SPRING 2023

MAY 5, 2023

EE 41440

PROF. MIKE SCHAFER

UNIVERSITY OF NOTRE DAME

BLUETOOTH SPEAKER DESIGN REPORT

BEN COLFER
MIA GORMAN
CALLIE KING
DEIRDRE SHERIDAN
TJ SIMS

Troubleshooting

To Market Design Changes

Conclusions

Appendices

Introduction

The Problem

Portable speakers have become an integral part of every person's day-to-day life. They're used socially on the beach, on bike rides, and at picnics. They come in all different shapes and sizes. But they lack the sound quality that other non-portable speakers provide. They also lack the surround sound experience that makes music and other media forms so enjoyable. Imagine yourself at a party and you are near the speaker. You can barely hear the person you are talking to! On the other hand, if you are far from the speaker you might not be able to hear the music at all. There is demand for a "goldilocks" solution to this problem, such that you are able to enjoy the music throughout a party, regardless of your positioning within the space.

The problem that this project aims to solve is simple: portable speakers cannot truly fill larger spaces with high-quality audio. Often, when choosing the portable option for a speaker, the consumer has to sacrifice high-quality audio. Speakers are plagued with audio distortion, limited loudness, popping or humming sounds, and so enjoyment of the media is lessened. When having an outdoor party, having one portable speaker means that only a small subset of the audience gets to enjoy the music. Creating a portable multispeaker system with synchronized outputs can solve this problem.

Presently, there are speakers that can work together via a wifi connection and an app to synchronize outputs, but this means that the speakers are not portable. This creates problems when guests want to play their music, because it is a hassle to download a new app just to take over a speaker. It also limits the user to enjoying this experience in their home only, they cannot set up this system in others homes, out in nature, or to a new home.. Using solely a bluetooth connection can modernize speakers, and make this surround sound music experience accessible in new spaces. .

The Solution

The goal of the Bluetooth WiFi Speaker Hub team is to design a speaker configuration that provides high-quality sound on the go for its users. The speaker design will be composed of smaller speaker units that will be flexible in their orientation. They can function as one large speaker or be broken up and distributed to create a more surround sound experience and allow music to be synced at various parts of a larger space. These speakers are powered by rechargeable batteries, allowing the user flexibility in the location they want to utilize the product. Currently, there are two peripheral speakers along with one main hub. This number can increase if the user wants. All the user would need is an extra peripheral speaker and another phone to connect to it.

The original idea required only one bluetooth connection from one phone to the main hub. Throughout the design process, we learned we needed to shift our solution. Now, in order for the speakers to work in tandem with each other, you need at least as many friends as you have speakers. Each speaker requires a unique bluetooth connection, and it is even better if one extra person can control the group Spotify session on a phone not connected to the speaker configuration. The speakers connect to the phone as simply as current state of the art bluetooth speakers do. You press a button on the speaker and look for the speaker name on the bluetooth settings of your device. Then, the group Spotify session will sync the audios and ensure that the speakers are playing the same songs. One person will control the group to ensure that the speakers stay in sync. This feature is also good because all members of the group are able to add songs to the queue and control which songs play next.

Meets Expectations

Our project met most of the requirements we had set for our projects. We learned, through much trial and error, that what we wanted to accomplish was not attainable for the scope of this class. The speaker system is portable. It runs off of rechargeable batteries, so there is no necessity to be close to a wall outlet. The speaker also uses bluetooth connection instead of a wifi connection so there is no need for a stable wifi router. This was integral for our requirements of design and for the speaker's value proposition as a more portable version of home speaker systems. We wanted this to be a truly portable speaker, with the only necessity being a bluetooth connection. The speaker system is able to provide surround sound if wanted. The peripheral speakers can be placed throughout a room to disperse the audio in the space.

We met most of the requirements with respect to the bluetooth connection. Originally, we wanted only one person to connect to the main hub, which would then communicate with the other speakers. Unfortunately, with the current bluetooth protocol, this was not possible. We were able to think of another solution that allowed us to utilize solely bluetooth connection. The connection is stable between the phones and the speakers in our new design. The speakers remember who was last connected and will look for that device if nearby. It will even ask the device if it will reconnect.

Our last requirement of our design was that it would be safe to use. The safety aspect was integral to the charging of the speakers. We needed to ensure that the correct amount of voltage was where it needed to be and that the batteries would not overcharge. We ensured this by having different transformers on the power management board to boost the voltage coming off of the hub batteries so that it would be large enough to charge the peripheral batteries. The power management board also cuts off the charging of the hub and peripheral batteries once they are fully charged. We checked the amount of voltage coming from the batteries to ensure that the appropriate amount of power is coming from each of them, which satisfies our safety requirement.

1 Detailed System Requirements

To accomplish our goal, our speaker first and foremost must produce high quality audio. Therefore, the speaker components like the mid woofers, tweeters, and the subwoofer, must be high quality but also within our budget. It is also required that the audio can reach a certain volume level that is loud enough. For the sake of this project, we required that the speaker must be at least loud enough to fill the space of Stinson 205 with music. We also require that there is limited noise (popping, static, non-musical output) when we play our audio. Because we imagine this speaker would be used for loud music for parties, we decided to require that there was no noticeable noise to the human ear when music is playing via the speakers.

Another important subset of requirements surrounds bluetooth connectivity. With the advancements in the audio industry throughout recent years, bluetooth is now a must-have feature for speakers. The ESP32 must have a Bluetooth interface that can connect to multiple peripheral speakers to play calibrated outputs. One important aspect to this requirement is for there to be a bluetooth connection from a phone to each speaker. This allows users to easily stream music or audio from their device. The speaker must connect to any mobile device with bluetooth, such as an iphone, laptop, or android cell phone. It must connect via the bluetooth section in the settings app, so it is necessary that the name of the speaker is shown to the user when the speaker is on and the user's device's bluetooth is on. Additionally, it is also important to have a stable bluetooth connection between each device and its speaker. This ensures that the audio is delivered seamlessly without interruptions or dropouts, as well as ensuring that it will play synchronously with the hub.

Portability is also majorly important to the proposed value of our speaker configuration. Portability is the major selling point that makes our speakers unique to similar existing systems. The intent of this speaker is that it can be used in any dorm, apartment, house, park, beach, or anywhere the user desires to use it. To be considered portable, the speaker should not have to be connected to an outlet to be producing music. This means that we require rechargeable batteries. It also needs to have certain physical characteristics that encourage portability, such as being relatively light and compact. A common portable speaker for outdoor parties is a Block Rocker Plus and this speaker is 9.85 x 17.33 x 14.57 inches and 20 lbs, so we used this as our guidelines for how large our speaker should be without being too big to be considered portable.

Another crucial aspect of the value proposition of our speaker configuration is the fact that the peripheral speakers should be removable from the hub to set up a surround sound experience. This greatly enhances the user experience from the current standard. For the best experience with surround sound, it was important for our project that the peripheral speakers and the hub could be separated and dispersed throughout a space. This way, the user can then optimize the sound and listening experience based on how they are using the speakers as well as the layout of the space they are using them in. We required that the music from the speakers be synchronized wirelessly, so that the peripheral devices can be placed wherever for the best experience.

Part of the challenge of designing this system was powering the system. We required that the system should be able to function while not plugged into wall outlets, so we required energy storage within the system. We require that the battery should hold a charge for a minimum of 5 hours, because this would more than account for the time of an average party. We also require that the time to charge these batteries should not exceed 12 hours, because this would be more than charging the system overnight. Finally, we require that the hub should act as an energy provider to the two peripheral speakers and recharge them when they are resting on top of the hub.

We require that the components of the speaker be protected from elements by an encasing. The encasing should not hinder the sound quality of the drivers, and should be sturdy enough to protect the speaker from general wear and tear.

An important constraint of this project is that the materials involved in the production of this prototype should not exceed $500. This will largely factor into the drivers, amplifier boards and batteries we select to purchase.

Last but certainly not least, we require that the speaker is safe. The power system needs to be safe for users, i.e. it must have the appropriate grounding and safety features in place to protect the users from electrical hazards. It was critical that we design a system with no exposed wiring or sharp edges that could cause injury. It is also important to have a power management system that ensures the batteries are not overcharged and not at risk of exploding. The components of our circuit that are prone to overheating and causing fire should be regulated to avoid this risk.

2 Detailed project description

2.1 System theory of operation

Our system connects three speakers together so that anyone can connect to them and play synced music. It all starts by a cell phone connecting to an ESP32 chip in our main hub via bluetooth. This hub is connected to two other peripheral speakers via bluetooth. In our hub the digital signal picked up from the ESP32 is sent to a bluetooth chip as well as to a surface mount digital to analog converter. This surface mount DAC outputs an analog signal which is then sent into an amplifier. The amplifier has low pass filtering so that it outputs only bass signals. From the amplifier we have a subwoofer connected which produces the bass sounds that were sent from the amplifier.

The digital signals that were also sent to the bluetooth chip within the hub are then sent to the bluetooth chips on the peripherals. The bluetooth chips on the peripherals receive the signal and output the digital data to surface mount DACs. These DACs then output the analog data via a left and right channel which is fed into an amplifier. The amplifier then sends the left channel data to one driver and the right channel data to the other driver. This happens within each peripheral as there are two drivers in each.

A main selling point of this speaker is the fact that it is portable. It is not like Sonos where it needs to have its own wifi mesh network, this speaker set can be brought anywhere and can even be used outside. For this to be possible the speaker set has to be battery powered. For our amplifiers to have enough power to be effective we settled on 12 volt batteries for both the hub and the peripherals. We wanted our system to be as safe as possible and through research we learned that lead acid batteries are the least dangerous as well as relatively less expensive compared to something like a lithium polymer battery.

All three speakers have a power management system which protects against overcharging through an LM317 voltage regulator, as well as handles up converting or down converting voltage. To charge the hub board we have a 24 volt AC to DC power supply that plugs into any wall outlet. These 24 volts are sent to the power management system which down converts the voltage to 15 volts. From here the battery is able to charge as well as power the rest of the hub with 12 volts. The amplifier takes in 12 volts and the main chip board has a voltage regulator (LM 117) which regulates the 12 volts down to 3.3 volts which is suitable for the chip. The Hub is the only part of the system that has to be plugged into the wall for charging. We have implemented electromagnetic coils at the top of the hub as well as on the bottom of the peripherals so that they touch charge off of the hub. The 12 volts from the hub battery also go towards these electromagnetic touch charging coils. These only output 5 volts so then in the power management system of the peripherals there is an upconverter which takes the 5 volts and boosts it to 15 volts so the peripherals can charge.

2.2 System Block diagram

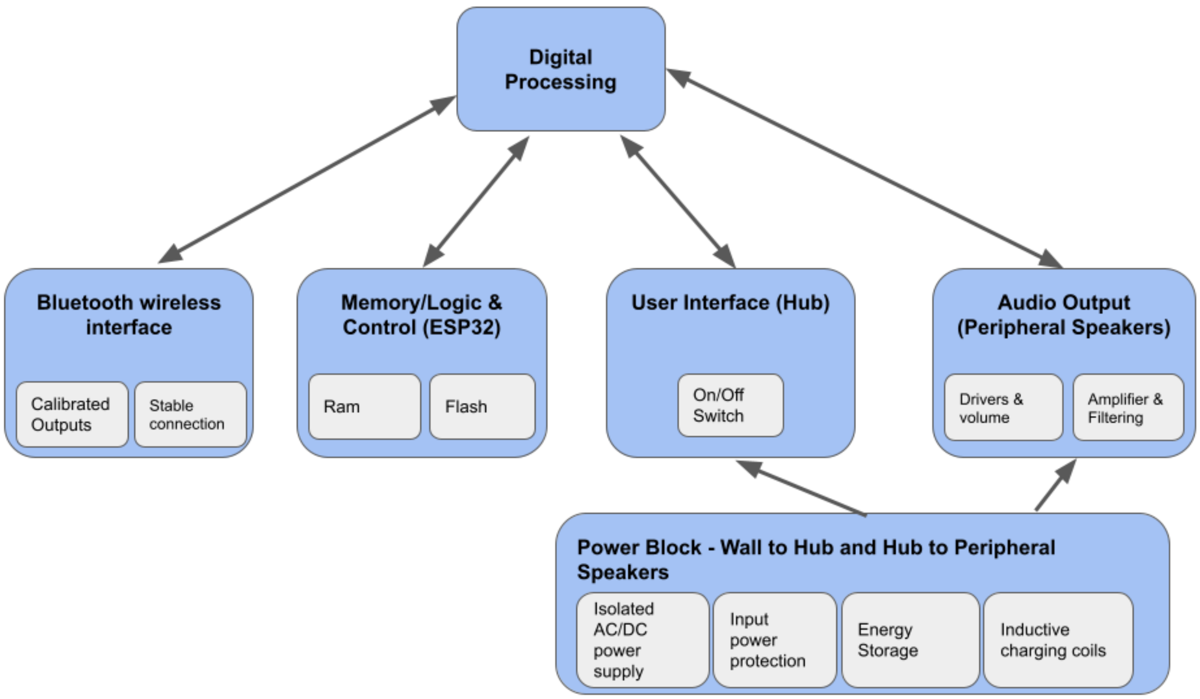


Figure 1: Overall system block diagram.

2.3 Detailed Design/Operation of Subsystem 1: Signal Processing System

The requirements of the signal processing system include the main bluetooth connection from phone to hub speaker as well as the unison sound sharing or passing of audio data between the hub and peripheral speakers. The subsystem entails being able to utilize bluetooth protocol to send audio data packets via the connected phone and the hub. Once the hub receives the data, it 1) enters subsystem two and begins the process of outputting audio and 2) sends the audio data to the peripheral speakers so that they can play it synchronously with the hub.

Another critical requirement for our audio system was for it to have bluetooth capabilities. With the advancements in the audio industry throughout recent years, bluetooth is now a must-have feature for speakers. One important aspect of this requirement is for there to be a stable bluetooth connection from a phone to the hub. This allows users to easily stream music or audio from their device. Additionally, it is also important to have a stable connection from the hub to the peripherals. This ensures that the audio is delivered seamlessly without interruptions or dropouts, as well as ensuring that it will play synchronously with the hub. Thus, our system requires connection stability. Unstable connections can cause audio interruptions, dropouts, or loss of connection. To avoid these issues, it is important to ensure that our bluetooth connection is stable, fast, and reliable.

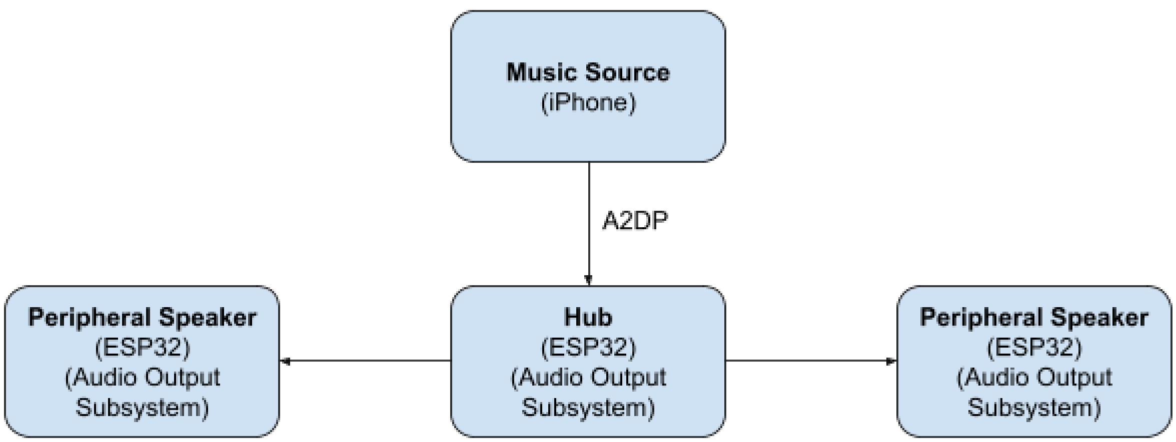


Figure 2: Diagram representing the signal processing system. This includes communication from a music source to our ESP32 hub via bluetooth. This also shows the ideal communication between the hub and the two peripheral speakers.

To go about creating this subsystem, there were multiple different ideas explored in depth. We started by getting the single connection going from the phone to the hub speaker system via bluetooth. This was achieved using the ESP32 A2DP protocol.

To connect a phone to the hub, the A2DP protocol is used. This allows for a bluetooth audio source (such as a phone or laptop) to connect a source to the speaker hub. There is an existing API for bluetooth A2DP. This is an audio streaming protocol that allows the wireless transmission of data packets from one device to another. This protocol uses a code to compress and decompress audio data which is then transmitted wirelessly via bluetooth. The codec's role is to take the raw audio data and compress it in such a way that it can be transmitted over the limited bandwidth available in bluetooth without losing too much quality. Once the compressed audio data is transmitted, it is received by the receiving device and decompressed by the codec. The uncompressed audio data is then played black through the speakers. This allows users to stream high quality audio from their phones to the speakers. The code we used is pictured below.

```
/*
  Streaming data from Bluetooth to internal DAC of ESP32

  Copyright (C) 2020 Phil Schatzmann
  This program is free software: you can redistribute it and/or modify
  it under the terms of the GNU General Public License as published by
  the Free Software Foundation, either version 3 of the License, or
  (at your option) any later version.
  This program is distributed in the hope that it will be useful,
  but WITHOUT ANY WARRANTY; without even the implied warranty of
  MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
  GNU General Public License for more details.
  You should have received a copy of the GNU General Public License
  along with this program.  If not, see <http://www.gnu.org/licenses/>.
*/

// ==> Example to use external 32 bit DAC - the 16 bit A2DP ouptut will be expanded to the indicated bits

#include "BluetoothA2DPSink.h"

BluetoothA2DPSink a2dp_sink;

void setup() {
  // you could also do this also with your own i2s config. But this is simpler
  a2dp_sink.set_bits_per_sample(32);
  a2dp_sink.start("BT32bit");
}


void loop() {
  delay(1000); // do nothing
}
```

Figure 3: Screenshot of our main code. We used the following github: https://github.com/pschatzmann/ESP32-A2DP.

Where it became more challenging to complete this subsystem was transmitting audio data from the hub to the peripheral speakers. We explored various ways of doing this including: ESP-NOW, using a RaspberryPi with snapcast, bluetooth low energy, Squeezelite, and creating a custom radio. While we were able to learn much about each of these options, unfortunately none of them ended up working successfully. Thus, we decided for demonstration purposes that it would be best to alter our original plan so that we could have a final product that worked as we originally intended it to.

The first thing we tried was using ESP-NOW. ESP-NOW is a communication protocol developed by Espressif. It is designed for low power and low latency applications that require a direct communication link between two or more ESP devices. This protocol works by creating a one-to-many or many-to-many connection between ESP devices. One device acts as a sender, while the other devices act as receivers. The source device can broadcast data packets to all of the sink devices. We were able to have two ESP32 communicate by sending a message using this protocol. However, we quickly discovered that the protocol was unable to send audio data packets due to the low bit depth and high latency of the communication. High quality audio requires a data rate of upwards of 128 kbps, while ESP-NOW maxes out at around 100 kbps under optimal conditions. This data rate would only decrease with the speakers separated and upon multiple speakers connecting to the link. Therefore this method was deemed inefficient for our high quality audio requirement.

Our next attempt was using Snapcast on a RaspberryPi that we then would interface with the ESP32. Snapcast is a multi-room client-server audio player, where all clients are synchronized with the server to play perfectly synced audio. In this protocol, all data that is fed into the server's audio input will be sent to the connected clients in a named pipe. The snapserver (our hub) reads the chunk from the pipe, and the chunk is encoded and tagged with the local time. Then, the chunk is sent via transmission control protocol connection to the snap clients (our peripheral speakers). TCP connections work by establishing a virtual circuit between two devices over which data is transmitted. Then, each snapclient does continuous time synchronization with the server, so the client is always aware of the local server time. The chunk is decoded and added to the clients' chunk-buffer before it is played out using advanced linux sound architecture at the appropriate time. ALSA is a software framework that provides an interface for handling audio devices. It is designed to provide a low-level hardware abstraction layer for audio devices. Unfortunately, Snapcast also proved unsuccessful in interfacing with the ESP32. This could be because the Raspberry Pi was using a USB connection while the ESP32 was using direct UART, although we would expect this not to be an issue. After much time spent troubleshooting, the interfacing issue could not be solved.

The next idea we had was to use bluetooth low energy. Like standard bluetooth, BLE uses RF technology that allows devices to communicate with each other without the need for cables or wires. It operates in the 2.4 GHz frequency band. However, it uses a different modulation scheme than normal bluetooth called gaussian frequency shift keying (GFSS) in order to reduce power consumption. Consequently, the data rate for BLE is limited compared to standard bluetooth and therefore it is not suitable for high quality audio applications that require a lot of bandwidth. Thus, BLE was not the right choice for our project.

Our next attempt at communicating between the hub and peripherals was using Squeezelite. This is a protocol made to run on ESP32s. It allows users to connect all major music providers and enjoy multi-room audio synchronization. The software offers a low power, high performing solution for audio streaming. However, the downside of this protocol is that it requires the use of WiFi. Since one of the main requirements for our project is portability/wirelessness, we were unable to use Squeezelite.

Since using different bluetooth and audio streaming protocols proved too difficult for our desired application, we decided to try avoiding the use of protocols altogether through the design of a custom radio system. We explored both a full analog radio as well as a digital radio. The analog radio would have included all necessary components of a standalone radio system.

The analog audio signal would be split to two paths, both with local oscillators generating a carrier signal of around 433 MHz within the experimental frequency band. The audio signals would combine with the carrier signals through on board mixers, where one of the signals would be sent to a 90 degree phase shifter. Both signals would then be sent through low pass filters to remove the original DC signals as well as decrease and LO and mixer leakage. With this, we would have the two I and Q signals in quadrature needed as inputs to an analog modulator chip. The modulator chip would then combine, modulate (using an amplitude modulation scheme), and send the output signal to an amplifier and antenna.

An antenna on the peripheral boards would receive the signal and pass it through a low noise amplifier and band-pass filter centered at our carry frequency. The signal would then be sent to a demodulator chip to be demodulated back to DC where it can then be sent directly to the peripheral amplifiers without the need of any digital processing. If we were to take a digital approach, much of the on board processes such as mixing and filtering could be done digitally and digital I and Q signals could be modulated directly. This method was appealing since it would not rely on any existing protocols, but the uncertainty around the method rendered it too risky. To design a stable and efficient radio system requires a large amount of testing and optimization, all the way to the size of the copper traces on our PCB. Without adequate time to do this testing and due to the large number and cost of the board components, we decided this approach would likely be unsuccessful and impractical.

With the deadline soon approaching and months of trial and error behind us, we turned to one final approach. This approach would combine aspects of all previous attempts by utilizing a separate, standalone bluetooth chip on each board to handle hub-to-peripheral communications. We settled on the TI CC2564MOD bluetooth chip as it was the most cost effective and promising chip. Unfortunately, the model of the chip with an integrated antenna was out of stock, so we needed to purchase a separate antenna. The idea was for a single bluetooth connection between the user and the hub ESP32 would provide an I2S signal via the I2S pins of the ESP32 to this bluetooth chip, which would then be programmed to link to both bluetooth chips on the board. With the bluetooth chips handling communication of the I2S signal between each speaker, the signal would be sent to external digital-to-analog converters before reaching the audio output system. By doing this, we would avoid the problem of the ESP32s being unable to send and receive simultaneously as well as avoid the uncertainty in designing a radio communication system in such a short period of time.

Unfortunately, this approach did not succeed after testing. We succeeded in providing the I2S signal to both the bluetooth chip on the hub board as well as the DAC, but we were unable to receive a clear signal on the peripheral boards. This could have been due to many things, but we guess that the error most likely occurred in the external antenna of the chip. Radio communication is sensitive and requires precision as was previously mentioned, so it is possible that the external antenna and trace were not accurate enough with the little testing and design we were able to do. Additionally, the TI bluetooth chip had very little documentation for use with microprocessors not also designed by TI like the ESP32. Therefore, errors could have also occurred in our programming and use of the bluetooth chips.

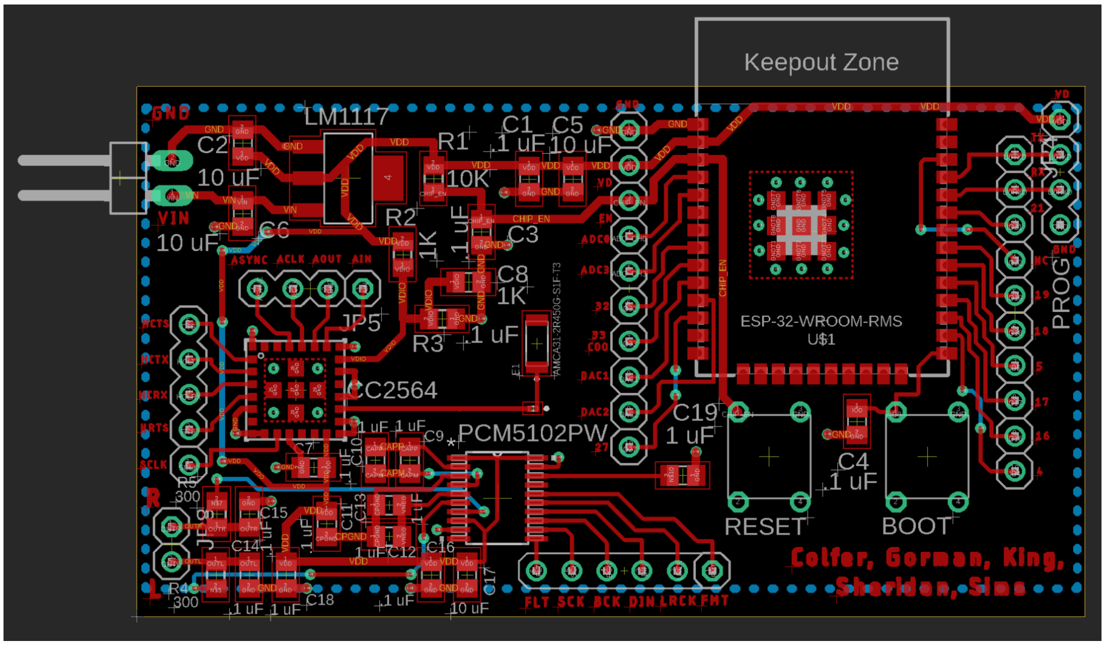


Figure 4: Board layout for our final design.

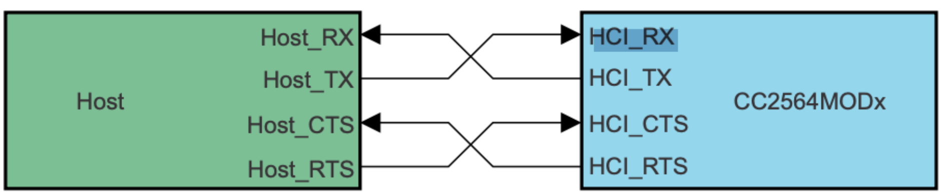


Figure 5: Diagram representing the connections between the ESP32 and the CC2564MODA bluetooth chip. We wired host pin 17 (U1RXD) to chip pin 2 (HCI_TX), host pin 18 (U1TXD) to chip pin 3 (HCI_RX), host pin 19 (URTS) to chip pin 4 (HCI_RTS), and host pin 20 (U1CTS) to chip pin 1 (HCI_CTS).

Nevertheless, all of these attempts did not result in our original intention working completely. Therefore, we decided to go with our backup plan. This plan included making use of a pre-existing Spotify protocol for group sessions. This protocol involves a single user initiating a group session and inviting others to join by sharing a link or scanning a code. Once in the session, all members can control playback and add or remove tracks to a shared queue. Each member's device sends playback information to the server, which synchronizes the playback across all devices to maintain a consistent listening experience. If a member leaves the group or loses their connection, the server automatically adjusts playback to ensure a seamless experience for the remaining members. So essentially, we had three people join a group Spotify session. Then, each would connect to their respectives speakers; one to the hub and one to each peripheral. This resulted in a final product that had the capability to stream audio data wirelessly while being completely portable.

We tested this subsystem as we went along. We could tell what was working by hooking this subsystem up with the audio output subsystem. When we finally could hear music playing synchronously, we determined that it was working to the best of our ability.

2.4 Detailed operation of subsystem 2: Audio Output System

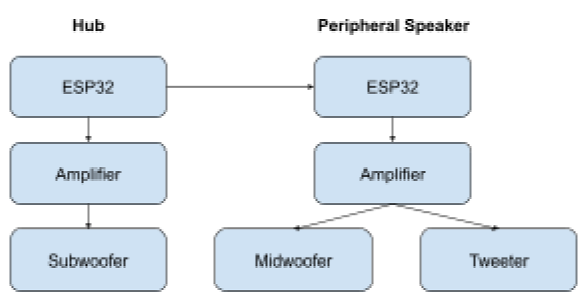


Figure 6: Diagram representing the audio output system. This image describes the major components in the hub: the ESP32, the amplifier, and the subwoofer for low frequency

components of the audio input. On the right handside are the components for one of the peripheral speakers: the ESP32, the amplifier, and the mid woofer and tweeter for higher frequency components of the audio input.

For high quality sound, we used high quality speaker components that fit our budget. The hub contained a subwoofer driver for low frequency audio. The analog signal from the ESP32 was sent to an audio amplifier board that contained on board low pass filtering to remove the unnecessary high frequency content before reaching the subwoofer. With the signal amplified and filtered, the signals were sent to the terminals of our subwoofer for high quality bass audio.

In the peripherals, the analog filter was sent to a two channel amplifier board. These amplifier boards split the signal into left and right channels after amplification, each of which were sent to mid-woofers and tweeters for mid and high range frequencies. Although these amplifiers had on board noise filtering, there was no high pass filtering, so unnecessary low frequency content was being sent to the high frequency drivers. However, audio quality was still perceptually high and peripheral filtering was deemed a task for future iterations. These components were all that was necessary for the analog signal output by the DAC to be output as high quality audio.

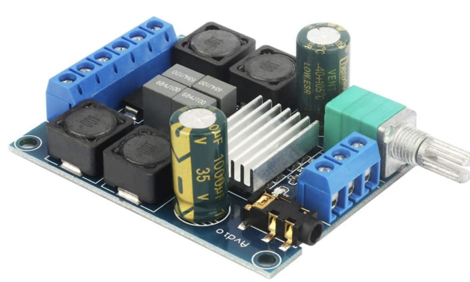


Figure 7: Peripheral amplifier board.

With the audio output electronics settled, the speakers required a housing or cabinet, not just for presentation but for audio quality as well. We decided to create wooden speaker boxes, rather than plastic. There are many advantages to using wood. First, it is naturally non-resonant, so there is minimal distortion. It is also naturally acoustic, which makes the sound very clear and pleasant to the listener's ear.

We chose a box shape for two reasons. First, with the limited resources available to us, it was the easiest type of enclosure to make out of wood. Second, the box shape produces the most accurate, detailed sound due to its lack of curves. We chose to create a port in our hub speaker box. The port allows for the pressure inside and outside of the speaker to equalize. This helps create a big bass sound that we were looking for in our design. To construct the boxes, we went to the EIH and cut our planned dimensions of the wood. We were then able to cut circles for the drivers to fit into with the help of Professor Schafer. We then screwed all of the pieces together and organized the hardware inside the boxes. Then we sanded the completed boxes down so that the wood would be smooth. Finally, we added some coats of spray paint to make the boxes more aesthetically pleasing.

2.5 Detailed operation of subsystem 3: Power System

The requirements of the power subsystem include energy storage, being able to hold charge for a minimum of 5 hours, no charge time longer than 12 hours, having the hub act as the energy provider for the two peripherals, and it has to be safe.

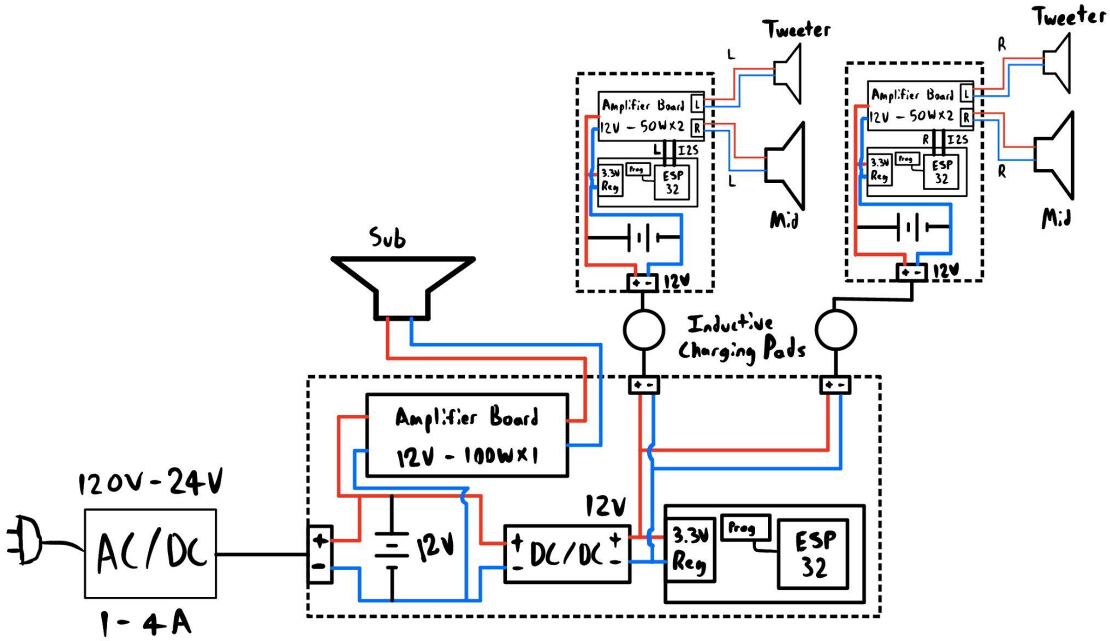


Figure 8: General power connections diagram.

We have an AC/DC power supply which takes the 120 volts AC down to 24 volts DC with 1 to 4 amps of current which goes into the hub. The 24 volts goes into the power management system of the first battery which converts the 24 volts down to 15 volts as well as regulates for overcharging. The 12 volt rechargeable battery powers the whole system when it is not plugged into the wall. The 12 volts power the amplifier within the hub as well as the main chip board. The main chip board has a surface mount regulator (LM117) which takes the 12 volts down to 3.3 volts. The 12 volts is also connected to an inductive charging set which will charge the batteries within the peripheral speakers when they are placed on top of the hub. The batteries within the peripheral speakers are also 12 volts and rechargeable. The inductive charging set outputs 5 volts so the power management boards in the peripherals boost the 5 volts back to 15 volts so that the peripheral batteries can charge. The 12 volts from the peripheral batteries powers the 2 output amplifiers for the midrange and tweeter drivers. The 12 volts also goes into the ESP boards which have a voltage regulator down to 3.3 volts.

In this power subsystem, we made use of inductive charging coils and lead acid batteries. Inductive charging is a way of powering a device without a direct wire connection. These chargers work by taking a power transformer and splitting it in half, an AC waveform is generated into one, and couples into the second coil. The coils need to be fairly co-axial, we will try to get them to be parallel and have the circles line up for the best power-transfer. For the batteries, we used a 12 volt rechargeable battery for the hub that will charge from a wall outlet. We did this by having two 6 volt batteries in series. This was a more budget-friendly option for our project. When the hub is plugged in it runs off of the outlet while the battery charges. The peripheral speakers will run off 12 volt rechargeable batteries. We used two 6 volt batteries in series for each of the peripherals. The 12 volt batteries recharge by inductive charging using the hub batteries as the charging source.

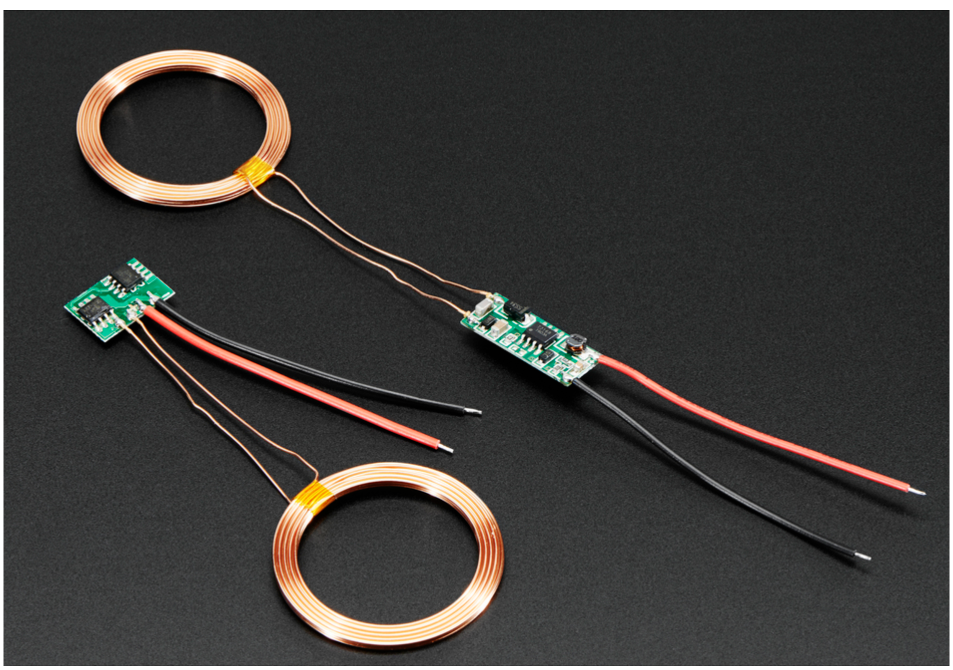


Figure 9: Inductive charging coils.

We require power to drive the speakers. This includes an isolated AC/DC power supply, input power protection, energy storage, and non isolated DC/DC power supply. This will ensure that the speaker hub can carry and store energy to charge the peripheral speakers. The hub will be charged through wall outlets. The peripheral speakers will charge from the main battery source at the hub through inductive charging. This way, when the peripheral speakers are back on the hub, they will be charging. We selected a 12V battery with 10 Ah so that the peripheral speakers will have approximately a ten hour battery life (calculations below).

Battery Life (in hours) = Battery Capacity (in Ah) / Load Current (in A)

$$BL = 5 / 0.417$$

$$BL = 12 \text{ hrs}$$

To be safe, the battery should be recharged after reaching 9.6 hours of use. This accounts for leaving twenty percent of battery life in the system so that it can recharge as expected. There will be an on board regulator to 3.3V for on board components, because of the ESP32. To power the hub, there will be an AC/DC adapter into the hub (120V AC to 24V DC).

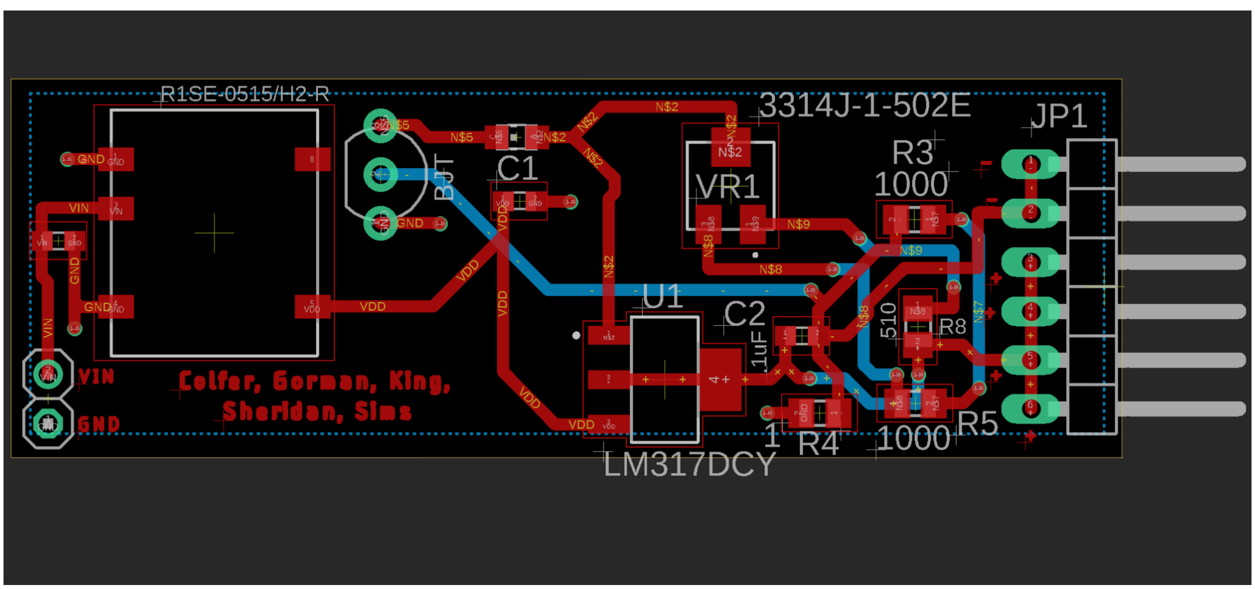


Figure 10: Peripheral speaker power management board.

The main components used on the peripheral power management system were the R1SE - 0515/H2 R which boosted the voltage from the inductive charging set of 5 volts to the 15 volts suitable for charging the peripheral batteries. This particular part was chosen because it was in stock, cost effective, as well as surface mount. Another major component was the LM317, this is a voltage regulator which would cut off the current coming from the inductive chargers when the peripheral batteries reached full charge. Finally we also implemented a potentiometer so that we could control the output of the power management system and make sure it was exactly 15 volts.

Figure 11: Image of R1SE - 0515/H2 R up converter.

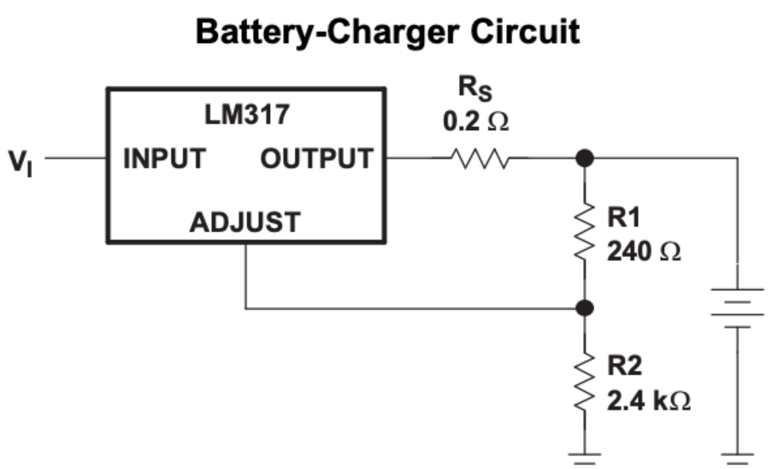


Figure 12: Image of battery charger circuit with LM317.

Figure 13: Image of potentiometer.

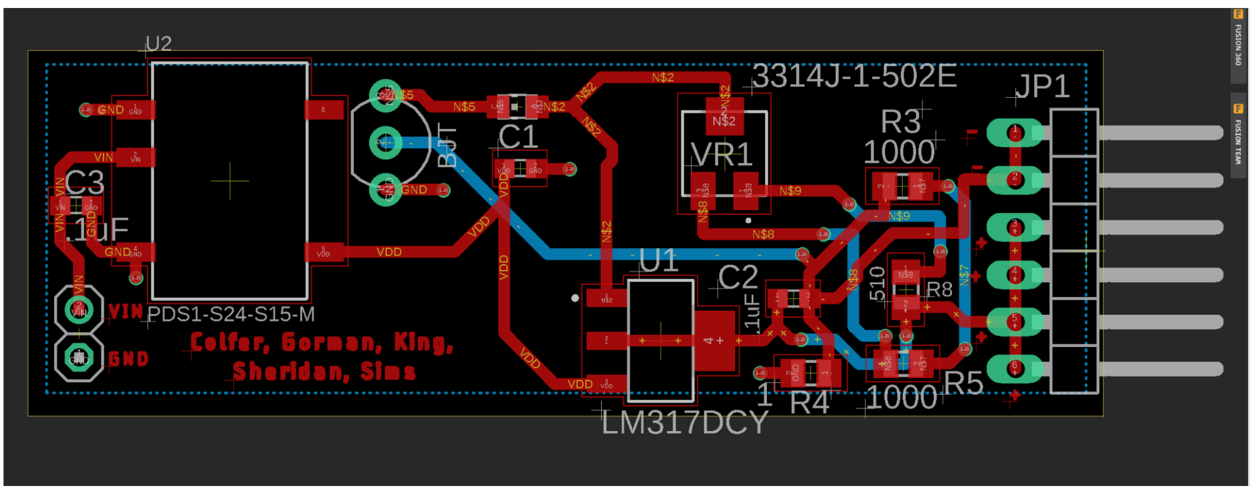


Figure 14: Hub speaker power management board.

The hub power management system is very similar to the peripheral power management system. It includes the same potentiometer as well as the same LM317. The difference is that the hub gets charge from a 24 volt AC/DC power supply, not from the 5 volt inductive chargers. This meant that we had to put the 24 volt to 15 volt down converter on the board instead of an up converter. We chose the PDS1-S24-S15-M because it was in stock, cost effective, and surface mount. These boards were not as complex as our main signal processing boards and for that reason all parts were hand soldered on instead of using the pick and place within the EIH.

Figure 15: Image of PDS1-S24-S15-M down converter.

2.6 Interfaces

As discussed earlier, the speaker boxes were designed with specific criteria in mind. We wanted the speaker boxes to promote the best listening experience. We chose to use wood as our material and chose a vented box for our subwoofer. Figure 16 shows our design.

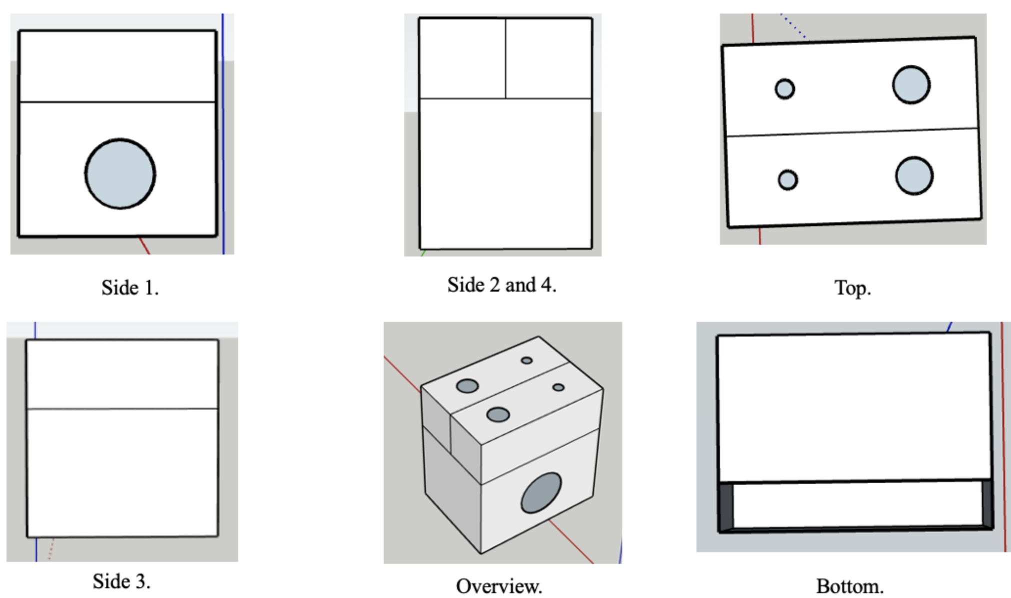


Figure 16: Speaker box CAD diagrams.

This setup allowed us to create holes for the drivers, as well as contain everything for the other subsystems inside of the boxes. The design was especially important for the power subsystem, as the electromagnetic coils for the inductive charging came out of both the hub and peripheral cases. This allows for the user to be able to place the peripheral boxes atop of the hub in order for them to charge wirelessly. Then once the peripherals are charged, they can be removed from the hub and placed wherever necessary.

3 System Integration Testing

3.1 Describe how the integrated set of subsystems was tested.

The integrated set of subsystems was tested in stages. The first step was to test each of the bluetooth boards individually. This was accomplished by programming our code onto the board and ensuring that a bluetooth connection could be established between our phones and the board. The first code we uploaded was intended to check only the functionality of the ESP32 programming on the board. We ran our code that established a bluetooth connection between our phones and the ESP32 and only interacted with the internal DAC of the ESP32. After that was determined to be working, we programmed the ESP32 to send the information received via bluetooth from the phone to the external DAC. We then hooked up the DAC to our amplifier boards which had previously worked with the class boards and found that the external DAC was not working. After troubleshooting, we learned that we had improperly soldered a mute pin. After fixing that, the DAC played music when connected to an amplifier and a driver. Because of our complications with the bluetooth chip as described above, we did many different tests and troubleshooting, we tried various programs but we think the problem stems from the facts that we are not using a Texas Instruments microprocessor and the RF trace and antenna were imprecise, so the communication between the bluetooth chip and the ESP32 made the bluetooth chip unusable for this application. We repeated testing these board components for all 3 of the bluetooth boards before we used our own power supply and power management board in combination with the boards.

The power management system and charging system was tested primarily using multimeters to see if we were getting the proper voltages and currents. To test that the inductive charging units were working we placed the coils on top of eachother and checked that the receiver was in fact outputting 5 volts. To test the peripheral power management system we made sure the voltage boosters were working by seeing if there was 15 volts running through the remainder of the board. We also used the potentiometer to make sure the output of the board was exactly 15 volts. To test the hub power management system we made sure the voltage down converters were working by making sure there was 15 volts running through the system and once again we used the potentiometer to make sure the output was exactly 15 volts. We tested the LM317 by seeing if it would allow current to flow through it when the power system was attached to a fully charged battery. The LM317 did not allow current to flow through which showed it was working.

Once the power and the bluetooth subsystems were tested, we made sure the boards were each programmed and had unique bluetooth names, so we knew which boards we were connected to successfully at any given time. We then wired up the Vdd and ground pins to the positive and negative terminals of the batteries to power the bluetooth boards, and then wired the amplifier board and driver to the internal DAC pins of the ESP32. Once power was connected, we hit the reset button on our board, and waited for the bluetooth name to pop up in the settings on our phones. When successful, this establishes that the bluetooth connection is working. If this were not the case, we would normally check the wirings first, and if the error was not found there, we would go back to stage one to ensure the proper program was uploaded to the board. Then once connected, we selected a song to send to the ESP32. If the song successfully plays through the driver, then we know that the bluetooth connection is successfully passing along audio packets that are interpreted by the DAC and then amplified by the amplifier board to produce sound at the driver. The amplifier board and the driver were both successful when used with other boards, so if there was a problem, we would assume it was not a result of the amplifier board or the driver, we would check our wiring and double check the programming of the board.

Our final step was to have multiple of the speakers connected and playing music in sync. We tested this by learning how to use Spotify Remote Listening to sync up music on multiple different devices. First, we tested synchronization on two phones. Once we learned that it is best to have one person in control of the pausing, playing, and skipping functions to keep the group listening session intact, we added a third phone to the mix. To ensure that the music was synchronized, we placed each phone next to each other and monitored how many seconds into the song each phone was playing at. Once we synchronized the music on three phones, we had those three phones connect to their own ESP32 in either the hub or the peripheral speakers. Once again, if the bluetooth name didn't present itself on our phones, usually hitting the reset button would fix this. From there, we learned that if the music starts to get out of sync, restarting the current song on one phone normally fixes this issue.

For future iterations, additional testing could be done on the audio quality of our speaker system. Perceptually, the audio quality was adequate for our high quality audio requirement. Audio measurement hardware and software such as the APX Precision Audio measurement family could be used to perform sin tone frequency sweeps and measurements for various audio metrics such as noise floor and frequency response. These metrics can then be used to adjust filtering and cabinet design as necessary.

3.2 Show how the testing demonstrates that the overall system meets the design requirements

This testing demonstrates that the overall system meets design requirements for each of our major subsets. First, this testing proves that a bluetooth connection between a user's phone can be initiated and maintained while playing music. This was tested on each of our phones, which proves that anyone can connect to our speaker. Second, the measurement of different voltages and currents ensured that our power management systems were functioning as designed. And third, our own perception was used to determine that our audio output system was performing well enough for high quality audio.

4 Users Manual/Installation manual

4.1 How to install your product

- No installation required!

4.2 How to setup your product

1. Flip the switch to turn power on to all of the speakers (hub and peripherals),
2. Press the reset button on all of the speakers.
    a. Look on your phone's bluetooth settings and wait for the names of the speakers to pop up.
        i. InternalDAC, red tape, black tape
    b. Have three friends connect to the speakers
3. Have a fourth friend start a group session on Spotify and invite the connectees to join.
4. Once everyone has joined the Spotify group session, the friend can start to control the music on his/her phone.
5. Control the volume with the knobs located near the speakers.
6. Move around the speakers (or keep them as a singular unit!) until you reach your desired listening placement.
7. To charge, simply put the speaker system back together and plug in the hub with the cord provided.

4.3 How the user can tell if the product is working

- If the user is able to connect to the speaker via the bluetooth settings on their phone and it displays “connected,” there is no problem with the bluetooth connection.

4.4 How the user can troubleshoot the product

- If the user does not see the name of the speaker in their settings app, they should press the reset button again and attempt to reconnect,
- If a user is booted off the connection, they should press the reset button and attempt to reconnect.

5 To-Market Design Changes

If going to market with this product, there are multiple aspects we would consider changing before selling our speaker system. One important thing to improve would be the quality of the speaker cabinets. This would include making them slightly smaller to make them weigh less. We would also have to improve upon the quality of craftsmanship. Due to our lack of resources with respect to woodwork, we were unable to create speaker boxes of the quality we would want. Another upgrade with the box would be to create holes so that the power switch and volume controls can be accessed from outside of the box, instead of being on the inside as we have now.

Another important upgrade to our system can come in waiting for new bluetooth protocols to be released. If the next generation of bluetooth protocols allow for only one connection and communication between speakers, our original goal would be met.

In the current design, we have two 6-volt batteries connected in series to create a 12-volt battery. We would want to cut out that extra step and buy 12-volt batteries to place inside the commercial product. This would streamline the production process. We would also like to implement an LED interface that shows when the battery level is high, mid, or low. Another facet of this interface would be an indicator that the batteries are charging while it is plugged in.

6 Conclusions

Our speaker ultimately meets the requirements we determined were vital to the success of our senior design project. Our speaker system is able to play synchronized music in multiple truly portable speakers. We found that there were many flaws with our original design, and designs we tried throughout the semester, but we were able to end with a final project that does what we wanted it to. With the addition of a more modern bluetooth protocol, this project may be able to work in its original design. Overall, this project demonstrates what we have learned throughout our time as electrical engineering students. We designed multiple boards that were successful in implementation. We also demonstrated problem solving skills as we ran into many problems throughout the semester.

7 Appendices

7.1 Datasheet Links

- Subwoofer:

  https://www.visaton.de/sites/default/files/dd_product/w130x_2x4.pdf

- Midwoofer:

  https://www.parts-express.com/pedocs/specs/292-640--visaton-frs-5x-8-spec-sheet.pdf

- Tweeter:

  https://www.parts-express.com/HiVi-B1S-1-Shielded-Aluminum-Mid-Tweeter-297-411?quantity=1

- Peripheral Amplifiers:

  https://www.amazon.com/TPA3116D2-Amplifier-Subwoofer-Suitable-Audio%E3%80%81car/dp/B08NFCCGC2/ref=sr_1_1?crid=15QS2A0MG011N&keywords=class+d+amplifier+board&qid=1677102618&sprefix=class+d+amplifier+boar%2Caps%2C96&sr=8-1

- Hub Amplifier:

  https://www.amazon.com/TPA3116D2-Amplifier-Subwoofer-Cancelling-Adjustment/dp/B081LXML5F/ref=sr_1_20?crid=2M0AY4AG43KIN&keywords=2+output+50+watt+class+d+amplifier&qid=1676587219&sprefix=2+output+50+watt+class+d+amplifier%2Caps%2C85&sr=8-20

- Inductive Charging Coils:

  https://www.adafruit.com/product/1407#description

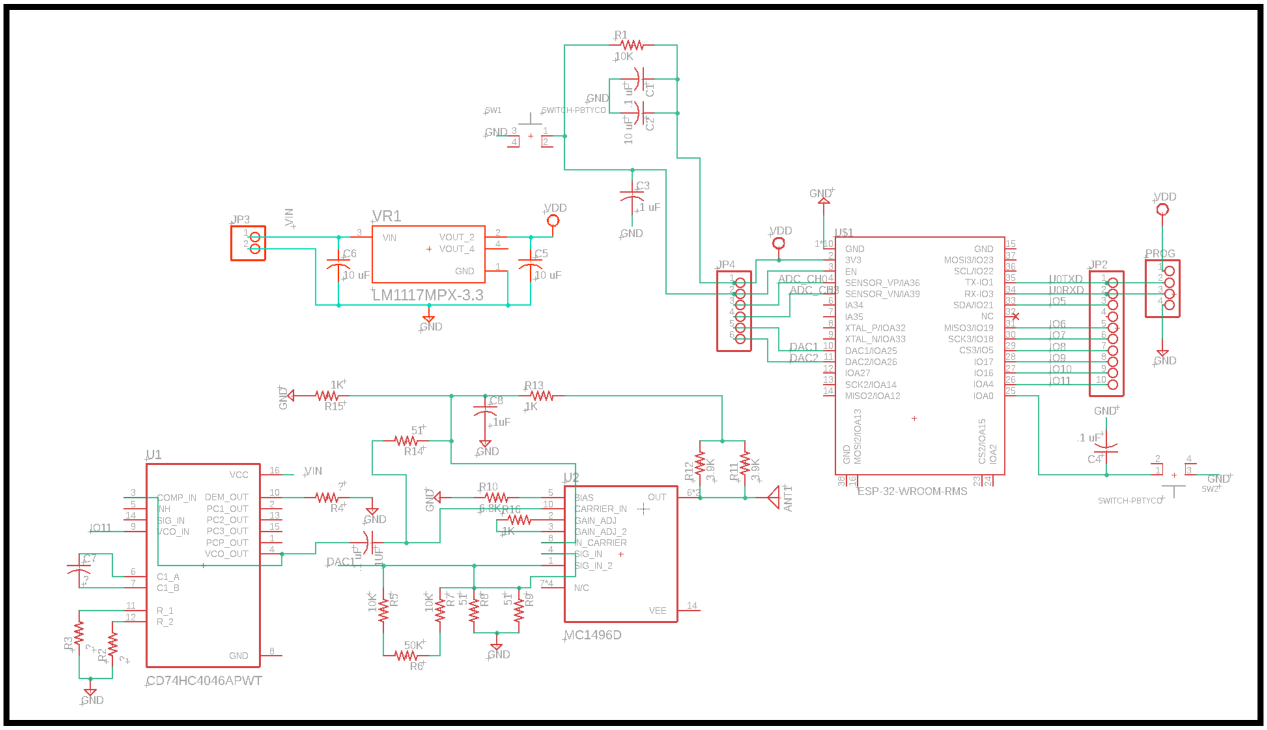
- Batteries:

  https://www.digikey.com/en/products/detail/zeus-battery-products/PC4-5-6F1/9828831

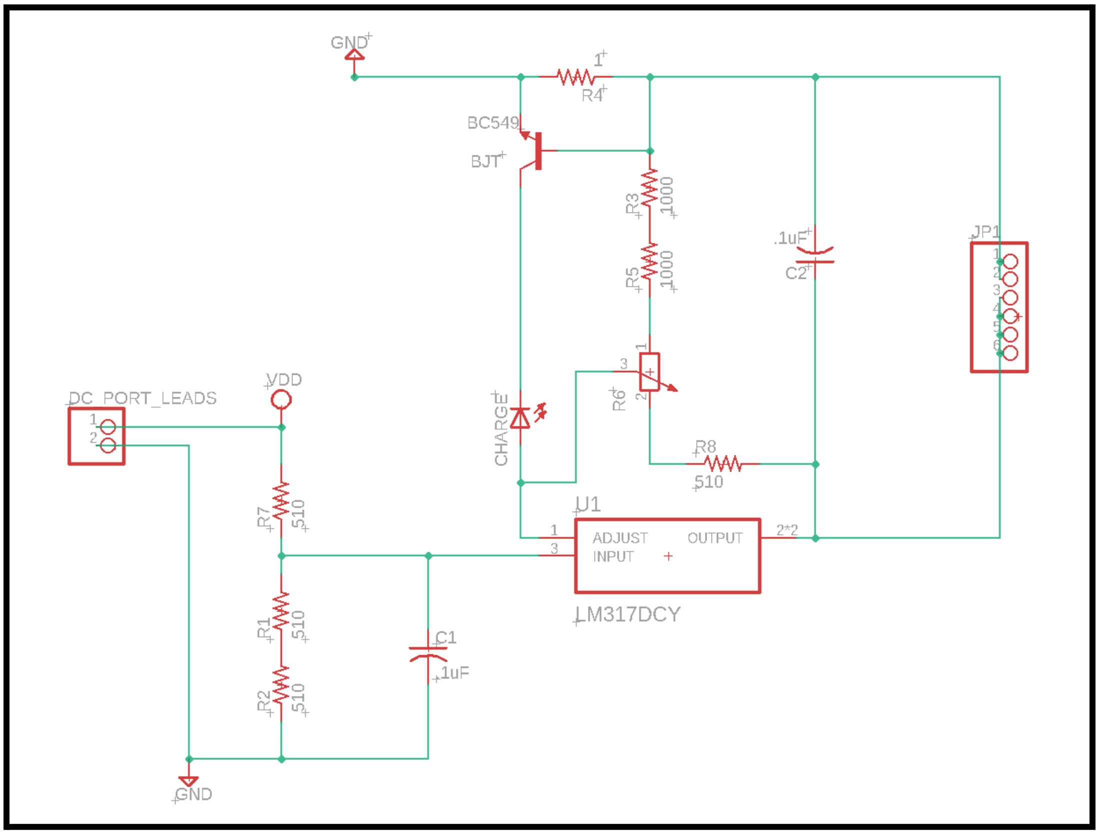
7.2 Code

- A2DP: https://github.com/pschatzmann/ESP32-A2DP
- ESP-NOW: https://github.com/espressif/esp-now
- Snapcast: https://github.com/badaix/snapcast
- Squeezelite: https://github.com/ralph-irving/squeezelite

7.3 Hardware Schematics



Main board schematic



Power management board schematic