Bin Diesel: Voice Guided Moving Bin

Motivation:

In this course, we have progressively built a deeper understanding of the fundamentals that drive modern electronic systems, from basic power regulation and motor control to closed-loop feedback and navigation. Through it all, one problem persisted: the workspace gets cluttered constantly with wire clippings, tape, trimmings, and many more tiny pieces of trash. It is, however, inconvenient to keep a trash can around the workspace, since there is not enough space to place one conveniently with two people working there.

For our final project, we want to extend these learned fundamentals into a system that solves this problem in a fun and exciting way: a robotic trash bin that comes when called! Thus, the base goal of our project is to augment our car (with a trash bin fitting to its top) to be able to detect a voice, estimate the direction of arrival of the caller, navigate in the direction of the voice, and stop at the foot of the user. After stopping for a few seconds to collect the discard, the car will then return to its point of origin, with the whole process being conducted in a timely manner.

Our motivation stems from seeing how real-world systems like the Roomba combine many of the subsystems we have learnt about in this course, like power electronics, sensor integration, control algorithms, and communication, into one coordinated design. By creating our own version of such a system from scratch, we hope to explore the intersection of hardware, sensing, and intelligent control in a way that reflects what we've learned in this course. It will be really cool to bring together a lot of tools and systems we have so far used in a "siloed" manner relative to each other.

Overview:

The system will work as follows: the user's voice will be detected via a mic array (the array is used so that we have relative directional information between each individual microphone), the audio signal will be sent to a Raspberry Pi where it will be processed (DOA estimation and smoothing), the Raspberry Pi will send steering and driving instructions to the PSoC, which will then interact with the car's hardware accordingly. We also intend to allow users to use natural language commands, and we will then use an LLM to assist us with converting this into commands for the car - the Raspberry Pi is also critical for this. Stopping will be triggered when a distance sensor fitted to the front of the car detects an obstacle, at which point it will stall for a predetermined amount of time, before the car turns 180° and returns to its original position.

We will take a few assumptions to make the project more tractable given the limitations of budget and time: 1) assume a straight path exists between the car and the caller, such that the first obstacle encountered by the car is the caller, and 2) assume a mostly noiseless environment so that minimal smoothing and filtering needs to be done during the DOA estimation phase. The first is assumed for the sake of simplicity, and the second is assumed to target the majority of our project towards ECE principles and not programming/COS principles.

From this base concept, we want to include a few auxiliary extensions: using a servo, we aim to configure our bin to be able to 'tip' over to empty trash into a larger receptacle, and using a speaker, we want the car to be able to audibly signal that it heard a call command by "calling back" (maybe trigger a pre-recorded phrase when it detects a call, like, "I'm coming!"), so that the user does not have to even look up from their work to ensure that it's on its way.

Links:

- This project builds a voice control module using the same kind of mic array/raspberry pi connectivity that we plan to use!
- This project uses an ESP32 to explain the logic behind receiving voice commands in a programming environment to control a robotic cars movements
- This project brings parts of our vision together, creating a moving bin! It also highlights the additional features we may want to incorporate into our trash can, ex. automatic closing and opening through hand detection

Table of parts:

Component	Estimated Price	Link for purchasing
Mic Array - sensor to detect sound and its approximate position	28.99	https://www.seeedstudio.com/ReSpeaker-Lite-p-5928.html
Foam Pads - to stabilize the mic array for maximum noise reduction	12.99	https://www.amazon.com/Soundproof-Self-Adhesive-Fire-Proofed-Soundproofing-Treatment/dp/B0B24HCCTH/ref=sr_1_9?dib=eyJ2IjoiMSJ9.9Ake0h8O89aM7txXFPTtY7Uek7Czg7oqg2Fn7vTSkQ6BQ7dMi9sOXOMjObdkTZadRO9uZ4E1Pn3nUMXd1dMIFJf4YMxO_ATycZ3L9zcDJUWIa

		FWWplRH8wpR6foq1-2MKzEL_nF3ASYXqgf29HUQPnyFClmt-w8YXjiozI3YEGfPSUFLJJqh9dgK-bgPrBSleQn8DQjnRYASFt4iNEAmvXIKXLyc0sIGHWGKuhxhWrZqpuRKRwuZmLA58dNPMgdJnhVvjcN0ludr862m82qTxAmJV6XGwPBkiXn14118W_U.rddsMh043zUNh3pjIPAUDQBzTAc93M1swlWJS39UmOE&dib_tag=se&keywords=sound%2Bproof%2Bpadding&qid=1759777119&sr=8-9&th=1
Raspberry Pi Model B 4GB - to connect with the mic array and process audio	55	https://www.pishop.us/product/raspberry-pi-4-model-b-4gb/
Distance Sensor - to determine stopping positions	60	https://www.adafruit.com/product/3317?utm_
Servo - for tipping the bin	19.95	https://www.adafruit.com/product/1142
Trash Can - to collect waste	19.99	https://www.amazon.com/Plastic-Wastebasket-Container-Bathrooms-Kitchens/dp/B0CYBW74TX/ref=sr_1_8?crid=1QFFEA414WQL6&dib=eyJ2IjoiMSJ9.E7oLLoSKTtV1HCLiKMMDtTl4e9bKtIPFXtndVymT4HMjoX3hH9EWZBoQ5lzffnLC5hvaous13HDnuQ9_5nfSOBlLrJlakEpP_ozo8yREyaArxht4PQLo6z8_-D2NZqNdqhmgLLWP36Xwmi3eTC8z3xB-f-rZp7z_EsM7sWwixsPTZYyvr09IADU4u055Sclen46l4Oj2wLL7iWhLQMYP7iGbcmwLnH-dhry9r0-F6ahO729rw0w13ct_vvcRqEUNSaLc3mE5DG7RS_FGgF_v1XzTzbj2a6lZspPG54JR8A.

		lPpTrRhtinkGJkH8y9Ah64y1XQgsdcgYyM646FzC82A&dib_tag=se&keywords=mini%2Btrash%2Bcan%2Bplastic&qid=1759782761&sprefix=mini%2Btrash%2Bcan%2Bplastic%2Caps%2C101&sr=8-8&th=1
Speaker - for “call back”	5.95	https://www.adafruit.com/product/3885

- Mic
    - ReSpeaker Mic Array v2.0
    - https://www.seeedstudio.com/ReSpeaker-Lite-p-5928.html
        - Mount rigidly but with vibration isolation (i.e. on foam pads) to isolate it from as much movement and motor noise as possible
- Foam pads
- Raspberry Pi 3 / Jetson Nano (for conversion of natural voice to commands using LLMs)
    - https://www.arrow.com/en/products/900-13448-0020-000/nvidia
    - https://www.adafruit.com/product/6125?src=raspberrypi
- IR sensors (Multiple for 360º coverage) / Ultrasonic
    - Adafruit VL53L0x
    - https://www.adafruit.com/product/3317?utm_
    - https://www.parallax.com/product/ping-ultrasonic-distance-sensor/
- H-bridge
    - https://shop.compressorcontroller.com/products/power-h-mini-dc-12a-h-bridge-driver-oem?variant=44195310928069&country=US&currency=USD&utm_medium=product_sync&utm_source=google&utm_content=sag_organic&utm_campaign=sag_organic
- Servo - in case we wanna make the bin tip forward and ‘empty’ itself
- Buck Convertor: will save us from building a new voltage reg board + we can use to power raspberry pi / in case we automate anything else like the bin movement
- Cute little trashcan (<1 lbs)

Hear a voice - just track voice to set direction, drive forward towards that direction until an obstacle is detected (assume the obstacle is the person that called).

2) Direction of arrival (DoA)

Goal: estimate the bearing of the caller to turn/drive toward them.

- **2-mic TDOA:** GCC-PHAT on 20–40 ms windows → peak lag τ → angle
  θ≈arcsin (c·τd)\theta \approx \arcsin\!\left(\frac{c\cdot\tau}{d}\right)θ≈arcsin(dc · τ)
  where ccc = 343 m/s, ddd = mic spacing. Median over 0.5 s.
- **4-mic array:** built-in beamformer/DoA (e.g., ReSpeaker) → smoother angles, better in noise.
- **Stability tips:**
  - Ignore frames with SNR < 6 dB.
  - Smooth DoA with EMA ($\alpha \approx 0.3$).
  - Freeze DoA while maneuvering at high motor PWM to avoid acoustic self-noise bias (sample during short "listen" micro-pauses).

1) Sensing options

- **Time-of-Flight (VL53L0X/L1X):** precise 3–200 cm; tiny FOV.
  - Mount **2–3 sensors**: front-left, front-right, and up-facing over the bin.
- **Ultrasonic (HC-SR04):** cheap, wider beam; good for "something's there."
- **(Optional) Up-facing camera:** simple brightness/skin-tone blob; use only if you already add a Pi/ESP32-CAM.

- Group number, name and email of the lab partners
- Motivation
- Overview and a brief explanation of what the project entails
- Links to similar projects (especially videos/pictures of a similar project) or any references that can help us better understand your goal (if any)
- A table including the components you need to purchase & their estimated price **(this is separate from the formal order placement).**

Component	Estimated Price	Link for purchasing

CleanOPetra/Elon Dust/Fleetwood Vac/Suckerberg: Voice-Guided, Vision-Assisted Autonomous Cleaner

Group Number: 308

- **Mayan Wasu (mw8270@princeton.edu)**
- **Callista Chong (cc0225@princeton.edu)**

Motivation

Throughout this course, we have progressively built a deeper understanding of the fundamentals that drive modern electronic systems — from basic power regulation and motor control to closed-loop feedback and navigation. For our final project, we want to extend these foundations into a system that feels both practical and exciting (we would definitely love to keep this even after Car Lab): an autonomous robotic cleaner that can interpret voice commands, make simple decisions, and perform physical actions in response. Our motivation stems from seeing how real-world systems like the Roomba combine many of the subsystems we have learnt about in this course like power electronics, sensor integration, control algorithms, and communication into one coordinated design.

By creating our own version of such a system from scratch, we hope to explore the intersection of **hardware, sensing, and intelligent control** in a way that reflects what we've learned in this course. It will be really cool to bring together a lot of tools and systems we have so far used in a "siloed" manner relative to each other. We will use **voice commands powered by a local LLM interface** (via laptop) that will be converted into instructions for the car and **computer vision for environmental awareness** for obstacle avoidance and to understand the environment it is cleaning.

Project Overview

Our final project, *Elon Dust*, is a **voice-guided, vision-assisted autonomous cleaner** built on the same car platform we have used throughout the semester. The robot will combine **onboard sensing, feedback control, and decision-making** to perform cleaning tasks autonomously.

The system's core functionality will center around three layers:

1. **Control and Movement:** Using our previously developed PID speed control system, the car will maintain **stable motion** and **execute navigation commands** such as forward, reverse, or turning.

2. **Perception and Navigation:** The robot will use a **Pixy2 camera** to detect lines on the floor (for navigation) and identify colored markers or "trash" objects. Additional **ultrasonic/time-of-flight/LIDAR sensors** will allow the robot to detect and avoid obstacles in its path.

3. **Voice and Decision Layer:** Instead of writing programs for each task, we will integrate a **laptop-based interface** that processes voice commands directly. Spoken instructions such as "clean the hallway" or "pick up the red cup" will be processed using a small local LLM (through an API - we will probably use langchain to to make API calls) that converts them into structured instructions (e.g., *task: clean, zone: my room, mode: vacuum*). These structured commands will then be sent to the robot over USB or Wi-Fi, allowing it to respond and execute tasks autonomously.

We have not decided yet on the exact cleaning functionality, but the robot will probably operate in either of these two modes:

- **Vacuum Mode:** Power a small handheld vacuum motor (that we can power using our board and also attach to the car).

- **Trash Pickup Mode:** Use the Pixy camera to recognize color-coded trash objects and pick them up using a small servo-based gripper arm before returning to a designated bin area.

To achieve these functionalities, we predict that we will need to make the following changes (and likely more) to the current car:

- A **custom power board** to safely step down the battery voltage to the different rails that we would need for the gripper arm or for the vacuum. We might integrate this onto the current board, or we might build a new one. If we build a new one we will use the same concept and likely have a similar layout with transient suppression, and decoupling capacitors (both high-frequency and bulk capacitors) to ensure stable operation.

- The existing **power MOSFET board** only allows the car to drive in one direction (forward). To allow it to move in all directions, we need to use a **H-bridge motor driver** that allows us to apply a voltage across it in both directions.

- **Integrated sensor suite**, including the Pixy camera, IMU (for heading stability), ultrasonic sensors (for obstacle avoidance) / LIDAR, and wheel encoders (for odometry feedback).

- **Software -** We will use a lot of open-source libraries like openCV, and we'd also be making a lot of LLM API calls. We will use either Raspberry Pi or Arduino.

Links to Similar Projects / References

- Autonomous RC Car
- Voice-Controlled Car Project
- Roomba-Inspired Cleaning Robot Project

These projects provided inspiration for aspects such as sensor interfacing and motion control, but our project will feature an original circuit design, a custom power system, and a unique integration of voice and vision.

Component List and Estimated Cost

Component	Estimated Price	Link for Pricing
Sensors - What exactly		
Vacuum?		
Gripper Arm (what subcomponents are needed)		

Summary

Elon Dust brings together every key aspect of what we have learned in this course power electronics, control systems, sensing, and integration into one unified system. It will demonstrate how an embedded platform can interpret natural human input, process sensor feedback, and perform a physical task autonomously.

Rough

Names lol : Suckerberg, Vaccie Chan, Elon Dust, Dobby, Wall-E, Cleanopetra, Fleetwood Vac

Possible Ideas:

- Autonomous Vacuum Cleaner: Takes in voice commands, i.e. “Clean Tait’s Room” and learns about the layout of its environment - essentially a voice activated vroomba / if the vacuum system is too complicated, we could replace it some sort of mechanical arm that picks up trash
- Glove that is gesture based - allows you to control the car, kind of gimmicky but would be a cool demo
-

Final Project: Here, you should design and implement an original system emphasizing

ECE principles rather than mechanical construction (MAE) or pre-built software solutions

(COS). You should submit a project proposal by Monday Oct. 6 th for approval before

placing orders. The course instructors will review proposals and give each team feedback.

You will then place orders for your projects by Monday, Oct. 13 th. See "Course Deadlines" on Canvas for key dates.

Requirements:

- Original implementation (no copy/paste projects or using pure AI or canned software)
- Emphasis on electronic design over mechanical construction (ECE-oriented instead of MAE or COS)
- Custom circuit implementation where feasible (e.g., build H-bridge rather than buy off-the-shelf)

Final project proposal

- Group number, name and email of the lab partners
- Motivation

- Overview and a brief explanation of what the project entails
- Links to similar projects (especially videos/pictures of a similar project) or any references that can help us better understand your goal (if any)

A table including the components you need to purchase & their estimated price (this is separate from the formal order placement).

Component Estimated Price Link for purchasing

We will review your final project proposals and provide feedback. Then you'll need to place your orders through the link that Radd will provide