Auto-EQ Sound System (with mobile application)

Noah Bixby, Xander Morris Levine, Jordan Riley, Matias Segura

Dept. of Electrical Engineering and Computer Science, University of Central Florida, Orlando, Florida, 32816-2450

Abstract* — *This **paper presents the design and development of a Smart Auto-Equalization Sound System intended to improve user audio playback through digital signal processing and customizable control. The system combines a 2.1 speaker platform, custom analog amplification, an ESP32-S3 microcontroller, a calibrated measurement microphone, and a mobile application for wireless interaction. The design supports two main operating approaches: an automatic equalization mode, where the system analyzes room response and generates corrective filtering, and a manual equalization mode, where the user directly adjusts frequency bands through the mobile app. This dual-mode design improves flexibility, preserves user control, and allows the system to remain practical even if full Auto-EQ implementation is limited.**

Index Terms* — *Audio **amplifiers, Bluetooth, digital signal processing, embedded systems, equalization, ESP32-S3, mobile applications, microphones.**

I. INTRODUCTION

High-quality audio playback depends not only on speaker quality, but also on the acoustic behavior of the listening environment. Room reflections, standing waves, speaker placement, and enclosure characteristics all affect the final sound heard by the user. As a result, a system that sounds balanced in one environment may sound overly bright, muddy, or bass-heavy in another. Many existing solutions require separate calibration tools, external microphones, desktop software, or a user with enough technical knowledge to manually interpret and correct the frequency response.

The Smart Auto-Equalization Sound System was developed to reduce that complexity. The purpose of the project is to create a 2.1 audio system that can analyze acoustic response and apply equalization with minimal user effort, while also allowing direct user control when desired. The system is intended to combine embedded processing, custom amplification hardware, and an accessible mobile interface into a single platform. In contrast to traditional manual equalization methods, the proposed system aims to provide a more user-friendly and self-contained calibration process, while still preserving manual tuning as an important feature rather than treating it as a fallback.

At the center of the design is the ESP32-S3 microcontroller, selected because it provides sufficient embedded processing capability for digital signal processing tasks while also supporting wireless connectivity for the application interface. The ESP32 is a very popular choice in many embedded projects, and it was selected because of its flexibility and adaptability, as well as the extensive support available online. In one operating mode, the system uses a calibrated measurement microphone (Dayton EMM-6) to capture the room's response when a test signal is played. The recorded data is then sampled by an external ADC unit (ADS8320) with a 16-bit (unsigned) resolution. This data is later processed using DSP techniques such as Fast Fourier Transform, windowing, smoothing, and filtering to identify peaks and dips in the response. In another operating mode, the user may bypass automatic correction and instead directly adjust equalizer levels within the application to tailor playback according to personal listening preference. The filters implemented are shared between these two modes, meaning that the user can still adjust the gain levels after the automatic calibration finishes. When music is transmitted to the ESP, the data is captured and stored in large chunks for filter processing. The microcontroller amplifies (or reduces) the desired frequency bands (following the current filter coefficients setting) and then sends the processed data to an external DAC (PCM5122) which outputs the analog signal to the amplification networks (subwoofer and mids/highs). The MCU also implements a digital Crossover Filter, which protects the subwoofer from receiving high frequencies and, on the other hand, the mids/highs from receiving low frequencies. This is a short but necessary software step used to protect the hardware.

This dual-mode capability is important to the overall project. While Auto-EQ is one of the project's most technically ambitious goals, manual EQ control ensures that the system remains functional, useful, and interactive even if the automatic calibration algorithm is not fully finalized. In many real-world audio products, users prefer to customize bass, mids, or treble on their own rather than rely entirely on automatic correction. For this reason, manual equalization should be viewed as a core feature of the system rather than a secondary compromise.

The project also includes two custom-designed amplifier paths: a dual-channel amplifier for the mid/high speakers and a dedicated mono amplifier for the subwoofer. These amplifier subsystems, together with the digital board, the microphone interface, the DAC/ADC modules, the power regulation circuitry, and the mobile application, form a complete engineering system rather than a single isolated subsystem.

The importance of the project lies in both technical depth and practical value. From an engineering standpoint, it combines analog audio design, embedded DSP, and software integration. From a user standpoint, it targets a familiar and frustrating problem: needing to retune speakers every time the system or listening environment changes. The goal is therefore not only to build a functioning audio system, but to make calibration and sound customization more accessible, repeatable, and user-friendly.

II. SYSTEM OVERVIEW

The Smart Auto-Equalization Sound System is structured as a 2.1-channel platform composed of two mid/high channels and one subwoofer channel. The overall goal is to receive user audio, provide controllable equalization, and reproduce sound through the speaker system with improved tonal balance. The design integrates an ESP32-S3-based DSP core, a calibrated microphone, custom analog amplifiers, DAC/ADC support, Bluetooth communication, and a mobile application for control.

The system operation can be understood in two primary modes: automatic equalization mode and manual equalization mode. In automatic mode, a reference or test signal is played through the speaker system, a calibrated microphone records the resulting acoustic response in the listening environment, and the ESP32-S3 processes this data to identify response irregularities. A corrective EQ profile is then generated and applied to playback. In manual mode, the user directly adjusts equalization bands through the mobile app, allowing custom control over low, mid, and high frequency content without requiring the microphone-based calibration process.

This two-mode structure is a major strength of the design. Auto-EQ aims to improve objectivity by correcting room-based coloration, while manual EQ preserves user preference and system usability. Some users may want a flatter sound, while others may intentionally prefer more bass or brighter treble. By allowing both approaches, the system is not locked into one interpretation of “correct” sound. Instead, it provides a technically guided option and a user-driven option.

The user interacts with the system through a mobile application. The app provides a control layer for initiating Auto-EQ, adjusting equalizer settings manually, switching between operating modes, and monitoring system behavior. The design also includes a status indication concept, so the user can tell when calibration is running and when the routine is complete.

A key strength of the system is that it is not pure software and not pure hardware. Instead, it is a mixed-signal embedded audio platform in which sensing, processing, control, and amplification are tightly integrated. That makes it appropriate for senior design because it requires contributions from both electrical and computer engineering. The hardware must reliably acquire and reproduce audio, while the software must perform analysis, coordinate control flow, and support user interaction.

The system is also guided by several important engineering targets, including wide audio-band coverage, acceptable distortion performance, separate power delivery for the subwoofer and mid/high channels, and sufficiently low latency for responsive control. Even when Auto-EQ is not active, these targets still matter because the system must remain in a solid and responsive audio platform under manual operation. These target specification are delineated in Table I.

Before the beginning of the double column section, as shown above.

TABLE I
KEY ENGINEERING SPECIFICATIONS

Specification	**Target**	**Purpose**
Frequency Response	20Hz – 20kHz	Full Audio Spectrum
THD	≤ 0.5%	Maintain Audio Fidelity
Subwoofer Power	50W RMS	Low-Frequency Output
Mid/Hihg Power	25W RMS/Channel	Stereo Speaker Output
MCU Processing	≥ 240MHz dual-core	DSP Execution
Microphone Accuracy	≤ ±1dB	Acoustic Measurement
Latency	≤ 20ms	Calibration Time

III. HARDWARE DESIGN

The hardware architecture combines digital control and analog audio circuitry on a custom prototype platform. The ESP32-S3 serves as the primary control and processing device. It manages wireless communication with the mobile application, receives measurement data from the microphone path, performs signal-processing operations, and coordinates playback-related behavior. This makes it suitable for both automatic processing tasks and real-time response to user-issued EQ commands.

One of the most important hardware subsystems is the microphone. The project uses a calibrated microphone to record the room's acoustic response during calibration. Because raw microphone output cannot be sent directly into the MCU in practical form, support circuitry is required to condition and amplify the signal before the ESP32-S3 can process it. This microphone path is especially important for Auto-EQ operation, but it does not prevent the rest of the system from functioning if automatic calibration is not used.

The microphone signal path requires careful conditioning to ensure that the recorded signal accurately represents the acoustic environment. This includes amplification, filtering, and biasing stages that prepare the signal for analog-to-digital conversion. Any noise or distortion introduced in this stage directly affects the quality of the measured response and, consequently, the effectiveness of the equalization process.

In addition, the placement of the microphone during calibration plays a critical role in measurement accuracy. The system assumes that the microphone is positioned near the primary listening location, allowing it to capture a representative response of the environment. Variations in placement can lead to different measured responses, highlighting the importance of consistent measurement conditions during Auto-EQ operation.

The playback side of the system uses separate amplifier paths for different parts of the audio band. A dual-channel amplifier is used for the mid/high speakers, while a dedicated mono amplifier is used for the subwoofer. This architecture reflects the different power requirements of low-frequency reproduction compared to mids and highs. These hardware choices benefit both automatic and manual equalization modes because the system still needs strong, clean, and properly distributed amplification regardless of how the EQ settings are selected.

Power regulation is another major hardware concern. Audio systems are sensitive to noise and supply instability, especially when digital logic and analog signal paths share the same prototype. The design therefore includes regulated rails for digital circuitry and audio stages, supporting the MCU, interface circuits, and amplifier sections while minimizing unwanted coupling and preserving signal quality.

The custom PCB work is also central to the project's scope. Rather than using only development kits and prebuilt amplifier modules, the team designed boards that combine regulators, main digital circuitry, microphone support, DAC connections, and amplifier-related sections. This supports the goal of producing a real engineering prototype rather than a loose bench-top demonstration.

Importantly, the hardware platform is useful even without a finalized Auto-EQ algorithm. The amplifier system, mobile control interface, embedded processor, and equalizer-adjustment capability still form a meaningful and functional product. This is significant because it shows that the project's value is not entirely dependent on one advanced algorithmic feature. The hardware remains capable of delivering adjustable, user-controlled sound enhancements even in a manually operating state.

In addition to the high-level architecture, amplifier topology selection was a critical design consideration. Class D amplifier ICs were selected due to their high efficiency (typically above 85–90%), reduced thermal requirements, and compact implementation compared to Class AB alternatives. While Class AB amplifiers offer lower distortion characteristics, their increased power dissipation and heatsinking requirements made them less practical for an integrated embedded system. By contrast, Class D amplifiers allow the system to deliver sufficient output power to both the mid/high channels and the subwoofer while maintaining manageable thermal conditions and PCB footprint.

The system power architecture was also designed with noise isolation in mind. Separate regulation stages are used for digital and analog subsystems to minimize coupling between high-frequency switching noise and sensitive audio paths. The ESP32-S3 and communication circuitry operate on regulated low-voltage rails, while the amplifier stages are supplied by higher-voltage rails appropriate for audio power delivery. This separation improves signal integrity and helps preserve audio quality during both manual and automatic equalization modes.

From a signal-chain perspective, the system follows a structured flow beginning with user audio input, which is processed by the embedded controller and optionally modified through equalization. The processed signal is then routed through DAC stages (or equivalent digital output paths), amplified by the dedicated analog stages, and finally delivered to the speaker drivers. This end-to-end integration of digital processing and analog amplification is central to the system's ability to provide both responsive manual control and automated correction.

In addition to topology selection, amplifier performance was evaluated based on total harmonic distortion (THD), efficiency, and output power capability under realistic operating conditions. While many amplifier datasheets specify power ratings at high distortion levels (e.g., 10% THD), the system was designed to operate within significantly lower distortion limits to preserve audio fidelity. As a result, amplifier operating points were selected conservatively to ensure that output power remains within a region where THD is minimized.

Thermal considerations were also important in the amplifier design process. Even with high-efficiency Class D operation, sustained audio playback at higher output levels can introduce thermal stress. The PCB layout and component placement were therefore designed to support heat dissipation, including the use of ground planes and thermal paths to distribute heat more effectively. These considerations help maintain stable operation and prevent performance degradation over time.

IV. EQUALIZATION AND SIGNAL PROCESSING METHOD

The equalization strategy for the Smart Auto-Equalization Sound System is based on two complementary approaches: embedded automatic correction and user-controlled manual adjustment. Together, these approaches make the system more flexible and more practical as a usable audio product.

The automatic equalization pipeline can be described as a multi-stage signal processing process. First, a known test signal, specifically a logarithmic sine sweep (ranged from 20 Hz to 20kHz), is generated from a wave file stored in the ESP32-S3's flash memory and played through the DAC (and after that the speaker system) using I2S communication. This signal is chosen because it provides uniform frequency coverage over the audible spectrum while maintaining controlled amplitude characteristics. The microphone captures the resulting acoustic response, which includes both the intended signal and the effects introduced by the environment.

The recorded signal is then sampled by the external ADC unit with a sampling rate of 48kHz and stored in the ESP32-S3's PSRAM (8MB) for future processing. The following steps described the Automatic Equalization Algorithm, which is shown in **Algorithm 1**.

Algorithm 1 Automatic Equalization Algorithm

Requires: Sampled data D_s, Frequency S_f
Parameters: # of Samples N_s, FFT Size N_{fft}
Ensure: Filter Gains B_{Gains}
WAV_{fft} ← Wave_file_to_FFT()
  Preprocessing
D_{fft} ← Compute_FFT(Ds, N_s, S_f)
D_{fft} ← Apply_Calibration(D_{fft}, S_f)
H ← Wiener_Deconvolution(WAV_{fft}, D_{fft}, N_{fft})
B_{Gains} ← Calculate_Gains(H, N_{fft}, S_f)
return B_{Gains}

It first computes the FFT of the sample signal, using **Algorithm 2**, directly from the wave file. The data is stored in a giant buffer (signal sampled at 48kHz, played for 4 seconds = 192,000 bytes). Then, the FFT is applied to the sampled data, to convert the time-domain captured signal into its frequency-domain representation. This allows the system to evaluate the magnitude response across the frequency range and identify deviations from the expected signal. Because FFT-based processing scales on the order of *O(N*log(N))*, it is computationally feasible to implement on an embedded processor while still achieving sufficient resolution for audio analysis. The ESP32-S3 also possesses a dedicated DSP library which includes optimization versions of each function specifically for the S3 version.

Additional processing steps in the FFT unit, such as windowing and smoothing, are applied to reduce spectral leakage and measurement noise, which can be seen in detail in **Algorithm 2**. The resulting frequency response is then analyzed to detect peaks, dips, and resonant behavior caused by room acoustics, speaker placement, and enclosure effects. These features directly inform the equalization adjustments applied to the system, allowing it to compensate for environmental coloration and improve perceived tonal balance.

Algorithm 2 Compute FFT

Requires: Sampled Data D of length N_s, Frequency S_f
Ensure: FFT of data D_{fft} of size N_{fft}
Allocate complex input buffers y_{cf}
Allocate window coefficients buffer W
Allocate accumulator buffer y_{acc}
W ←Generate_window_coefficients(N_{fft})
h ← NUM_BINS ▶ Hop size between frames
k ← $\lfloor (N - M) / h \rfloor$ ▶ Number of overlapping frames
for i=0 to K-1 **do**
  for j=0 to N_{fft} **do**
    $x_j \leftarrow (D_{ih+j} - 32768) / 32768$
    $y_{2j} \leftarrow x_j \cdot W_j$
    $y_{2j+1} \leftarrow 0$
  end for
  y_{cf} ← FFT(y_{cf}, N_{fft})
  y_{cf} ← BitReverse(y_{cf}, N_{fft})
  for k=0 to $2N_{fft} - 1$ **do**

$y_{acc\ 2k} \leftarrow y_{acc\ 2k} + y_{cf\ 2k}$
$y_{acc\ 2k+1} \leftarrow y_{acc\ 2k+1} + y_{cf\ 2k+1}$
end for
end for
if k > 0 **then**
for m=0 to $2N_{fft} - 1$ **do**
$y_{acc\ k} \leftarrow y_{acc\ k}$ / k ► Average over all frames
end for
end if
return y_{acc}

Once both FFTs are found, the algorithm computes the transfer function which compares the played signal (wave file) and the captured signal (from the microphone). This is done through a Wiener Deconvolution Filter, which not only computes the transfer function, but also aims to reduce noise from both the analog and digital sides. The Wiener Filter is implemented as

$$\boldsymbol{H} = \frac{\boldsymbol{X^T Y}}{\boldsymbol{X^T X} + \boldsymbol{\delta}}$$

where $\boldsymbol{\delta}$ is the regularization factor used to avoid division by zero. This algorithm is mostly mathematically implemented as shown in Algorithm 3.

Algorithm 3 Wiener Deconvolution Filter

Requires: FFT array of data D, wave file W, FFT size N_{fft}
Ensure: Transfer function H
$\delta \leftarrow 0.01$
if $X = NULL$ **or** $Y = NULL$ **then return** $NULL$
Allocate TF buffer H
for i=0 **to** $2 \cdot N_{fft}$ **do**
$x_{mag} = D_i \cdot D_i + D_{i+1} \cdot D_{i+1}$
$d = x_{mag} + \delta$
$H_i = (W_i \cdot D_i + W_{i+1} \cdot D_{i+1}) / d$
$H_{i+1} = (W_{i+1} \cdot D_i - W_i \cdot D_{i+1}) / d$
end for
return H

This Transfer Function H now compares the captured signal with the test signal, which yields the first response of the integrated system. This H is now used to calculate the frequency band gains for the equalizer setting.

The filter implemented in this system is actually a series of IIR filters, one for each frequency band. The implementation follows a traditional 8-band Equalizer, with the following bins: 60, 125, 250, 500, 1000, 2000, 4000, 8000, and 16000. The function to implement the digital filters is already implemented in the ESP-DSP library, with optimized versions for the S3 chip. Once **Algorithm 1** finishes, the EQ coefficients are updated in real time and applied in the audio playback loop.

In automatic equalization mode, the process begins with playback of a known test signal used for microphone measurement. For this project, the test signal is a logarithmic sine sweep, sampled at 48 kHz, 16-bit wave file played through the speaker system. This signal provides a consistent sweep from 20 to 20,000 Hertz (normal hearing range for an adult human) and repeatable audio source for testing because its frequency and digital format are fixed. A wave file is characterized for representing a signal using its raw format, meaning we can send bytes to the DAC without having to decode or encode the data, making it much simpler than other formats. As the wave file is played, the microphone captures the resulting acoustic response at the listening position. The microphone pipeline ends at the input of the ADC unit, which converts the analog captured signal to a 16-bit unsigned value. The recorded signal reflects the combined behavior of the speaker system, amplifier chain, microphone path, and surrounding environment. By using a fixed sine sweep, the system can determine how the measured response differs from the expected playback and use that information during calibration and analysis. This provides a simple and straightforward method for verifying whether the audio chain is reproducing the signal correctly and whether adjustment may be needed.

The ESP32-S3 then performs embedded signal processing on the recorded microphone data, as described previously in **Algorithm 1**. This includes signal acquisition, sampling, and interpretation of the measured waveform so that the system can evaluate how accurately the output matches the intended test signal.

However, the system is not restricted to this automatic method. In manual equalization mode, the user directly changes EQ settings through the mobile application. Rather than relying only on microphone measurements and automatic correction, the user selects preferred levels for different frequency ranges. This gives direct control over tonal balance and allows the system to remain useful even when Auto-EQ is unavailable, incomplete, or intentionally disabled.

Manual control is valuable for several reasons. First, listener preference is subjective, and some users may intentionally want enhanced bass or brighter treble rather than a neutral response. Second, manual EQ is easier to demonstrate and validate because it does not depend entirely on environmental measurement accuracy. Third, it gives the project a reliable operating mode even if the automatic calibration pipeline is still under development. In this sense, manual EQ is both a practical feature and a risk-reduction strategy.

The dual-mode processing concept also improves the overall project design. Instead of framing the system as successful only if full Auto-EQ is achieved, the project can be presented as an embedded audio platform that supports both measurement-based analysis and direct user-controlled equalization. This better reflects the actual engineering value of the system and ensures that it remains functional and useful under multiple operating conditions.

The overall system architecture can be understood as an interaction between sensing, processing, control, and output subsystems. The microphone and optional sensors provide environmental input, while the ESP32-S3 serves as the central processing unit responsible for signal analysis and control decisions. The mobile application functions as the user interface layer, issuing commands and receiving system feedback through wireless communication. Processed audio data is then routed through the amplification stages and delivered to the speaker system. This closed-loop structure enables both measurement-driven correction and user-directed adjustment within a unified platform.

From a physical perspective, peaks in the measured response often correspond to constructive interference or room resonances, while dips may result from destructive interference or absorption effects. By identifying these features, the system can apply targeted attenuation or gain adjustments to specific frequency regions. This interpretation bridges the gap between mathematical analysis and real-world acoustic behavior, reinforcing the practical relevance of the signal processing approach.

V. MOBILE APPLICATION AND COMMUNICATION

The mobile application is a major part of the project because it provides the main user interface to the sound system. Without the app, the equalization system would still exist as an embedded prototype, but it would be far less practical and less aligned with modern consumer expectations. The application gives the user direct access to both system operation and sound customization.

One of the most important roles of the app is enabling manual adjustment. Through the app, the user can modify frequency-band settings themselves rather than depending entirely on automated processing. This means the app is not just a start button for Auto-EQ; it is also a direct sound-control interface. Users can shape the system output according to preference, listening material, or room conditions.

The communication link between the app and the embedded hardware is based on the ESP32-S3's wireless capabilities, with Bluetooth emphasized as the main phone-to-device control path. This communication design supports calibration initiation, EQ updates, connection status, and manual control changes. The embedded system receives these commands and applies them to playback behavior.

From a software-architecture perspective, the app serves as a companion control interface rather than the location of the primary DSP workload. The user may choose whether to run Auto-EQ or manually adjust settings. This is an important design distinction. The embedded system remains responsible for processing and execution, while the app improves accessibility and usability.

The app also improves the project from a demonstration standpoint. Even if automatic equalization is only partially implemented, the user can still connect to the system, adjust sliders or bands, hear changes in the audio output, and observe that the embedded platform is responding correctly. This makes the system more robust as a senior design project because it guarantees meaningful interaction and visible functionality.

User experience is especially important here because one of the project's main goals is reducing the burden of tuning. For some users, that means automatic correction. For others, it means having a simple and convenient app that makes manual adjustment much easier than using traditional external equalizers or complicated calibration software. The system supports both interpretations.

VI. PROTOTYPE TESTING AND EVALUATION

Prototype testing focused on reducing system risk through staged validation of the most important subsystems. These tests do not need to prove a fully finalized consumer-ready product to demonstrate engineering progress. Instead, they show that the project's core hardware and software directions are feasible.

Microphone testing was one of the most critical validation efforts because the success of Auto-EQ depends on accurate acoustic measurement. The microphone chain must provide a clean enough signal for the ESP32-S3 to analyze meaningfully. This required testing the ADC behavior, the microphone hardware, the conditioning circuitry, and the combined signal path. These tests are directly tied to the automatic calibration goal.

At the same time, the prototype evaluation should also emphasize that manual EQ functionality does not depend on the same level of microphone-processing maturity. Even if the Auto-EQ algorithm is still being refined, the system can still operate as a user-adjustable equalization platform. This gives the project a strong and demonstrable baseline mode of operation.

From a user interaction perspective, the mobile application is designed to minimize complexity while still providing meaningful control over the system. The interface prioritizes clarity by grouping controls into logical sections, allowing users to quickly switch between automatic and manual modes without navigating through complex menus.

Responsiveness is also a key consideration. The system is designed so adjustments made in the application are reflected in the audio output with minimal delay. This real-time feedback is important for manual equalization, as it allows users to iteratively refine their preferred sound profile based on immediate auditory results. This interaction model aligns with common user expectations for modern audio control systems.

Embedded prototype testing also addressed communication between the Flutter app and the ESP32. This was important because the app must reliably send commands, switch modes, and update EQ values. In the context of manual adjustment, this communication path becomes even more central, since real-time user control depends on the app and embedded system exchanging commands correctly.

Audio-related testing addressed the practicality of the analog output path and amplification design. Since the project depends on separate mid/high and subwoofer amplification sections, signal quality, power delivery, and distortion remain important under both automatic and manual operation. A system with user-adjustable EQ still requires clean playback and stable amplification if it is to be considered successful.

The prototype results therefore support two conclusions. First, the long-term Auto-EQ direction is technically meaningful and worth pursuing. Second, the system already has clear value as a manual user-controlled equalization platform. These two conclusions together make the project stronger, because they show that useful functionality exists even while the more advanced automation features continue to develop.

Additional testing focused on validating system responsiveness and usability under manual control conditions. In particular, the interaction between the mobile application and the embedded system was evaluated to ensure reliable communication and real-time adjustment capability. Changes made through the application interface resulted in immediate and perceptible differences in audio output, confirming that the control pathway between the user interface and the DSP processing layer was functioning as intended.

Audio testing also included qualitative evaluation of output characteristics across different equalization settings. Adjustments to low-frequency bands produced noticeable changes in bass response, while mid and high-frequency adjustments affected clarity and brightness. These observations demonstrate that the system can deliver meaningful and controllable sound modification even without relying entirely on automatic calibration.

Together, these results support the conclusion that the system functions as a completely embedded audio platform. Both automatic and manual modes contribute to overall system capability, with manual equalization providing a reliable and demonstrable baseline while automatic processing continues to be refined.

VII. SYSTEM LIMITATIONS

While the Smart Auto-Equalization Sound System demonstrates strong integration of hardware and software components, several limitations remain in the current implementation. One limitation is the dependence on accurate microphone measurements for Auto-EQ functionality. Environmental noise, microphone placement, and hardware imperfections can all influence the measured response and reduce calibration accuracy. An amplification range had to be introduced in order to prevent clipping the signal, and also to protect the hardware from any issues overamplification could cause.

Another limitation is the computational constraint imposed by the embedded platform. Although the ESP32-S3 can perform FFT-based analysis, more advanced signal processing techniques may be restricted by available processing power and memory. This limits the complexity of real-time filtering and adaptation that can be implemented in the current system.

Additionally, the system currently relies on predefined equalization structures rather than fully adaptive or continuously learning algorithms. While this approach ensures stability and predictability, it may not capture all dynamic variations in real-world listening environments. These limitations represent opportunities for future improvement and further development.

VIII. CONCLUSION

The Smart Auto-Equalization Sound System presents a practical embedded approach to speaker calibration and user-controlled sound customization. The project was motivated by a common audio problem: sound quality changes with the listening environment, yet correcting that behavior usually requires time, expertise, and additional equipment. By combining a 2.1 audio platform, a calibrated microphone, an ESP32-S3 processing core, custom analog amplification, and a mobile application, the system moves toward a more accessible audio-adjustment solution.

A major strength of the project is that it does not rely exclusively on fully automatic correction to remain useful.

The system supports both microphone-based Auto-EQ and direct manual equalizer adjustment through the mobile application. This means that even if the automatic algorithm is not completely finalized, the project still delivers meaningful functionality, interactive user control, and a complete embedded audio platform.

The project is valuable because it integrates multiple engineering disciplines into one prototype. The analog hardware must support clean amplification and reliable signal conditioning. The embedded system must perform control, communication, and potentially real-time DSP. The mobile application must provide an intuitive interface for both automatic and manual operation. Because these pieces must all work together, the project demonstrates both breadth and depth in electrical and computer engineering design.

Several design tradeoffs influenced the final system implementation. One key decision was the selection of Class D amplification over Class AB, balancing efficiency and thermal performance against potential distortion concerns. Another tradeoff involved the emphasis on both automatic and manual equalization modes. While Auto-EQ provides a technically guided correction based on environmental measurements, it introduces additional complexity in signal processing and system validation. Manual equalization, by contrast, offers immediate usability and reliability, ensuring that the system remains functional regardless of algorithm maturity. Additionally, performing DSP on an embedded microcontroller rather than an external processor reduces system cost and complexity, but places constraints on computational resources. These tradeoffs reflect a design approach that prioritizes practicality, integration, and demonstrable functionality.

Future work may continue refining the Auto-EQ algorithm, expanding testing across more listening conditions, and improving real-time response. At the same time, the manual EQ mode ensures that the system remains practical, demonstrable, and user-oriented throughout development. Overall, the project demonstrates that an embedded sound system with both automatic and user-adjustable equalization is technically feasible and appropriate as a senior design effort.

Future work may also explore the integration of more advanced DSP techniques, such as adaptive filtering algorithms and real-time feedback systems that continuously adjust equalization based on environmental changes. Additional improvements could include enhanced user interface features, expanded wireless communication options, and further optimization of the embedded processing pipeline. These developments would move the system closer to a fully autonomous and adaptive audio platform.

Acknowledgement

This project was conducted with the support of the Senior Design lab at the University of Central Florida. We wish to acknowledge the assistance and support from Dr. Weeks for his technical expertise and practical assistance in the design and implementation of our circuit boards. We would also like to acknowledge our mentors, Dr. Sreeram Sundaresh and Dr. Chung Yong Chan, for their constant guidance and administrative support. In addition, we'd like to recognize the time and attention given to our group and our project by our reviewers Professor Maddox, Dr. Das, and Dr. Fasihi

References

[1] Espressif Systems, *ESP32-S3 Series Datasheet*, Espressif Systems, 2024.