R.O.A.M.

Robotic Observer of Abandoned Materials
Group 14

Daniel Claassen
CpE
Computer Vision Hardware & Software

Claire Persinger
EE
PCB & Hardware

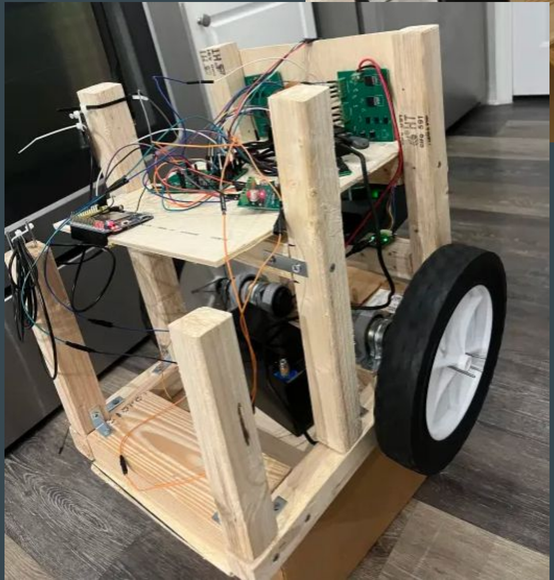
Nathan Little
CpE
Robot Guidance & Web Development

Ethan Robotham
CpE
Systems Integration & Communication

What is R.O.A.M.?

R.O.A.M. - Robotic Observer of Abandoned Material

- R.O.A.M. is an autonomous driving robot
- Wanders the sidewalks in neighborhoods detecting for trash and treasure
- Uploads this information to our website
- Allows users to interact with website



What Sets it Apart

- Autonomous sidewalk navigation using LiDAR
- Real-time trash vs. treasure detection (YOLO on Pi 5)
- GPS-tagged uploads to roambot.dev via LTE
- Dual-processor architecture (Pi for vision, ESP32-S3 for control)

Why is it Important

- Supports cleaner, safer public spaces
- Automates the detection and reporting of abandoned materials
- Helps cities identify valuable or hazardous items quickly
- Reduces reliance on manual inspection and labor
- Creates real-time data for environmental decision-making
- Scales easily across campuses, parks, and urban areas
- Demonstrates the potential of autonomous systems for community benefit

Overview and motivation

R.O.A.M. - Robotic Observer of Abandoned Material

- Autonomous robot to detect and report abandoned objects
- Aim to keep useful items out of landfills and clean up neighborhoods
- Created an interactive website for people to participate in the collection of it items and trash

Goals

Basic / Advanced

- Able to identify treasure and trash while navigating public environments.
- Autonomously follow sidewalks to patrol designated areas.
- Record and upload item locations and classifications to the website.
- Request human intervention when needed (assist).

Stretch

- Categorize treasure into more specific types (e.g., metal, electronics, valuables).
- The website should allow advanced search by location, item classification, or treasure subtype.
- Capture and upload images of identified objects.
- Support extended field operations using a solar charging system.
- Monitor and display battery life to improve mission reliability.

Objectives

Basic / Advanced

- Achieve ≥70% classification accuracy for differentiating treasure vs. trash.
- Maintain continuous sidewalk detection with smooth path-following for at least 25 feet without manual correction.
- Successfully POST item data (GPS, timestamp, classification) to the website within 10 seconds of detection.
- Request assistance via LCD screen when robot can not move

Stretch

- Recognize ≥3 additional treasure subtypes with ≥80% accuracy.
- Implement a searchable database allowing queries by ZIP code, GPS radius, classification, or subtype with <1-second response time.
- Capture and upload at least one photo per detected artifact.
- Maintain robot power for >4 hours using a solar-augmented charging system under normal daylight.
- Display battery percentage with ±5% accuracy on the website or onboard interface.

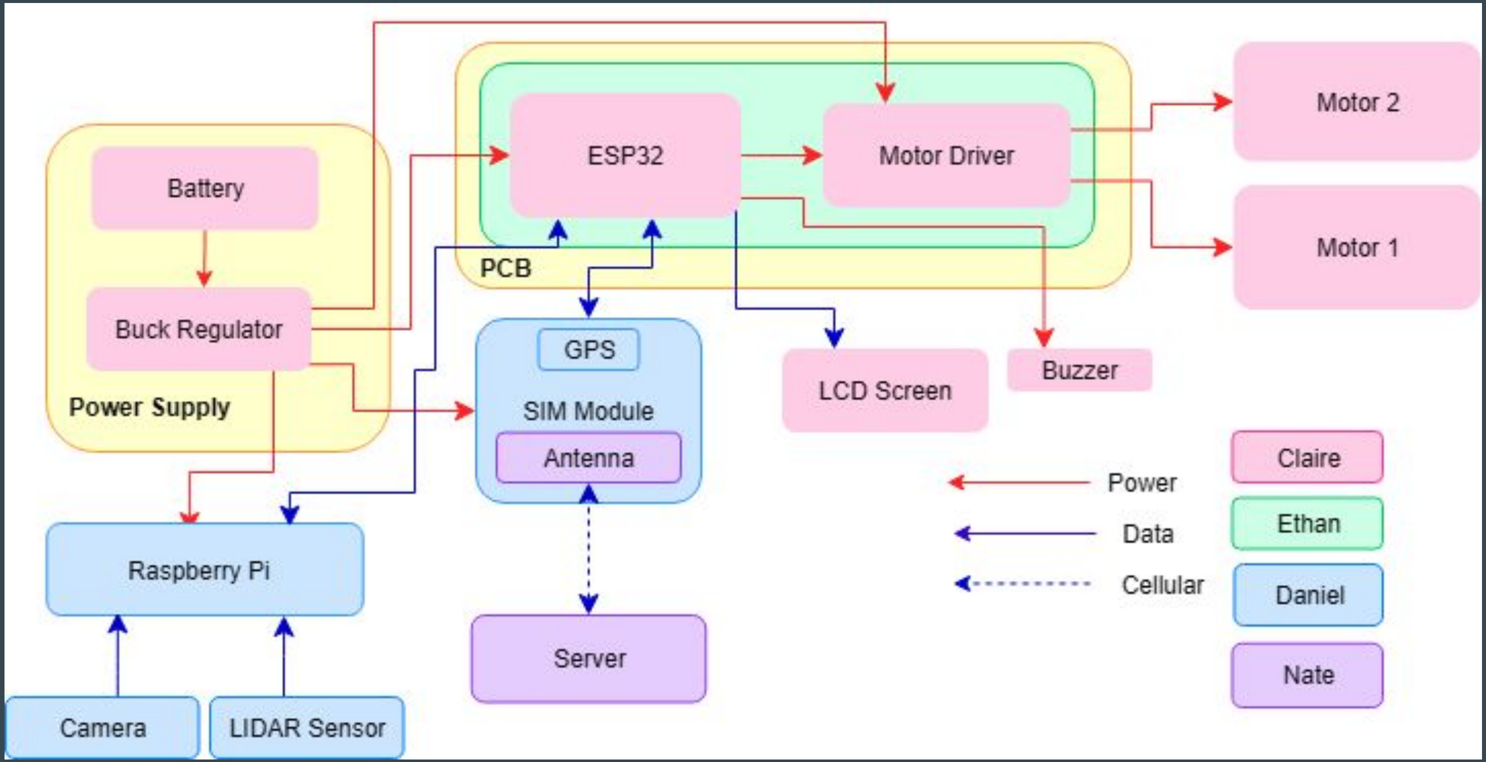
Engineering Specifications

ROAM should have:

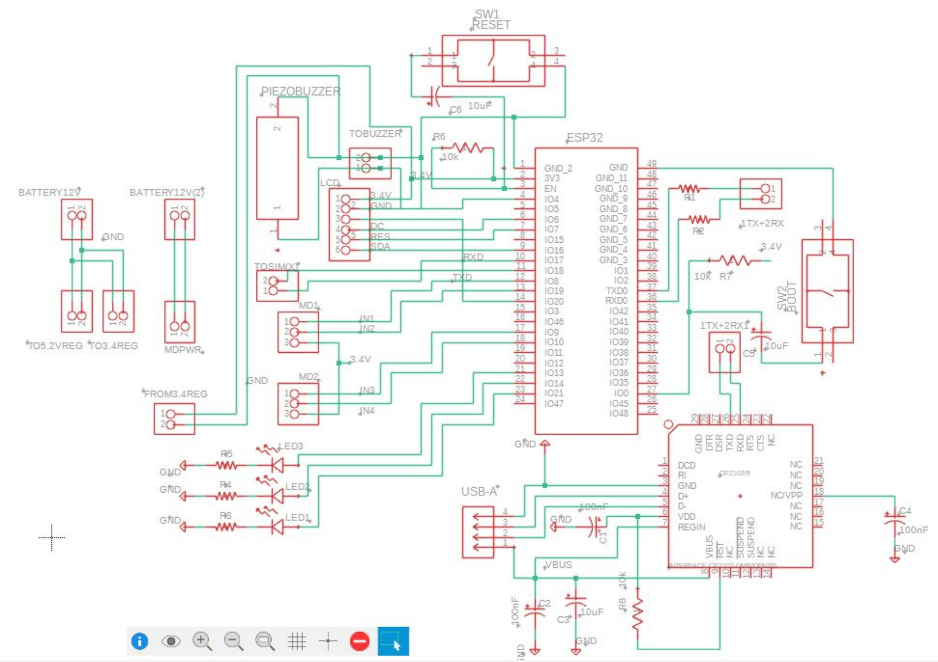
- Treasure identification accuracy greater than 70%
- Ability to travel more than 25 feet without crossing boundaries
- Object detection range between 0.5 and 3 meters

Parameters	**Specifications**
Speed (legal requirement)	< 10 mph
Width (wheel to wheel)	< 46 cm
Traversable Obstacle Height	< 35 mm
Website Data Upload Time	< 10 seconds
Object Detection Response Time	< 2 seconds
Treasure Identification Accuracy	> 70%
Battery Life	> 15 minutes
GPS accuracy	Within 20 m
Distance traveled without going out of bounds	> 25 feet
Object Detection Range (R)	$0.5m < R < 3m$
Power consumption	< 55 W
Cost	< $1500

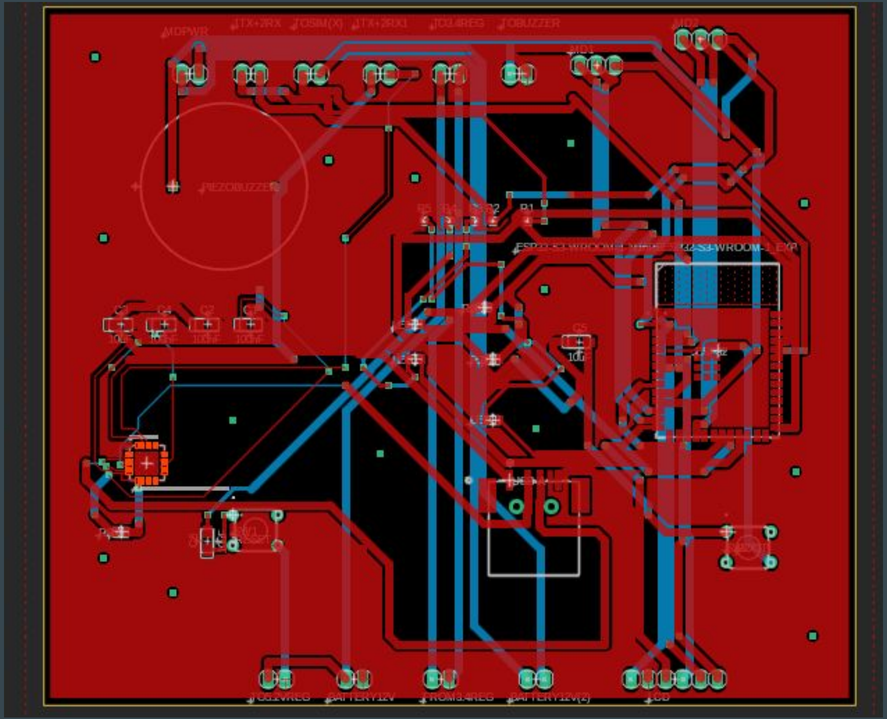
Hardware Block Diagram



PCB Design



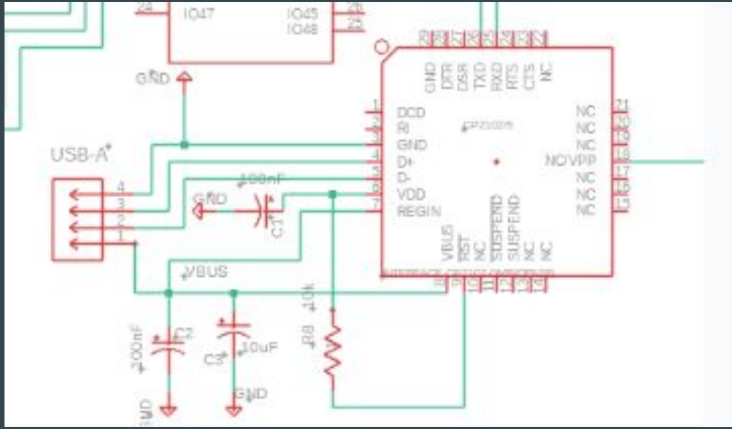
SW1
RESET
PIEZOBUZZER
TOBUZZER
ESP32
C6 10uF
R6 10k
LCD
BATTERY12V
BATTERY12V(2)
GND
TO5.2VREG
TO3.4REG
MDPWR
TOSIM(X)
MD1
MD2
IN1
IN2
IN3
IN4
RXD
TXD
3.4V
FROM3.4REG
R1
R2
1TX+2RX
R7
SW2
BOOT
1TX+2RX1
C5 10uF
LED1
LED2
LED3
R3
R4
R5
USB-A
100nF
C1
VBUS
C2
C3
R8
C4
CP2102



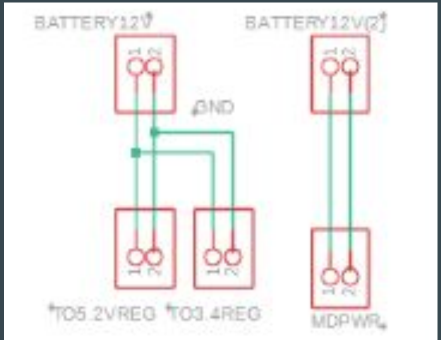
MDPWR
PIEZOBUZZER

PCB Design

UART to USB

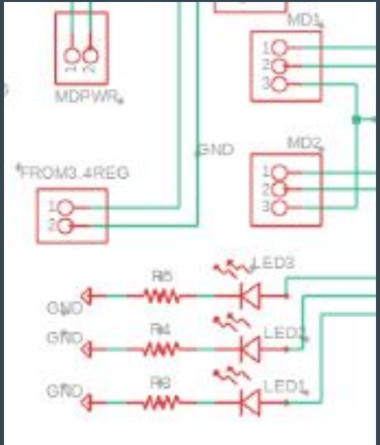


Battery to Reg

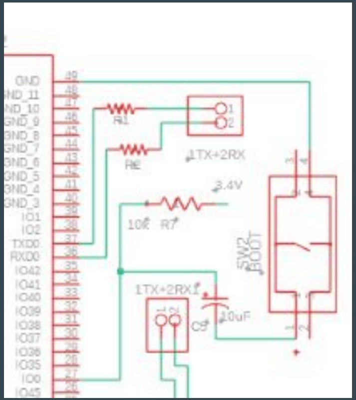


MD + LEDs IO

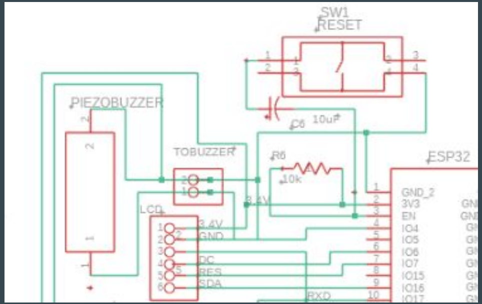
MD + LEDs



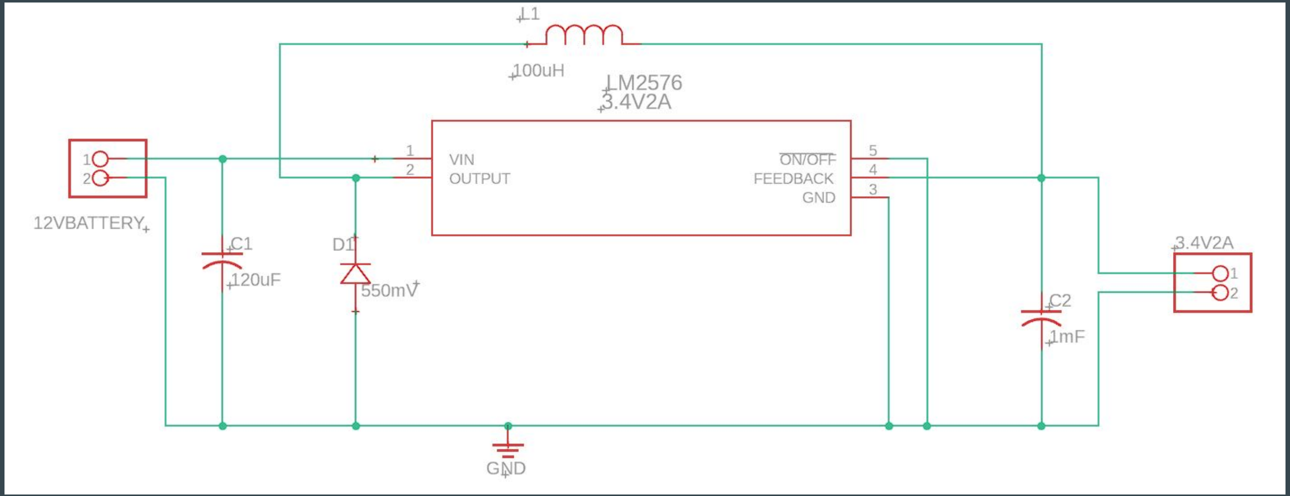
BOOT SW

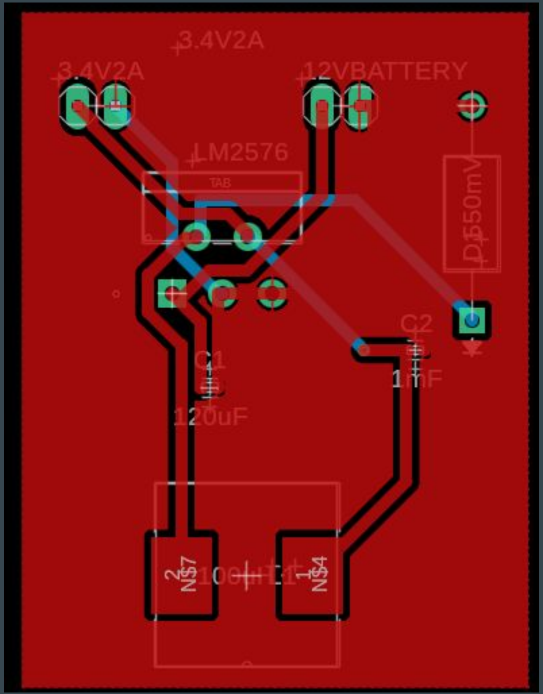


Buzzer & LCD

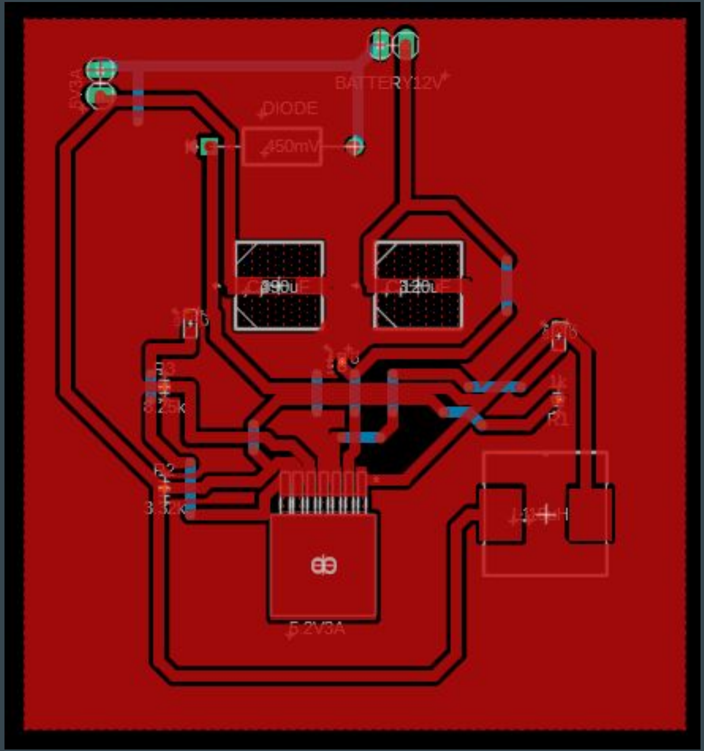


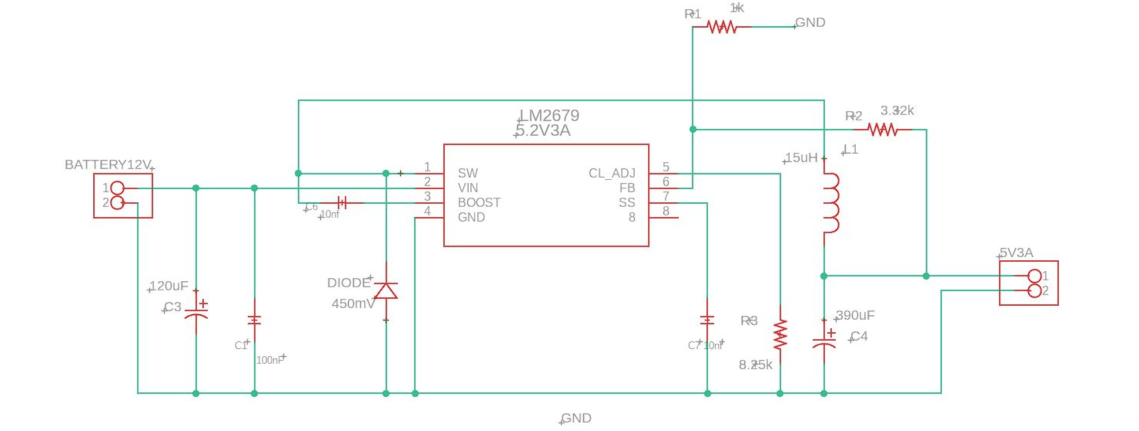
PCB Regulator Design 3.4V 2A



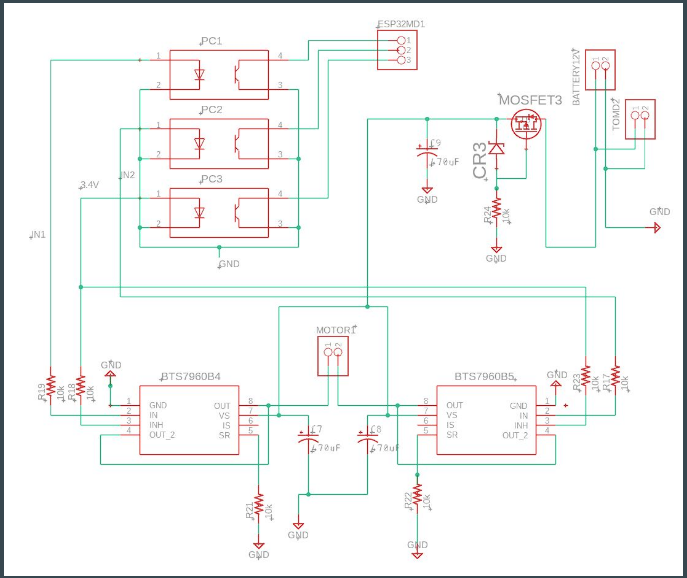


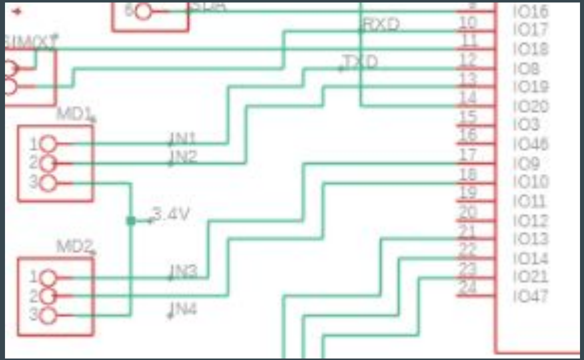
PCB Regulator Design 5.2V 3A

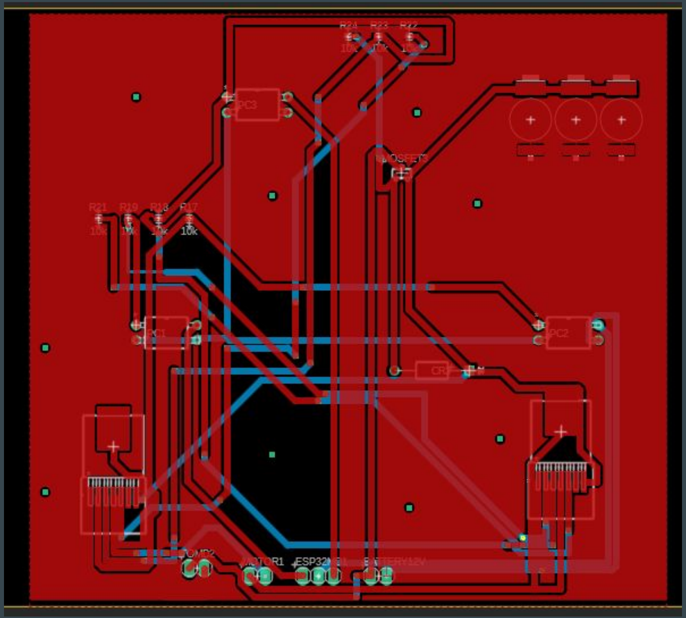




PCB Regulator Design Motor Driver





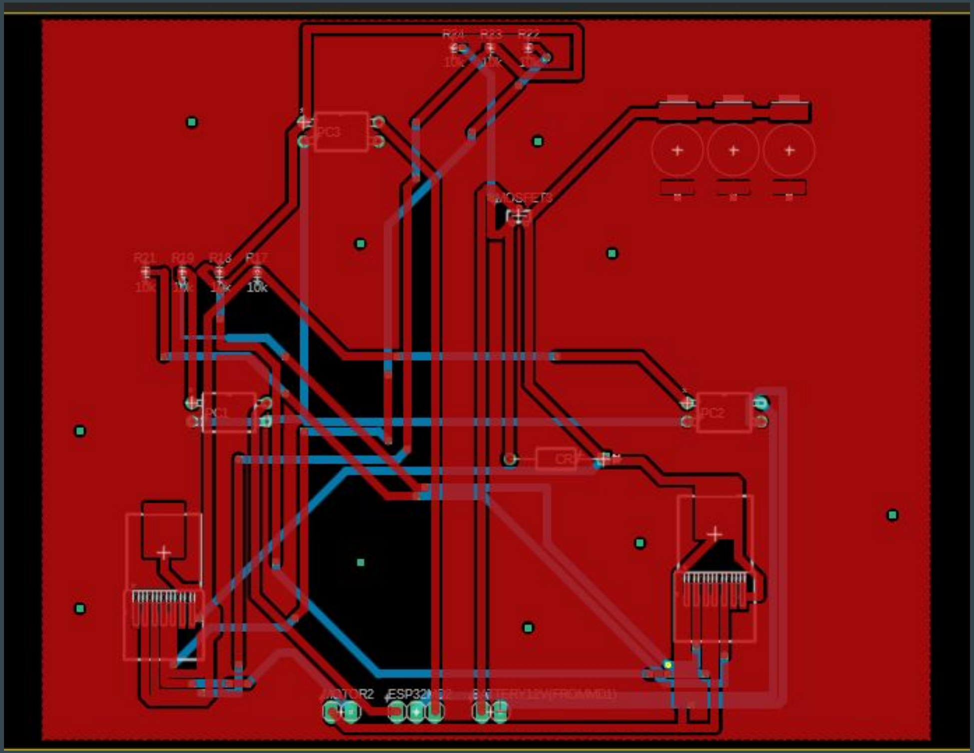


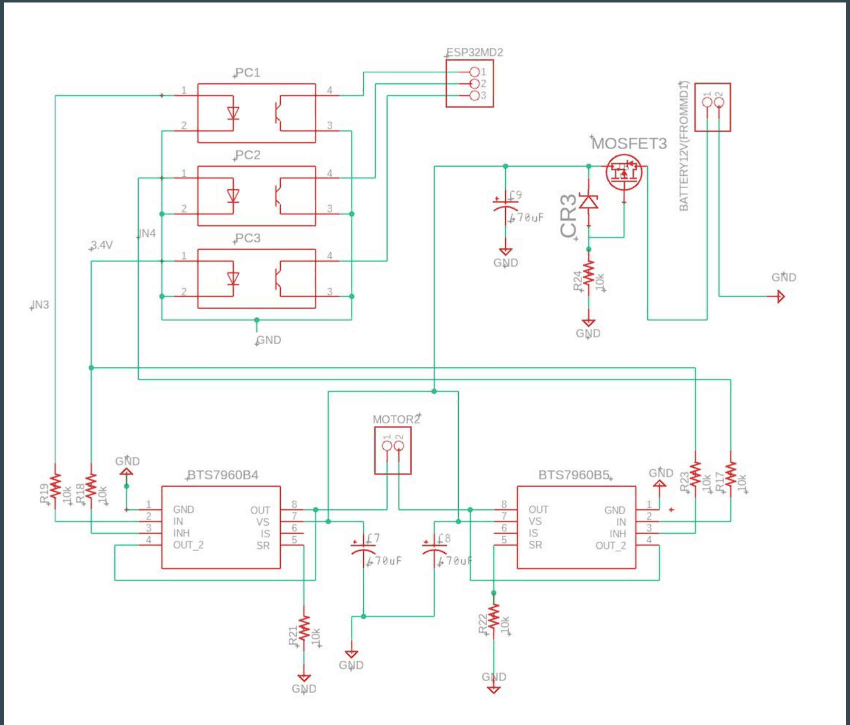
Logic

- IO8/IO19
- IO9/IO10

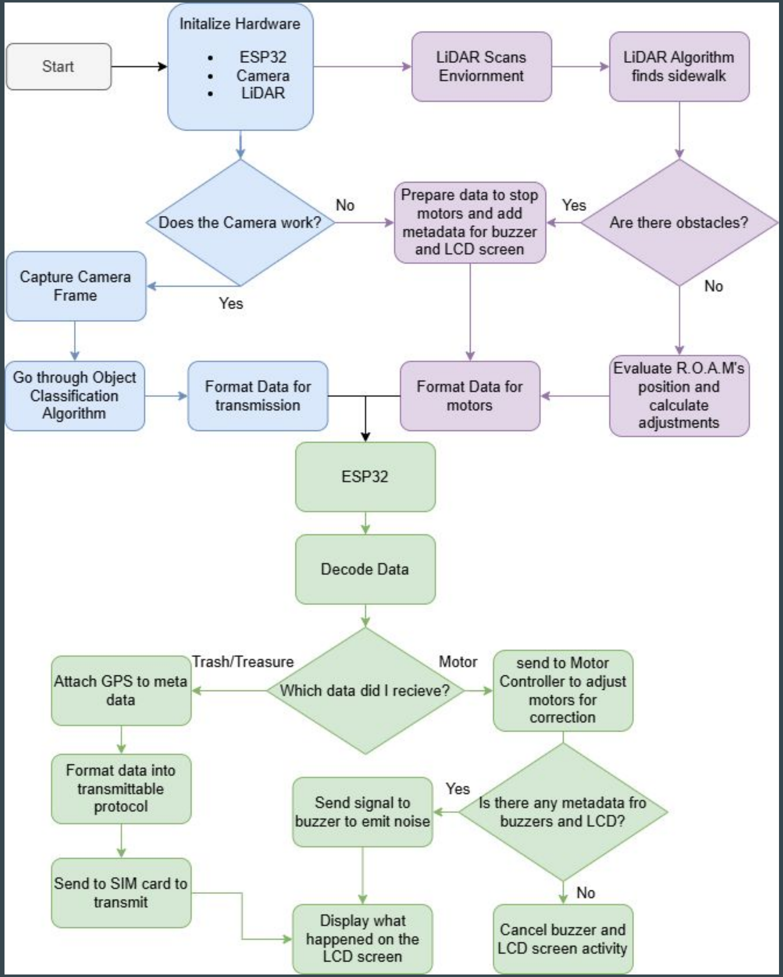
3.4V Logic Pwr

PCB Regulator Design Motor Driver



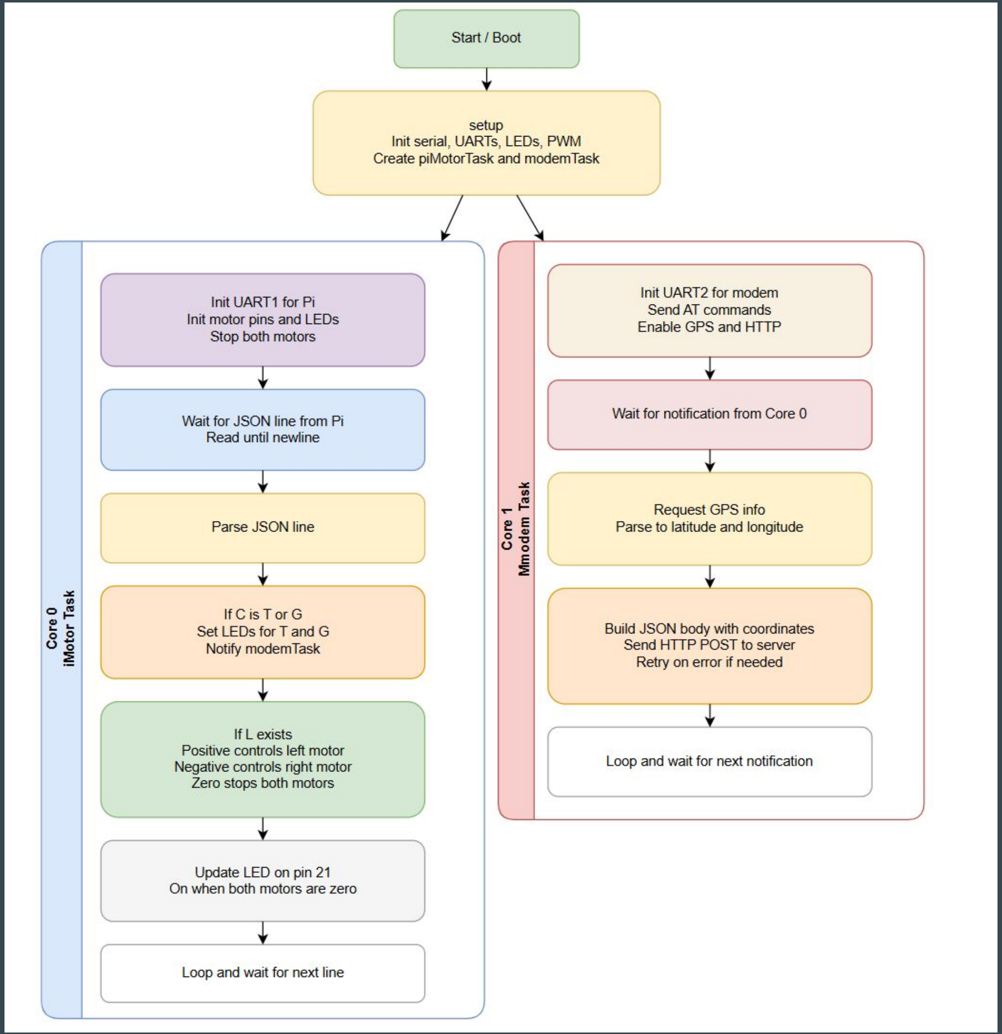


Software Flow Chart

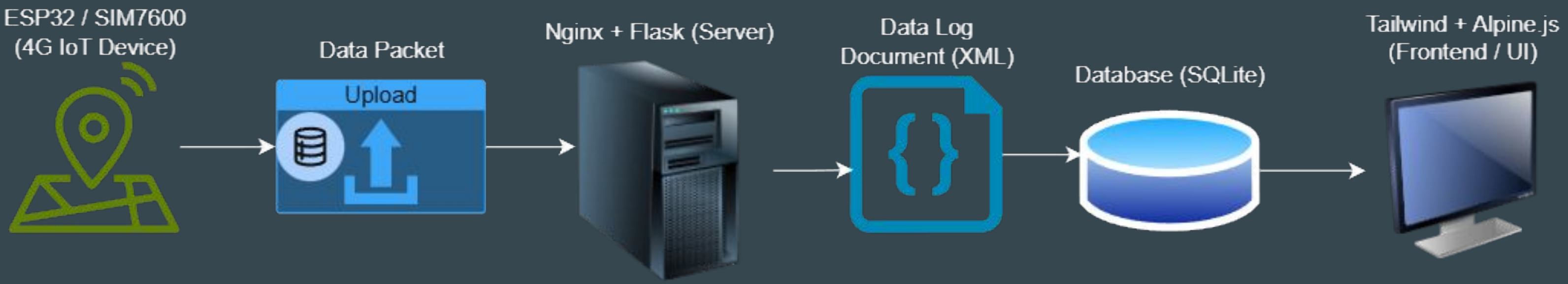


ESP32 software flow chart

- Responsible for:
  - Motor control
  - Sending data to website (using AT commands/data to SIM module)
  - Parsing data from Pi
  - Controlling user interface



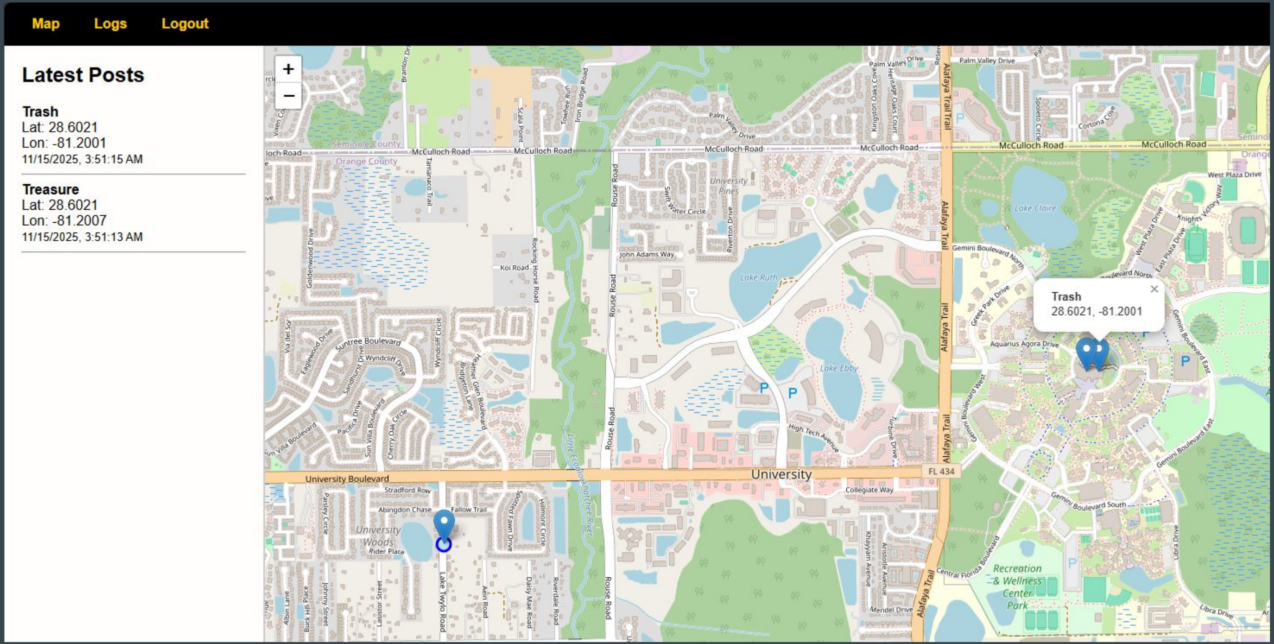
Web Service Pipeline



- Device → Server: ESP32/SIM7600 sends GPS + classification data over HTTPS
- Server → Database: Nginx + Flask handle requests, log activity, and store results in SQLite
- Database → Frontend: Tailwind/Alpine.js UI with Leaflet maps displays real-time data

UI/UX Design

- Current system uses Tailwind + Alpine.js for a lightweight 4G-friendly UI
- OpenStreetMap tiles for free cached GPS mapping
- Flask backend for real-time updates.



Object Classification

Object classification accuracy is based on:

- Dataset size and quality
- Captured frame quality
- Environmental Variability

Object Identification Accuracy

	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5	Trial 6	Trial 7	Trial 8	Trial 9	Trial 10
Trash Detected	7/7	4/5	4/6	6/6	4/4	7/7	7/8	10/10	9/10	0/2
Furniture Detected	1/3	3/5	3/4	3/4	1/6	2/3	2/2	0/0	0/0	6/8
Total Objects Detected	8/10	7/10	7/10	9/10	5/10	9/10	9/10	10/10	9/10	6/10
Accuracy	80%	70%	70%	90%	50%	90%	90%	100%	90%	60%

Total Accuracy: 79%


ID1:Trash 0.84
ID2:Trash 0.73
ID3:Trash 0.59

ID1:Trash 0.74
ID2:Trash 0.68

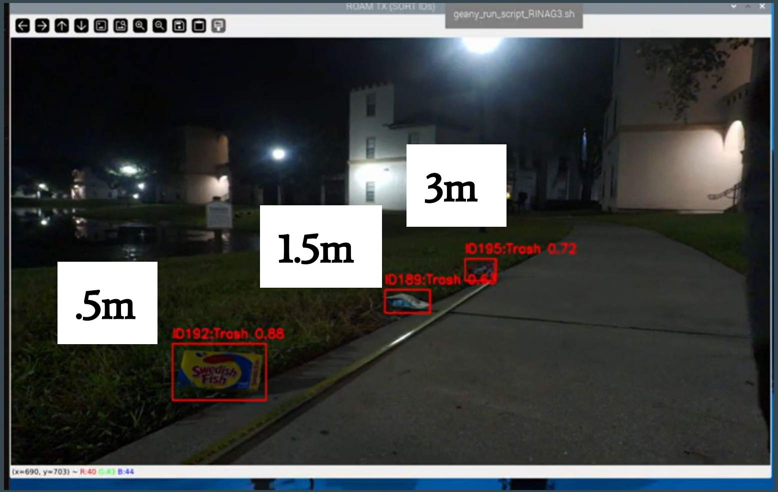
ID1:Furniture 0.32

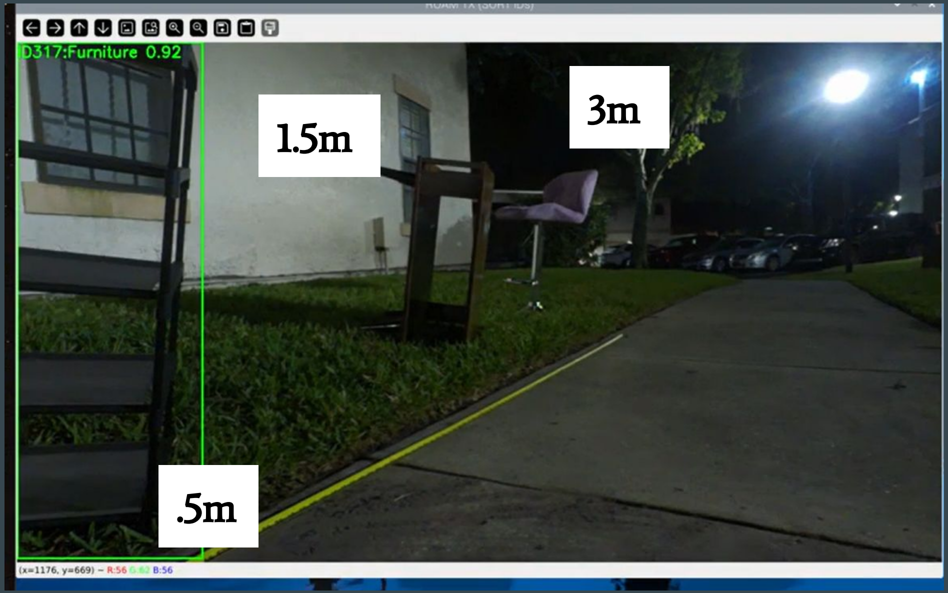
ID279:Furniture 0.60

Object Identification Distance

	Trash #1	Trash #2	Trash #3	Trash #4	Trash #5	Trash #6	Shoe Rack	Coffee Table	Bar Chair	Dining Chair
0.5m	Y	Y	Y	Y	Y	Y	Y	Y	X	Y
1.5m	Y	Y	Y	X	Y	Y	Y	Y	X	Y
3m	Y	Y	Y	X	X	X	X	X	X	Y
Score	3/3	3/3	3/3	1/3	2/3	2/3	2/3	2/3	0/3	3/3

Object Identification Distance .5M < R < 3M



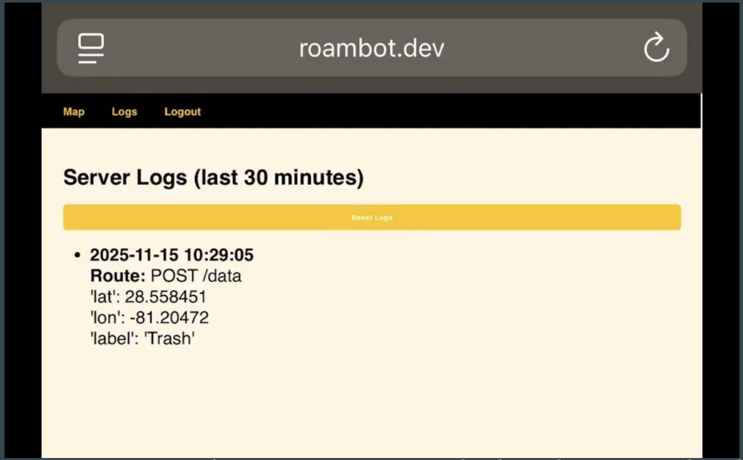


System Services and Process Control

- .service files autostart ROAM's programs on boot
- Ensures those programs run in correct order
- Automatically restart programs if they crash
- Allow the use of systemctl commands
- Allow the use of journalctl for debugging

ESP32 and SIM Upload Test

- ESP32 successfully sent GPS data to website when Trash/Treasure was detected

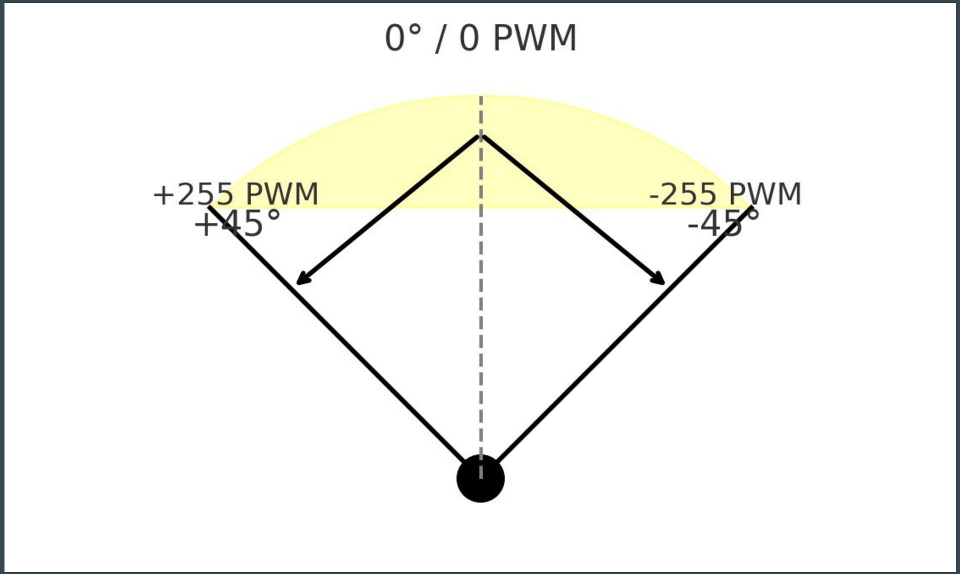


Data sent from sim module that includes GPS coordinates, time sent, and If the object was trash or treasure

Front and Backend Interfaces

- Flask + Gunicorn backend came online quickly on a small(free) Oracle VM, with SQLite for lightweight storage of GPS and classification data.
- Nginx reverse proxy configured with relaxed TLS (TLSv1.1–1.3, !DH ciphers) to ensure SIM7600/ESP32 compatibility, while also handling HTTPS termination.
- Automated certificate management is a best-practice step for production which we don’t have right now.
- Added logs page in Flask for quick debugging of ESP32/SIM7600 uploads
- Verified end-to-end data flow: device → Nginx → Flask → SQLite → map display
- Stress-tested API endpoints with malformed and valid requests to confirm resilience and reliable Flask responses.

LIDAR Testing



- Used the LiDAR to scan a ±45° field of view ahead of the robot and identify the base level of the sidewalk in front of it.
- Converted the angular offset of the detected “center” into a steering value from –255 to +255, where:
- +255 = strongest left turn
  –255 = strongest right turn
  0 = straight ahead
- Steering values were sent to the ESP32, which adjusted motor output by ramping PWM to the left or right wheels.

Peripheral testing

Peripherals:

- Controlling motors
- LED
- LCD screen
- Sending data to website

Motor Technology and Part Selection

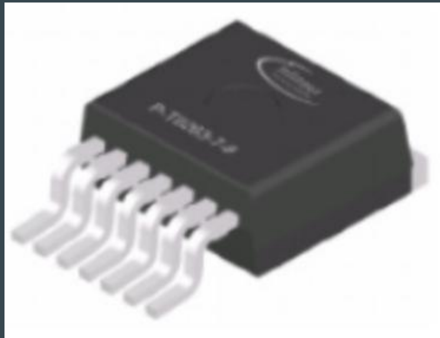
Motor	Pros	Cons
25D 12V 117 RPM Spur Gearmotor	• Compact and lightweight • Inexpensive • Easy to mount in small frames	• No encoder for feedback • Spur gears wear faster under load • Lower long-term durability • Less torque than planetary types
36mm 12V 170 RPM Planetary Gearmotor w/ Encoder	• Good balance of torque and speed • Planetary gearbox • Built-in encoder	• Generic • Encoder resolution is low • Shaft may require adapter for some hubs • Limited official documentation
goBILDA 5202 Yellow Jacket 19.2:1	• High-quality, branded motor • Built-in encoder • Strong planetary gearbox • Excellent documentation and support	• More expensive • 8mm REX shaft may not be compatible with non-goBILDA • Slightly larger and heavier • Possibly overbuilt for early prototypes

- Brushed DC 12V Motor
  - PWM control, high torque options
  - Acceptable stall current
  - Ease of integration
  - Excellent documentation

Motor Driver Part Selection

Technology	Supply (V)	Logic In (V)	Continuous Current (A)	Voltage Drop per Leg	Integration Effort
L298N Breakout	5–35	5	2.0	~2.0 V	Plug-and-play, not 3.3 V compatible
DRV8833 Breakout	2.7–10.8	1.8–6	1.5	~0.3 V	Low effort, but supply is marginal
TB6612FNG Breakout	2.5–13.5	1.8–5	1.2	~0.2 V	Ideal for 3.3 V GPIO, thermal limits apply
IRF3205 Dual-MOSFET Board	3–36	Requires ≥10 V	10	~0.05 V	No PCB needed, but not logic-compatible
IR2104 MOSFETs (Custom PCB)	5–20	3.3–20	≥1.5	~0.025 V	Medium effort, full layout control
Bts7960	6-27	3.3-5	2	~.2	Easy layout to incorporate into this design

- BTS7960
  - Can handle up to to 43A
  - Overcurrent protection
  - Undervoltage lockdown
  - Bidirectional use
  - PWM speed control
  - Smooth logic integration
  - Information widely accessible



Battery Part Selection

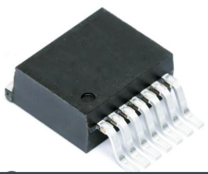
Battery	Pros	Cons
Bioenno 12V 12Ah $LiFePO_4$	• Steady voltage under load • Built-in BMS for protection • Long cycle life • Lightweight and compact • Reliable for mobile use	• More expensive • Needs $LiFePO_4$-specific charger • Heavier than smallest $LiFePO_4$ packs
TalentCell 12V 12Ah $LiFePO_4$	• Regulated 12V output • Lightweight and compact • Built-in protection • Plug-and-play wiring • Good cycle life	• Limited current output • Not ideal for high draw setups • Requires correct $LiFePO_4$ charger
Mighty Max 12V 12Ah SLA	• High capacity • Inexpensive • Easy to find • Simple to charge with basic charger	• Heavy and bulky • Voltage drops under load • Shorter lifespan • Poor energy efficiency for mobile use

- TalentCell 12V
  - Compact
  - 2000+ Cycles
  - 153.6Wh
  - Moderate Price

Regulator Part Selection

Type	Pro	Con
MP1584	• Fast switching freq • 4V-28V	• 2A • Thermal efficiency
TPS5430	• 3A • Moderate Efficiency	• Higher voltage range
LM2576	• 3A • 1.23V-37V • Variability	• Slower switching freq
LM2679	• 5A • 1.2V-37V • Variability	• Sensitive to noise

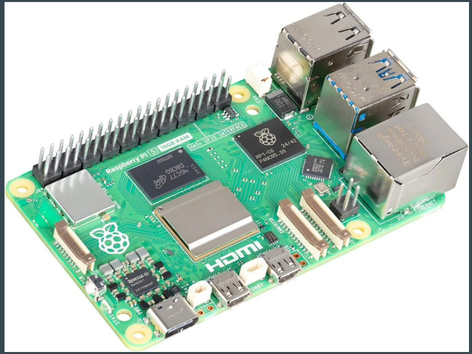
- LM2576SX-ADJ/NOPB
  - 3.4V 2A
  - Fixed 52kHz
  - Built in current limit
  - Availability
- LM2679SX-ADJ/NOPB
  - 5.2V 3A
  - 260kHz
  - Current limiting, thermal shutdown
  - Availability



Edge Device

Raspberry pi 5 was selected for its ease of use, affordability, and adequate performance

Device	Cost	Familiarity	Power Consumption	Performance
Raspberry PI 5	~ $80	Very	< 10W	Medium
NVIDIA Jetson Orin Nano	~ $250	None	< 15W	High
Arduino Portenta H7	~ $100	Little	< 2W	Low



The 16gb RAM version is required for reliable functionality

Camera Part Selection

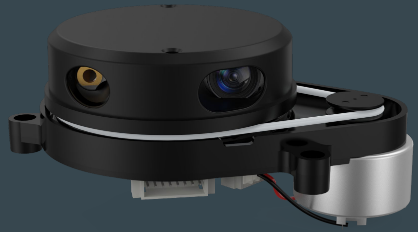
Ideal Camera
> 7 MP ~ 100° HFOV Global Shutter HDR Support

Device	MP	HFOV	Shutter Type	HDR Support	Cost
Raspberry PI Camera Module 3	12 MP	102°	Rolling	Yes	~ $25
Arducam OV9782	1 MP	70°	Global	No	~ $60
Intel RealSense D435	2 MP	85°	Global	No	~ $300

LIDAR

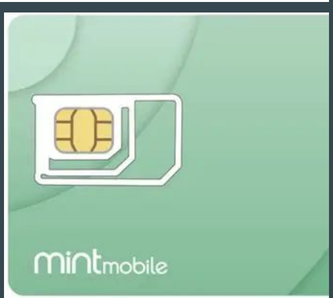
LiDAR Model	Pros	Cons	Cost
RPLIDAR A1	• Extremely affordable • Straightforward integration	• Coarse scan • Limited refresh rate	$100
YDLIDAR X4	• Higher scan frequency • Low integration effort	• Shorter range • Similar angular fidelity	$100
HLS-LFCD2	• Lowest price point • Ideal for testing hardware stacks	• Unstable data • Lacks official support	$80
Benewake TF02	• Best balance of range and detail for field trials	• Requires parser • UART wiring considerations	$220
Yahboom LD06	• High-density output • Stable DTOF performance	• Heavier • Specialized control circuitry needed	$250

- No significant scan rate difference in the 2D LIDAR market.
- ROS1 and Docker support.
- Same price as the A1 but with 2Hz higher scan rate.
- More expensive models add range, not data.



SIM Card Module

Component	Pros	Cons
SIM7600G-H	• Fast upload speeds (150 Mbps) • GNSS support • Includes antennas and cables • Official support	• Larger footprint than other boards • Slightly higher power consumption
SIM7000G	• Energy efficient • Built in GNSS	• Slow upload speeds (375 Kbps) • Limited network availability
A7670E	• Cheaper than CAT4 modules • Multiple interfaces (UART,USB)	• Slower upload rates (10 Mbps) • Can be unreliable for multiple simultaneous protocols



- Best data speeds in price range.
- Accurate and Built-in GNSS for gps position data
- Comes as a kit complete with a breakout board and antennas.
- Documentation on AT commands and detailed datasheet.

Microcontroller

Microcontroller	Pros	Cons
ESP32-S3-WROOM-1U-N16R8	• Three UART pins • 16 MB flash + 8 MB PSRAM = huge head-room • Massive Arduino/ESP-IDF ecosystem	• About 50 mA active, higher than ultra-low-power counter parts
MSP-EXP432P401R	• Three UART pins • Ultra-low-power modes	• Lower clock rate (48 MHz) • Low SRAM (64kB)
Nordic nRF52840	• Tiny current draw when in deep sleep (Micro amps) • BLE radio could be enabled later for field debug	• Only 2 UART pins

- Well supported
- Lots of space
- Cost efficient

Bill Of Materials

Component	Sub-system	Quantity	Unit cost	Total
Raspberry Pi	Autonomous driving	1	$300	$300
LIDAR	Autonomous driving	1	$100	$100
Esp32 chip	Autonomous driving	1	$6	$6
PCB	All	5	$100	$250
PCB Components	All	*	$250	$300
Camera	Trash Detection	1	$25	$25
Chassis car kit	All	1	$120	$120
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				$1273

Bill Of Materials

- Under $1500
- Attempting to be cost efficient
- Software (~$200)
- Hardware (~$1000)
- Overhead (~$280)

Division of labor

Person	Role / Work Distribution
Ethan (main)	Embedded software and cellular communication.
Claire	
Claire (main)	Power supply, PCB design, and autonomous driving hardware.
Ethan	
Daniel (main)	Object classification CV software.
Nathan	
Nathan (main)	Autonomous driving software and website.
Daniel	

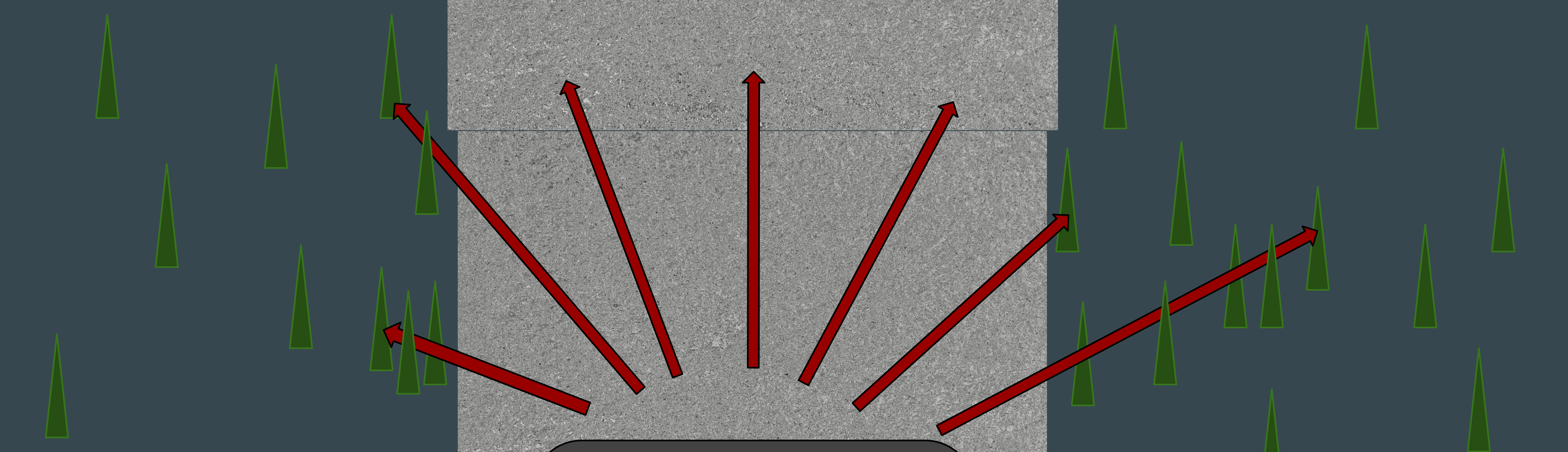
Thank you!

Questions?

LiDAR

2D LiDAR Trials

2D LiDAR Drawbacks:



Grass is a noisy texture and there is no way to reliably tell the difference between grass and sidewalk

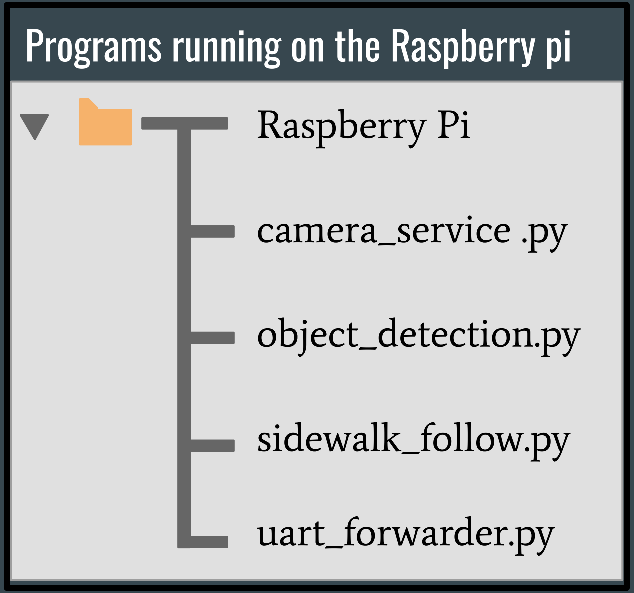
A 2D LiDAR beam fired into grass will hit random blades of grass at varying distances provided too little information to infer reliably

VIDEOS OF 2D LIDAR

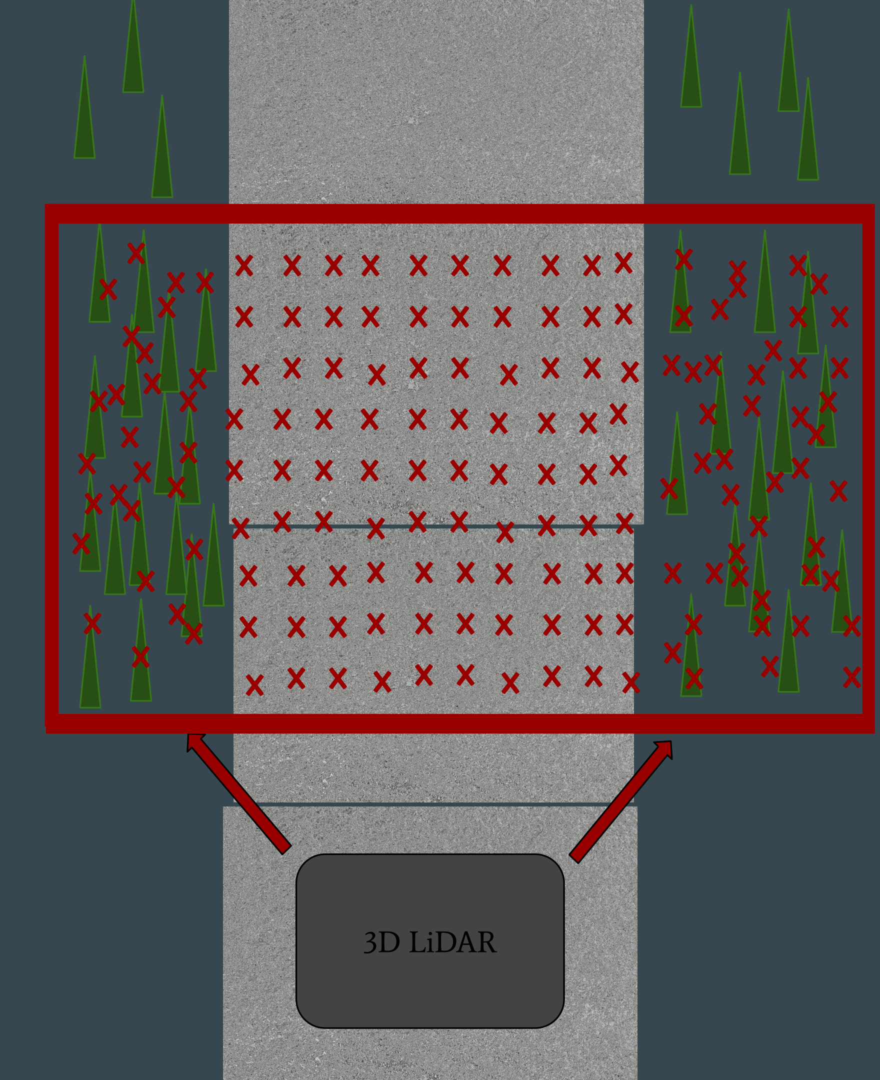
Alternative Solution

Use 2D LiDAR only for obstacle avoidance and use the captured frames to detect and follow the sidewalk

We would need a program to control the camera and send frames through a Unix socket to the other programs for individual analysis



3D LiDAR



The 3D LiDAR would be able to give us enough information two identify two planes:

- A noisy grass plane
- A flat sidewalk plane

And use those planes to stay on the flat plane that represents the sidewalk using

- RANSAC
- Elevation Map with Roughness-Based Terrain Classification