Roomify: A Smart Room System

ECE 445 Final Report - Spring 2026

Project # 67 - *Roomify*

Benjamin Chang, Warren Lam, Owen Wang

Professor: Craig Schultz

TA: Lukas Dumasius

Abstract

Recently, room decor and aesthetics have become increasingly popular, with the hashtag #HomeDecorTikTok gaining over 10 billion views [1]. However, room devices such as LED lights, screens, and music systems are hard to coordinate, with separate device interfaces and remotes. In addition, existing smart room systems like Google Home are complex, expensive, and difficult to integrate with. We propose *Roomify*, a centralized device control system that copies and stores infrared (IR) remote codes that can then be grouped together into synchronized room "presets." For example, a preset could include green lights, Christmas music, lavender mist from an aroma diffuser, and a fireplace video on the TV. When a preset is selected, *Roomify* broadcasts the necessary IR signals across the room, providing a lightweight, extensible system that can control multiple devices simultaneously. *Roomify* allows users to create and set cohesive room atmospheres, giving them full control over their room.

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 [1]. Solving this problem with a cohesive, extendable room system would reduce remote clutter, enhance room lifestyle, and establish a new standard for integrated rooms.

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 (Demo Video). For example, users will be able to press a single button on the *Roomify* screen and put on the cooking channel, turn the LED string lights yellow, diffuse lavender, and play jazz music. *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 square capacitive touch RGB TTL TFT display driven by an ESP32-S3 board. The core functionality of the board will be controlling the display, infrared (IR) signal receiving and decoding to store device remote codes, omni-directional IR transmission to transmit device remote codes, and WI-FI communication (to make Spotify API calls). In addition, we will design “repeaters” placed throughout the room to relay IR signals 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. 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 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 square display screen). When the user selects a preset, Roomify will transmit the necessary IR signals in all directions. Aside from presets, users can also use the virtual *Roomify* remotes to change individual device settings.

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.

1.3 Visual Aid

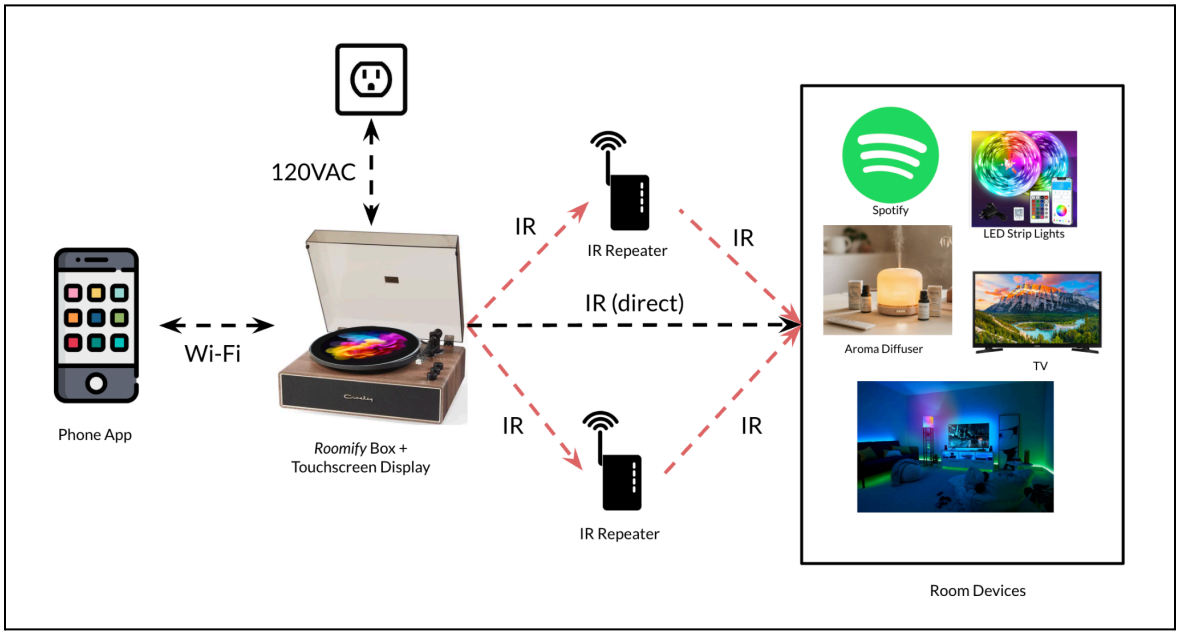

Figure 1 - Visual Aid

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.

1.5 Subsystem Overview

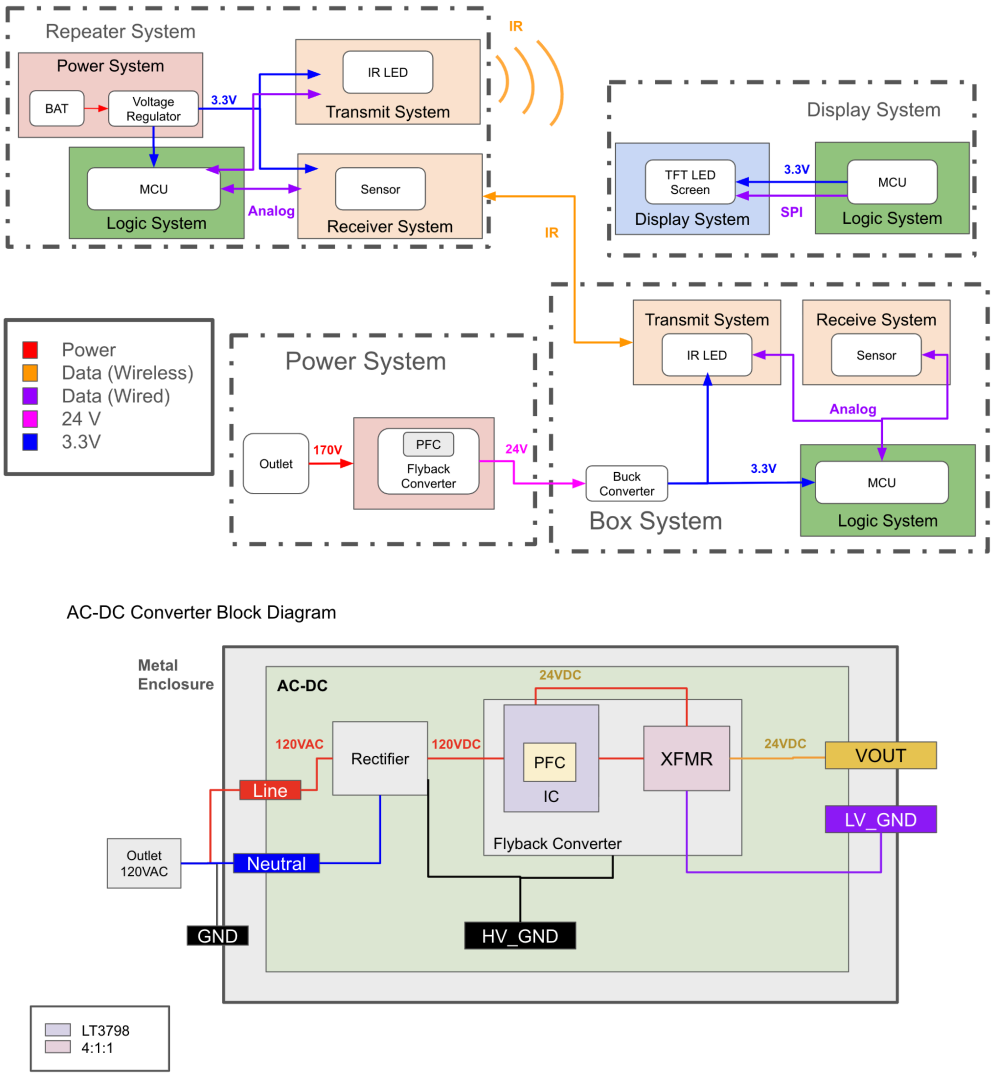

Figure 2 - System Block Diagram

Users interact with *Roomify* through the Display System (top right), which has a touchscreen display for selecting room device presets, adding remotes, and controlling music. The Display System is controlled by the Box System, which is the central unit of *Roomify*. The Box System contains the main MCU, which controls infrared signal transceiving and Wi-Fi operations. The Repeater System redirects IR signals from the Box System to increase coverage within the room. Repeaters will be separate, battery-powered units placed throughout the room to relay IR signals to devices not in direct line of sight of the Box System. Finally, the Power System converts offline AC power to 3.3V for the Box and Display Systems.

2. Design

2.1 Design Procedure

2.1.1 Infrared Transceiver System

The general approach our team took to the IR transmission system was to maximize the amount of current that would flow into our IR LED in order to maximize the range of the entire system. The overall system of the box was to have 14 IR LED in total to control and allow every IR controlled device in a room to be able to receive a signal and be controlled by the devices. Some alternatives we were deciding on during the implementation was to have a single IR LED attached to a box and aim it at IR repeaters to relay the message. This would allow us to use only 1 LED rather than 14, but it would require more precision. With 14 IR LEDs in total, it allowed us to cover more area. This was a major reason for why we chose to allow 14 transmitting devices to be on a single board, which required a very high power supply as each LED consumed around 1A of current from our overall system. The major equation used to determine the power of our overall system was the amount of current each LED consumed. By increasing the amount of current through our diodes, it increased our range and allowed us to cover more area throughout our room. The major equation we used was $I_D = \frac{V_{DD} - V_{D,on}}{R_{total}}$ as it aimed to minimize total impedance for our signal.

2.1.2 Repeater System

Our repeater system had the same governing principle behind the way our box system worked when transmitting signals. The same goal of maximizing the current through the LED was needed in order to transmit across a large room. The main difference between the repeater and the main box was the fact that this board included only a single LED. The reason for a single LED is to repeat a signal in a single direction. With the same equation of $I_D = \frac{V_{DD} - V_{D,on}}{R_{total}}$ The impedance seen by the signal was very small and around 1 ohm.

2.1.3 Display System

Within the Display system, the overall objective was to allow the user to interact with a screen device on board the box in order to reduce the need for external remotes and to enable synchronized control of multiple devices. The major design choices for this was to have a stable power supply to reduce the chances of any brownouts from the MCU or the display device. There was a debate between allowing the display system to connect via UART or WiFi to the main box as UART would be a more reliable option without the need for a middle man. However, since the repeater system was also on WiFi, the addition of a display device on the same network allowed for an easier development system, as well as a more robust system overall since the repeater no longer needed to be in line of sight of the *Roomify* box (allowing control of devices in multiple rooms, etc.). Reducing the amount of complexity across connections and communication protocols allowed for a smoother development process on both the hardware and software side. Due to limitations in the graphics of the touchscreen display, the design for the display was also simplified to only include preset execution buttons to keep the user interface clean and intuitive.

2.1.4 Power System

Within the Power System, designing an AC-DC power supply that was able to pull power from the line was the biggest challenge. A major factor that dictated the design process was following the UL 62368-1 standard as that would allow for creating a safe system when pulling off the line. The topology we chose was a flyback converter as it was an isolated system and allowed for safety on the secondary side of the transformer, which would provide power for the entire box and display system. The other topologies we looked at were not as desirable as the flyback offered both an isolated topology as well as several resources online for implementation and development. The main equation that is satisfied is the duty cycle of the switching of the NMOS transistor on the low side of the primary winding. $D = \frac{\frac{N_p}{N_s}V_{Load}}{V_{supply}+\frac{N_p}{N_s}V_{Load}}$ is the duty cycle the PMIC aims to set for the overall control.

2.1.5 Box + Logic System

For the Box and Logic System, the main considerations were designing a clean housing for *Roomify*'s internal electronics, while also providing a simple control unit capable of managing all of the other subsystems like infrared transceiving and driving the touchscreen display. We decided to handle the control flows over WiFi, with a central AWS server that all of our subsystems interact with to queue commands, fetch commands, and view user presets. The box was modified in Fusion360 from an online build from Concept Bytes to match our project specifications and mount our specific touchscreen display [3].

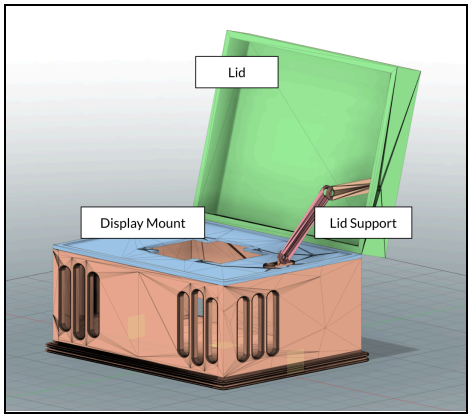

Figure 3 - Front View of Physical Box Design

2.1.4 Mobile Device Control System

For the Mobile Device Control System, the main design considerations were making a streamlined, intuitive user interface that made it easy to add remotes and room devices and also create grouped presets. We kept the design minimal, with only two pages (remotes and presets). Users can also link their Spotify accounts to access their playlists, and create presets through a simple two-field form. All updates are immediately propagated through the *Roomify* system without any user action necessary. For our system design, we selected standard frameworks, using React, TypeScript, and Next.JS for the

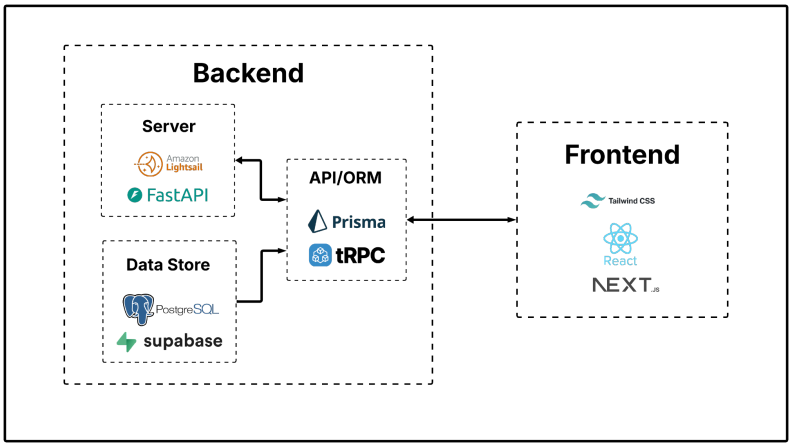

Figure 4 - System Design for Roomify Web-App and Server

frontend, and Supabase for our database (Figure 4). We used Amazon Lightsail to host our central FastAPI server, which handles all communication from the frontend and our ESP32s.

2.2 Design Details

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, we will use the built-in Remote Control Transceiver (RMT) peripheral on the ESP32-S3 [5]. However, we will use our own transmitters instead of using the RMT peripheral to provide multi-directional transmission. The transmit unit consists of fourteen 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 (Figure 5) [4]. 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.

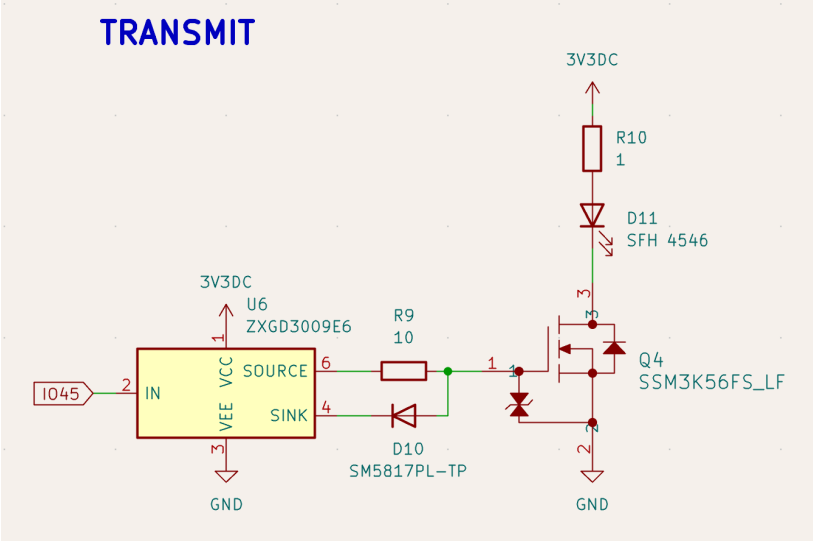

Figure 5 - Schematic Diagram For Infrared Transmission

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 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 has a simplified transceiver and microcontroller unit. Like the main *Roomify* MCU (2.2.5), the repeater polls (every 250ms) a central server for any queued commands to execute. The repeater also maintains a buffer of recently transmitted IR signals to avoid retransmitting commands multiple times before the main MCU marks the command as complete. 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.

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 preset visuals. The Display System communicates bidirectionally with the main *Roomify* MCU via a WiFi connection to our central server, enabling synchronized operation between all the *Roomify* interfaces and MCUs for preset execution. When the user selects a preset, the display MCU sends a POST to our AWS Lightsail server, which then queues the necessary IR command codes in a database and plays the linked Spotify playlist. The *Roomify* MCU (2.2.5) can then GET the commands and transmit the IR signals to operate room devices.

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 24V DC (Figure 7) [6]. The box power system converts 120VAC from the outlet into regulated DC rails [7]. A buck converter then steps this down to 3.3V for logic circuits like MCU, IR subsystems, and Display System (see Figure 6).

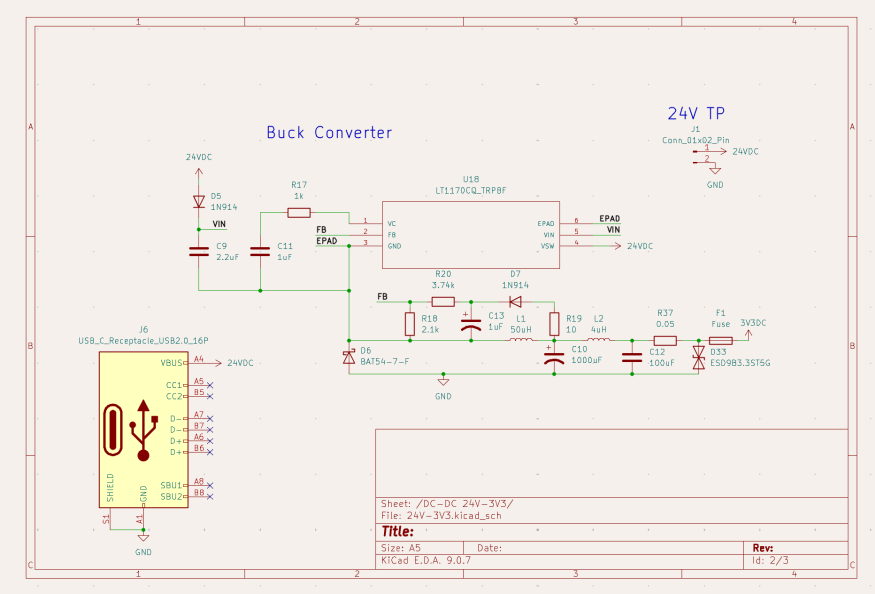

Figure 6 -Schematic Diagram for 24V-3.3V Buck Converter

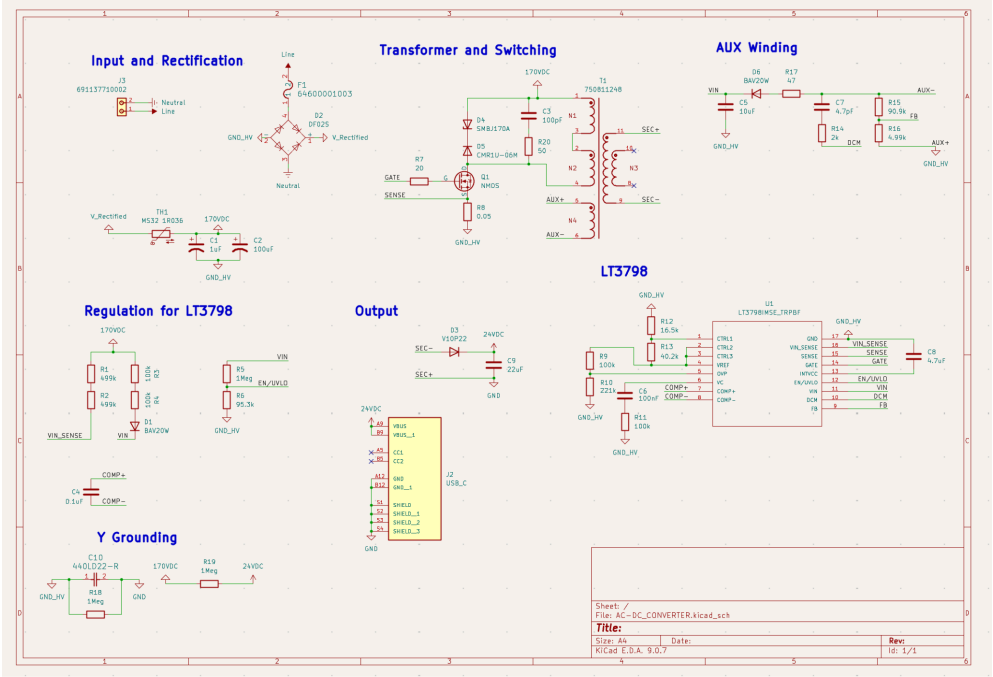

Figure 7 -Schematic Diagram for AC-DC Converter

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 (24V) and logic-level (3.3V) domains to ensure stable operation and safe interfacing between subsystems.

2.2.5 Box + Logic System

The Box System is the central hardware unit of Roomify and contains power conversion, logic control, and IR communication. The *Roomify* box will be 3D printed and painted to look like a vinyl player with the square display acting as the spinning record (Figure 1, Figure 3).

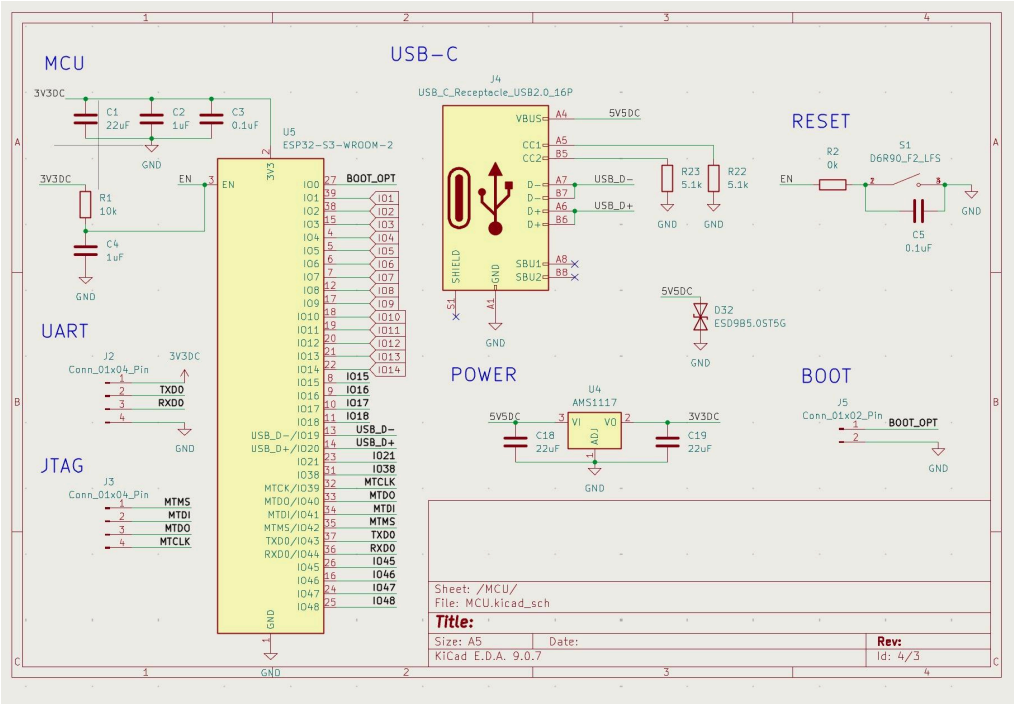

Figure 8 - Schematic Diagram for ESP32-S3 MCU

The central controller for the Logic System is an ESP32-S3 MCU (Figure 8), coordinating IR transmission/reception, 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 using the IRremote Arduino library [8]. 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 be able to receive user instructions from mobile devices via Wi-Fi, allowing users to use *Roomify* through a web application on their phones.

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.

3. Verification

3.1 Infrared Transceiver System

Throughout the testing portion of our IR transceiver, we focused on verifying the amount of current our overall system was delivering to each LED and resulting irradiance (Appendix A, Table 1). This was to ensure maximum range and to comply within our calculations and high level requirements of a range over 10 feet. In Figure 9 (below), we measured the voltage between V_{DD} and the cathode of our IR LED to calculate the voltage drop across the LED and total current.

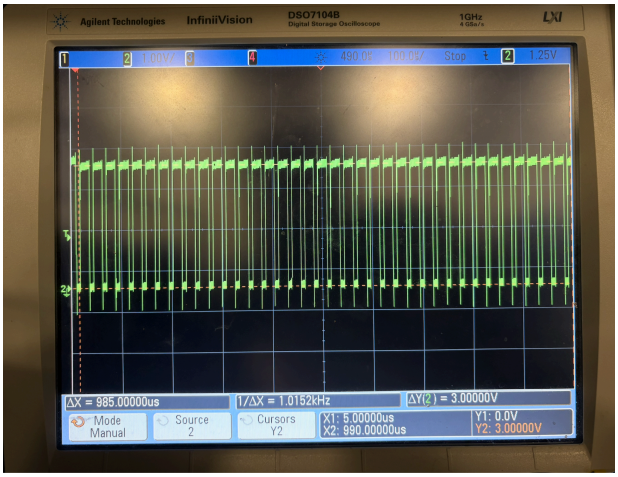

Figure 9- NEC Transmission of 0x50 (Yellow)

By measuring the voltage drop and overall impedance of the signal path, we were able to calculate the current going through the infrared LED as 0.917A (Figure 9). Following the

specifications for the IR LED, the emitted irradiance of the transmitter is around $E_e \sim 7 \frac{mW}{m^2}$ at a seven meter distance (corner to corner of a standard 15 ft. x 15 ft. x 10 ft. room) [9].

The threshold irradiance of the receiver is $E_e = 0.15 \frac{mW}{m^2}$ [10]. We were able to calculate a link budget of $|E|_{dB} = 10log_{10}(\frac{E}{E_{min}}) \rightarrow |E|_{dB} = 9.37dB$. This gave us the theoretical verification that we were transmitting with enough power to reach each end of a room, which was further validated by testing infrared signal transmission between our transmitter and a commercial LED light strip device at up to 25 ft.

The receiver unit was validated by using commercial infrared remotes, receiving and decoding their signals with our receiver unit, and retransmitting the signal to verify identical operation to the commercial remote.

3.2 Repeater System

The repeater system uses the same circuit design as the Infrared Transceiver, but rather than 14 LED in total, it uses a singular LED to retransmit in a signal direction. To validate this design, we did a manual test of aiming our PCB from a distance to an IR receiver and measuring the distance between the two devices (Appendix A, Table 2). This would allow us to calculate the peak distance our repeater can be placed without the loss of any signal and data.

From our testing, we validated that we had over 20 ft. of transmission range, and correctly relayed queued infrared signals. In addition, we repeated the same oscilloscope measurements to observe the transmitted signal and measure current flowing through the LED (same calculations as 3.1).

3.3 Display System

The Display System (Figure 10) was validated by measuring the time to preset execution completion after selecting a preset by logging out timestamps to the serial monitor and verifying that it was under 500ms (Appendix A, Table 3). Across 15 trials, we observed an average completion time of 387.2 ms, passing our 500ms requirement.

Figure 10 - Roomify Touchscreen Display

3.4 Box + Logic System

The Box System is the main control and logic unit of *Roomify*. As a result, correct operation was mostly verified through testing other subsystems, since the Box System controls them (Appendix A, Table 5). To validate latency, we logged timestamps from the time a user request was made in the web application to the time the request was processed in the box logic system. We obtained an average request processing time of 312ms, which satisfies our 500ms requirement. To validate correct HTTP request functionality, we made GET requests to our AWS server to several endpoints, verifying status 200 OK responses.

3.5 Power System

The power system took several iterations of bringing up to validate the overall system. At first, we tested it with a $50V_{DC}$ signal before plugging it into a wall outlet. Since our transformer has a 4:1 step down ratio, we set the output to be a fraction of the 24V output and tuned our board to output around 2V (Figure 11).

Figure 11 - 50 V_{DC} to 2V output

Once we were able to validate our design at 50V, we plugged it into the wall to supply the 120VAC to our board. A major issue arose and showed that there was a large ringing that was traced back to the PFC part of our board (Figure 12). The frequency of this oscillation was around 120Hz, which was double the input frequency of a wall outlet.

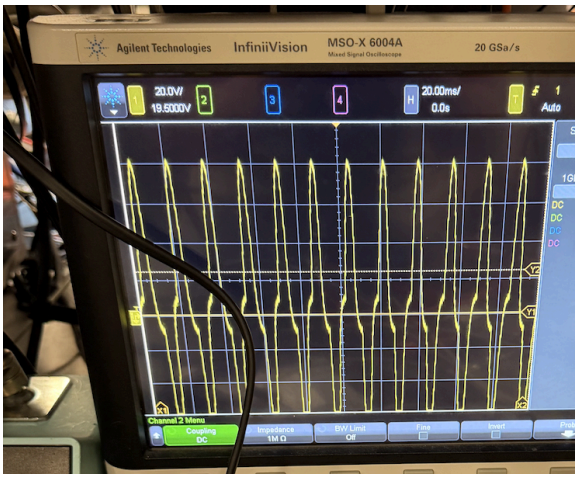

Figure 12 - Parasitic Ringing in AC/DC Output

This allowed us to narrow down to the PFC portion of our board. We were able to determine that the output capacitance which would supply a reference voltage to the auxiliary terminal of the transformer did not have a high enough capacitance. With a 22uF output, the capacitor would not have enough capacitance to store the large transient and current draw from the load causing a massive discharge and oscillation. To resolve this problem, we added a 1000uF capacitor in shunt as well as TVS diodes to clip any transients. From these changes, we were able to generate a stable 24V output from 120V AC (Figure 13), satisfying our power system requirements (Appendix A, Table 4).

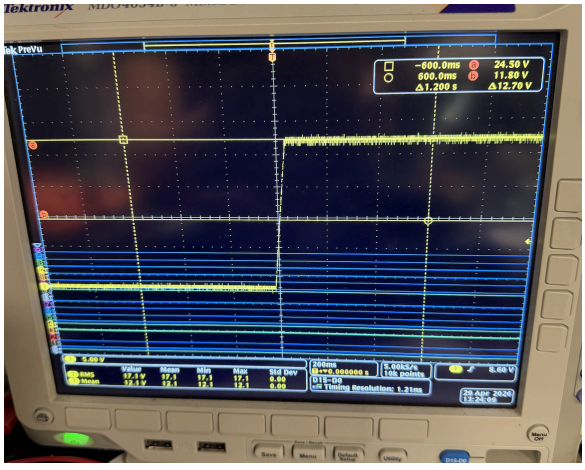

Figure 13 - 120V AC to 24V DC Output

3.6 Mobile Device Control System

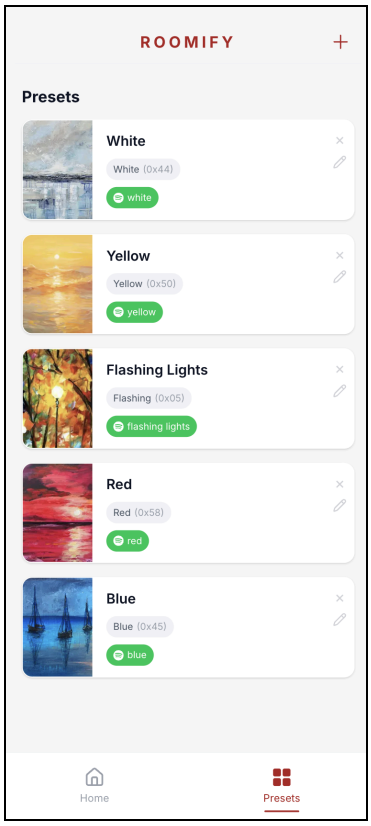

Figure 14 - Roomify Web App

The Mobile Device Control System (Figure 14) was validated by measuring the time to preset execution completion after selecting a preset in the web app by logging timestamps and verifying that it was under 500m (Appendix A, Table 6) . Across 10 trials, we observed an average completion time of 412.2 ms, passing our 500ms requirement. In addition, we performed user testing with 5 test users who had no knowledge of the project. We asked them to use the web application to pair a device, add remote buttons, and create presets, which they successfully completed with minimal instruction.

4. Costs

The total cost for parts before shipping is $321.47 (Appendix B - Parts List). Adding 10% for sales tax and shipping gives a total of $353.62. Assuming a $40/hour salary and 50 hours to complete the project for each team member (Appendix C - Work Schedule) and the standard 2.5 multiplier for employee overhead, the total labor cost is:

$$40\frac{dollars}{hour} \times 50\frac{hours}{person} \times 3\,people \times 2.5 = \$15,000$$

Summing up materials and labor, the total cost to develop and manufacture *Roomify* was $15,354. If *Roomify* were produced commercially, costs could be decreased by replacing the custom AC/DC power supply ($122.46, Table 7 - AC/DC) with an off-the-shelf version.

5. Conclusions

5.1 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) [11]. Given this context, the main considerations in designing *Roomify* were electronic safety and data privacy. *Roomify* was designed according to the UL 62368-1 standards for audio/video technology to mitigate electrical risks through electrical isolation. The offline power supply was isolated in an aluminum box acting as a Faraday cage, which was further insulated and inaccessible inside of another box. In addition, we collect minimal user data and store everything in authenticated accounts.

5.2 Future Work

Roomify successfully met all planned requirements and project goals but should be redesigned to become commercially viable. We went slightly over our budget, with the touchscreen display, *Roomify* box and custom power supply costing more to manufacture than expected. Instead, using an off the shelf power supply (e.g., a simple USB-C charger) and replacing the main *Roomify* box with more repeaters would drastically reduce cost (~$10 per repeater). The main *Roomify* box with omnidirectional transmission and touchscreen display could become an add-on, with the core system being replaced with one repeater per room device (that can be placed in direct proximity to the device receiver) and the mobile app as the main user interface. This would make *Roomify* cost around $30 per user, which is in the affordable range and much more practical than the current ~$200 design.

5.3 Conclusion

With over 20 ft. of near omnidirectional range, a scalable repeater system, a touchscreen display, and a mobile web application, *Roomify* gives users flexibility to set synchronized room devices and cohesive atmospheres while eliminating the need for multiple handheld devices (Demo Video). All requirements were met and verified, with a working $120V_{AC}$ to 24 V_{DC} offline power supply, high-range infrared transceiving (> 7 meters), signal repeaters, a 3D printed hinged box, and two user interfaces (touchscreen display and app) that allow users to add remotes and room devices and create presets. *Roomify* gives full control over room decor, lights, and sound, making it easy for students and apartment owners to manage their room devices.

6. References

[1] Chan, Manila. “Tiktok Made Me Buy It! Tiktok Home Decor Trends.” *The Daily Well*, 18 June 2025, https://thedailywell.life/gadgets-gear-for-families/tiktok-home-decor-trends/. Accessed 27 Feb. 2026.

[2] Amendola, Caterina, et al. “Optical Characterization of 3D Printed PLA and ABS Filaments for Diffuse Optics Applications.” *PloS One*, U.S. National Library of Medicine, 16 June 2021, https://pmc.ncbi.nlm.nih.gov/articles/PMC8208549/. Accessed 27 Feb. 2026.

[3] *Spotify Record Player STLS + Build Manual | Concept Bytes*, https://www.patreon.com/posts/spotify-record-127390794. Accessed 27` Feb. 2026.

[4] “Data Formats.” *Vishay Semiconductors.* https://www.vishay.com/docs/80071/dataform.pdf. Accessed 27 Feb. 2026.

[5] “Remote Control Transceiver (RMT).” *Espressif Systems*, docs.espressif.com/projects/esp-idf/en/stable/esp32s3/api-reference/peripherals/rmt.html. Accessed 27 Feb. 2026.

[6] “LT3798 Isolated No Opto-Coupler Flyback Controller with Active PFC” *Linear Technology*, www.analog.com/media/en/technical-documentation/data-sheets/3798fa.pdf. Accessed 27 Feb. 2026.

[7] “LT1170/LT1171/LT1172 High Efficiency Switching Regulators”. *Linear Technology*, www.analog.com/media/en/technical-documentation/data-sheets/117012fi.pdf. Accessed 27 Feb. 2026.

[8] “IRremoteESP8266.”*Arduino Docs*, https://docs.arduino.cc/libraries/irremoteesp8266/. Accessed 27 Feb. 2026.

[9] “5mm Infrared LED Technical Data Sheet.” *Adafruit*, cdn-shop.adafruit.com/datasheets/IR333_A_datasheet.pdf. Accessed 27 Feb. 2026.

[10] “IR Receiver Modules for Remote Control Systems.” *Vishay Semiconductors*. https://www.vishay.com/docs/82491/tsop382.pdf. Accessed 27 Feb. 2026.

[11] “IEEE Code of Ethics.” *IEEE*, www.ieee.org/about/corporate/governance/p7-8. Accessed 27 Feb. 2026.

Appendix A - Requirements and Verification

Table 1 - Infrared Transceiver Subsystem RV

Requirements	Verification
• Receive and decode 38kHz NEC infrared signals with irradiance > 0.15 $\frac{mW}{m^2}$.	• Point and click a button with a known hex code on a device remote (LED light remote) from ~20ft. away at the *Roomify* infrared receiver. • Compare decoded signal from *Roomify* Transceiver System to known remote hex code value and verify accuracy.
• Transmit 38kHz NEC infrared signals at > $130\frac{mW}{sr}$.	• Test transmission of a valid remote code signal with a device receiver at up to 20 ft. and verify correct, predictable operation (e.g. turn LED lights red, then green, and then blue). • [Oscilloscope] Connect the transmit pin of the transceiver unit to the oscilloscope and observe the waveform of the transmitted signal (check for noise, voltage difference between highs and lows, etc.)
• (Near) Omni-directional transmission of infrared signals, > 300° coverage along any axis.	• Place the *Roomify* box 10 ft. away from a device receiver at the same height. • Continuously rotate the box by 90 degrees, testing correct reception of the signal at the oscilloscope (no bit flips, relatively unnoisy signal).

Table 2 - Repeater Subsystem RV

Requirements	Verification
• Receive and decode 38kHz NEC infrared signals with irradiance > 0.15 $\frac{mW}{m^2}$.	• Point and click a button with a known hex code on a device remote (LED light remote) from ~20ft. away at the *Roomify* infrared receiver. • Compare decoded signal from *Roomify* Transceiver System to known remote hex code value and verify accuracy.
• Transmit 38kHz NEC infrared signals in a single direction at > $130\frac{mW}{sr}$	• Test transmission of a valid remote code signal with a device receiver at up to 20 ft. and verify correct, predictable operation (e.g. turn LED lights red, then green, and then blue). • [Oscilloscope] Connect the transmit pin of the transceiver unit to the oscilloscope and observe the waveform of the transmitted signal (check for noise, voltage difference between highs and lows, etc.)

Table 3 - Display Subsystem RV

Requirements	Verification
• Process user touch and execute action within 100ms. Complete preset execution within 500ms of the user selecting a preset.	• Measure average end-to-end latency across >10 trials of selecting a preset and timing the response. • Requirement passes if worst-case latency < 500ms for present execution and < 100ms from when the preset begins executing.
• Display stored user presets and settings.	• Preload multiple presets involving >3 room devices into memory on the *Roomify* MCU. • Verify correct preset names, correct icons, and correct associated IR mapping. • Use *Roomify* touchscreen display to select a preset and confirm matching operation of room devices.

Table 4 - Power Subsystem RV

Requirements	Verification
• Convert 120V AC from outlets to 24VDC.	• Connect the Roomify box to a 120V AC outlet and power on the system. • Using PCB test points, place the multimeter negative probe on GND. • Measure the 24V rail with the positive probe and confirm voltage is 24V ± 0.5V. • Measure the 3.3V rail and confirm voltage is 3.3V ± 0.3V. • Use an oscilloscope to probe 24V and 3.3V rails. • Use the oscilloscope to measure ripple voltage and confirm that it is within ±0.5V
• Power system should provide peak 1A for the Roomify system at 3.3V +/- 0.3V.	• Connect an electronic load to the 3.3V rail. • Increase load current gradually up to 1A. • Monitor voltage using a multimeter. • Confirm voltage remains within 3.3V ± 0.3V under full load.
• The battery system should provide peak 500mA at 3.3V +/- 0.3 V for *Roomify* repeaters.	• Power repeater using a fully charged battery. • Connect electronic load to 3.3V output. • Increase current draw up to 500mA. • Confirm voltage remains within 3.3V ± 0.3V.

Table 5 - Box Subsystem RV

Requirements	Verification
• Control the Transceiver System	• Send IR transmission commands to the Transceiver System. • Verify correct operation of room devices. • [Oscilloscope] Use an infrared receiver connected to an oscilloscope to validate signal transmission (check that bits match).
• Receive and process user requests over Wi-Fi within 500ms	• Connect system to Wi-Fi network. • Trigger a request from the mobile/web application. • Confirm that average action execution latency is < 500ms.
• Store user information (presets or remote signal codes) to a database over Wi-Fi reliably	• Restart the *Roomify* system and reload the application. • Verify stored data persists and matches expected values.
• Make valid HTTP requests over Wi-Fi to Spotify API	• Use ESP32-S3 to send a HTTP request to Spotify to play a song. • Verify HTTP status code is 200. • Confirm correct behavior on the Spotify account (song plays).

Table 6 - Mobile Device Control Subsystem RV

Requirements	Verification
• Communicate with the *Roomify* Box System over Wi-Fi with < 500ms latency	• Connect the mobile device and *Roomify* Box to the same Wi-Fi network. • Verify that average response latency is < 500ms after pressing a button in the app.
• Display stored user presets and settings.	• Create a user account with predefined presets stored in the database. • Log into the web application using the account. • Verify that all presets, names, icons, and settings are correctly displayed.

• 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).	• Select at least 5 users unfamiliar with the system.
	• Provide a short 1 minute explanation of the application's purpose (no walkthrough).
	• Assign tasks:
	  ○ Add a new remote
	  ○ Create a preset
	  ○ Trigger a device
	• Measure the time required to understand and complete tasks and verify that < 3 minutes of instruction were necessary.

Appendix B - Parts List

AC/DC

Table 7 - AC/DC Parts List

Order	Link	Quantity	Cost	Total Cost	Notes
TERM BLK 2POS SIDE ENTRY 5MM PCB	Link	4	$0.37	$1.48	
BRIDGE RECT 1P 200V 1.5A 4-SDIP	Link	2	$0.98	$1.96	
NTC THERMISTOR ICL P13/10	Link	2	$1.10	$2.20	
FUSE BLOCK CART 250V 6.3A PCB	Link	2	$1.15	$2.30	
CAP ALUM POLY 2.2UF 20% 250V TH	Link	2	$1.37	$2.74	
CAP ALUM POLY 220UF 20% 250V TH	Link	2	$28.08	$56.16	
DIODE STANDARD 150V 400MA SOD123	Link	2	$0.15	$0.30	
CAP CER 2200PF 760VAC Y5U RADIAL	Link	2	$2.82	$5.64	
DIODE SCHOTTKY 200V 3.1A TO277A	Link	2	$1.37	$2.74	
CONN RCPT TYPE C 6P SMD RA	Link	2	$0.60	$1.20	
TVS DIODE 170VWM 275VC DO214AA	Link	2	$0.38	$0.76	
DIODE STANDARD 600V 1A SMA	Link	2	$0.80	$1.60	
RES 0.5 OHM 1% 2W 2512	Link	2	$0.37	$0.74	
RES SMD 100 OHM 5% 2W 2512	Link	2	$0.44	$0.88	
MOSFET N-CH 600V 21A TO263-3	Link	2	$4.70	$9.40	
XFMR LED DR AC/DC CONV 300UH TH	Link	2	$7.48	$14.96	
RES 200K OHM 1% 1W 2512	Link	2	$0.24	$0.48	
IC OFFLINE SWITCH FLYBACK 16MSOP	Link	2	$8.30	$16.60	
RES SMD 5.1K OHM 5% 1W 2512	Link	2	$0.16	$0.32	
				$122.46	

MCU

Table 8 - Microcontroller Parts List

Order	Link	Quantity	Cost	Total Cost	Notes
ESP32-S3-WROOM-1	Link	1	$5.92	$5.92	
CONN RCP USB2.0 TYP C 24P SMD RA	Link	1	$1.67	$1.67	
TVS DIODE 5VWM SOD923	Link	1	$0.10	$0.10	
IC REG LINEAR 3.3V 1A SOT-223-3L	Link	1	$0.29	$0.29	
CONN RCPT TYPE C 6P SMD RA	Link	1	$0.60	$0.60	
DIODE SCHOTTKY 20V 1A SOD123FL	Link	2	$0.15	$0.30	
DIODE SCHOTTKY 30V 200MA SOT23-3	Link	1	$0.10	$0.10	
TVS DIODE 3.3VWM 11.5VC SOD923	Link	1	$0.16	$0.16	
FUSE BOARD MOUNT 5A 32VDC 0603	Link	1	$0.31	$0.31	
IC REG BUCK BOOST ADJ 5A DDPAK-5	Link	1	$19.96	$19.96	
TERM BLK 3POS SIDE ENTRY 5MM PCB	Link	1	$0.61	$0.61	
TERM BLK 2POS SIDE ENTRY 5MM PCB	Link	15	$0.37	$5.55	
IC GATE DRVR LOW-SIDE SOT26	Link	14	$0.32	$4.48	
MOSFET N-CH 20V 800MA SSM	Link	14	$0.32	$4.48	
FIXED IND 50UH 2.6A 71 MOHM TH	Link	1	$2.34	$2.34	
				$46.87	

Box

Table 9 - Box Parts List

Order	Link	Quantity	Cost	Total Cost	Notes
Square 4" TTL TFT RGB-666 Display Screen with Touchscreen	Link	1	$44.95	$44.95	
Adafruit Qualia ESP32-S3	Link	1	$19.95	$19.95	
Super-bright 5mm IR LED	Link	25	$0.68	$17.00	
IR Receiver Sensor	Link	3	$1.95	$5.85	
Box 3D Print	Link	1	$35.00	$35.00	
				$122.75	

Repeater

Table 10 - Repeater Parts List

Order	Link	Quantity	Cost	Total Cost	Notes
TERM BLK 3POS SIDE ENTRY 5MM PCB	Link	1	$0.61	$0.61	
IC GATE DRVR LOW-SIDE SOT26	Link	1	$0.32	$0.32	
DIODE SCHOTTKY 20V 1A SOD123FL	Link	4	$0.15	$0.60	
MOSFET N-CH 20V 800MA SSM	Link	2	$0.32	$0.64	
TERM BLK 2POS SIDE ENTRY 5MM PCB	Link	2	$0.37	$0.74	
ESP32-S3-WROOM-1	Link	1	$5.92	$5.92	
CONN SIL HDR MALE PIN 32POS TIN	Link	1	$7.54	$7.54	
IC REG LINEAR 3.3V 1A SOT-223-3L	Link	1	$0.29	$0.29	
TVS DIODE 5VWM SOD923	Link	1	$0.10	$0.10	
CONN RCP USB2.0 TYP C 24P SMD RA	Link	1	$1.67	$1.67	
IC REG BUCK BST ADJ 1.25A 8PDIP	Link	1	$8.36	$8.36	
TVS DIODE 3.3VWM 11.5VC SOD923	Link	1	$0.16	$0.16	
DIODE SCHOTTKY 30V 200MA SOT23-3	Link	1	$0.10	$0.10	
FIXED IND 50UH 2.6A 71 MOHM TH	Link	1	$2.34	$2.34	
				$29.39	

Appendix C - Work Schedule

Week	Task	Person
March 1 - March 7	Make BOM and order parts	Everyone
	Complete Box Design	Benjamin
	Complete Repeater PCB Design	Owen
	Complete Power Supply & MCU PCB Design	Warren
March 8 - March 14	Solder Power Supply & MCU PCBs	Benjamin and Owen
	Revise Power Supply & MCU PCB Designs	Warren
	Revise Repeater PCB Design	Owen
March 15 - March 21	Complete physical design for Repeater (3D Print)	Benjamin
	Set up user accounts, server, and data storage.	Owen
	Revise Power Supply & MCU PCB Designs	Warren
March 22 - March 28	Complete add remote flow (receive and store infrared remote codes)	Benjamin
	Begin working on *Roomify* web-app interface	Owen
	Working offline power supply	Warren
March 29 - April 4	Add transmission capability (operate LED string lights with *Roomify*)	Benjamin
	Finish V1 of *Roomify* web-app interface and link with server and user accounts.	Owen
	Modify aluminum enclosure to encase power supply and connect with MCU board	Warren
April 5 - April 11	Working UART communication between *Roomify* MCU board and display screen board	Benjamin
	Adapt *Roomify* web-app interface to square display screen	Owen
	Assemble *Roomify* box (3D print + assembly), working repeaters	Warren
April 12 - April 18	Working touchscreen control of *Roomify* through the square display screen	Benjamin
	Display Spotify/music controls on *Roomify* square display screen	Owen
	Test and verify subsystems	Warren
April 19 - April 25	Finalize assembly + integration tests, prepare for demo	Everyone
April 26 - May 2	Final Presentation + Demo	Everyone

Appendix D - Project Documents

Demo Video

Demo Video Link (Youtube)

Presentation

Senior Design Final Presentation

Demo Handout

ECE 445 - Roomify Demo Handouts