaces in post-disaster scenarios, such as earthquakes, building collapse.

Safety Enhancement: Decreases hazards for first responders by using advanced sensing technologies.

Impact: Aims to enhance both the efficiency and safety of disaster relief efforts by detecting human life under debris.

Key Stakeholders & Customers

- **First Responders and Search and Rescue Organizations:**
These are the primary potential end-users who would directly benefit from the deployment of such technology in disaster-affected areas.

- **University or Academic Department:**
The academic department sponsoring the project could also be considered a customer, especially in terms of evaluating the project's success and its contribution to the students' learning outcomes.

- **External Sponsors or Partners:**
Collaborators like government agencies, NGOs, and companies in disaster relief.

Project Impact on Customers

Enhanced Search and Rescue Operations:	Utilizes LiDAR and sensors for efficient navigation and survivor localization in disasters.
Reduced Risk for First Responders:	Minimizes first responder exposure to hazards by remotely navigating dangerous areas.
Operational Efficiency:	Speeds up disaster area mapping and strategic planning for rescues.
Cost-Effectiveness:	Automates search tasks, reducing mission duration and personnel needs.
Versatility in Disaster Response:	Adapts to various disaster scenarios, from earthquakes to building collapses.
Innovative Use of Technology for Social Good:	Innovates in technology application for humanitarian efforts, aligning with broader social impact goals.

Business case

Key Partners

- **Technology Suppliers:** Companies providing LiDAR sensors, thermal imaging cameras, and other critical components.
- **Research Institutions:** Universities or research labs contributing to product development and testing.
- **Search and Rescue Organizations:** End-users providing insights into operational requirements and feedback on prototypes.
- **Government Agencies:** For regulatory guidance, funding, and potential procurement.

Key Activities

- **R&D for Advanced Sensing Technologies:** Developing and integrating LiDAR and life detection technologies.
- **Manufacturing:** Building durable and reliable RC cars capable of operating in harsh environments.
- **Marketing and Sales:** Reaching out to potential customers and demonstrating the value proposition.

Key Resources

- **Technical Expertise:** Engineering skills in robotics, sensors, and software development.
- **Intellectual Property:** Patents on unique aspects of the technology and design.
- **Manufacturing Facilities:** Capable of producing high-quality, durable RC cars.

Value Propositions

- **Enhanced Efficiency in Search and Rescue Operations:** Quickly locating survivors in disaster zones.
- **Safety for First Responders**: Minimizing the risk by reducing the need for human presence in dangerous areas.
- **Cost-Effectiveness:** Reducing the operational costs of search and rescue missions.
- **Innovation:** Offering a high-tech solution for disaster response efforts.

Customer Relationships

- **Consultative Sales Process**: Working closely with search and rescue organizations to tailor the product to their needs.
- **Training and Support:** Providing comprehensive training on operating the RC car and ongoing technical support.
- **Feedback Loop**: Incorporating customer feedback into continuous product improvement.

Channels

- **Direct Sales:** Selling directly to search and rescue organizations and government agencies.
- **Trade Shows and Conferences:** Demonstrating the technology at industry events.
- **Online Marketing:** Leveraging digital platforms for product promotion and customer engagement.

Customer Segments

- **Search and Rescue Organizations:** Government and non-profit organizations involved in disaster response.
- **Military:** For use in combat search and rescue (CSAR) missions.
- **Industrial Customers:** For surveillance and inspection in hazardous environments.

Cost Structure

- **Development Costs:** High initial investment in R&D and prototype development.
- **Manufacturing and Materials:** Costs associated with production and assembly of the RC cars.
- **Marketing and Sales:** Expenses related to customer acquisition and market penetration.

Revenue Streams

- **Sales of RC Cars:** Primary revenue through direct sales to customer segments.
- **Service Contracts:** Ongoing revenue from training, maintenance, and support services.
- **Customization Services:**

Additional revenue from customizing the RC cars to specific customer requirements.

Requirements Specifications

Functional Requirements

A user interface for remote controlling the vehicle will be viewable on the central computer (laptop).

The remote-control UI will be intuitive with easy controls for steering and acceleration.

The operator will be able to view a detailed, real-time 3D map of the vehicle’s surroundings to facilitate navigation of complex terrain.

The system must be able to reliably communicate sensor data back to the operator at a central computer (laptop).

The system shall be able to detect human movements/vibrations underground.

The system shall be able to detect heat signatures underground.

The car shall be rugged and durable.

The car will be able to navigate constrained environments and uneven terrain.

Non-Functional Requirements

A 3D produced UI map will update itself with a latency of no more than 500ms.

The operator shall be able to control the car’s servo motors with a latency of less than 30ms using a remote control.

Sensor data will be transmitted with a latency of at most 30ms.

The system should be able to detect human presence with at least 95% accuracy.

The system will be powered by a 12V battery.

The battery life of the system should be at least 3 hours.

The car will be smaller than 20cm in width and 15cm in height.

The system will be equipped with tracks to provide better traction on uneven terrain.

Constraints

The car’s design will incorporate a remote emergency shut-off mechanism to prevent accidents should the car end up in a dangerous situation.

The vehicle will have a physical on/off switch and a start command on the central computer to help avoid injuries during deployment.

The system will be portable and easily deployed to not impede ongoing rescue efforts or endanger rescue personnel working in the affected areas.

The RC car will be properly grounded to eliminate the possibility of sparking to minimize the risk of explosion in the presence of a gas leak.

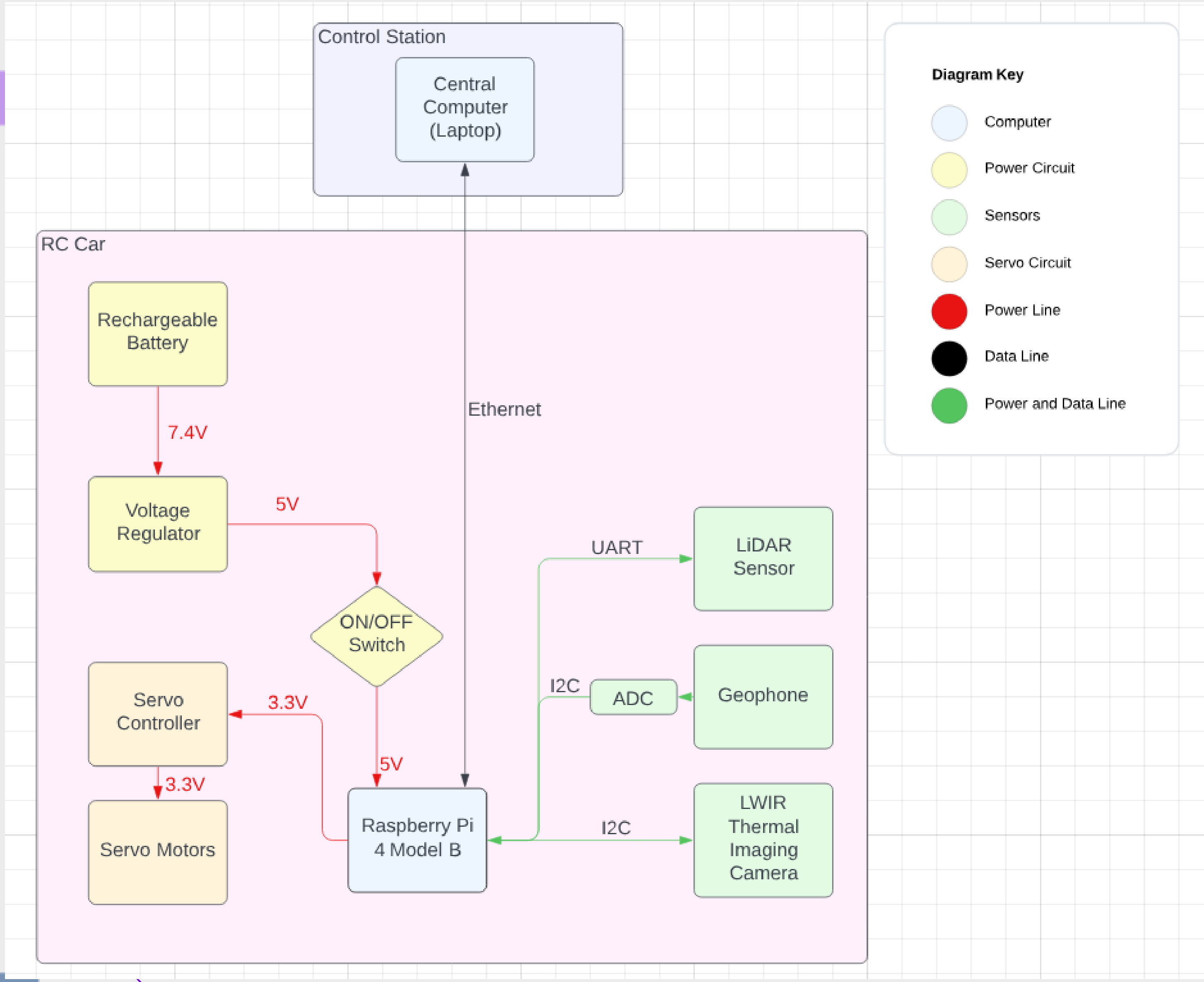
The car’s chassis will include emergency lighting LEDs to enhance visibility.

The vehicle’s design will adhere to Canadian safety standards for electrical and material safety

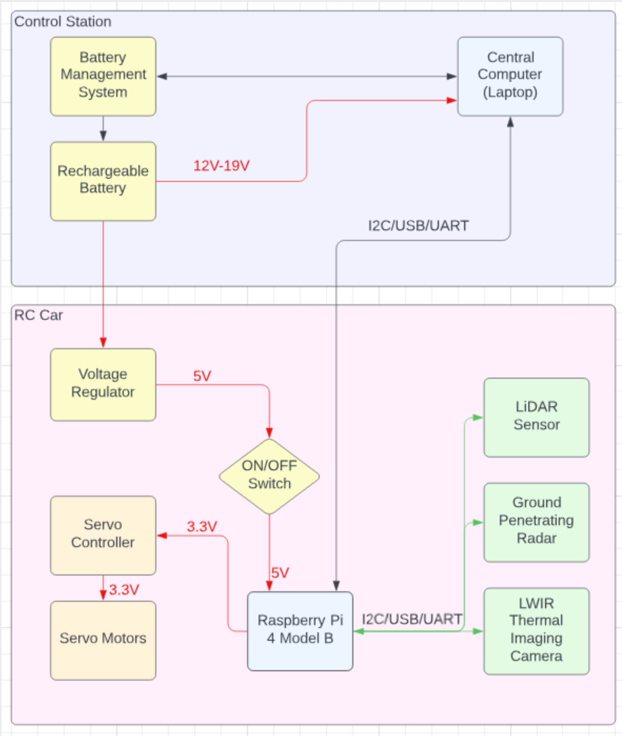
Breakdown of Components

Components Name	Usage
Raspberry Pi	The Central Controller for the project.
Slamtec RPLIDAR Sensor	3D mapping of the environment.
FLIR Lepton LWIR Thermal Camera	Detect heat signatures underground.
Geophone SM-24	Geophones are used in geological exploration to detect seismic activity.
Servo motors	Control of angular position, velocity, and acceleration.
Acrylic glass base with 3 pairs of wheels	The wheels enable movement, and the acrylic base provides a sturdy, transparent structure.
Ultrasonic Sensor	Obstacle avoidance.
Ethernet Cable (100ft)	Used to establish a wired connection for reliable long-distance communication.
ADS1115 ADC	An Analog-to-Digital Converter that allows microcontrollers to use analog sensors by converting the sensor output to a digital format for processing.
Servo Motor Drivers	Controlling the power supplied to the car's motors from the battery, control speed and direction.
Voltage Regulator	Maintain a consistent voltage supply to the sensors and control systems.
Battery 7.4V	Provide portable power to the car.
Switch on/off	Shut down or activate equipment, helping to prevent accidents, damage to the system, or further escalation of the emergency.

Hardware Design



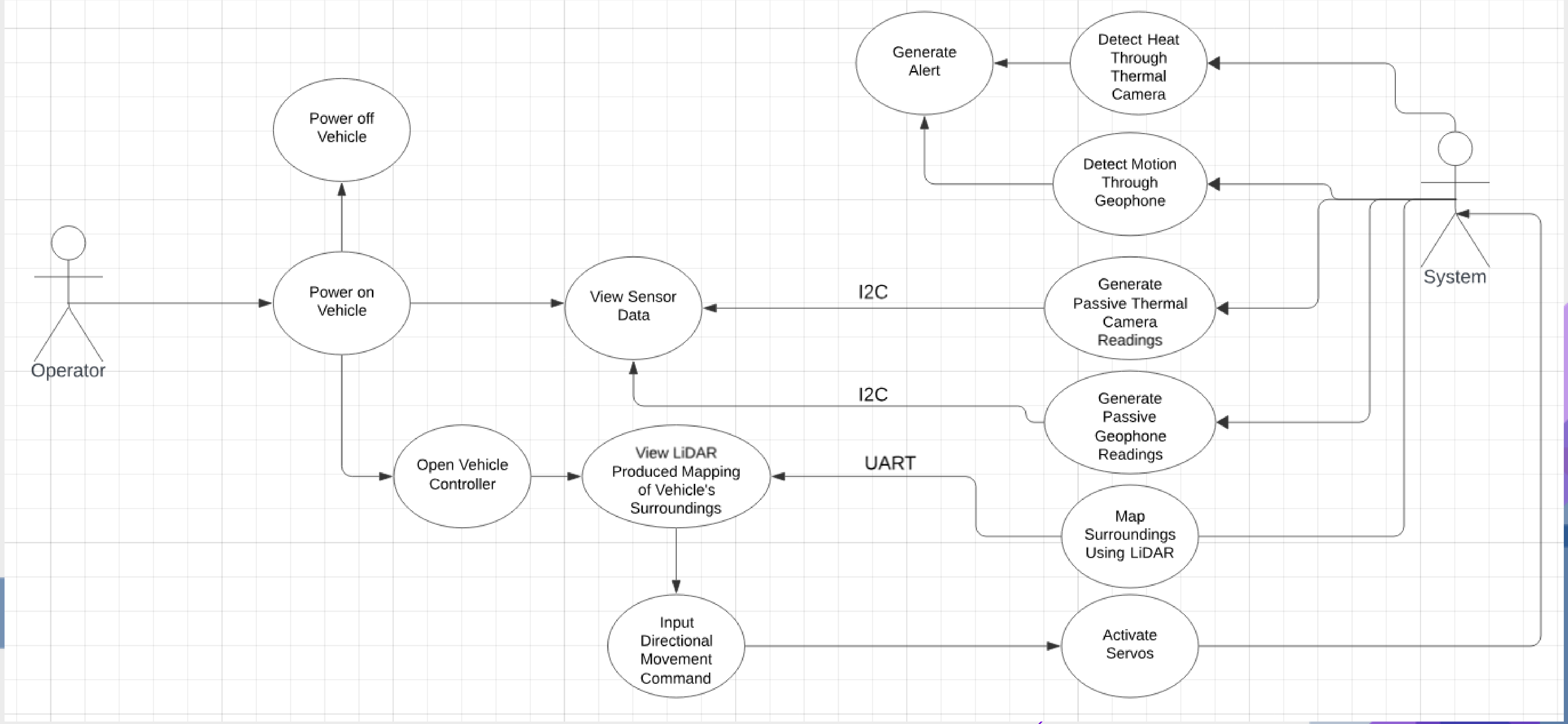
Midterm Design



<u>List of Changes:</u>
-Removed battery management system
-Moved rechargeable battery from control station to the vehicle
-Changed ground penetrating radar to geophone (requires ADC).
-Updated serial communication protocols

Software Design

- Software will be developed in Python.



Anticipated Risks and their Mitigation

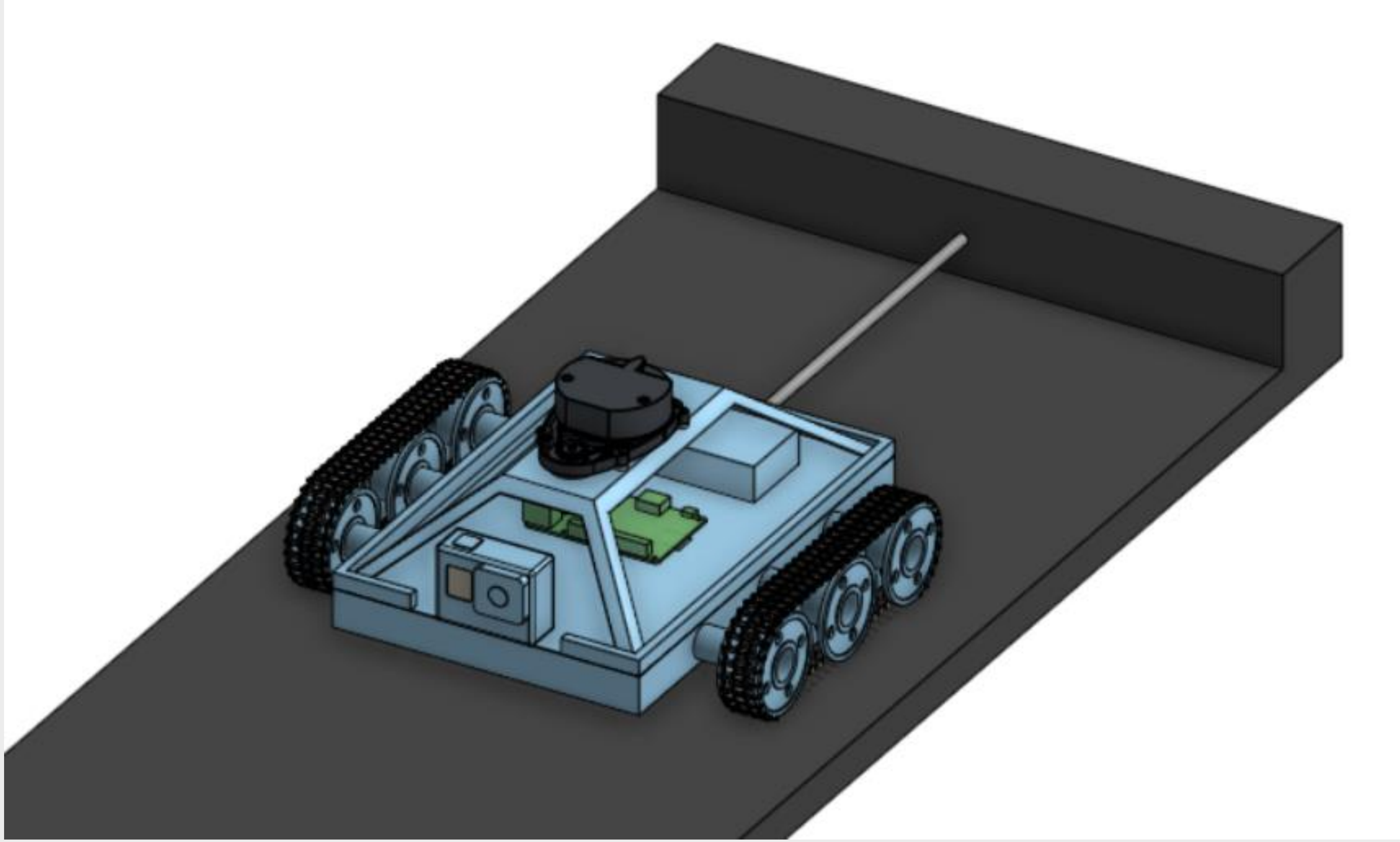
No.	Rsk Description	Probability	Impact	Planned Mitigation
1	Lazer Hazards	*Low*	*Low*	Avoid direct eye contact with laser for prolonged periods of time
2	Battery Incineration	*Low*	*High*	Heat sensor
3	Treads – Moving Parts	*Low*	*Medium*	Shut systems down before handling device
4	Electrical Damage – Short Circuits	*Medium*	*Low*	Test circuit/Heat sensor
5	Troubleshooting Issues	*High*	*High*	Good planning and external support
6	Conceptual Revisions	*Medium*	*High*	Good planning

Assumptions

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

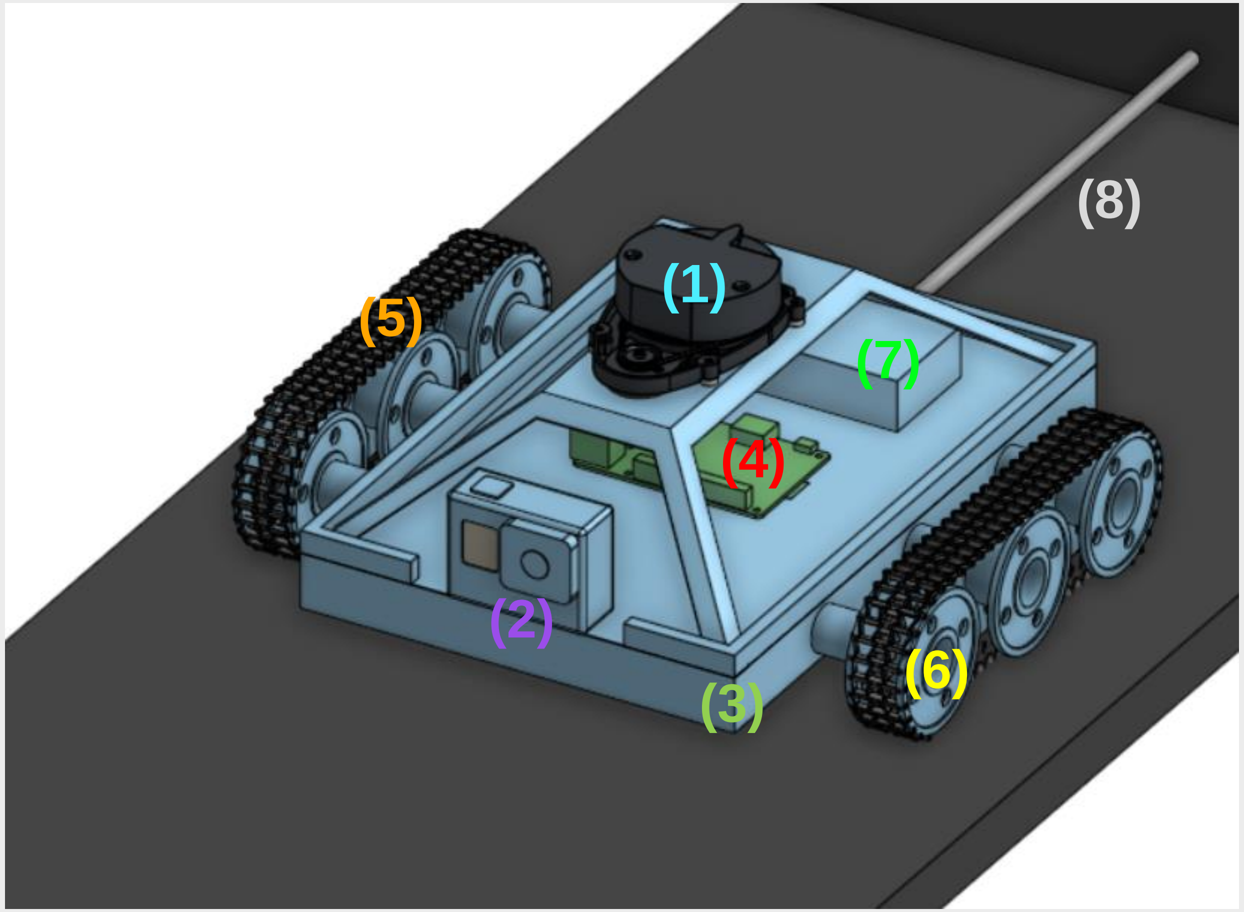
Proof of Concept

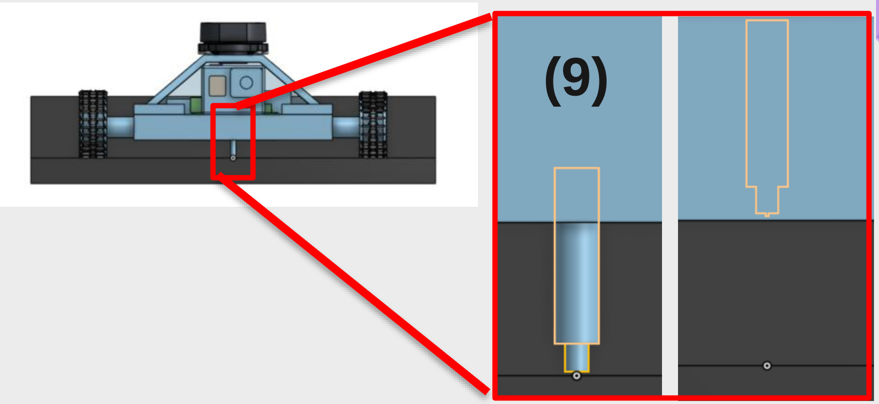
3D First Prototype – CAD



https://cad.onshape.com/documents/fb32e11b8d84719f547e33a7/w/6dca8c35242ddf8d58317f99/e/9b5a9b049bd657a6c0488e62?renderMode=0&uiState=66031dc0c03fc734aef8f41a

3D First Prototype – CAD



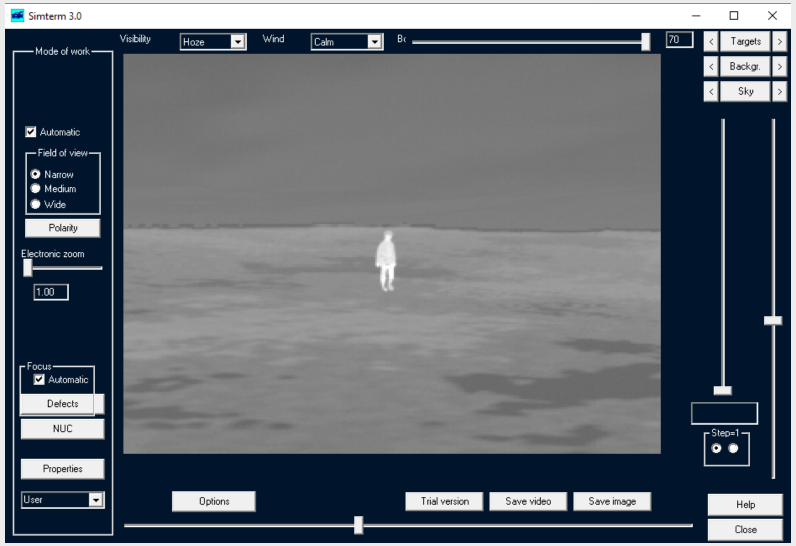


- (1) LIDAR
- (2) LWIR Thermal Imaging Camera
- (3) Chassis
- (4) Raspberry Pi
- (5) Tank Tracks
- (6) Wheels
- (7) Power components
- (8) Ethernet Cable
- (9) Geophone

Long-Wave Infrared Camera Simulation 1: Simterm

- Simterm is a basic simulator that produces realistic images reflecting the quality of thermal images in varying conditions.

Thermal Camera Image of Person in Ideal Condition



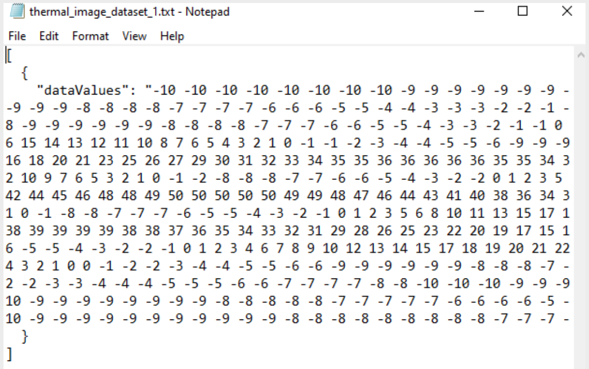
Thermal Camera Image of Person in Haze (combination of smoke/fog/other dry particles)

Thermal Camera Image of Person in Heavy Rain

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

Long-Wave Infrared (LWIR) Camera Simulation 2: FLIR Lepton Python Image Processing

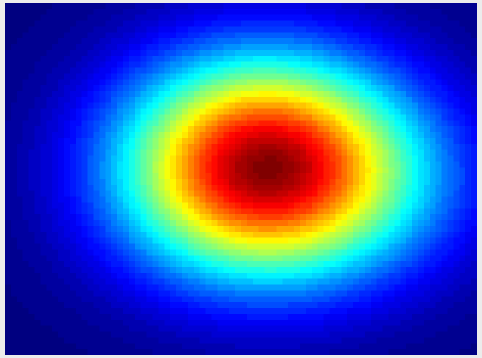
- This simulation simply produces example output files in the same format and resolution (80x60) that our FLIR Lepton LWIR would produce, and then creates a heatmap image based on the data.



thermal_image_dataset_1.txt - Notepad

File Edit Format View Help

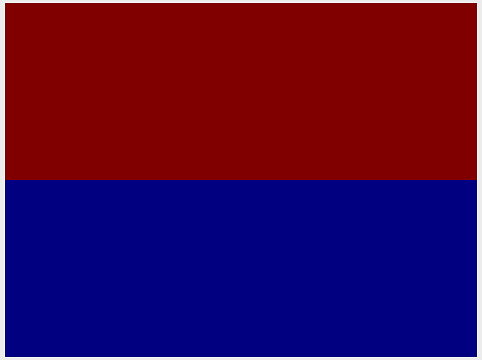

```
[
  {
    "dataValues": "-10 -10 -10 -10 -10 -10 -10 -10 -9 -9 -9 -9 -9 -9 -9 -
-9 -9 -9 -8 -8 -8 -8 -7 -7 -7 -7 -6 -6 -6 -5 -5 -4 -4 -3 -3 -3 -2 -2 -1 -
8 -9 -9 -9 -9 -9 -9 -8 -8 -8 -8 -7 -7 -7 -6 -6 -5 -5 -4 -3 -3 -2 -1 -1 0
6 15 14 13 12 11 10 8 7 6 5 4 3 2 1 0 -1 -1 -2 -3 -4 -4 -5 -5 -6 -9 -9 -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 -
  }
]
```

two_colours10,11.txt - Notepad

File Edit Format View Help

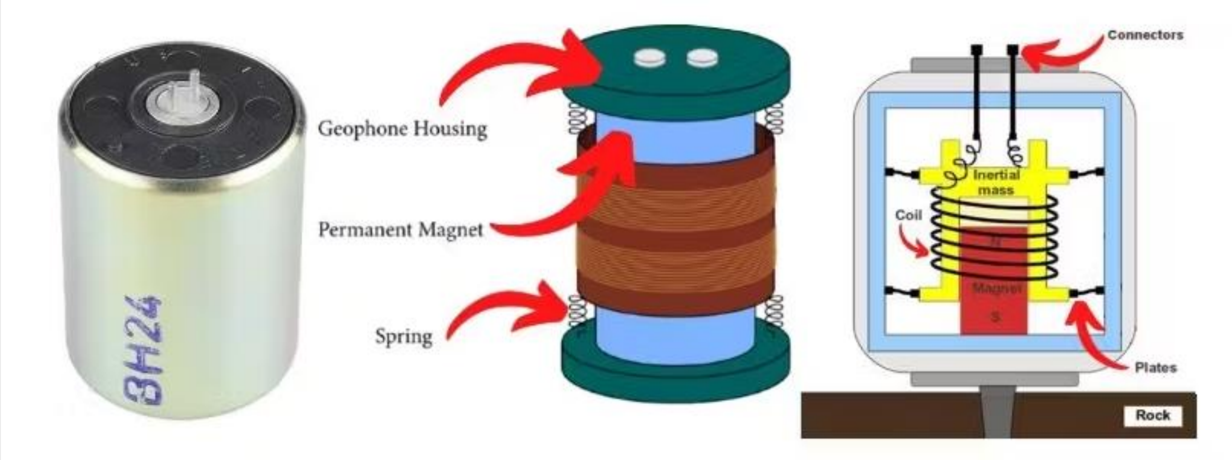

```
[
  {
    "dataValues": "11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11
11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 1
1 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11
11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 1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

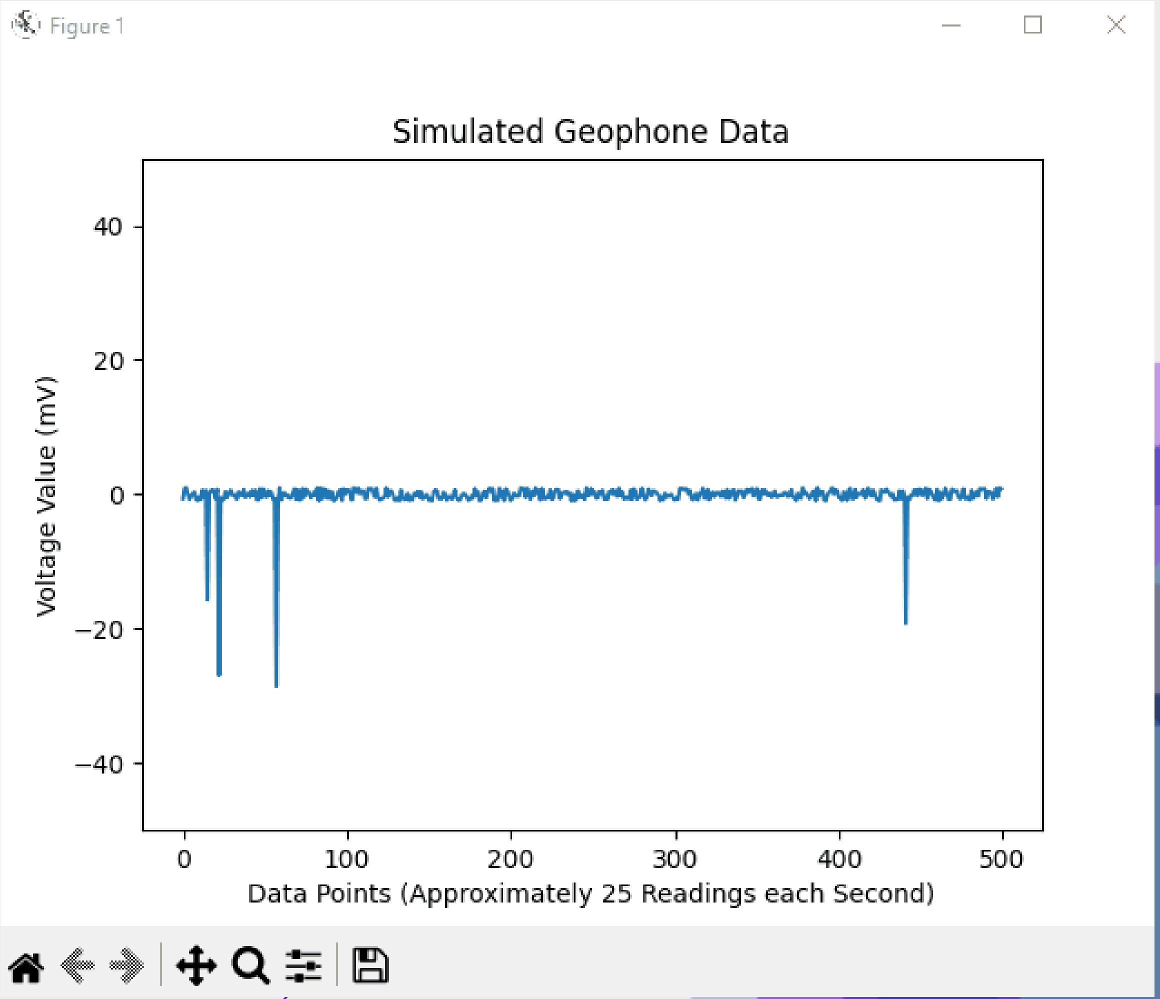



- Credit for the original code of this simulation goes to KhaledSaleh and the GitHub repository can be found at https://github.com/KhaledSaleh/FLIR-human-detection

SM-24 Geophone Simulation

- Python code to translate seismic data from the SM-24 geophone into a consistently updating plot. The original code was edited to replace the I2C input with random data between -1mV and 1mV to represent the zero-noise state, with a 0.5% chance of producing a voltage spike representing movement.






```python
# 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
```

- The source code for this project can be found at https://core-electronics.com.au/guides/raspberry-pi/geophone-raspberry-pi/

LIDAR Simulation in 2D

- The purpose of the Lidar simulation is to replicate the functionality of a real-world LiDAR sensor within a controlled virtual environment.

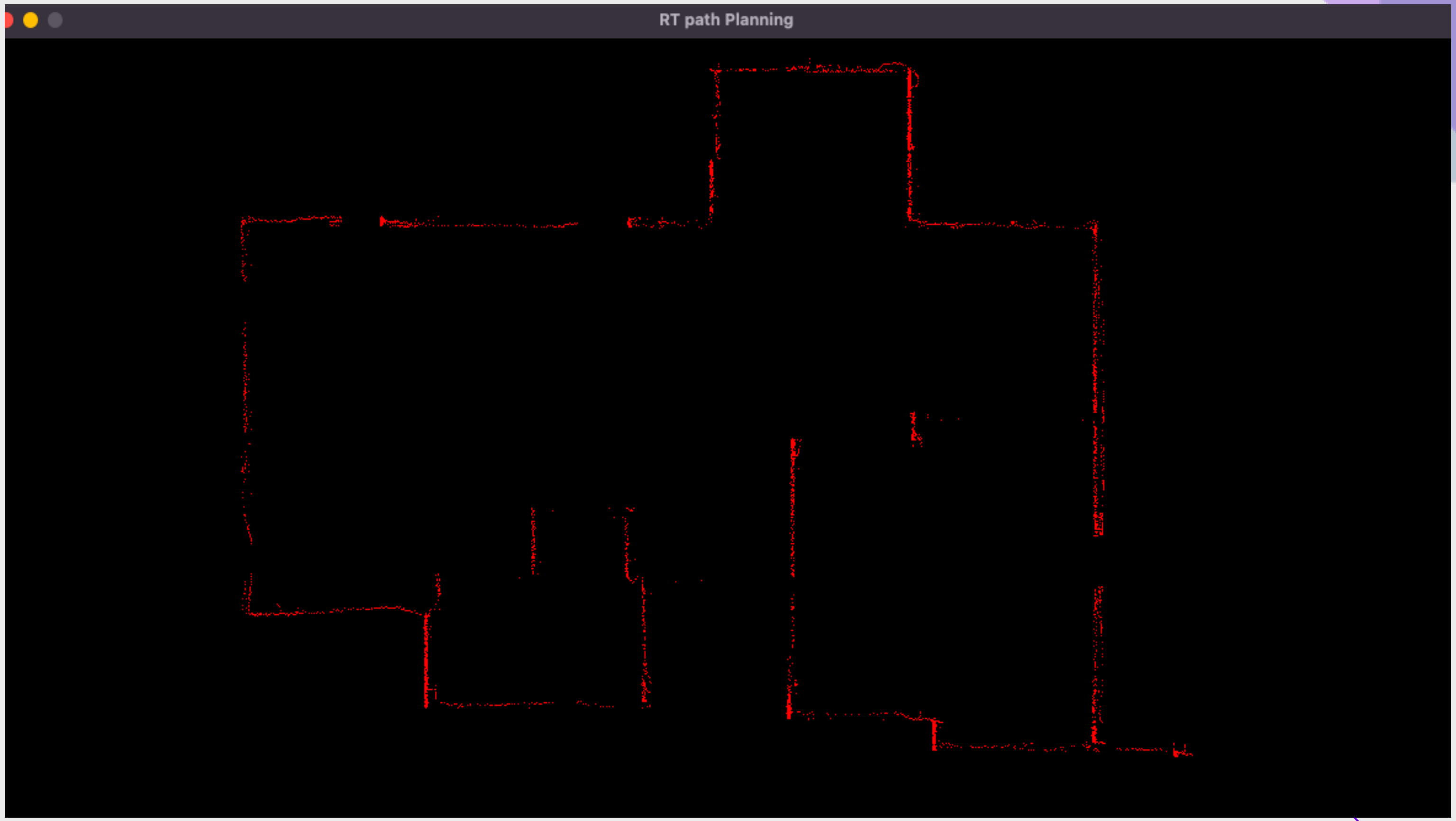
The graphical display resembles a typical occupancy grid used in SLAM (Simultaneous Localization and Mapping) techniques, which are commonly utilized for vehicle navigation to create maps of unknown environments while tracking the location of the sensor within it.

```python
from env import Buildenvironment
from sensors import LaserSensor
import pygame
import math
import random



if __name__ == "__main__":
    environment = Buildenvironment((600, 1200))
    environment.originalmap = environment.map.copy()
    laser = LaserSensor(200, environment.originalmap, uncertainty=(0.5, 0.01))
    environment.map.fill((0, 0, 0))
    environment.infomap = environment.map.copy()

    running = True

    while running:
        sensor_on = False
```

Python code for the LiDAR simulation

```
data --> []
1651
data --> [[6.785234122896453, 0.010137133153482658, (201, 183)]]
1652
data --> [[7.331230439865883, 0, (201, 181)]]
1653
data --> [[5.694491950360028, 0, (201, 180)]]
1654
data --> [[6.51513690261883, 0.0020061986540293383, (201, 178)]]
1655
data --> [[6.298220741745315, 0, (201, 177)]]
1655
data --> [[6.279147081590955, 0, (201, 176)]]
1656
data --> [[5.841012828793886, 0.02504917370055199, (201, 177)]]
1657
data --> [[6.737417995731053, 0.0007235942580894283, (201, 178)]]
```

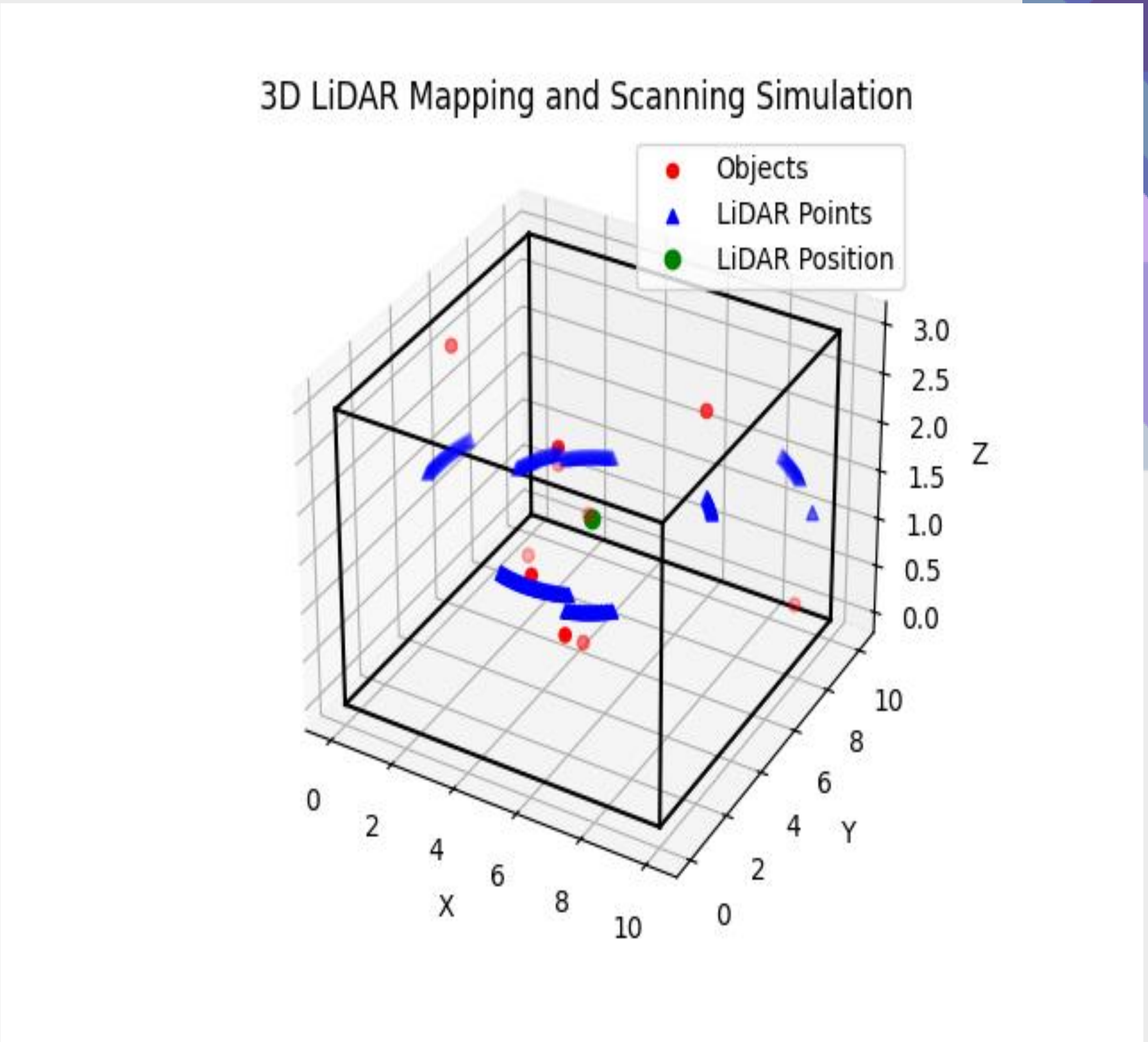
Sample of the data generated during simulation



Lidar Sensor Simulation (2D)

LIDAR Simulation in 3D

- The simulation depicts a controlled space with dimensions along the X, Y, and Z axes.
- The grid represents the extents of the 3D space, where LiDAR mapping is being simulated.
-The green dot symbolizes the position of the LiDAR sensor within the room.
-The red dots indicate the positions of objects within this space. These could be any physical items that the LiDAR sensor can detect.
-The blue triangular markers represent the points detected by the LiDAR sensor. These are the specific locations on the objects' surfaces where the LiDAR's laser beams have bounced back to the sensor, indicating the presence of an object.



Lidar Sensor simulation 3D

Sample Results

The Python code that was used for the simulation, which outlines the parameters set for the environment, the Lidar sensor, and the simulation process.

```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 = []
```

Python Code used for simulation

- The simulation confirms the LiDAR device's static location at coordinates (5, 5, 1.5), serving as a reference point for all measurements.
- A set of (X, Y, Z) coordinates illustrate the varied locations of objects within the room, demonstrating the simulation's effectiveness in identifying obstacles at multiple points in space.
- Detailed scan points from the LiDAR's rotation are provided, indicating the precise locations where the LiDAR detected surfaces, crucial for constructing a 3D environmental map.

```
LiDAR Position:
(5, 5, 1.5)

Objects' Positions (X, Y, Z):
[2.66448596 7.1405144  1.38841636]
[4.67437283 3.76313183 2.41526122]
[7.46695905 7.03148014 2.47310007]
[5.12557843 1.56863103 1.61695487]
[6.42756663 1.13948872 1.25708217]
[5.01774346 4.50187274 0.30654119]
[2.2360291  6.14834288 0.56926305]
[9.21179196 9.25593151 0.22283231]
[3.51288527 7.37810506 0.90526097]
[0.08303299 5.64268931 2.61597015]

LiDAR Scan Points (X, Y, Z):
[10.18700282  8.25243042  1.5        ]
[7.79875461 6.82392295 1.5        ]
[7.76640546 6.87262466 1.5        ]
[7.73320894 6.92075276 1.5        ]
[7.69917522 6.96829252 1.5        ]
[7.66431471 7.01522938 1.5        ]
[7.62863809 7.06154895 1.5        ]
[7.59215631 7.10723705 1.5        ]
[7.55488052 7.15227968 1.5        ]
[7.51682215 7.19666305 1.5        ]
[7.47799285 7.24037357 1.5        ]
[7.43840452 7.28339784 1.5        ]
```

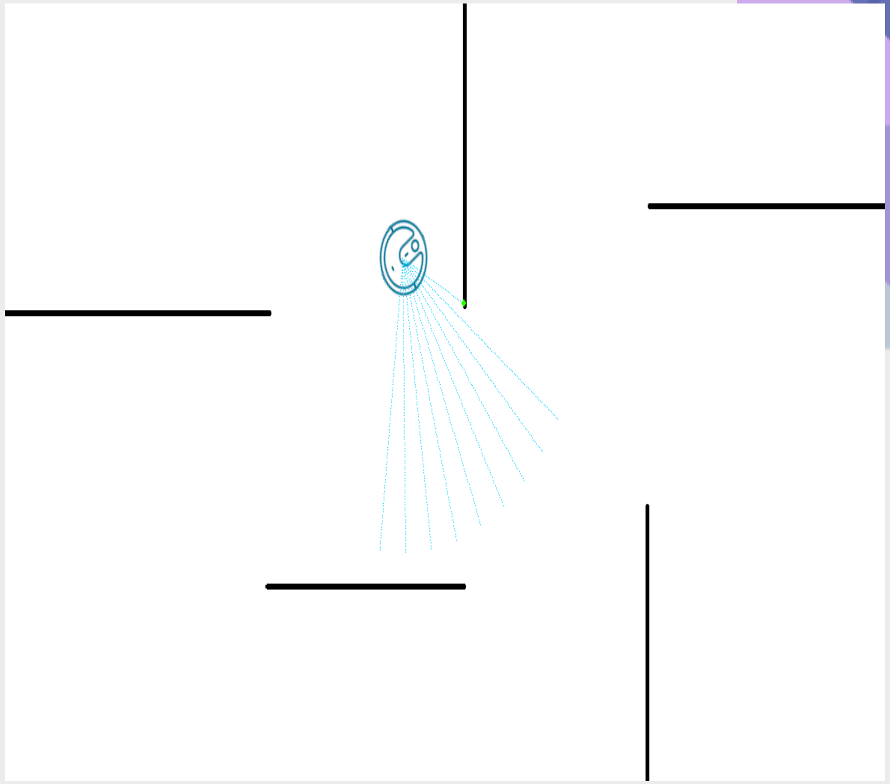
Sample Results

Ultrasonic Sensor Simulation

```python
def __init__(self, startpos, width):
    self.y = startpos[1]

    self.heading = 0

    self.velocity_l = 0.01*self.meters_to_pixel
    self.velocity_r = 0.01*self.meters_to_pixel

    self.max_speed = 0.02*self.meters_to_pixel
    self.min_speed = 0.01*self.meters_to_pixel

    self.min_obs_dist = 100
    self.count_down = 5


def avoid_obstacles(self, point_cloud, time_step):
    closest_obs = None
    dist = np.inf

    if len(point_cloud) > 1:
        for point in point_cloud:
            if dist > distance([self.x, self.y], point):
                dist = distance([self.x, self.y], point)
                closest_obs = (point, dist)
```

Part of the python code used in the simulation

- This simulation illustrates how the ultrasonic sensor simulation operates within a virtual environment.
- These lines represent the sensor's detection range. Objects within this range reflect the ultrasonic waves back to the sensor, allowing it to perceive the environment.”
- The ultrasonic sensor simulation gives us insights into sensor behavior and helps in developing algorithms for obstacle detection and navigation."



Ultrasonic sensor simulation

Test Plan

Test Plan

Test basic car functionality, LiDAR accuracy, and sensor integration.

Assess navigation accuracy and sensor reliability.

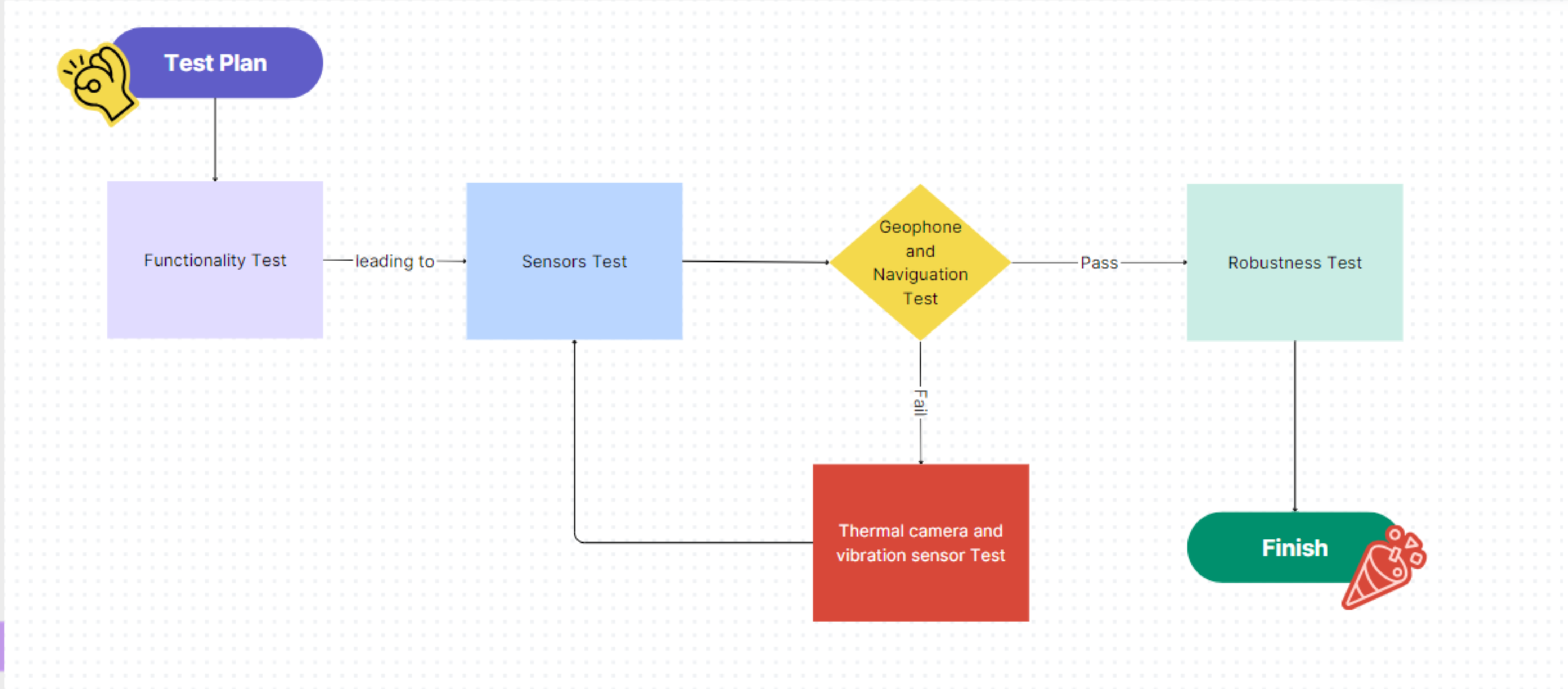
Identify and adress any defects or issues in our system

Test Cases

Test Cases	Description	Status
Functionality Test	-Test basic car functionality, such as movement and responsiveness to commands. -Validate the accuracy of LiDAR technology in obstacle detection and mapping.	-To be Tested -Pass
Sensor Integration Test:	-Verify seamless integration of LiDAR, vibration sensors, and thermal cameras. -Ensure effective interpretation of sensor data for decision-making.	-To be Tested -To be Tested
Thermal Imaging Test	-Test the thermal camera's ability to detect life signs in various conditions. -Verify temperature readings and heatmap accuracy.	-Pass -Pass

Test Cases

Test Cases	Description	Status
Navigation Test:	Assess the car's ability to navigate through simulated obstacles. Evaluate the accuracy of route planning and obstacle avoidance algorithms.	-Pass -Pass
Geophone Test:	Evaluate the geophone's sensitivity and reliability in detecting vibrations. Test performance in different noise and surface conditions.	-Pass -Pass
Robustness Test:	Subject the car to extreme environmental conditions to assess durability. Test impact resistance and terrain capability.	-To be Tested -To be Tested

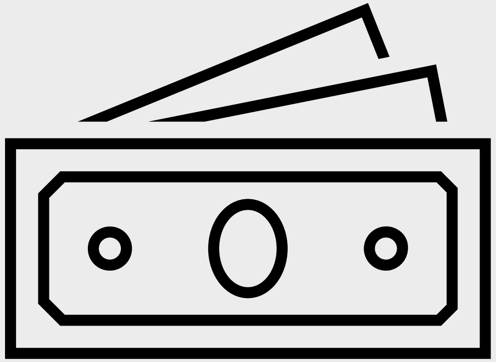


Test Plan
Functionality Test
leading to
Sensors Test
Geophone and Navigation Test
Pass
Robustness Test
Fail
Thermal camera and vibration sensor Test
Finish

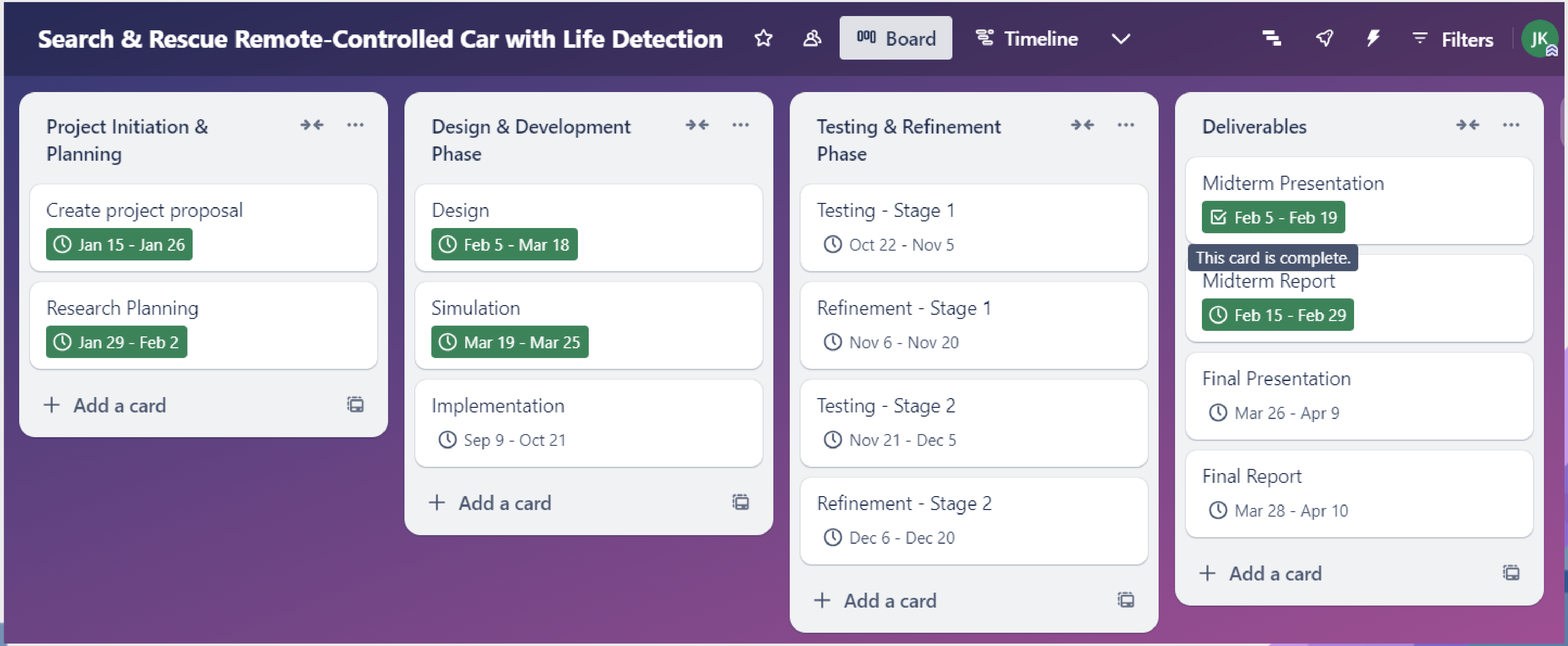
Budget Overview

Component	Cost
Slamtec RPLIDAR Sensor	$ -
Thermal Camera	$ 100.21
Geophone SM-24	$ 87.68
ON/OFF Switch	$ 1.15
Servo Controllers	$ -
Raspberry Pi	$ -
Software (SBC, simulation environments...)	$ -
Servo motors	$ -
Acrylic glass base with 3 pairs of wheels	$ -
Battery 7.4V	$ -
Voltage Regulator	$ 2.69
Ethernet Cable (100ft)	$ 25.88
ADS1115 ADC	$ 20.32
Ultrasonic Sensor	$ -
Total	**$ 237.93**

- Components in blue are given to us by the university.



Schedule

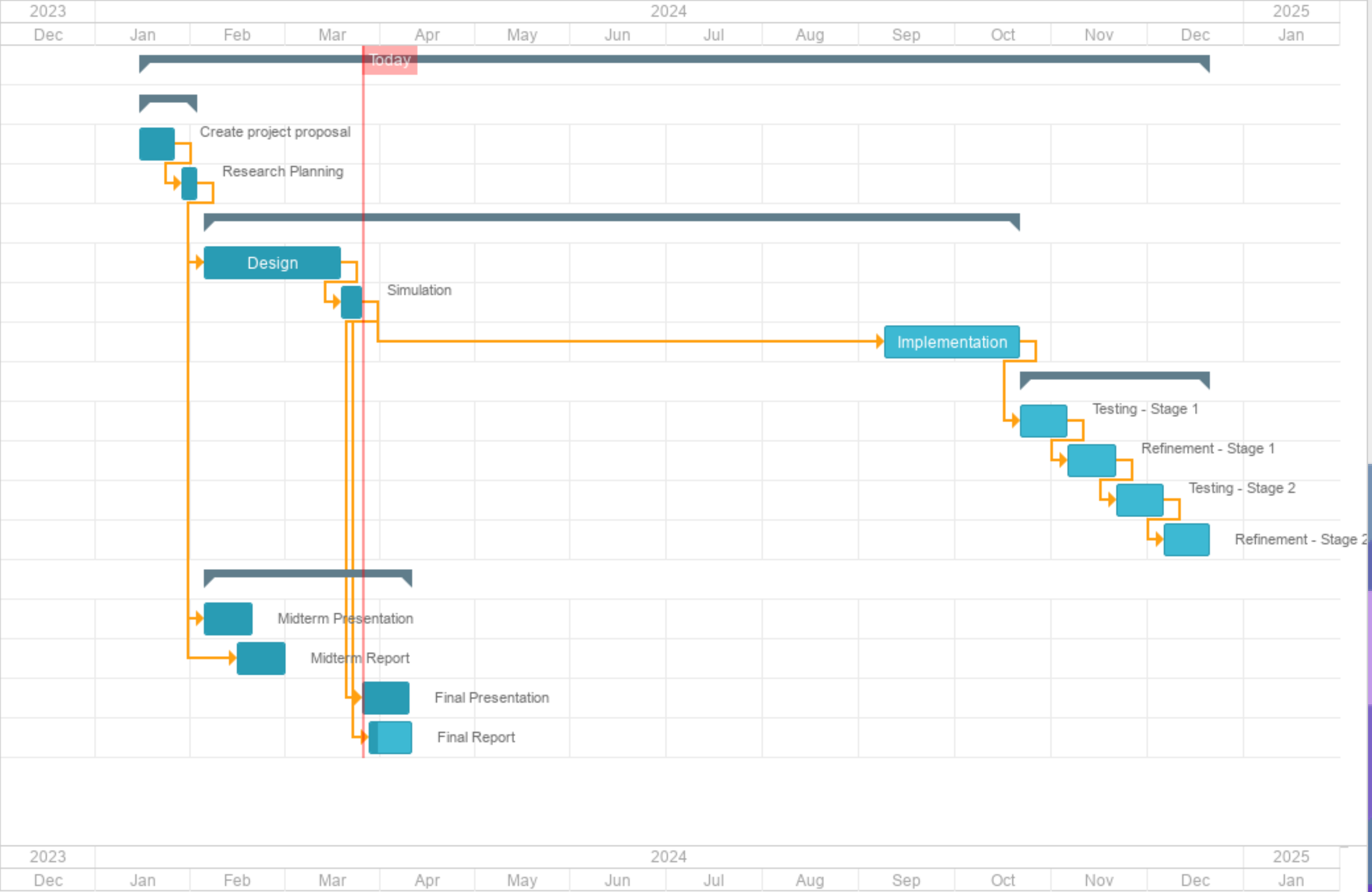


Search & Rescue Remote-Controlled Car with Life Detection
Board
Timeline
Filters
JK
Project Initiation & Planning
Create project proposal
Jan 15 - Jan 26
Research Planning
Jan 29 - Feb 2
Add a card
Design & Development Phase
Design
Feb 5 - Mar 18
Simulation
Mar 19 - Mar 25
Implementation
Sep 9 - Oct 21
Add a card
Testing & Refinement Phase
Testing - Stage 1
Oct 22 - Nov 5
Refinement - Stage 1
Nov 6 - Nov 20
Testing - Stage 2
Nov 21 - Dec 5
Refinement - Stage 2
Dec 6 - Dec 20
Add a card
Deliverables
Midterm Presentation
Feb 5 - Feb 19
This card is complete.
Midterm Report
Feb 15 - Feb 29
Final Presentation
Mar 26 - Apr 9
Final Report
Mar 28 - Apr 10
Add a card

Schedule

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

Name	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



Name

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

Internal Releases

Internal Releases
Removed Battery Management System (BMS)
Added Ethernet Cable
Swapped the Ground Penetrating Radar (GPR) for Geophone

Thanks for Listening!