Robotic Observer of Abandoned Materials
R.O.A.M.

Creators:

Daniel Claassen CpE

Nathan Little CpE

Claire Persinger EE

Ethan Robotham CpE

Group 14

Semesters:

Summer 2025 - Fall 2025

Reviewer Committee:

Dr. Xin Xin
ECE Assistant Professor

Dr. Justin Phelps
ECE Assistant Professor

TBD

Mentor:
Dr. Chung Yong Chan

Chapter 1: Project Description

1.1 Background and Motivation

R.O.A.M. (Robot Observer of Abandoned Materials) is an autonomous robot that detects and reports discarded and salvageable items near residential sidewalks. The identified items are then uploaded to a website for people in the area to view. Objects are categorized into the two groups using a trained object classification model. The salvageable items, known as treasures, include items such as couches, chairs, or bookshelves. The classification and reporting aims to keep salvageable items out of landfills and to connect them with people who may find good use for them.

The successful operation of this system is dependent on several key functions and components: interfacing a camera to identify objects, LiDAR to direct the autonomous driving, connection to the internet, and an interactable website. The trained object classification model breaks down the images into the different types of objects, trash and treasure. Both classifications are reported to the website with the object classification type and approximate location of the object. The approximate location is sent as a coordinate using a sim card module. We will have to select an object identification model and use OpenCV to implement it. A considerable portion of this project will be training the model with images of trash and treasure within view of sidewalks.

R.O.A.M. is a small, four-wheeled robot meant to traverse neighborhood sidewalks. This restricts the size R.O.A.M. could be, limiting it to about half the width of a sidewalk. R.O.A.M. would house cameras on the top of the chassis so that it can take pictures or videos of the objects it finds on the neighborhood sidewalks with minimal obstruction.

We came up with this idea after seeing how often people leave salvageable or functional items on the side of the road. R.O.A.M. provides a second life to discarded objects before they reach landfills, reducing the garbage on the streets. R.O.A.M. was created to help connect individuals with functional items

1.2 Existing Products and Related Work

1.2.1 Clearbot

Pranjal Malewar, covering the story behind Clearbot, writes about a solar-powered, marine-trash-collecting robot used to identify and collect trash from waterways. It uses AI vision to detect floating debris while navigating an aquatic environment. Not only can it collect trash, but it has the ability to map underwater terrain, depth, and shape. Clearbot uses AI technology, surveys and computer models to track an estimate of how much trash enters the ocean and finds it to collect it. In the collection process, this bot has the ability to identify and log the types of trash it collects. This process is done through a two-camera system which also uses the AI machine learning system. One of the cameras is to survey the water for trash as well as hazards. The second camera is used for the identification of the rubbish it finds. Once the trash is collected, there is an image and GPS location sent to a data collection site. The obvious objective of this project, to clean up the ocean, is not the only solution it provides. This robot is important to collect data about the types of trash that ends up in the ocean, this data can help researchers and people in the environmental protection sectors.

1.2.2 GULP, Garbage Utility Locator & Picker

Another garbage identification project found while researching is GULP. GULP is a solar-powered smart garbage that separates trash to increase the functionality of the waste management process. It uses machine learning image processing to separate the waste and even suggest a recommended compressor to create more room for that type of garbage. This item also has the ability to monitor the garbage bins using a SMS notification system to prevent overflow. This was all created using a Raspberry Pi and multiple ultrasonic sensors to aid in the separation of garbage as well. Being able to identify trash types, separate trash, suggest compactors, and detect overflow are relatively intense requirements of a system. When looking into this project, we see an interesting way that they handled their workloads with the image processing. We also see that they used ultrasonic sensors to separate the trash types. The processes they used in this project is something we could consider to look further into as we build our own.

1.2.3 Takeaways

While ROAM shares ideas with the two projects discussed above, It differs in how each part is applied. Clearbot focuses on identifying trash in the ocean and autonomous navigation, while G.U.L.P. focuses on identifying and sorting trash. Using these projects as references, R.O.A.M. is going to extend these ideas to neighborhoods by detecting trash and valuables while driving autonomously on the sidewalk. By referencing projects that work, R.O.A.M. offers a solution to waste management in neighborhoods.

1.3 Goals and Objectives

The overall goal of our project is to be able to deliver the information we have in a safe and affordable way. By combining object detection, autonomous driving, wireless transmission and more, we aim to create a device that can increase accessibility to second hand items. This design is a practical and reliable device that can be implemented in communities without extensive maintenance.

1.3.1 Our Goals

A. Basic Goals

a. Identify salvageable and discarded items near the sidewalk
b. Autonomous sidewalk following.
c. Upload object's classification and location to a server from a distant location.

Identifying reclaimable items is at the center of R.O.A.M. using computer vision R.O.A.M. can classify objects on the side of the sidewalk. This gives the ability to relay that information to the website. Uploading the information is a vital next step to display it to any potential users on the website. The location must be attached once sent to the website, so it is retrievable. Being able to autonomously follow a sidewalk gives R.O.A.M. functionality and purpose. Combining with the other basic goals this gives R.O.A.M. a full range of features that allows R.O.A.M. to roam the sidewalks and gather information without human control or intervention.

B. Advanced Goals

a. Use a Map API to visualize data collected on our website.
b. Ask for help when path is obstructed

Once data has been collected, it will be displayed on a website for readability. Using a Map API to put icons on the map at the coordinates provided. This gives information an intractable element to the website, allowing the users to not just read coordinates of the objects but see where those coordinates place on a map. When R.O.A.M. becomes blocked by an object in the sidewalk, it will call out for help, communicating that its path is blocked so it can be assisted.

C. Stretch Goals

a. Further classification to identify treasure type.
b. Integrate a search via location or treasure type on the website.
c. Solar battery charging system.
d. Battery life indication

To broaden R.O.A.M.'s abilities, we have added goals to a stretch section in an attempt to expand on the non-critical functions of our robot. One way is by improving how it classifies the items it detects. Rather than simply categorizing items into trash and treasure, the system will support more detailed classification of items. These items will be displayed on the website with additional abilities to filter or search. To extend its use time, we plan to incorporate solar capabilities. This would allow more time between charges, as we can supplement it with solar energy. Additionally, we plan to have a battery life indicator to the LCD screen to improve functionality and monitor the robot's status in real-time.

1.3.2 Objectives

A. Identify salvageable and discarded items on sidewalks in neighborhoods.
   a. Camera Communicating with Edge device.
   b. Trained object identification.
B. Autonomous sidewalk following.
   a. LiDAR Sensor that communicates with Edge Device.
   b. Algorithm to make a 2D point cloud and interpret data, identifying Sidewalk.
   c. Further data interpretation to adjust motors.
C. Upload an object's classification and location to a server from a distant location.

    a. Server to host a website.
    b. GPS Locator working on R.O.A.M.
    c. Ability to communicate information from R.O.A.M. to server.

The objectives of R.O.A.M. are derived from our project goals. Each goal requires several different components to achieve. The basic goals outline the core functionality of R.O.A.M. which include achievable items that we believe can be completed within our timeframe. The first objective is to identify items and classify them as either salvageable or discarded, to do this we will need to employ a camera that can communicate with our edge computing devices. The robot will use a trained machine learning for object identification. Additionally, we want our robot to function as an autonomous entity which will have a few objectives to meet. Autonomous functionality will entail having LiDAR sensors that will be able to communicate with our edge device. These will be responsible for sending information to our ESP32 to then use further interpretation to adjust our motors. Another goal is to upload information about detected objects and their coordinates to a website. This would require GPS tracking and communication protocol between the edge and the website. We also need to host a website to provide that information.

A. Use a Map API to visualize data collected on our website.
    a. Upload GPS coordinates with trash or treasure markers.
    b. Employ map API.
B. Asks for help when the path is obstructed.
    a. Use a buzzer to audibly signal an object in its path.
    b. Use an LCD screen to visually display this information.
    c. Website notification to alert the robot is stuck.

The advanced goals of R.O.A.M. build on the basic goals to enhance the core functionality, improve ease of use, and provide insurance for edge cases. These objectives are more difficult but remain achievable. We aim to display the date collected on a map. To execute that idea, we would need the GPS coordinates of those items, as well as a map API. Additionally, since our robot is autonomous, we need to be able to handle a case where our robot's path gets blocked. To do this, we plan on adding a buzzer to alert people nearby, as well as an LCD screen with signals for assistance on it. Finally, we will send a signal to our website that the robot needs assistance.

A. Further classification to identify treasure type.
    a. Another object classification model running on the base server.
    b. Ability to send treasure images to the database for analysis.
B. Integrate a search via location or treasure type on the website.
    a. Categorized Database.
    b. Further classification of treasure
    c. GPS coordinate translation and organizer.
C. Solar battery charging system.
    a. Charge Controller.

        b. Access to battery.
    D. Battery life indication
        a. Voltage divider to “read” battery.
        b. Sending it to the LCD screen.

The stretch goals of R.O.A.M. add features that build off existing concepts within R.O.A.M. but aren't required for core functionality. Instead, they improve R.O.A.M. by working alongside existing components. To enhance its classification capabilities, one of our objectives involves running a more advanced model on the server to allow hyper specific identification. This would also include sending photos that the robot takes of the objects to the website. Another improvement includes integrating searching on our map, enabled through additional object categorization and better GPS data structuring. Solar battery charging was another topic we discussed, allowing the autonomous robot to spend more time on the road without intervention. This would require a solar charge controller and access to that battery to charge it. Finally, we want to add a battery life indicator on the LCD screen, to provide a real-time update for users that interact with our robot. For this we would need to use a voltage divider and connect it to our ESP32, which with some code can allow us to read the voltage of our battery.

1.4 Functionality

1.4.1 Autonomous Driving

R.O.A.M. will use LiDAR and an analyzing algorithm like RANSAC. LiDAR’s main function is to provide the information that makes it possible to identify the sidewalk, allowing for R.O.A.M. to stay on the sidewalk. The LiDAR will interact with our edge device that will run the algorithm and interpret the current position with the sidewalk to provide directional correction to the ESP32. The ESP32 will control the information sent to the motors.

1.4.2 Object Detection & Classification

A large goal of this robot is to use an object classification machine learning algorithm implemented on an edge device fitted with a camera to identify and classify objects on the side of the sidewalk as trash or treasure. This feature drives our main goal of allowing users to reclaim discarded objects that would be useful, such as a used chair for a college student. The information gathered would be accessible on our website for people to interact with. This will reduce the amount of unnecessary waste on the side of the road. R.O.A.M. will also track the trash it sees, so areas with high volumes of trash are noted and cleaned up.

1.4.3 Interactive Website

The website will organize and provide information for the public to view and use to their discretion. The website will be sent information by R.O.A.M. depending on what is seen by the robot. If treasure is seen like couches, chairs, or otherwise usable furniture, a package from R.O.A.M. will be sent to the website containing the treasure classification, an image, and GPS location of the treasure. This could show up on an interactive map or in a list format. Information provided about the trash in the area will be displayed differently from treasure, since the exact location of a piece of trash is not required, R.O.A.M. will send the amount of trash it's seen every 5 minutes with a GPS location and then reset the trash seen variable. This will aggregate the amount of trash seen by R.O.A.M. between two GPS locations.

1.5 Key Specifications

Table 1.1 — Key Specification

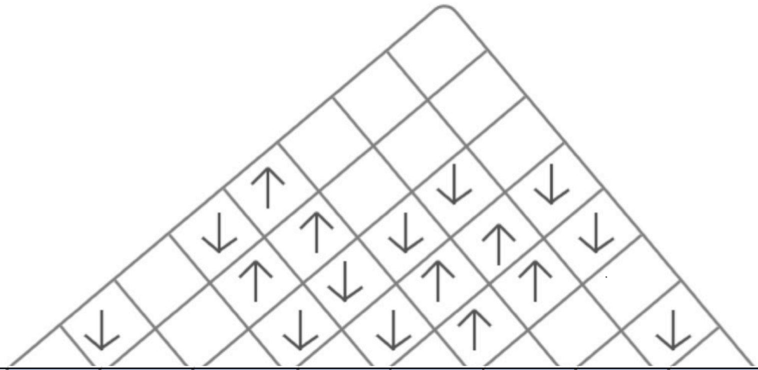
Parameters	**Specifications**
Speed (legal requirement)	< 10 mph
Width (wheel to wheel)	< 46 cm
Traversable Obstacle Height	< 35 mm
Website Data Upload Time	< 15 seconds
Object Detection Response Time	< 2 seconds
Treasure Identification Accuracy	> 70%
Continuous Operation Time	> 15 minutes
GPS accuracy	Within 20 m
Distance traveled without going out of bounds	> 800 meters
Object Detection Range	< 3 meters > 0.5 meter
Power consumption	< 55 W
Cost	< $1500

1.6 House of Quality

Our marketing requirements in Figure 1.1 are aimed at a safe and easy-to-use autonomous robot. With this, a lot of consideration will be placed upon the reliability of this functioning bot. This is not the only driving force behind our engineering development as GPS accuracy and operating time will be important deliverables on our reliability front, while remaining under $1500 and with a width to remain within 46 cm.

Table 1.2 — House of Quality

Legend:
o = no correlation
↓ : Negative Correlation
↑ : Positive Correlation
↓↓ or ↑↑ : Strong Correlation



Engineering Requirements			Battery Capacity	Prototype Width	Object Identification Accuracy	Cost	Battery Life	Obstacle Detection Response Time	Website Data Upload Time	GPS Accuracy
Marketing Requirements			+	-	+	-	+	-	-	+
	Portability	+	↓↓	↑↑	o	o	↓	o	o	o
	Durability	+	↑↑	↓	o	↑↑	↓	o	o	o
	Reliability	+	o	o	↑	↓	↓	↓	↓	↑
	Affordability	+	↑↑	↓↓	↑	↓↓	↑↑	↓	↓	↑↑
	Usability	+	↑	↓	↑	↓	↑	o	o	↑↑
	Safety	+	o	↓	o	o	o	↓↓	o	o
Targets for Engineering Requirements			> 25 Ah	< 46 cm	> 70%	< $1500	> 15 minutes	< 2 seconds	< 2 seconds	< 12 meters

1.7 Hardware Description

Figure 1.2 describes the basic system of our project's Hardware. Beginning with the power supply section, there is a battery that will be the main contributor to the rest of our hardware. That will supply the power through our regulators which will be a part of our PCB board, this is the connection point to the rest of the system. Going through the regulators allows us to protect the rest of our equipment in case of an unwanted supply of amps. The three paths from the regulators are the ESP32, the Jetson, and the motor controllers. From there we have a multitude of different functions that will happen. With the regulators supplying power to the ESP32, it then can send and receive data. One of these data paths is to the Jetson, that Jetson is also sending data to the ESP32, this data would include images and data about the object identification for the website. On top of that, the ESP32 will also be receiving information about how to adjust the motors based off of the LIDAR sensors. LIDAR, in conjunction with the Jetson will determine the movement of the robot based on the environment, and send that information to the ESP32 to then control the motors.We additionally aim for the ESP32 to communicate with the Jetson as with a communication protocol, they will need to acknowledge each other. The last thing interacting with the ESP32 is the GPS. The GPS is going to be tracking the coordinates of our objects which we will then post to the server. Since the ESP32 communicates with the server we will send the GPS information to the ESP32 as a middle connection. From the esp32 and sim card module we will be communicating information wirelessly via cellular data to reach the main server hosting the website. The server is our website, which will store the coordinates and classifications that it receives. This sensor is to allow our robot to autonomously drive.

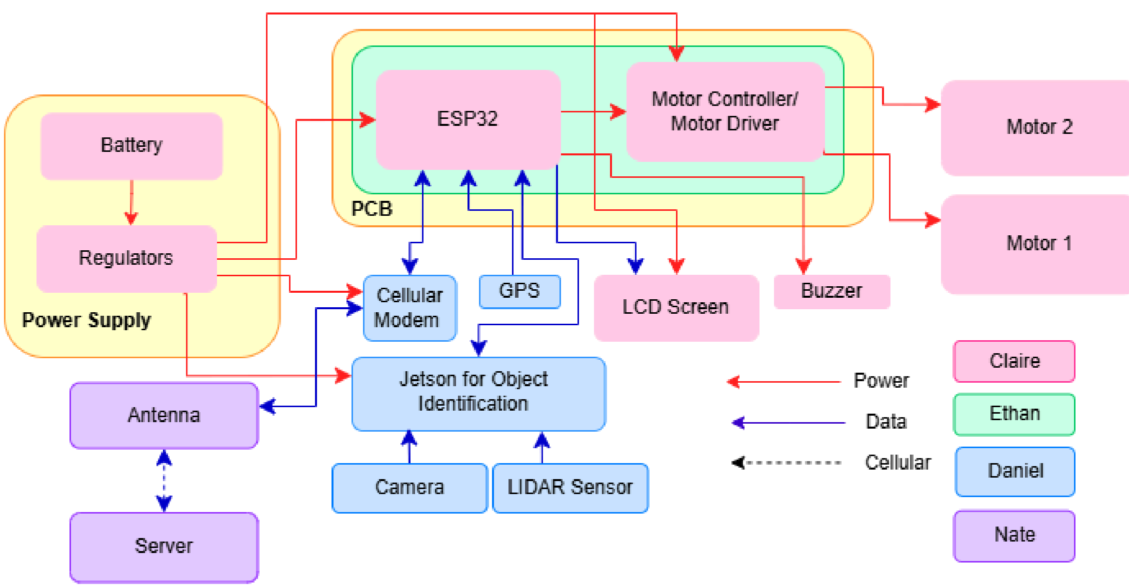


Figure 1.1 — Hardware Block Diagram

1.8 Software Description

1.8.1 Software High-Level Flowchart

Figure 1.3 is the outline of how the software of R.O.A.M. behaves overall and how the subsystems interact with each other. The software serves three major purposes: autonomous driving, object classification, and data transmission. The autonomous driving works by using LiDAR to scan the local environment in front of R.O.A.M. and create a 3-D point cloud of the surroundings. This point cloud will be used to identify the sidewalk and thus provide directional adjustments to the motors. R.O.A.M. will only have two motors, directional adjustments will be provided in the form of the voltage supplied to the left and right motor allowing for turning as well as smaller path corrections. Data transmission will only happen in two scenarios, either a treasure is found or the trash amount needs to be updated. If treasure is found then a package is sent to the ESP32. The ESP32 will take the package and attach a GPS location to it which will be sent to the website making the information and location viewable. If trash is found an internal counter will increment by one, this will be set on a timer that would periodically update the website with how much trash was found.

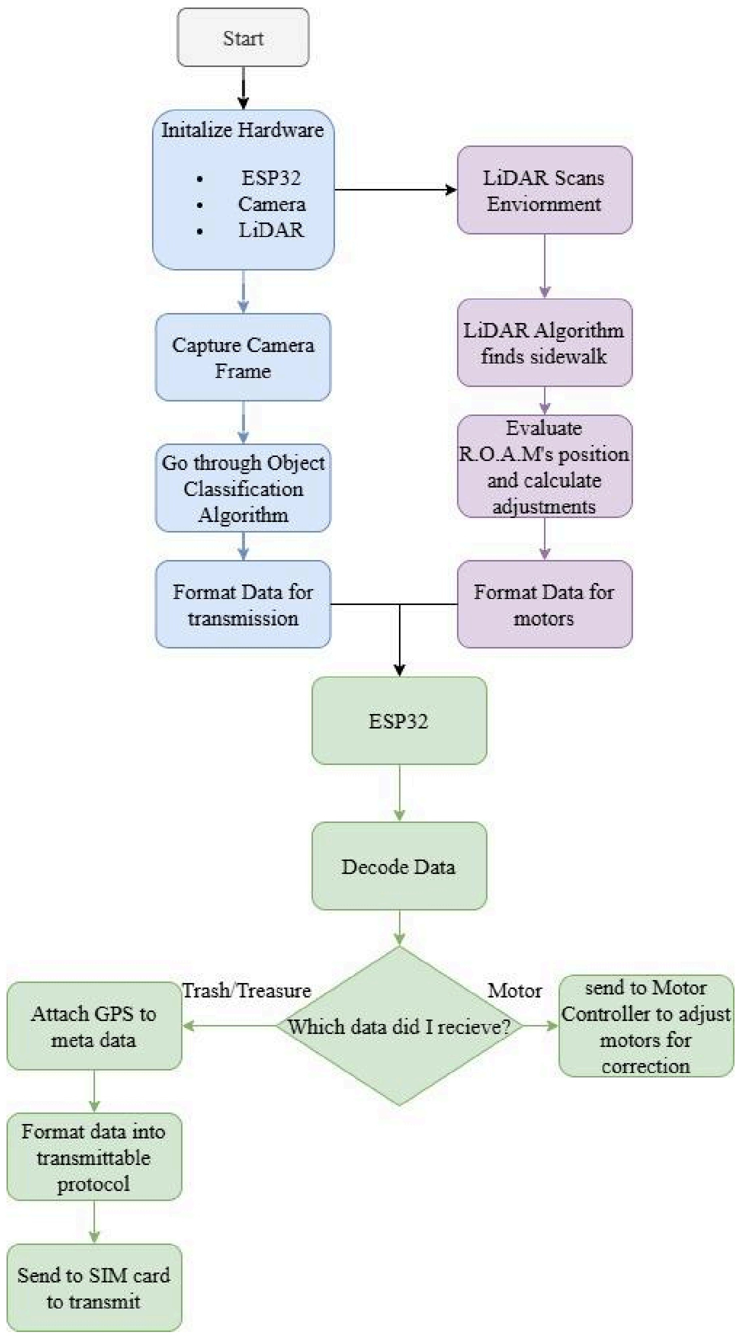


Figure 1.2 — Software Flowchart

1.8.2 Object Classification Flowchart

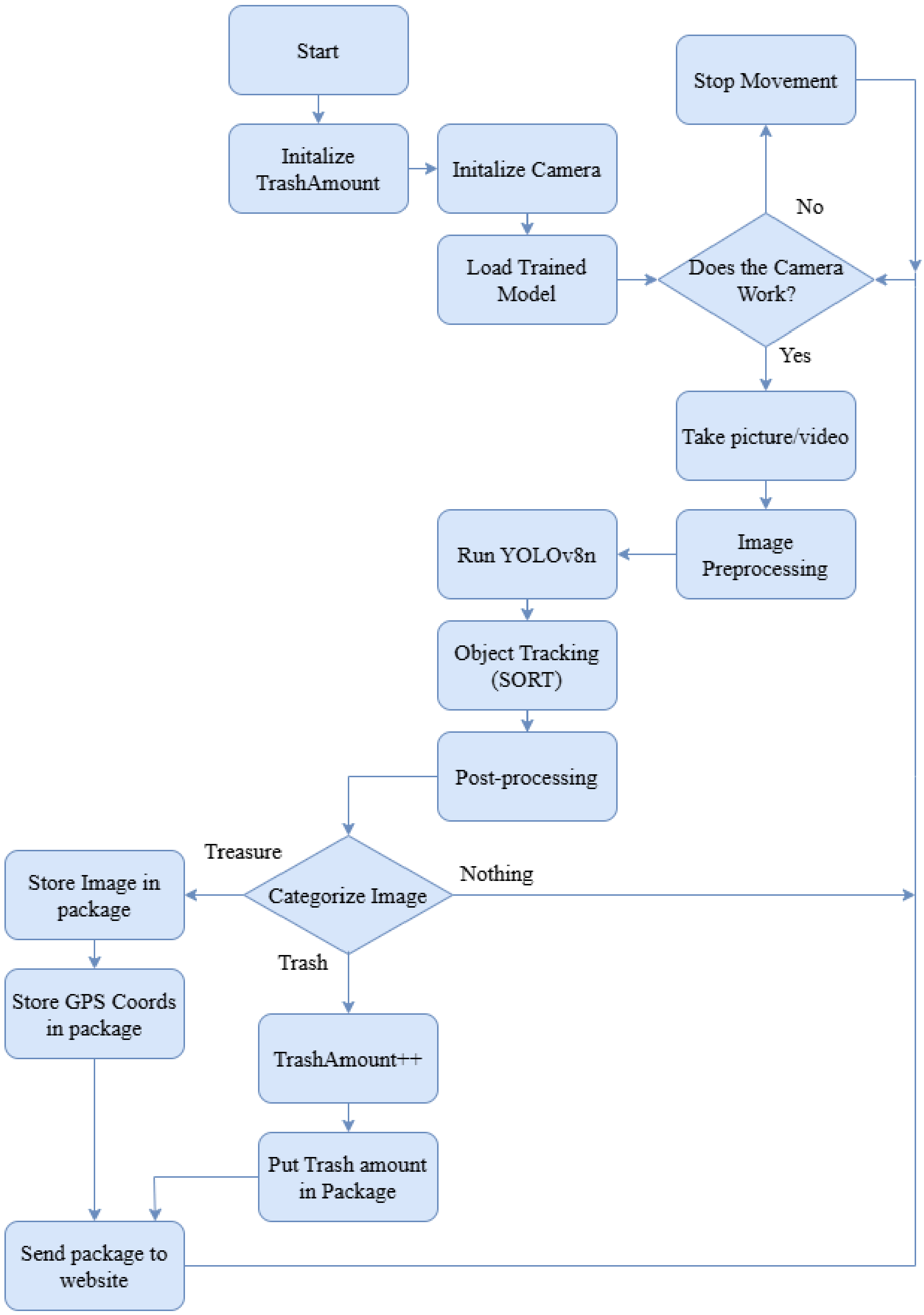


Figure 1.3 — Object Classification Flowchart

Object Classification will provide the core functionality to our project. This will detect and identify objects on the side of the sidewalk as trash or treasure using a machine learning model. The model will be trained on a dataset of images originating from both online and taken in person, this helps narrow down difficult edge cases that R.O.A.M. will specifically encounter. This process will require a camera mounted near the top of R.O.A.M. If the camera stops working R.O.A.M. will be programmed to stop moving to avoid missing any objects. The binary object classification will be done by the YOLOv8n model. This model is equipped to detect and classify objects. YOLOv8n requires preprocessing of images, including changes to the color channel orderings and resolution.YOLOv8n also requires post-processing. YOLOv8n will feed into a SORT object tracking algorithm. This will ensure that objects identified will not be repeatedly identified, leading to less reliance on a stable internet connection, which is a limiting factor. Once the image is identified, it is categorized and handled according to its classification. Then packaged to be sent to the server to be viewed on the website.

1.8.3 ESP32 Software Flowchart

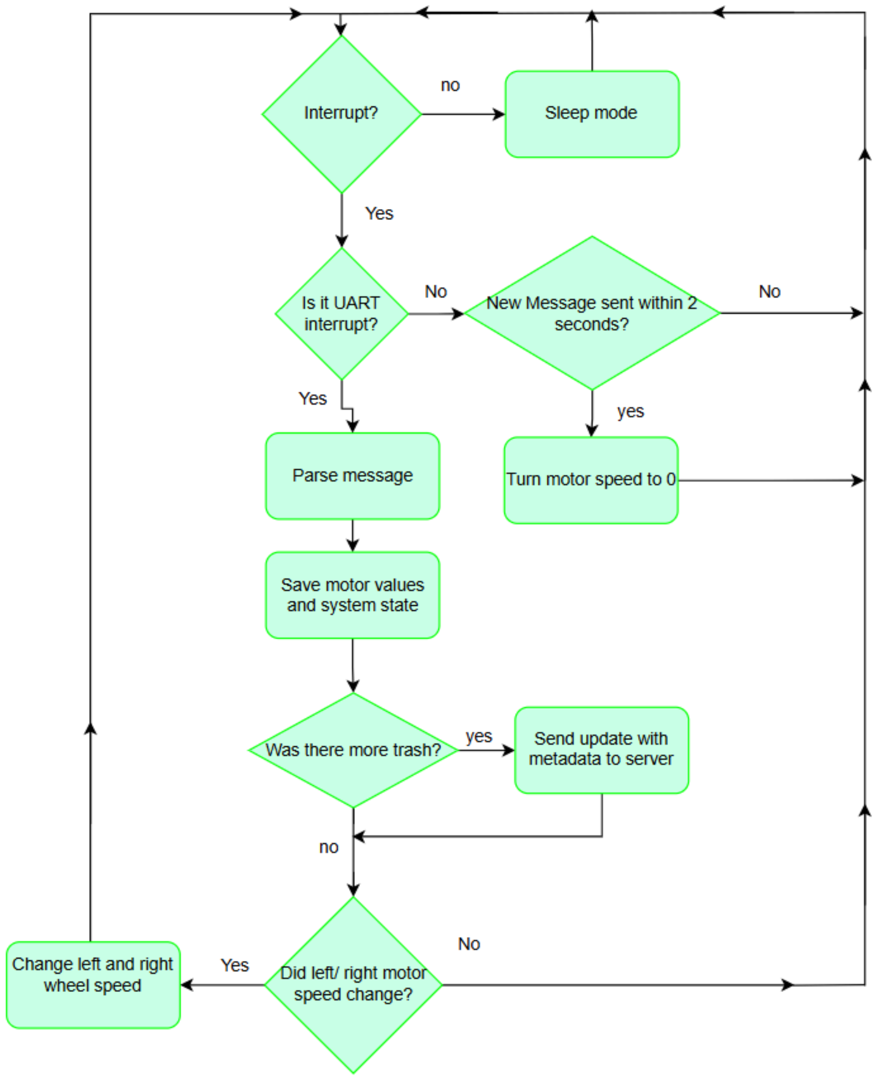


Figure 1.4 — ESP32 Software Flowchart

The two main jobs of the ESP32 module are to control the motors and send meta-data to the server, over cellular data. The meta-data consists of. The flowchart starts by checking if an interrupt occured. If there was no interrupt, the processor goes to sleep mode to conserve power. If there was an interrupt, the program checks to see if it was a UART interrupt. UART will have the highest priority to ensure its data is handled first.. This occurs when the Jetson sends data to the ESP32. If there was a UART interrupt, the message is then parsed, and the motor values are updated. The program then checks if more trash was detected. If more trash is detected it will send the location and the amount of trash or treasure that R.O.A.M. found to the server. The program then checks if the motor speed changed, if so, it updates accordingly, if not the ESP32 goes back to sleep. If the interrupt was unrelated to UART, the system checks if a message has been sent within 2 seconds. If no message has been sent, the program changes the speed of the wheels to 0. This protects R.O.A.M. if a communication failure occurs between the ESP32 and the Jetson.

1.9 Interactive Website

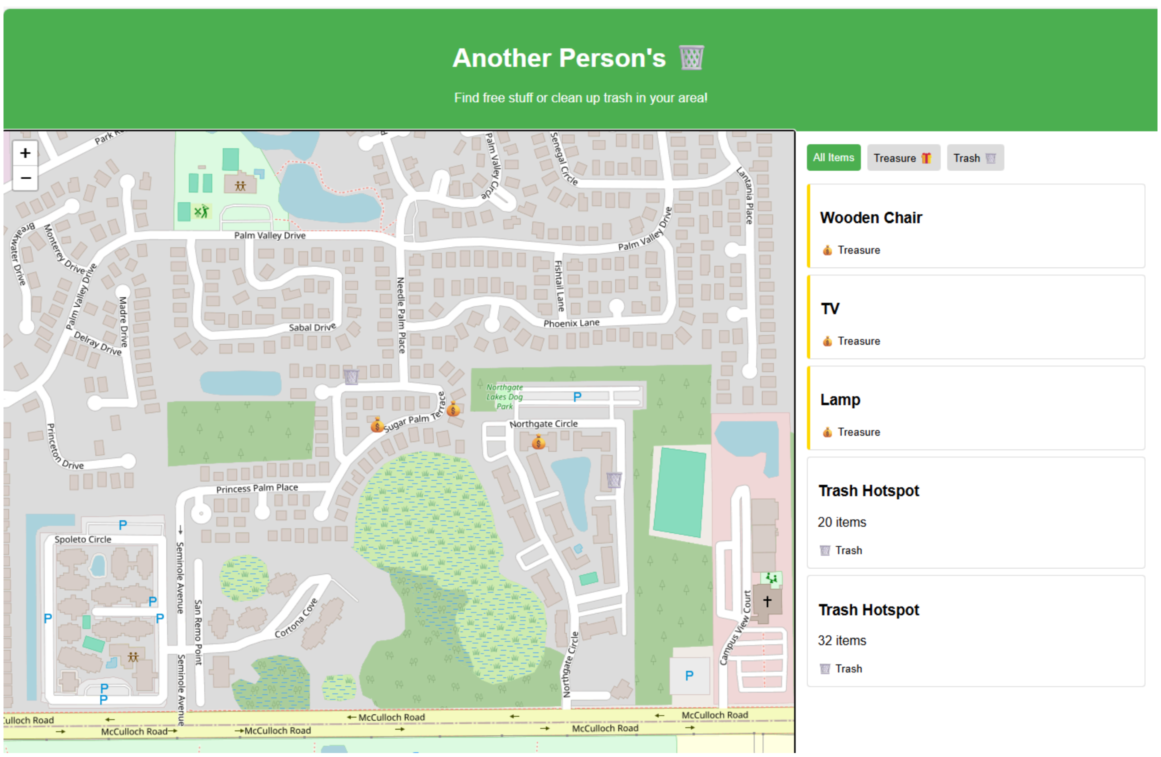


Figure 1.5 — Website Example

The website will be the main source of information gathered by R.O.A.M. for The website will list available treasures in an area with a picture and coordinates of the treasure, so the user can assess the object and deem it worthy of pick-up. As it traverses through a neighborhood, it will identify these objects and report to the website. This feature should focus on ease of use and accessibility. In figure 1.4 we have an example of the website that we will create, this was made in javascript and openstreetmap where we were able to model an idea of what we would like this to look like. This example is without a dedicated database. Trash symbols show the aggregated trash numbers in the area, and the treasure bag is the individual furniture. The meta-data and more details will be included at a later date.

1.10 Prototype Illustration

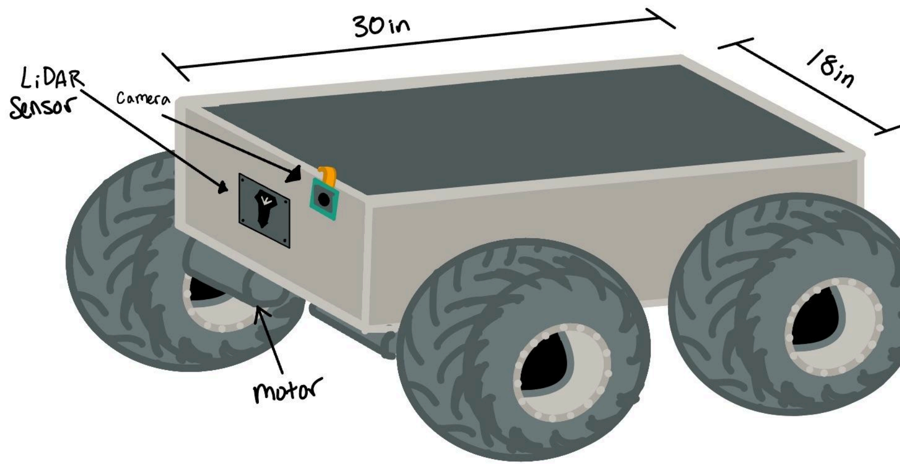


Figure 1.6 — Prototype Illustration

This image above is our proposed idea of what our product will look like. A few things to define are the larger wheels to compensate for cracks in the road, the LIDAR sensor, and the Jetson camera. The camera will be used for object detection, and the LIDAR sensor will be used for autonomous driving. These will both be attached to our Jetson to assist with the image processing before sending the information to the ESP32 for our website. This product has a few things that are contingent upon more research. For example, what specific types of tires are used, the material for the body frame, and even the exact measurements of the body. This image is an approximate description that gives our audience a general idea of what we are creating, along with approximate expectations to consider.

Chapter 7 - Administrative Content

7.1 Budget table

Table 1.3 — Budget Table

Category	**Subcategory**	**Cost**
Software	Autonomous driving	$50
	Object Classification	$50

	Website	$100
Hardware	Motors + Motor Control	$40
	PCB Design	$150
	Processing	$400
	Chassis	$120
	Power	$65
Overhead	Shared Resources	$20
	Misc Supplies	$50
	Wear and Tear	$10
	Contingency Reserve	$80
Total		$1135

7.2 Bill of Materials

Table 1.4 — Bill of Materials

Component	Sub-system	Quantity	Unit cost	Total
Jetson	Autonomous driving	1	$300	$300
LIDAR	Autonomous driving	1	$100	$100
Esp32 chip	Autonomous driving	1	$6	$6
PCB	All	2	$75	$150
Regulators	All	10	$10	$100
Camera	Trash Detection	1	$25	$25
Chassis car kit	All	1	$120	$120
Rechargeable Battery	Power Supply	1	50	50

Solar panel	Power Supply	1	15	15
Antenna	Website	1	50	50
Sim Card	Website	1	$20	$20
Cell Plan	Website	1	$25/month	$25/month
Domain	Website	1	$5	$5
Miscellaneous	All		$64	$64
Ethernet Cable	Autonomous Driving + Trash Detection	1	$8	$8
Total				$1038

7.3 Work Distribution

Given our motivations and backgrounds, we all have chosen our topics to work on and split the workload with. As Claire is the only electrical engineer, we wanted to make sure she had enough with creating a PCB as well as enough power distribution work. As for Ethan, he has shown interest in the embedded software task, he has had a bit of experience in it and wants to do work to strengthen that. Daniel has shown interest in learning more about the machine learning tasks, and the object classification interested him. Lastly, Nathan has shown that he wants to learn more about autonomous driving and website development, which is why he was tasked with this. Across four members, we have further split into a ‘buddy system’ with Claire and Ethan in one pair and Daniel and Nathan in another. This was done to focus each person’s efforts on one field at a time, and was brought up because of some similarities and differences in our tasks. When getting advice about senior design we were always told since it was such a long project, to find something that was enjoyable, something you were interested in. When you want to do the task at hand it will not feel like work, so we all found something we all would find enjoyable.

Table 1.5 — Work Distribution

Person	Role / Work Distribution
Ethan (main)	Embedded software for autonomous driving.
Claire	

Claire (main)	Power supply, PCB design, hardware for RC/autonomous driving.
Ethan	
Daniel (main)	Object classification CV software.
Nathan	
Nathan (main)	Autonomous driving software and website.
Daniel	

7.4 Milestones

Table 1.6 — Senior Design 1 Milestones

Senior Design I				
Start Date	Planned End Date	Req. End Date	Task	Status
5/12/2025	5/18/2025	5/20/2025	Team Recruitment	Completed
5/12/2025	5/20/2025	5/23/2025	Project Idea	Completed
5/25/2025	5/28/2025	5/30/2025	D&C Report	Completed
5/25/2025	7/01/2025	7/25/2025	CV Algorithm Design	Started
6/03/2025	6/05/2025	6/06/2025	Update D&C on group website	Started
6/02/2025	6/24/2025	6/28/2025	Prototype PCB design	Incomplete
6/21/2025	7/04/2025	7/07/2025	Midterm Report	Started
5/25/2025	7/11/2025	7/25/2025	Accomplish Basic Driving	Incomplete
5/25/2025	7/11/2025	7/25/2025	Accomplish Basic Website	Incomplete
6/28/2025	7/21/2025	7/25/2025	Build Prototype	Incomplete
5/25/2025	7/28/2025	7/29/2025	SD1 Final Report (120 Page) & small demo video	Incomplete

Table 1.7 — Senior Design 2 Milestones

Senior Design II				
Start Date	Planned End Date	Req. End Date	Task	Description
		TBD	Review Report	Incomplete
		TBD	Finalize PCB	Incomplete
		TBD	Fix Prototype issues	Incomplete
		TBD	Assemble Final Product	Incomplete
		12/25	Live Demo	Incomplete

Appendix A

References

[1] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000. [Online]. Available: http://opencv.org/

[2]Pranjal Malewar, "Clearbot, a solar-powered, marine trash-collecting robot," *Inceptive Mind*, Jan. 10, 2021. https://www.inceptivemind.com/clearbot-solar-powered-marine-trash-collecting-robot/17108/?utm_source (accessed May 29, 2025).

[3]J. B. Sigongan *et al.*, "GULP: Solar-Powered Smart Garbage Segregation Bins with SMS Notification and Machine Learning Image Processing," *arXiv:2304.13040 [cs]*, Apr. 2023, doi: https://doi.org/10.25147/ijcsr.2017.001.1.142.