ECE 445

Spring 2026

Project Proposal

Roomify: A Smart Room System

Project No. 67

TA: Lukas Dumasius

Benjamin Chang, Warren Lam, Owen Wang

Contents

1. Introduction

1.1 Problem

Room devices such as LED lights, music systems, TVs, and other decorations are hard to coordinate, leading to unsynchronized room atmospheres and an excess of handheld remotes. People interested in room decor face a fragmented ecosystem where each device operates independently. This makes it difficult to create cohesive environments with a unified mood (e.g., slow music with flashing LED lights), and the need for multiple remotes and is inconvenient for managing room environments.

For college students and apartment residents, budget and room modification restrictions make purchasing and installing integrated smart home systems unrealistic. In addition, existing products like Google Home and Amazon Echo only support specially designed devices, making it impossible to integrate use with ordinary room decor devices and appliances. These smart home compatible devices are also much more expensive and harder to find. The lack of solutions for smart room systems indicates a need for an affordable, flexible solution that can synchronize home devices without requiring special hardware.

As social media use rises, room decor has become increasingly popular as people post their setups. For example, the hashtag #HomeDecorTikTok has over 10 billion views, making the problem of improving room ecosystems extremely relevant. Solving this problem with a cohesive, extendable room system would reduce remote clutter, enhance room lifestyle, and establish a new standard for integrated rooms. Because people spend a lot of their time at home, the ability to easily create personalized room environments without purchasing expensive smart devices would be very impactful.

1.2 Solution

We propose *Roomify*, a centralized room control system where users can easily operate devices in their room like LED lights, TVs/projectors, aroma diffusers, Spotify, or other remote-controlled decorations and displays. For example, users will be able to press a single button on the *Roomify* screen or on their phone and turn on the cooking channel on their TV, turn LED string lights yellow, diffuse lavender, and play jazz music, all in one click. *Roomify* introduces an extendable, cohesive room system that coordinates independent devices and removes the need for multiple remotes.

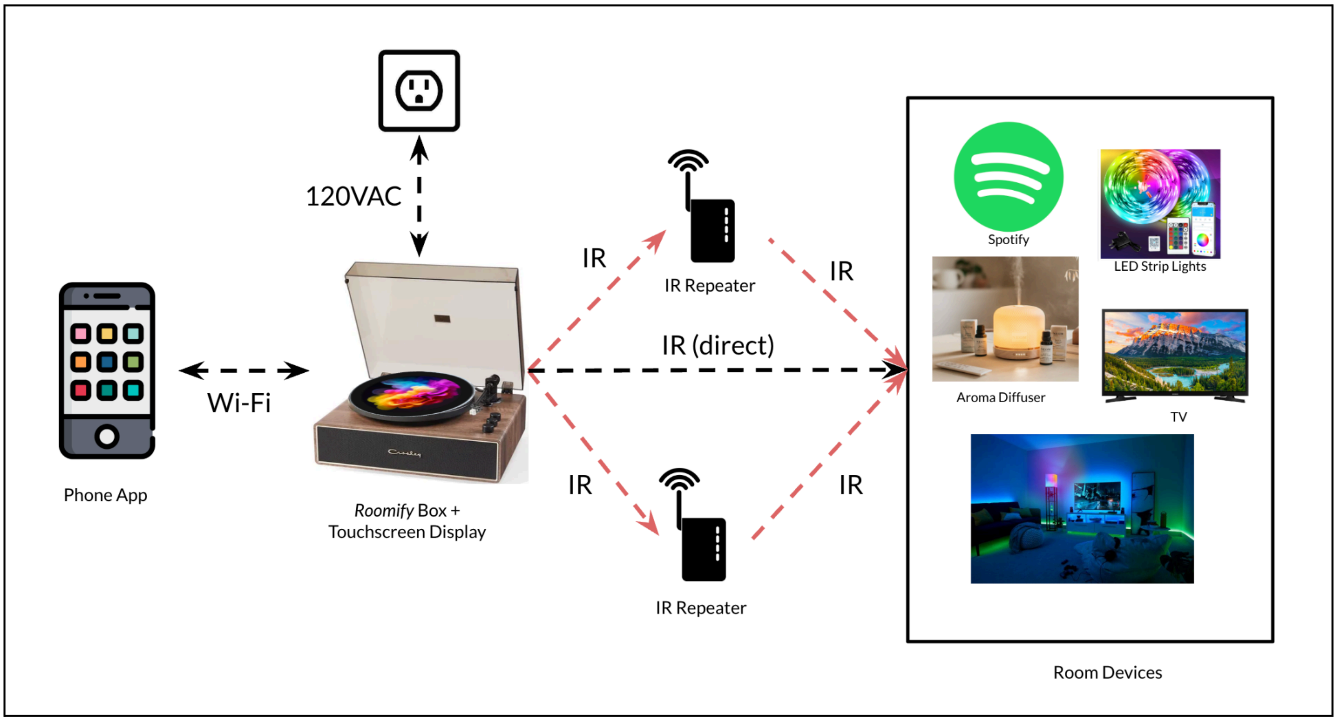
Roomify will appear like a vinyl player with a hinged wooden box, with a round capacitive touch RGB TTL TFT display driven by an ESP32-S3 board. The core functionality of the board will be controlling the circular display, infrared (IR) signal receiving and decoding to store device remote codes, omni-directional NEC protocol IR transmission to transmit device remote codes, and WI-FI communication(to make Spotify API calls. In addition, we will design “repeaters” that can be placed throughout the room to receive and relay IR signals to increase effective range and transmit to devices not in direct line of sight.

By copying and storing IR remote codes as a “virtual remote”, users can map buttons on the *Roomify* touchscreen display and their phone (web-app) to IR signals that interact with devices in the room. To accomplish this, *Roomify* will have an "Add Remote" mode where users can store remote information within *Roomify*. For example, a user would add the "red" button on a LED string light remote by aiming the remote at the *Roomify* box. After *Roomify* decodes the signal, users can label and save the button code data. The user could then repeat the process for other buttons on the remote, or for other remotes. Any device that uses infrared remote signaling or has an API (like Spotify) will be usable with *Roomify*.

After adding all room device remotes to *Roomify*, users will be able to create quick-start presets (e.g. green lights with Christmas music and a Snoopy Gif display on the round screen). When the user selects a preset, Roomify will transmit the necessary IR signals in all directions (14 IR LEDS, 6 to cover both directions along the XYZ axes, and 8 pointed along the center of each octant) and make Spotify API calls. Aside from using presets, users can also use the virtual remotes in *Roomify* to change individual device settings, removing the need to use multiple remotes and apps.

Roomify is an extensible, centralized room system that allows for full control over room decor, lights, and sound, allowing students and apartment owners to easily set cohesive room atmospheres without hassle.

1.3 Visual Aid



1.4 High-Level Requirements List

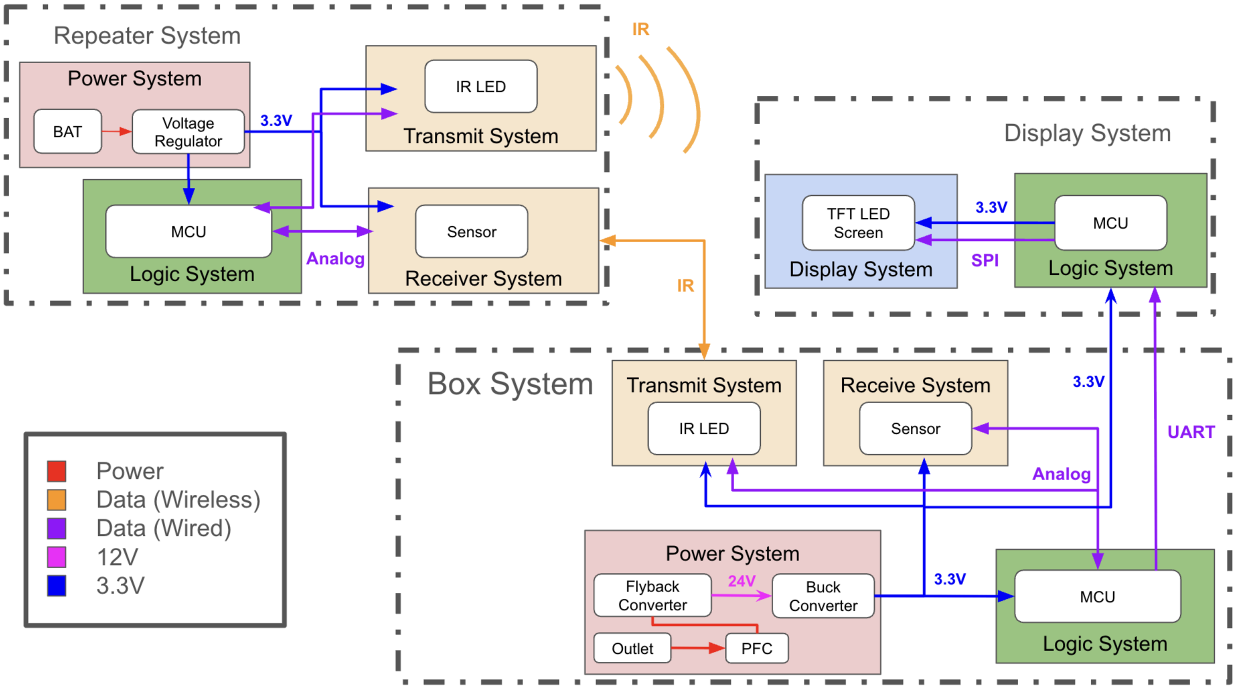
1. **Infrared Remote Control Transmission.** *Roomify* should be able to decode, store, and retransmit at least 10 unique IR commands for at least three different devices with accuracy ≥95% compared to using the original remote.

2. **Preset Execution.** *Roomify* should be able to operate multiple devices in a user-selected preset. Execution should finish within 1 second, measured from user input to the correct activation of all devices in the preset. The preset should include at least one device in the following categories: IR transmission (LED lights), API calls over Wi-Fi (Spotify), and *Roomify* screen display.

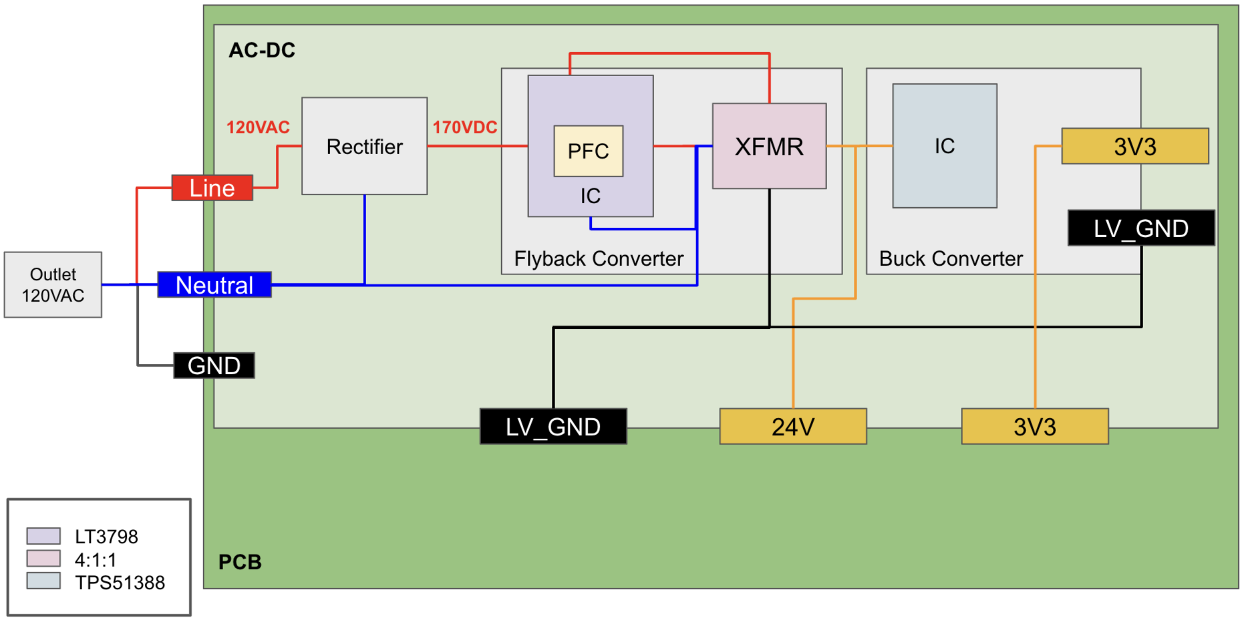
3. **Range and Coverage.** *Roomify* must provide reliable omnidirectional infrared coverage capable of controlling three separate IR-based devices within a 5-meter radius across a ≥300° horizontal field using no more than three infrared repeaters.

2. Design

2.1 Block Diagram



AC-DC Converter Block Diagram



2.2 Subsystem Overview and Requirements

2.2.1 Infrared Transceiver System

The Infrared Transceiver System includes a unit for infrared transmission and a unit for infrared signal receiving and decoding. To learn device remote codes during pairing, the receiver unit will have an infrared sensor that can pass raw signal data to the MCU for decoding and storage. The transmit unit consists of multiple infrared LEDs (six along XYZ axes, and eight along the diagonal of each octant) to provide near-omnidirectional transmission. The IR LEDs are driven directly by MCU GPIO outputs using the standard NEC encoding. The transmit subsystem communicates wirelessly via IR with both room devices and the Repeater System (orange paths in the block diagram), forming the core communication mechanism for device control.

Requirements:

- Receive and decode 38kHz NEC infrared signals with irradiance $> 0.15\,\frac{mW}{m^2}$
- Transmit 38kHz NEC infrared signals at $> 130\,\frac{mW}{sr}$
- (Near) Omni-directional transmission of infrared signals, > 300° coverage along any axis

2.2.2 Repeater System

The Repeater System extends the infrared coverage of *Roomify* by receiving IR commands from the main Transceiver System and retransmitting them in a different direction (user can decide which direction to point the repeater in). Essentially, the repeaters will act as “mirrors” that can relay infrared signals to devices that are not in direct line of sight of the main transmission unit (e.g., behind a wall). Each repeater will have simplified receiver, transmitter, and microcontroller units. The receiver unit will have an IR sensor that passes the raw signal to the MCU. To avoid amplifying noise, the MCU will decode the signal and verify that it is a valid NEC remote code. The decoded signal is then re-encoded and re-transmitted. The repeater operates independently using battery power and is designed to be lightweight and simple so users can place several of them throughout their room.

Requirements:

- Receive and decode 38kHz NEC infrared signals with irradiance $> 0.15\,\frac{mW}{m^2}$
- Verify received signals are valid NEC signals and block transmission of noise (>95% accuracy of retransmission)
- Transmit 38kHz NEC infrared signals in a single direction at $> 130\,\frac{mW}{sr}$

2.2.3 Roomify Display Interface System

The Display System is the primary visual user interface for *Roomify*. The Display System consists of capacitive touch TTL TFT screen and a special ESP32-S3 MCU that can drive the RGB-666 display. The display MCU communicates with the TFT screen using a SPI interface (purple wired data path), enabling high-speed graphics updates such as spinning vinyl animations or preset visuals. The Display System communicates bidirectionally with the main *Roomify* MCU via UART (wired data path), allowing for full control over the display and for touchscreen triggers to be sent to the central MCU. In response, the *Roomify* MCU can execute IR transmission presets. All execution is still handled by the *Roomify* MCU, and the Display System simply acts as the user interface.

Requirements:

- Provide user touchscreen data to the main MCU via UART with latency < 50ms
- Display animations and images at > 15 FPS.
- Display stored user presets and settings.
- Provide an interface that allows users to use all functions of *Roomify*, such as selecting a preset and pairing new remotes.

2.2.4 Power Systems (Box and Repeater)

There are two independent power architectures shown in the diagram. The power system for the *Roomify* box uses an AC outlet input, followed by power factor correction (PFC) and a flyback converter to generate 12V DC. The box power system converts 120VAC from the outlet into regulated DC rails. A buck converter then steps this down to 3.3V for logic circuits. The 12V rail powers the Motor System, while 3.3V powers the MCU, IR subsystems, and Display System.

The power system for the repeaters uses a battery source regulated down to 3.3V for its MCU and IR circuits.

Both systems maintain electrical isolation between high-power (12V) and logic-level (3.3V) domains to ensure stable operation and safe interfacing between subsystems.

Requirements:

- Convert 120V AC from outlets to 12VDC and 3.3V.
- Impose current limits for safe operation and outlet usage.
- Power supply should provide peak 3A for the *Roomify* system at 3.3V +/- 0.3V.
- The battery system should provide peak 500mA at 3.3V +/- 0.3 V for *Roomify* repeaters.

2.2.5 Box + Logic System

The Box System is the central hardware unit of Roomify and contains power conversion, logic control, IR communication, and a motor system (optional) to automate opening and closing the box. The *Roomify* box will be constructed from wood and painted to look like a vinyl player with the round display acting as the spinning record (see Visual Aid).

The central controller for the Logic System is an ESP32-S3 MCU, coordinating IR transmission/reception, motor control, display communication, and Wi-Fi (for making API calls to Spotify, mobile web application control, etc.). The Transceiver System allows the box to learn remote codes and send omnidirectional NEC-encoded commands wirelessly to room devices and repeaters. The Motor System, powered by 12V and driven through a motor driver, mechanically opens and closes the vinyl-style box enclosure. The Box System also connects to the Display System via wired 3.3V power and UART communication, acting as the master controller for the entire Roomify device. Finally, the MCU will able to receive user instructions from mobile devices via Wi-Fi, allowing users to use *Roomify* through a web application on their phones.

Requirements:

- Control the Transceiver, Display, and Motor systems.
- Receive and process user requests over Wi-Fi within 500ms
- Store user information (presets or remote signal codes) to a database over Wi-Fi reliably
- Make valid HTTP requests over Wi-Fi to Spotify API

2.2.6 Mobile Device Control System

In addition to the *Roomify* display interface, users will also be able to control *Roomify* with their phones via a web application. User presets and virtual remotes will be stored in a database linked to user accounts. After signing in to the web application, users can access the same interface as the *Roomify* display to select presets and control room devices. This system will use Wi-Fi to allow for wireless control of the *Roomify* box. When users select an option on their phone, a request will be sent to the *Roomify* ESP32 MCU, triggering appropriate actions.

Requirements:

- Communicate with the *Roomify* Box System over Wi-Fi with < 200ms latency
- Display stored user presets and settings.
- Provide a clean, intuitive interface for users to add new remotes, create presets, and control devices through *Roomify* (time to teach someone how to use the application should be < 3 minutes).

*2.2.7 Motor System (within Box System, **optional**)*

The Motor System controls the automated mechanical opening and closing of the wooden vinyl-style enclosure. It consists of DC motors powered by the 12V rail generated in the Box Power System. A motor driver interfaces between the 3.3V logic-level control signals from the Box MCU and the higher-power 12V motor supply. When triggered by the Logic System (for example, during preset activation or startup), the MCU sends control signals to the motor driver, which regulates current to the DC motors. The Motor System is controlled by the box Logic System, and can be activated by users on their mobile devices through a web application. For example, when users select a preset on their phone, the *Roomify* box would automatically open via the Motor System. Because the Motor System is not critical to *Roomify*'s core functionality, it is an **optional**, aesthetic subsystem that we will implement if we have time.

Requirements:

- Generate 0.4 Nm +/- 0.5 Nm torque to prop open a 310 gram lid.
- Open/close the lid within 1 second after receiving instructions from MCU.

2.3 Tolerance Analysis

2.3.1 Infrared Signal Transmission

Infrared signal transmission is a core part of *Roomify*. The infrared signals from the Roomify box must be strong enough to reach the device receivers, either directly or through the *Roomify* repeaters.

From the datasheet for the infrared LED we plan to use, the IR LED has a radiant intensity of 130 mW/sr when using 200mA pulses. To determine how much power reaches the receiver, we multiply this value by the solid angle subtended by the receiver.

Assuming a worst case distance of 10m (for a 15 ft. x 15 ft. x 10ft. room, the distance from opposite corners of the room is 7 meters), we can calculate the irradiance of the signal at the receiver as follows:

$$E = \frac{I_e}{r^2}$$

$$E = \frac{130\frac{mW}{sr}}{100m^2}$$

$$E = 1.3\ \frac{mW}{m^2}$$

Standard infrared receivers have $E_{min} > 0.15\ \frac{mW}{m^2}$. Calculating the link margin as 10 $\log_{10}(E/E_{min})$, we get a link margin of 9.37 dB.

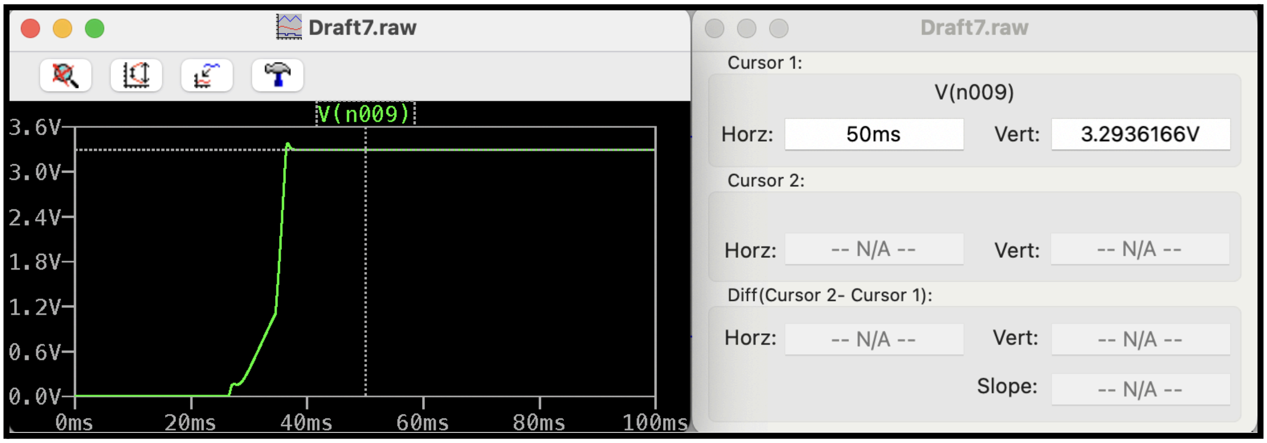
To account for insertion loss from the PLA box, we assume signal attenuation of approximately -0.5 dB, resulting in a substantial margin of 8.87 dB for our device. In addition, the maximum rating for the infrared LED is 1A pulses (for 750 mW/sr), meaning we can increase signal strength significantly compared to using 200mA pulses if necessary.

With 1A pulses, the irradiance at a receiver 10 meters away is about $7.5 \frac{mW}{m^2}$.

2.3.2 Box Power System

Another critical component of *Roomify* is the power subsystem and its ability to deliver enough power and current. The voltage regulator IC we chose for our microcontroller unit is the LT1170. This IC allows us to have a wide input voltage and an output of 3v3. Because *Roomify* will have two MCUs (one “main” MCU and one special MCU for controlling the touchscreen display, we will need around 2.4A to power both (assuming a peak current draw of 1.2A for an ESP32-S3). *Roomify* will drive at most three infrared LEDs at a time at 200mA pulses, requiring peak draw of 600mA.

Combining the current requirements for the MCUs and the IR LEDs gives a total budget of 3A. The rating for the LT1170 is 5A at 3v3. For our 12V system that powers our 3v3 system, it is required to output around 10 Watts. Our flyback converter can deliver up to 40 Watts of power, so we will have an ample amount of power for our system. Below is a simulation of our LT1170 component that delivers enough voltage and current.



Cursor 1:

V(n009)

Horz: 50ms Vert: 3.2936166V

Cursor 2:

Horz: -- N/A -- Vert: -- N/A --

Diff(Cursor 2- Cursor 1):

Horz: -- N/A -- Vert: -- N/A --

Slope: -- N/A --

2.3.3 Motor System (**optional**)

The motor system will control automatic opening and closing of the *Roomify* box. We plan to have a crank-slider-lever system where the motor rotates a crank, crank pushes a vertical linkage, the linkage connects to the lid some distance from the hinge, and the lid moves up and down with the linkage.

We are assuming that we have a 5.5in. x 7in. x 1in. hollowed polycarbonate encasing of dimensions with walls that are 0.2in thick. This has a density of 1.20 g/cm^3 resulting in a total mass of about 310 grams.

We can calculate the weight of the encasing as follows:

$$F_{weight} = mg$$

$$F_{weight} = 0.31 * 9.81$$

$$F_{weight} = 3.0411\ N$$

We can then calculate the required torque for movement from the hinge for the encasing:

$$|\tau_{req}| = r\ Fsin\theta$$

$$|\tau_{req}| = 0.1778/2\ \ * 3.0411 * sin(90)$$

$$|\tau_{req}| = \ 0.2704\ Nm$$

The above calculation also relies on the fact that the center of mass is in the exact center.

The 5V DC motor has a rated load of 8.0 g/cm which is a torque of 0.000785 Nm. If we attach it to a 50:1 gear system the torque would increase by 50 times, and with two motors, we could achieve a torque of 0.000785*50*2 = 0.0785 Nm.

Now assuming that the motor is connected to a 1cm crank, we can calculate the force generated by it:

$$F_{total} = \ \tau_{hinge}/r_{crank}$$

$$F_{total} = 0.0785/0.01$$

$$F_{total} = 7.85\ N$$

Assuming the linkage attaches 2 inches from the hinge, we can calculate the torque at the linkage:

$$|\tau_{linkage}| = F_{total} * r_{linkage}$$

$$|\tau_{linkage}| = 7.85 * 0.0508$$

$$|\tau_{linkage}| = 0.398\ Nm$$

This is greater than the counter torque due to the weight by 47%, so even if the motor cannot perform with 100% efficiency, the system should be more than able to lift the encasing.

3. Ethics

As engineers, we have an ethical and professional responsibility to design safe and transparent systems. The IEEE Code of Ethics states that engineers should "hold paramount the safety, health, and welfare of the public" (IEEE Code of Ethics, Principle 1). Given this context, the main considerations in designing *Roomify* are electronic safety and data privacy. Because *Roomify* will be constantly powered on in users' rooms, it is extremely important to mitigate any electrical risks through electrical isolation, safe operating voltages, fuses, and controlled signal transmission.

The most critical component in *Roomify* is the power subsystem. Because we are designing our own power supply to work with standard 120V AC household outlets, we have adequate safeguards to guarantee safe operation. The power subsystem will use an isolated AC-DC converter to separate high and low voltage components. To reduce shock risk, the *Roomify* PCB will be housed in a wooden box to prevent user access. For development, the box will be easily disassembled (using sliding panels and walls) so that we can extract the PCB if necessary. The only electrical component that users can interact with is the capacitive touchscreen (purchased component), which is designed to be safe to touch. In addition, we will use fuses to enforce current limits and prevent overheating and component failure. Mechanical components, such as the motorized hinges, will be enclosed within the wooden box to prevent pinch hazards.

By identifying potential hazards, classifying risk, and implementing necessary safeguards, *Roomify* will adhere to the UL 62368-1 standard for audio, video, and ICT equipment. When working with AC mains power and high-voltage circuits during development, the team will follow UIUC lab safety policies such as wearing appropriate PPE, ensuring circuits are de-energized before interacting with them, and using one hand instead of two.

Furthermore, the ACM Code of Ethics requires computing professionals to "avoid harm" (ACM Code 1.2) and to "respect privacy" (ACM Code 1.6). To align with these principles, *Roomify* will completely avoid invasive data collection. The system does not include cameras or microphones, and it only stores user-provided IR remote codes and preset configurations. Any network communication, such as Spotify API usage, is limited to authenticated user requests and follows the terms of service of the external platform. Though we are not storing private data, we will implement user accounts to protect user information. For the web application, we will implement proper authentication protocols, input validation, and protection against common web vulnerabilities such as SQL injection and cross-site request forgery. To prevent misuse, Roomify will only transmit IR commands that have been explicitly learned and labeled by the user, reducing the risk of unintended control of nearby devices.

Additionally, IEEE emphasizes that engineers must “be honest and realistic in stating claims or estimates” (IEEE Code of Ethics, Principle 3). Throughout this course and design process, we will ensure that all documentation and demonstrations accurately represent the capabilities and limits of our system. For example, instead of implying universal compatibility, we will clearly specify which IR devices and API platforms *Roomify* supports.

From a societal and economic perspective, *Roomify* provides a safe, low-cost alternative to existing smart home ecosystems, improving accessibility for students and renters who cannot modify their living spaces or invest in smart devices. Environmentally, *Roomify* will not increase waste or have a significant carbon footprint since it is a reusable, low-power system. Above all, *Roomify* is a product for improving users’ lifestyles, and all design choices will prioritize safety, regulatory standards, and user privacy.