UCF Senior Design 1 (Divide and Conquer)

Smart Auto-Equalization Sound System

Group 31 Authors:

Xander Levine
Computer Engineering

Noah Bixby
Electrical Engineering

Jordan Riley
Electrical Engineering

Matias Segura
Computer Engineering

Review Committee:

Mark Maddox
Instructor

Dr. Sonali Das
Lecturer

Dr. Maedeh Sadat Fasihi
Adjunct Lecturer

Mentor:

Dr. Sreeram Sundaresh

Chapter 2 – Project Description

2.1 Motivation and Background

Audiophiles all over the world spend thousands of dollars on audio equipment and multiple hours of tuning to achieve the best performance for their respective systems and environments. In many cases, these consumers are buying multiple subsystems to manually tune the system. Each tune is only good for the exact environment and system the tune was made for, meaning that any small change to the environment or system calls for the entire system to be re-tuned. This can be quite a headache for audiophiles who want a plug and play system. This is where the Smart Auto-Equalization Sound System idea strives.

The Smart Auto-EQ Sound System aims to integrate a microcontroller (MCU), audio amplifier, distance sensors, and microphones to measure the environment's dimensions and natural frequency response to automatically adjust the audio file's equalization (EQ) and feed the signal to the audio amplifier. While in addition by implementing a mobile app interface or web-based connection, we can provide the user with a more tailored sound EQ experience.

2.2 Goals and Objectives

2.2.1 Basic Goals:

- Design and build a functional 2.1 speaker system.
- Develop a custom 2 channel amplifier using the MCU as the input for the mid/highs.
- Develop a mono channel amplifier using the MCU as the input for the subwoofer.
- Integrate a microcontroller (ESP32-S3) for DSP-based EQ adjustment.
- Devise an algorithm that uses analog input signals to construct an auto-EQ system.
- Develop a mobile app that works alongside the system that provides a manual override for customer preferences and/or monitoring.

2.2.2 Advanced Goals:

- Perform a pink noise sweep to assist the MCU and LiDAR sensors in ensuring a flat EQ response to start with when tailoring the adjustment for each location the speakers are in.
- Combine DSP with analog circuitry for low latency situations.
- Implementing adaptive filters such as Least Mean Squared, or Recursive Mean Squared algorithms to stabilize given noisy or crowded environments.

2.2.3 Stretch Goals:

- Support Bluetooth audio streaming that incorporates EQ.
- Implementation of voice control/assist, such as Siri, Alexa, Bixby, to help with EQ control settings.
- Prevent clipping or distortion by actively monitoring the Total Harmonic Distortion and adjusting in real time.
- Prevent latency when switching between a Wi-Fi or Bluetooth connection.

2.2.4 Basic Objectives:

- Utilize LiDAR sensor(s) to measure room dimensions, as well as a calibrated microphone to pick up the speakers' response. This data will help us with the signal processing.
- Create an algorithm that is fast, reliable, and that will ensure the best performance, with the use of the ESP-32's DSP library.
- Implement an LED that will blink to let the user know it is working on Auto-EQ. When it finishes, the LED will stop blinking, which will indicate to the user that the run is complete.
- Investigate the correct amplification factor to achieve sufficient power to the system.
- Achieve a base equalization setting. This will help us and the algorithm to have a starting point, to make the equalizer settings easier for the user and the MCU.

2.2.5 Advanced Objectives:

- Code an app using Java/Python where the user can choose between running Auto-EQ and manually changing the settings to their convenience.
- Provide a fast and reliable connection between the app and the system. The ESP32-S3 comes with built-in Bluetooth and WI-FI modules that will help achieve this task. A separate module could be considered if necessary.

2.2.6 Stretch Objectives:

- Achieve a runtime of N*log(N), where N is the number of samples. We believe that the ESP-32S3 can handle the computational power required for this task. We can achieve this by using FFTs, as well as the ESP's dedicated library for DSP.
- Provide the user with real-time data. Histograms and charts are helpful for displaying data.

2.3 Features and Functionalities

- Calibration using a calibrated microphone and LiDAR sensors.
- Custom amplifier built for the subwoofer and mid/high drivers.
- Real-time DSP EQ on an embedded MCU.

- Custom PCB with modules including MCU, DAC/ADC, LiDAR sensors, and amplifiers, LEDs, ON/OFF button, voltage regulators.
- Mobile app interface for user control.
- Multicolor LED to indicate to the user the current state of the system.

2.4 Related Work

2.4.1 Commercial Systems:

- DSPeaker Anti-Mode X2 (1)
- DSPeaker Anti-Mode 2.0 Dual Core Room Correction Floor Sample (2)
- IK Multimedia ARC Studio Room Correction System (3)

These commercial systems contain similar functionality to the project we are designing here, however on a smaller scale. These products are designed for car stereo systems while we are designing a system for a much larger scale with rooms. In addition, these products are no longer sold through their website, giving us an edge in the market.

- REW: Room Equalizer Wizard (4)

This software aims to help the user to find the most suitable equalizing settings based on their room's acoustic response. The user must have access to a computer to download the software. In addition, the program requires a microphone that is used to pick up the signals. Our project differs from this software by first integrating an entire sound system together with the software; this way, the user does not need to get any other equipment.

2.4.2 Prior Academic Projects:

- Pegasus: A Multi-Effect Audio Processor (5)

This was a project that was completed last fall, 2024, in the Senior Design Capstone course, here at UCF. Like our project, this group is adjusting their audio outputs to create a new signal. The difference, however, is that this project is creating different effects, such as adjusting the pitch, tempo, and reverb of the output signal, which would mean the sound quality is fixed and is the same from one location to the next.

2.5 Key Engineering Specifications

Specification	Target Value	Notes
Frequency Response Range	20 Hz – 20 kHz	Full audio spectrum
Total Harmonic Distortion (THD)	≤ 0.5%	At rated power
Amplifier Output Power	50W RMS (subwoofer), 25W RMS (mid/high)	Per channel

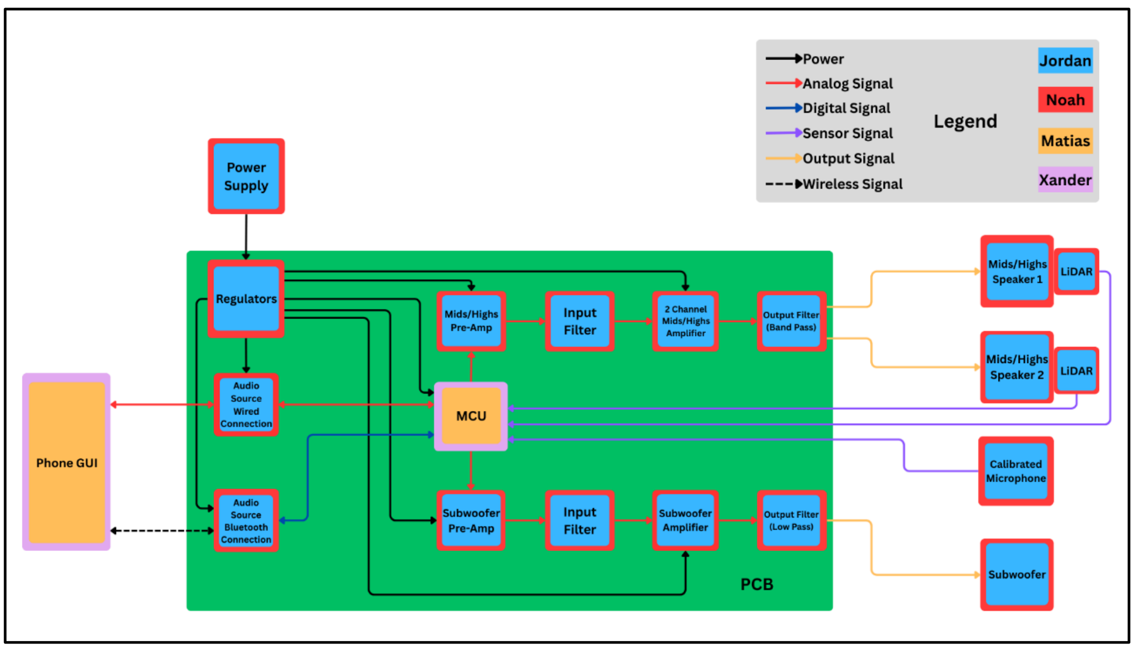
MCU Processing	≥ 240 MHz dual-core with FPU/DSP	ESP32-S3 candidate
LiDAR Accuracy	≤ ±30 cm for ToF	Room measurements
Microphone Accuracy	≤ ±1 dB	Acoustic measurements
Power Supply	±12V to ±24V (audio), +5V/+3.3V (logic)	Linear + switching regulation
Latency (Mic-to-DSP-EQ)	≤ 20 ms	Real-time correction

Table 2.5.0.1: Engineering Specification Table

2.5.1 Highlighted Demonstrable Specs:

- Achieving a frequency range of 20Hz – 20kHz for a fully enjoyable audio spectrum for the user.
- Having an output power of 50W RMS for our subwoofer and 25W RMS for the mid/high speakers.
- With customer satisfaction in mind, we want to ensure our speakers produce clear crisp sound, hence highlighting the limit of distortion.

2.5.2 Hardware Block Diagram



Pictured above is the hardware system diagram. The diagram illustrates the main hardware components required to process an audio signal from a source and provide an output to a 2.1 stereo system.

Figure 2.5.2.1: Hardware Block Diagram

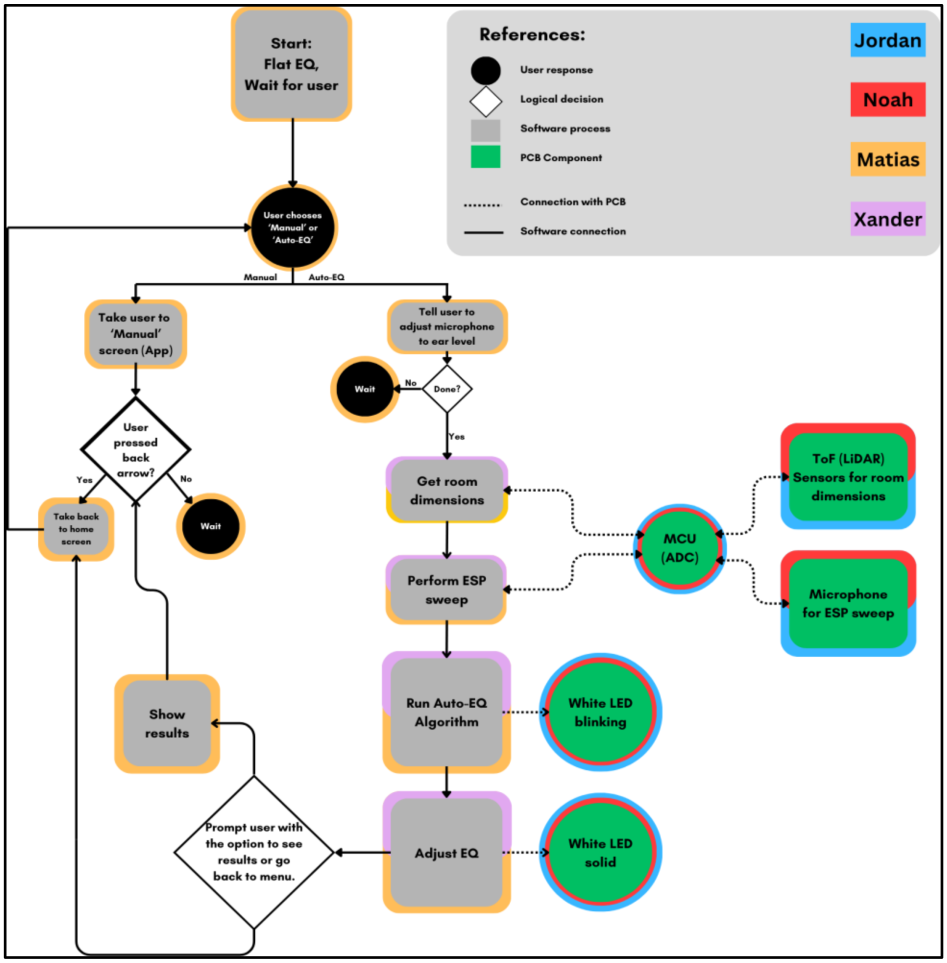
The input to the system includes three total sensors, Bluetooth, wired device connectivity, and power. Two sensors are LiDAR, that should be positioned on the speakers to detect the room's dimensions and send the information to the MCU for processing. The third sensor is a calibrated microphone that will record the room's natural acoustics when white noise is played from the speakers. The recording is sent to the MCU for processing and filtering of the audio file to create a flat frequency response across the audio band. Additionally, the system will accept either Bluetooth or wired phone connection to communicate with the GUI and send the user's preferred audio to the device. The power supply should supply the power for the entire system from a standard wall outlet.

The system also has five total outputs, three of which are to drive the speakers using the onboard amplifiers. This output should accept speaker wire gauges ranging from 12 to 16-gauge to ensure adequate compatibility with a variety of speaker combinations. The final two outputs are for connectivity to the phone GUI, whether it be Bluetooth or a wired connection. These outputs were also mentioned as inputs as the phone GUI should be able to send and receive data from the system.

On the PCB, there are a variety of subsystems. The MCU is at the heart of the PCB as it oversees receiving commands sent from the phone and three sensors, as well as the reception of audio files from the phone. Additionally, the MCU must process the data sent from all inputs, aside from the power, and execute a DSP algorithm to filter out peaks and valleys in the frequency response. The MCU must also be capable of sending the audio, whether digital or analog, to the amplification section on the board.

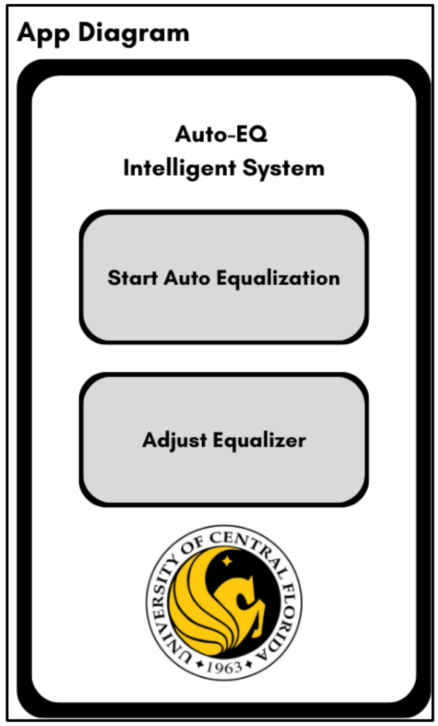
The amplification section of the board has multiple stages and channels to achieve an efficient, high quality, powerful output. All channels in the amplification section have the same flow but are separated in the hardware diagram as the mids/highs and subwoofer because they are amplifying separate sections of the audio band and with different amplification factors due to the larger power requirements to accurately and effectively reproduce low frequency signals. At the forefront of these sections is the pre-amp, this section ensures the audio signal has a large enough amplitude for the rest of the amplifier to work efficiently. Next, the input filter will be a low pass filter to remove unnecessary signals that cannot be heard by human ears or frequencies that will cause damage to the drivers. The channels' respective amplifier sections do most of the amplification to prepare to drive the speakers. The final stage is responsible for filtering out any unwanted frequencies that passed through the input filter and got amplified in the amplification stage.

2.5.3 Software Flowchart



The figure above represents the Software Diagram. Here we show the complete software flow to acquire the necessary data to run the Auto-EQ algorithm, as well as the required hardware components.

Figure 2.5.3.1 Software Flowchart



Above is the User Interface intended to form a nexus between the user and the system. This app gives the user a simple and accessible way to connect and enjoy the Auto-EQ system.

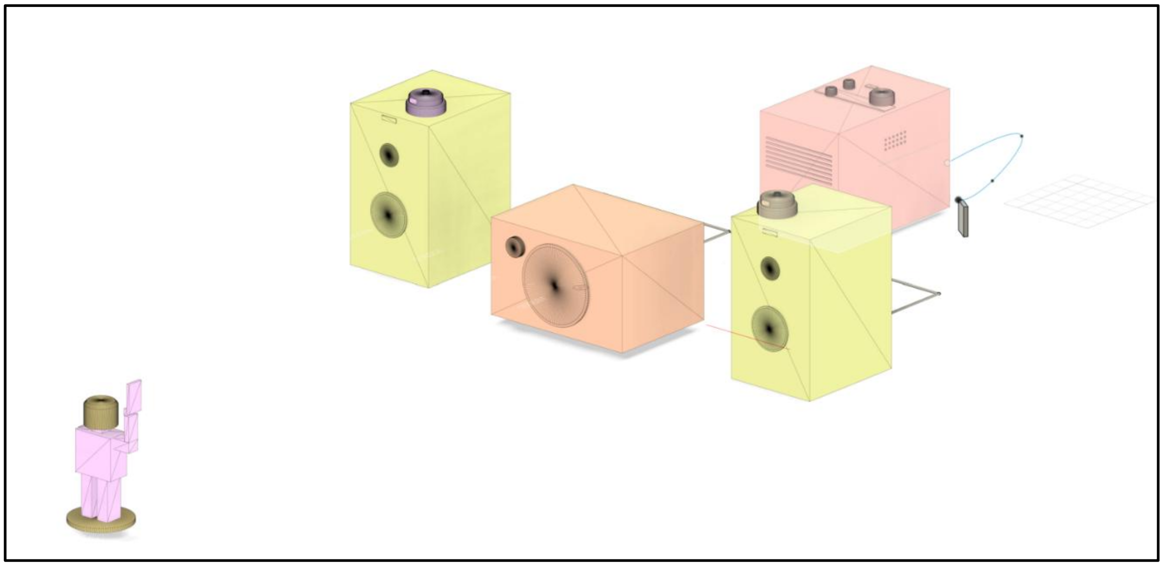
Figure 2.5.3.2 User Interface

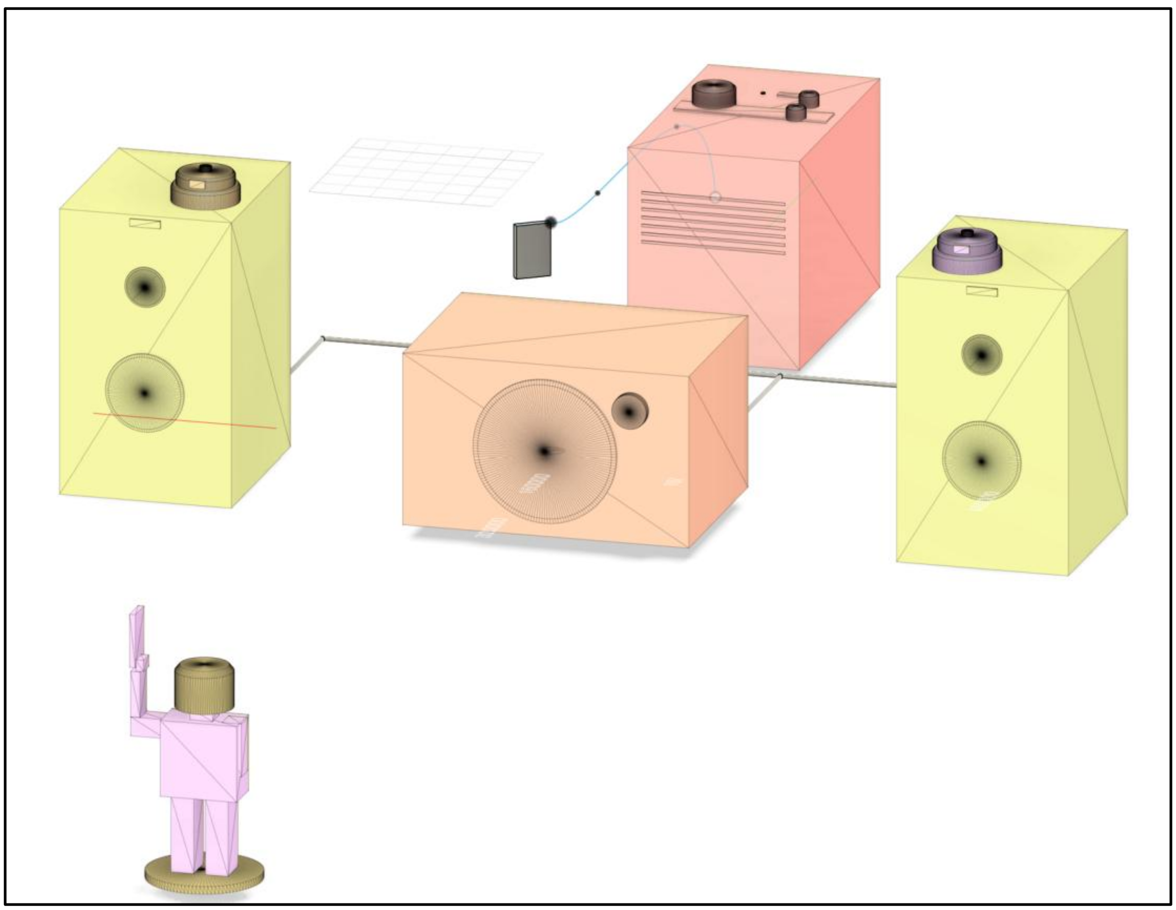
The software model intends to integrate an app that the user can use as a way to start the system. At first, we load a flat EQ file into that will give us a starting point from where we then adjust the EQ settings when we run the algorithm, or if the user desires to change them manually. This is the first logical decision our flow takes: the user can either do it themselves or let the machine do its job. If the user decides to manually adjust the settings, then we give them the option and wait for them to go back by pressing the back arrow.

If, instead, they choose to run the Auto-EQ, then we first let them know that they need to adjust the calibrated microphone to where they will be listening. We will also add a multi-color LED to indicate it; at this point the LED will be blue. Once they do that, we then acquire the necessary signals (using the ToFs and microphone) to run the algorithm. When the algorithm is running, we will make the LED blink in yellow to let the user know that the system is working. This algorithm will be implemented using the ESP-32's dedicated ESP-DSP library [(6)], as well as complex mathematical analysis such as Fast Fourier Transforms (FFT), which are still under investigation.

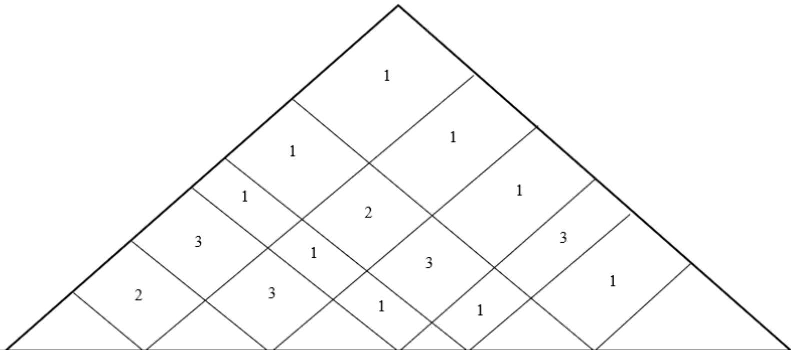
One of our end goals is to give the user information while the system is running. This means, with the use of diagrams such as histograms, being able to show the user what is actually happening inside the system. We aim to achieve this, however, at the end of the run, so that the user can look deeper into the results. Otherwise, they can choose to go back to the main screen on the app and enjoy their intelligently equalized sound system.

2.5.4 Prototype Illustration





2.5.5 House of Quality



	Total Harmonic Distortion -	Capability +	Digital Signal Processing +	Cost -	Calibration Time -	Manufacturability +
Audio Quality +	3	2	3		1	
Ease of Use +		1	2		3	1
Affordability +				3		2
Connectivity +		3	1		1	1
Size -	1	1		1		3

Above is the House of Quality, here we can see the correlation between the engineering and customer requirements for our project (denoted in the bottom section of the graphic) and the tradeoffs between each of the engineering requirements (denoted by the roof of the document). Here, 3's denotes a favorable connection, 2 denotes a medium connection, and 1 denotes a negative connection

Figure 2.5.5.1 House of Quality

Chapter 10 – Administrative Content

10.1 Budget and Financing

Item	Estimated Cost	Notes
MCU/DSP	$25	Dev Board Testing
Amplifier IC	$15	Stereo + Subwoofer
Microphones	$10	2 Units
LiDAR Sensors	$65-$100	2 Units
Multi-Colored LED	$5.95	1 Unit
ADC/DAC Components	$15	Precision Audio Converters
Power Supply	$30	Linear + Switching Regulator

PCB Fabrication	$100	Multi-Layered Board
Enclosure Material	$100	Wood/Acrylic
Misc. PCB Components	$50	Passive Components

Table 10.1.1: Tentative Bill of Materials

10.2 Project Milestones

10.2.1 Senior Design 1

Weeks	Completed Activity
1-2	Project Idea Created and Group Member Formation
3	Project Proposal and Committee Formation
4	Project Proposal Meeting Held with the Group and Mentor / Revision of Project Proposal
8	Midterm Report
9	Revision of Our Midterm Report
14	Final Report

Table 10.2.1.1: Senior Design 1 Milestones

10.2.2 Senior Design II

Weeks	Completed Activity
1-3	Fabricate PCB and Begin Enclosure Construction
4-8	Integrate DSP Algorithms Onto MCU
9-12	Full System Integration and Testing
15	Showcase Final Functional Prototype

Table 10.3.2.1: Senior Design 2 Milestones

10.3 Work Distributions

Electrical Engineering	Responsibilities
Noah Bixby	PCB Fabrication
	Amplifier Design
	Power Supply
	Regulators
Electrical Engineering	**Responsibilities**
Jordan Riley	Amplifier Design
	Bluetooth Hardware
	USB Connectivity
	Speaker Fabrication
	Sensor Integration
Computer Engineering	**Responsibilities**
Matias Segura	GUI Design
	UI/UX Design
Computer Engineering	**Responsibilities**
Xander Levine	DSP Algorithm
	Auto-EQ Algorithm

This is a tentative table for each team member's responsibilities. Matias's and Xander's work will be related so the two will be sharing some of the other's responsibilities. The same principal applies for Jordan and Noah, especially with the amplifier design.

Table 10.3.1

Appendices

Appendix A:

(1) "Anti-Mode X2." DSPeaker, www.dspeaker.com/anti-mode-x2. Accessed 4 Sept. 2025.

(2) "DSPeaker Anti-Mode 2.0 Dual Core Room Correction Floor Sample - Bing." Bing, 2016, www.bing.com/search?q=DSPeaker+Anti-Mode+2.0+Dual+Core+Room+Correction+Floor+Sample&cvid=41c7e0c6efe6420399faa51acf909d8e&gs_lcrp=EgRlZGdlKgYIABBFGDkyBggAEEUYOTIGCAEQRRg8MgcIAhDrBxhA0gEHNTYzajBqNKgCCLACAQ&FORM=ANAB01&PC=HCTS. Accessed 4 Sept. 2025.

(3) "ARC Studio." Ikmultimedia.com, 2025, www.ikmultimedia.com/products/arcstudio/?irclickid=0%3Am27Ex4BxycRcp0o4yNPy4SUkpw0hxnwWLgxI0&sharedid=EdgeBingFlow&irpid=2003851&prodsku=&irgwc=1. Accessed 4 Sept. 2025.

(4) Mulcahy, John. "REW - Room EQ Wizard Room Acoustics Software." Roomeqwizard.com, 2018, www.roomeqwizard.com/.

(5) Pegasus: - UCF Ece - University of Central Florida, www.ece.ucf.edu/seniordesign/su2024fa2024/g13/SD1_final_report_Group13.pdf. Accessed 4 Sept. 2025.

(6) Espressif Systems Co, "ESP-DSP Library". Accessed Sep. 3, 2025. https://docs.espressif.com/projects/esp-dsp/en/latest/esp32/index.html

Appendix B: Copyright permissions (if external diagrams/images used).