PORTABLE BLUETOOTH AMP FOR HOME SPEAKERS

By

Nicholas Jew

Austin Palanca

Anthony Pham

Final Report for ECE 445, Senior Design, Spring 2018

TA: Zhen Qin

02 May 2018

Project No. 53

Abstract

This document explains in-detail the design and technical verification of the Portable Bluetooth Amp for Home Speakers. The final design is capable of outputting 20 watts continuously to external speakers with a Total Harmonic Distortion of less than 1% while being powered by an internal battery for over 3 hours. It supports power and battery charging with an external DC input. Digital audio can be amplified via a 3.5 mm auxiliary cable directly to the amplifier or streamed to the CC2564MODA Bluetooth chip and pipelined through the TLV320AIC3109 digital-to-analog converter then to the amplifier.

Table of Contents

1. Introduction

1.1 Background

Bluetooth speakers are increasingly more common as people find convenience in a wireless and portable speaker [1]. However, the market lacks more powerful and affordable Bluetooth speakers for those who need them for larger applications, such as theatrical and dance rehearsals. Large companies such as Bose and JBL have boombox style speakers, however their price point is over $300 [2][3]. Since speakers have become more common in households in the past couple of decades, we have created a portable amp with Bluetooth capability that can convert common household speakers into a Bluetooth speaker. As users can repurpose their speakers, they can purchase our device to use with their own speakers rather than be locked into a speaker and amp all in one that is provided by other companies. This gives the user more freedom at a lower price point. We expect the device to cost under $100 to be manufactured.

1.2 Functional Description

We define portable as something that can be easily moved, for example by carrying it in your hands or storing it in a backpack. Users of our product can unplug a household speaker and transport both the speaker and amp with ease.

Owners can connect to the amplifier with any Bluetooth-capable device or a 3.5mm audio jack. Since the device has its own battery, the user can play music anywhere. A user can plug in an external power adapter to both charge the battery and use the device without draining the battery.

We define our product with the following specifications:

- The device can output at least 20 watts continuous for an 8-ohm speaker. *
- The device can operate for at least 3 hours using the battery with the amp outputting 20 watts continuously. *
- The device is small enough to be carried in a backpack or in one's hand (more details under physical design). *
- The device supports Bluetooth audio streaming with a stable signal at ranges up to 20 feet.
- Total harmonic distortion of less than 1% while outputting a 20-watt continuous signal.

** Denotes unchanged high-level requirements from the original design*

1.3 Block Diagram

The final design consists of three blocks, differentiated by the colored modules in Figure 1. There is the Power Unit (PU), which handles the charging of the battery and supply of the necessary voltages for all the other modules. The Digital Logic Unit (DLU) handles the initialization of the device's chips, Bluetooth event callbacks, and volume control. Finally, the Audio Output Unit (AOU) accepts the audio signals from the Bluetooth module or an external AUX input, amplifies the signal, and outputs sound to an external speaker.

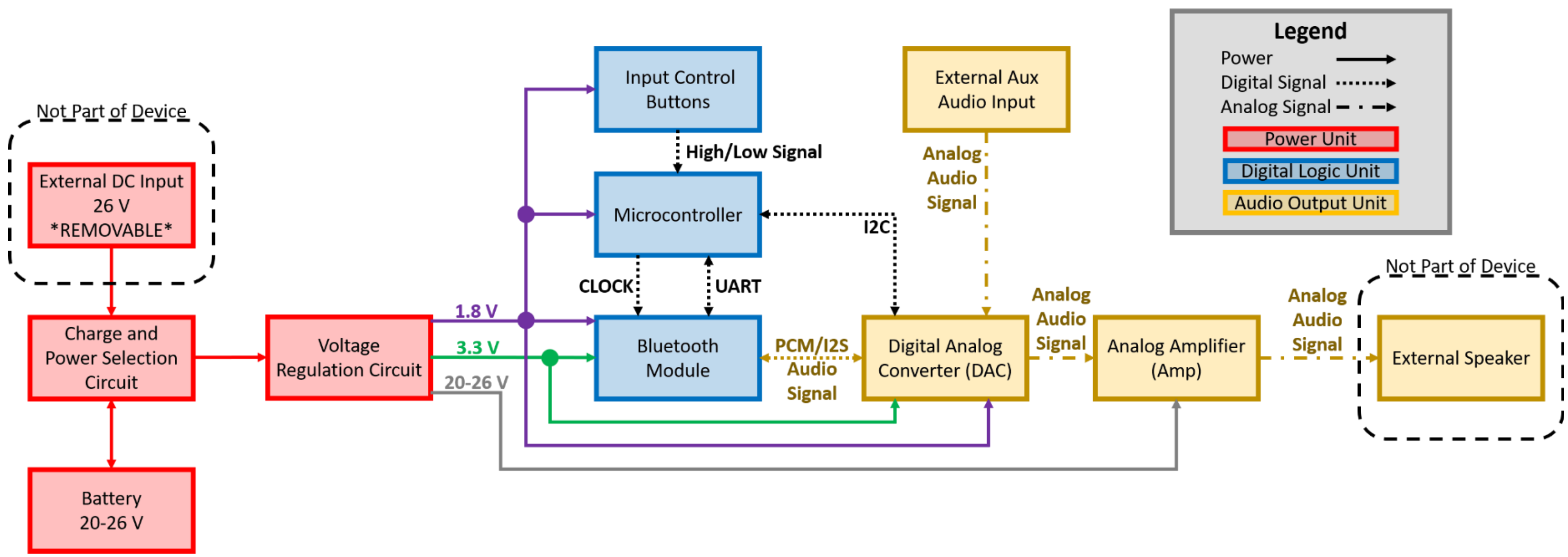

Figure 1. High-Level Block Diagram

1.4 Physical Specifications

The finalized physical format of the device consists of a rectangular box. All components are secured within the chassis. The rear of the box houses the banana plug connections for the speaker outputs, a 3.5mm audio input, and the DC power supply input. The front of the device contains the input buttons for easy accessibility. To yield a portable product, the appropriate dimensions should be 10 in. x 5 in. x 5 in. or smaller. The final physical product has dimensions of 9.42 in. x 4.98 in. x 3.53 in. (including the protruding buttons and jacks) which falls within the dimensional constraints of portability. Most backpacks will hold this volume comfortably. Figure 2 shows the 3D-Model of the product with major components labeled.

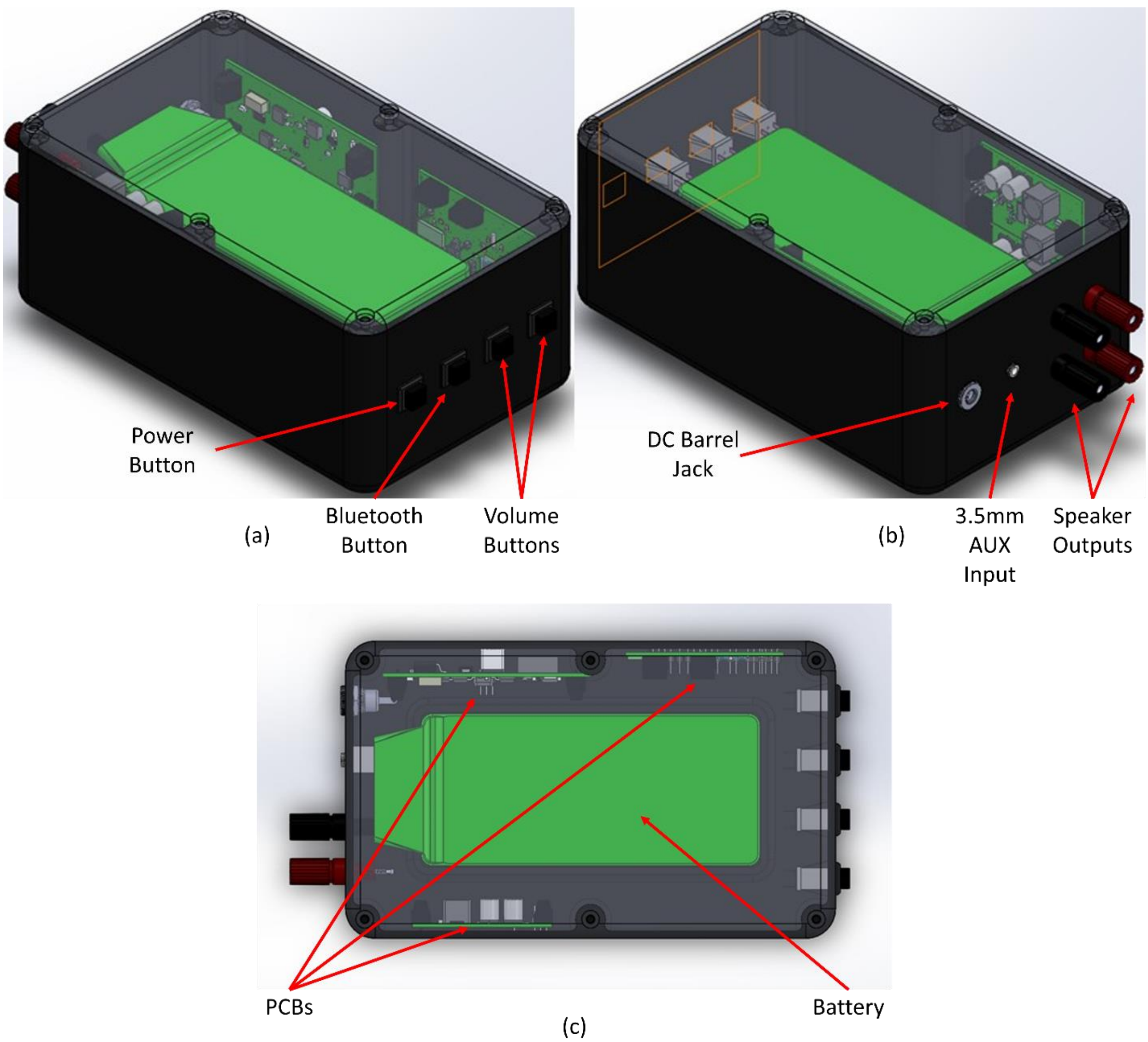

Figure 2. Physical Design 3D-Model

2. Design

2.1 Design Procedures

2.1.1 Power Unit

The Power Unit consists of four main blocks as shown in Figure 3. The Battery and External DC Input feed power into the Charge and Power Selection Circuit. The output of this circuit is fed to the Voltage Regulation Circuit to step down the voltages to the necessary values needed for the rest of the system.

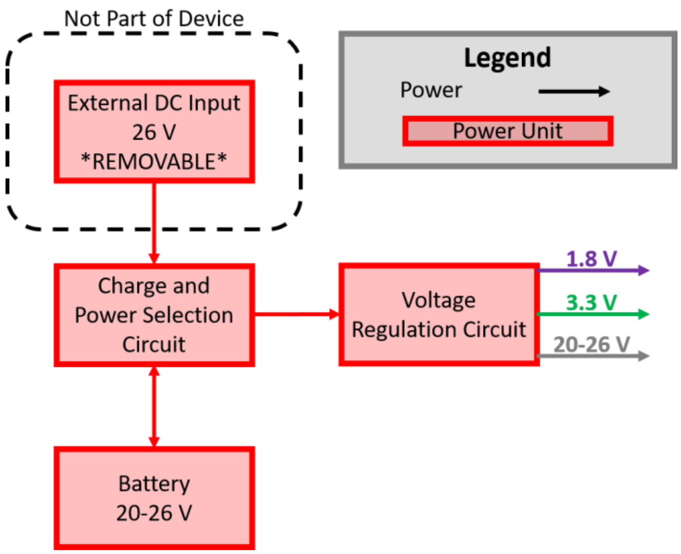

Figure 3. Power Unit Block Diagram

The Charge and Power Selection Circuit needs to be able to automatically select between the two sources for whichever is present (preferring the External DC Input when both are available) and charge the six-cell lithium polymer battery when the external is connected. Based on these requirements, the BQ24610RGER IC was chosen, since it supports auto-power selection and charging of lithium batteries up to six cells.

The primary purpose of the PU (from the perspective of the other units in the system) is to supply 20-26 V, 3.3 V, and 1.8 V. The 20-26 V comes directly from the battery/external DC, so the supply is directly passed to the output for that voltage, eliminating the need for a voltage regulator at that voltage. For the 3.3 V supply, the input voltage of 20-26 V is much greater than the 3.3 V output, so a switching voltage regulator is needed to handle the large voltage drop with little sacrifice in efficiency and minimal heat generation, thus the LM2576SX-3.3/NOPB voltage regulator was chosen. This regulator also supports current draw up to 3 A, which is well-above the 2 A maximum current needed for the system. As for the 1.8 V regulator, the 3.3 V supply is used as the input to this regulator, allowing for the TLV1117LV18DCYR linear regulator to be chosen for its smaller and cheaper form factor. This regulator supports a maximum of 1 A current draw, which is also above the required 500 mA for the system.

2.1.2 Digital Logic Unit

The DLU has two equivalent forms: the prototype and the final circuit. Because the prototype form is fully verified, it is the current state of the DLU. Software for both forms is developed in Code Composer Studio version 8.0.

At the high level, the final DLU circuit (depicted in Figure 4) consists of the MSP430F5659 microcontroller united with the CC2564MODA Bluetooth module, with push-button physical interface. These chips are readily integrated with royalty-free stack software and well-document Bluetooth APIs, both provided by Stonestreet One. This design is partially tested and partially verified.

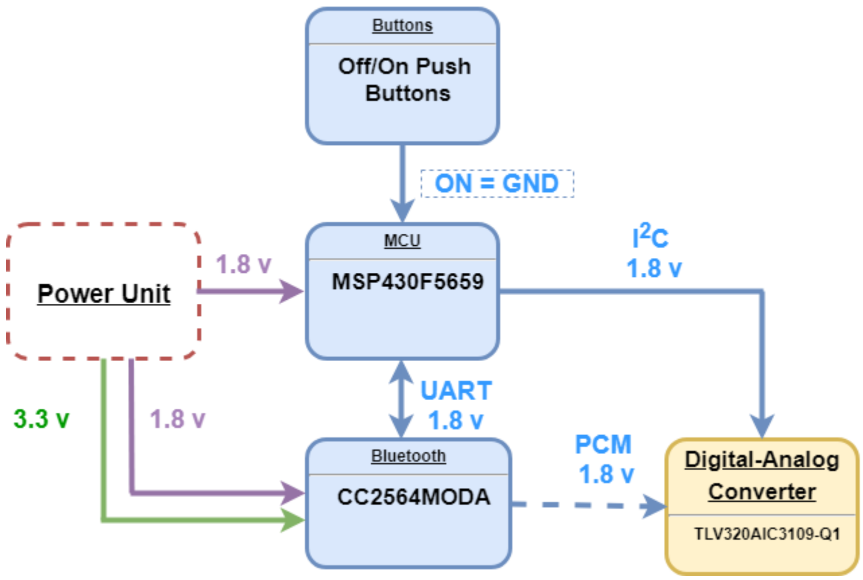

Figure 4.   Digital Logic Unit Final Circuit Block Diagram

The prototype (depicted in Figure 5) consists of two development boards made by Texas Instruments: the MSP432P401R Launchpad and the BOOST-CC2564MODA; this form is completely tested and verified. The MSP432 comes readily integrated with the CC2564MODA using the same stack software, except designed for the MSP432 chip-type. The power configuration is different for the prototype, but the change in total power consumption is negligible, and is not considered in power calculations since it is only the prototype.

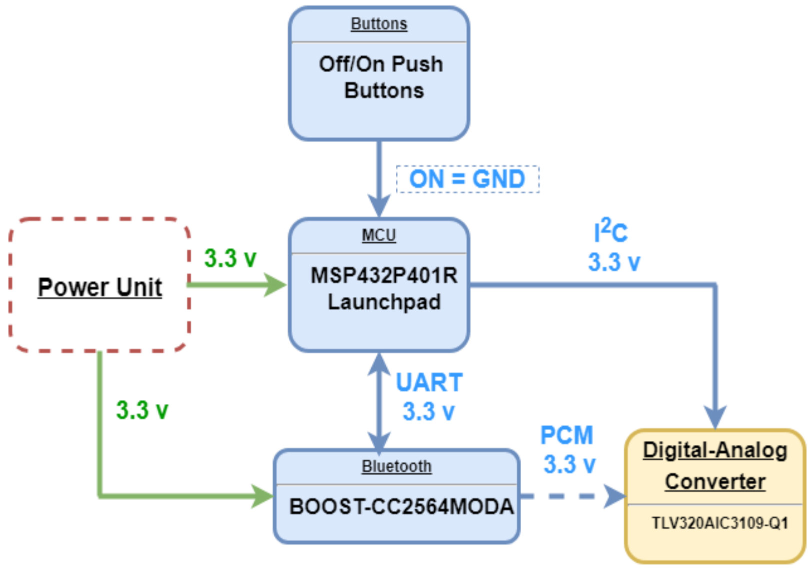

Figure 5. Digital Logic Unit Prototype Circuit Block Diagram

2.1.2.1 Final Circuit Microcontroller and Bluetooth Design Choices

We hinged our microcontroller choice on the choice of the Bluetooth module. We decided on the CC2564MODA as the Bluetooth module because it has a fully documented, built-in digital codec interface that supports a wide variety of codecs, as well as royalty-free stack software with documented Bluetooth profile APIs, allowing for a quick development process. Consequently, the MSP430F5659 was chosen because it supports the stack software, has 512 kB of flash storage, and runs actively with very low power consumption.

2.1.2.2 Circuit Development Tools

In final circuit development, an ez-FET lite debugger located on an MSP430F5529 Launchpad was used to flash software onto the MSP430F5569 using a serialized JTAG protocol called Spy-Bi-Wire.

During prototype development, the MSP432P401R was flashed and debugged using the onboard XDS-110ET emulator. The BOOST-CC2564MODA module is designed to fit onto the MSP432P401R launchpad using jumper blocks.

2.1.3 Audio Output Unit

The Audio Output Unit consists of two main blocks, the digital analog converter (DAC) and the amplifier (Amp), in Figure 6. The DAC receives programming and digital audio signals from the Digital Logic Unit. The audio signal can then be mixed by an external analog signal and sent to the amplifier. The amplifier then amplifies the output of the input signal. Both devices are powered from the Power Unit.

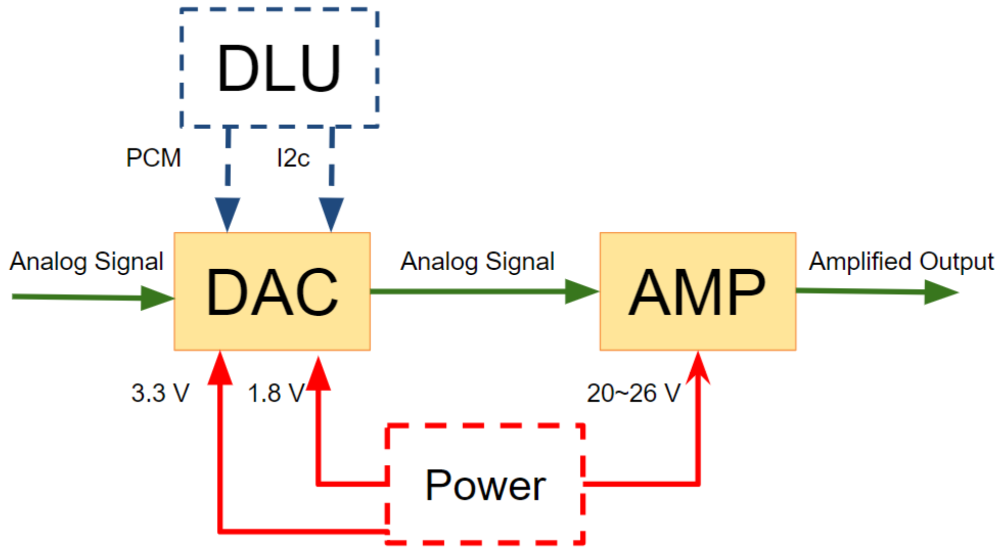

Figure 6. Audio Output Unit Block Diagram

The amplifier is a part of a battery-powered device, so we investigated finding an amplifier that is the most efficient in its power usage. Therefore, we chose to look for a Class-D amplifier. Other alternative amplifiers that exist are Class-A, Class-B, and Class-A/B amplifiers. Although these amplifiers have a cleaner output than a Class-D amplifier they produce a lot of heat [16]. As we did not want to design our chassis to have a fan and are running the device on a battery we chose to sacrifice audio quality for a longer battery life. For this reason, we chose the Texas Instruments TPA3112D1 Class-D amplifier, which is the cheapest amplifier found on Digi-Key that can meet our requirements to satisfy our Amplifier Requirement and Verifications found in Appendix A.

As the amplifier we chose did not have a programmable gain or multiple inputs, we needed to look for a DAC that can operate as a mixer for both the analog signals and the digital signals. Since we know the digital signal type will be PCM we chose the cheapest digital codec that supported mixing and PCM by Texas Instruments on Digi-Key and came to the result of the TLV320aic3109-Q1.

For the construction of the device we decided to solder the IC chips onto compatible breakout boards and breadboard the circuit before creating a PCB. This way we can verify if our circuit design works as intended. However, this resulted in problems such as trace inductance and will be discussed in Section 2.2.3.1.

2.2 Design Details

2.2.1 Power Unit

2.2.1.1 BQ24610 Charge IC Circuit

The most complex circuit in the PU was for the charge IC, which required multiple MOSFETs for power switching, as well as specific capacitor and resistor values for setting parameters. The full finalized schematic of the charge IC (along with the voltage regulators) is shown in Appendix B Figure 12. The guidelines of the circuit were based upon the BQ24610 Datasheet [8]. Major parameters included the battery voltage regulation, battery current regulation, input adapter current regulation, precharge/termination current configuration, and charge safety timer configuration.

2.2.1.2 Voltage Regulation Circuit

The voltage regulation circuit, although not as complex as the charge IC circuit, still required some specific components, including specifically sized capacitors, diodes, and inductors because of the switching voltage regulator used for the 3.3 V line.

2.2.2 Digital Logic Unit

The CC2564MODA was designed onto the DLU final circuit PCB with several considerations in mind. Table 1 shows important design details and the reason(s) they were implemented.

Table 1. DLU PCB Design Details

Detail	Reason for Implementation
All digital bus traces have no other traces running underneath them.	To minimize noise or interference from any other signal.
CC2564MODA external slow clock is generated by a dedicated oscillator.	To generate the clock as close as possible to the CC2564MODA to minimize drift.
CC2564MODA is located on the edge of the PCB with no ground planes or traces underneath the antenna.	To maximize the RF signal strength.
Digital PCM from CC2564MODA lead to jumper pins.	To allow for a disconnected DAC module.

2.2.2.3 Bluetooth Audio Streaming

The royalty-free stack for the CC2564MODA is employed on the MSP432P401R. The stack allows for quick and easy Bluetooth profile activation on the CC2564MODA. An audio sink sample program comes with the stack software, which activates the Advanced Audio Distribution Profile (A2DP) profile, defines callbacks for pairing and audio streaming events, and starts a scheduler to run indefinitely. Due to the length of the source code, it has not been included in the document. Modifications were made to the source code to accommodate the needs of this project. These modifications are in Appendix C as code, with file names derived from the file names in the stack and sample program.

Modification 1: In the sample program, set the Bluetooth device name after start-up so the amp becomes discoverable immediately after power-up.

Modification 2: In the HAL platform, disable the codec initialization of the CC3200 AUDIO BOOST module. In this prototype, the DAC is initialized by an external microcontroller, in this case a Raspberry Pi. The reason for this is defined in Section 5.2 of this report. In a final design, we would modify the original codec initialization to initialize the TLV320AIC3109 via I2C.

Modification 3: In the HAL platform, modify the Bluetooth Audio Codec configuration to output DSP timed PCM data configured for the TLV320AIC3109.

2.2.3 Audio Output Unit

2.2.3.1 Amplifier

The amplifier layout is based from the layout example provided by the TPA3112D1 datasheet [11]. The modification we made to that example was to use inductors as an output filter instead of ferrite beads due to the component being unfamiliar. We originally set it up on a breadboard for quick prototyping and testing, however we realized that long traces between the power pins and the decoupling capacitors caused transient voltage spikes which damaged our chips. When consulting a TI employee about the issue, it was stated that evaluating the device on a breadboard is violating the recommendations provided by the datasheet for evaluating the unit [17]. Following this, we designed a PCB with the decoupling capacitors as close as possible to the power pins on the amplifier. After assembling, the amplifier was able to operate to specifications and our requirements without faulting.

2.2.3.2 Digital Analog Converter

We prototyped our DAC using a protoboard and a breakout board. We determined that it was safe to evaluate the DAC using this setup without damaging it because the transient voltage spikes caused by the amplifier was because the amplifier outputs a significant amount of power compared to most IC chips. For example, the DAC's electrical characteristics has a power consumption of less than a watt compared to the 20-watt output of the amplifier.

The design of the DAC is based from the layout example provided by Texas Instrument's documentation [12]. As we only have a mono input for our analog input and we are not using a microphone we excluded those connections from the designs. As we are only working with I^2S we chose to program the DAC to generate a clock from the word-clock using PLL, thus removing the need of a master clock to the DAC.

3. Design Verification

The design of this device needed to be verified with the Requirement and Verification Table, which is listed in Appendix A Table 8. Each unit has its own requirements and verification procedures listed in the table. In this section, major verification results are shown for each unit in the system.

3.1 Power Unit

3.1.1 External DC Input

The External DC Input (Mean Well RS-150-24) needed to be able to supply 26 V with a tolerance of +0.2 V and -1 V at a peak current draw of 6 A. Using an Agilent 6060B DC Electronic Load, a constant load of 6 A was pulled from the external supply and the voltage was measured using a Keysight 34461A Multimeter. The results are shown in Table 2.

Table 2. Experimental Results of the External DC Input Test

Current Draw (A)	Output Voltage (V)
(No Load) 0.0000	25.9985
6.0609	25.9865

Based on the results in Table 2, the verification passed since the output voltage stayed well-within the tolerance of +0.2V / -1 V when at full load of 6 A.

3.1.2 Battery

The internal lithium polymer battery (Multistar 6S 8000 mAh) needed to be able to supply 20.0 - 26.0 V to the system at a peak current draw of 3 A. Following the same steps for the External DC Input described in Section 3.1.1 External DC Input, the output voltage was measured at no load and full load. The results are shown in Table 3.

Table 3. Experimental Results of the Battery Test

Current Draw (A)	Output Voltage (V)
(No Load) 0.0000	25.1792
8.0115	25.1135

From the results in Table 3, this battery passed the verification by maintaining an output voltage of 25.1135 V when pulling over 8 A, which is well-over the peak current draw of 3 A needed from the device.

The battery also needed to have a capacity large enough to support 3 hours of continuous use. To verify this, the battery was connected to an external lithium battery charger/discharger in discharge mode with a current draw of 2 A. The time elapsed for the battery to discharge to 21.5 V was recorded. This time should be greater than three hours. The results are shown in

Table 4.

Table 4. Experimental Results of the Battery Life Test

Current Draw (A)	Starting Voltage (V)	Ending Voltage (V)	Elapsed Time (hours)
2.0	25.19	21.54	3.7589

Based on the results listed in

Table 4, the battery capacity is verified, since it took over 3.75 hours for the battery to discharge.

3.1.3 Voltage Regulation Tests

Each of the three voltage lines (20-26 V, 3.3 V, and 1.8 V) have their respective required current draws from the system. The system needed to draw a maximum of 3 A at the 20-26 V line, 2 A with a tolerance of ±0.2 V at the 3.3 V line, and 0.5 A with a tolerance of ±0.2 V at the 1.8 V line. The PU was powered by the External DC Input, and each of the lines were individually connected to a DC Electronic Load set to constant current mode with the respective current draw. The results are shown in Table 5.

Table 5. Experimental Results of the Voltage Regulation Tests

<table>
<tr><th>Voltage Line (V)</th><th>Current Draw (A)</th><th>Output Voltage (V)</th></tr>
<tr><td rowspan="2">20-26</td><td>0.0000</td><td>25.9959</td></tr>
<tr><td>3.0711</td><td>25.7756</td></tr>
<tr><td rowspan="2">3.3</td><td>0.0000</td><td>3.32030</td></tr>
<tr><td>2.5230</td><td>3.30115</td></tr>
<tr><td rowspan="2">1.8</td><td>0.0000</td><td>1.79331</td></tr>
<tr><td>0.8201</td><td>1.80122</td></tr>
</table>

From the results in Table 8, all voltage lines passed verification since their output voltages stayed well-within the voltage tolerance of ±0.2 V when under full current draw.

3.2 Digital Logic Unit

There are several design qualities that had to be verified for the DLU to allow the user stable control of the device while playing music. The two most important requirements of the DLU from the original design are shown in Table 6.

Table 6. Major Requirements for the Digital Logic Unit

<table>
<tr><th>Module</th><th>Requirement</th></tr>
<tr><td rowspan="2">Bluetooth</td><td>Bluetooth module must be able to transmit PCM data via I^{2}S with:
1. a sampling frequency of at least 44.1 kHz
2. and a 16-bit frame width.</td></tr>
<tr><td>Device should support a range of at least 20 feet in unobstructed sight. Passed maximum distance means constant interference/drop-out occurs; finite/unpredictable interference/drop-out is acceptable.</td></tr>
</table>

3.2.1 Outgoing PCM Data Rate

While in development, we made the decision to use DSP timing instead of I^2S timing for lower power consumption. We reduced the verification of data rate to verification of two clocks that the CC2564MODA supplied to the DAC during PCM transmission: the bit clock and the word clock. To supply PCM data with a sampling frequency of 44.1 kHz, the word clock must have a rising edge at a frequency of 44.1 kHz. Figure 7 captures the frame clock during an audio stream, showing the correct frequency with minimal probing error.

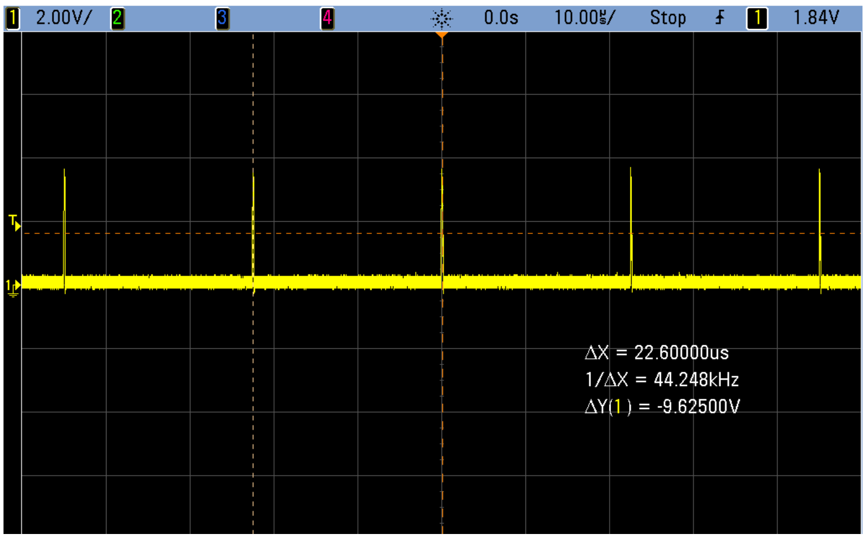

Figure 7. Word Clock Frequency During Audio Stream

Finally, we reduced a 16-bit frame width to the bit clock having a frequency equal to a multiple of 16 times the sampling frequency. We verify a bit clock frequency of 5.6448 MHz in Figure 8 using the same software build used during the word clock verification.

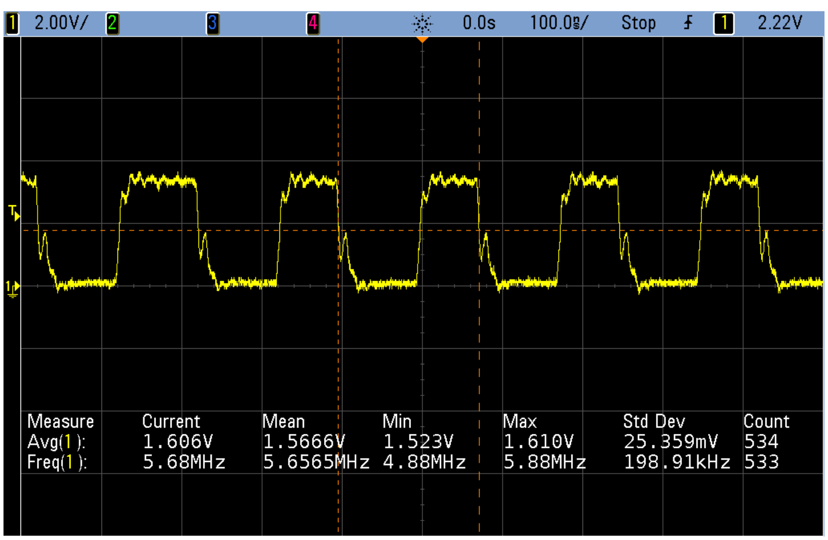

Figure 8. Bit Clock Frequency at 5.6448 MHz During Audio Stream

3.2.2 Bluetooth Signal Strength Versus Distance

To verify a stable Bluetooth signal with the device at ranges of at least 20 feet, we placed the DLU in the 3D-printed case and used a Samsung S9 smartphone to scan for Bluetooth devices and gather Received Signal Strength Indicator (RSSI) values. We used the Bluetooth Signal Strength Meter application by NeoFrontier Technologies to measure RSSI values of the BOOST-CC2564MODA, and since we followed all PCB guidelines for the CC2564MODA in our final circuit, we can use the gathered values as verification. Figure 9 shows the estimated RSSI values against distance.

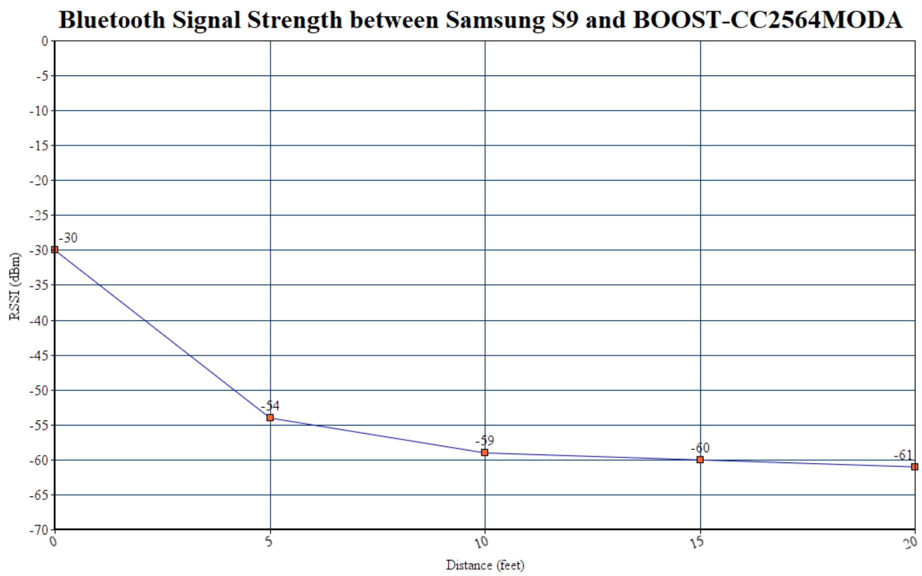

Figure 9. RSSI Values of the CC2564MODA at Various Ranges

Since a Bluetooth RSSI value of -67 dBm and greater is considered stable, we conclude that the device supports ranges of at least 20 feet.

3.3 Audio Output Unit

3.3.1 Amplifier

3.3.1.1 Output Power

To verify output power of the amplifier we input a 1 kHz sine wave into the amplifier and record the output of the amplifier, connected to an 8-ohm resistive load, with an oscilloscope. We then increase the input level of the sine wave until the recorded voltage cycle RMS hits the threshold voltage which results in an output power of over 20 watts continuous. This threshold can be calculated using the equation below.

$$\frac{V_{RMS}^2}{8} \geq \text{Target Wattage} \tag{1}$$

$$V_{RMS} \geq \sqrt{20 \cdot 8} = 12.649 \text{ W} \tag{2}$$

We were able to achieve a voltage of 14.99 V, which satisfies the requirement for the output.

3.3.1.2 Total Harmonic Distortion

To calculate the total harmonic distortion, or THD, of the amplifier we used an Agilent 89441A Vector Signal Analyzer. We used the signal generator in the analyzer to generate a 1 kHz sine wave and measured the FFT of the output using the analyzer. We then measure the fundamental harmonic up to the 4th harmonic, where each harmonic increments by 1 kHz. The results we receive from each harmonic is the power in dBm.

Table 7. Signal Analysis Results for THD

Harmonic	Frequency (kHz)	Power (dBm)	Power (W)
Fundamental	1	43.044	20.16
2^{nd}	2	0.466	0.001113
3^{rd}	3	-6.566	0.000220
4^{th}	4	-12.029	0.000062

We must then convert power dBm to power watts to use the THD power equation (3).

$$\text{THD} = 100 \times \sqrt{\frac{P_{Total}}{P_{Fundamental}}} \tag{3}$$

To convert to power watts, we use the following using power dBm:

$$P_{watts_n} = \frac{10^{\frac{P_{dBm}}{10}}}{1000} \tag{4}$$

$$P_{Total} = P_{watt_2} + P_{watt_3} + P_{watt_4} + \ldots = 0.001113 + 0.000220 + 0.000062 = 0.001396 \text{ W} \tag{5}$$

Using Equation (3), the THD percentage can be calculated:

$$\text{THD} = 100 \times \sqrt{\frac{0.001396}{20.15579}} = 0.83\% \tag{6}$$

As the total harmonic distortion is less than 1%, we have met our requirement.

3.3.1.3 Frequency Response

To calculate the frequency response of the amplifier we used an Agilent 89441A Vector Signal Analyzer. We begin by connecting the source generator of the analyzer into the inputs of the amplifier and probe the amplifier's outputs with the analyzer. We then create a frequency sweep between the frequency range of 1 Hz to 25,000 Hz. We then markers at 20 Hz and 20,000 Hz, the spectrum that we are observing.

We recorded that the highest response was 13.735 dB at 235 Hz, while the lowest response was 12.833 dB at 16 Hz. This is within the 4dB range that we have specified in our requirements. The frequency response graph of the amplifier is shown in Figure 10.

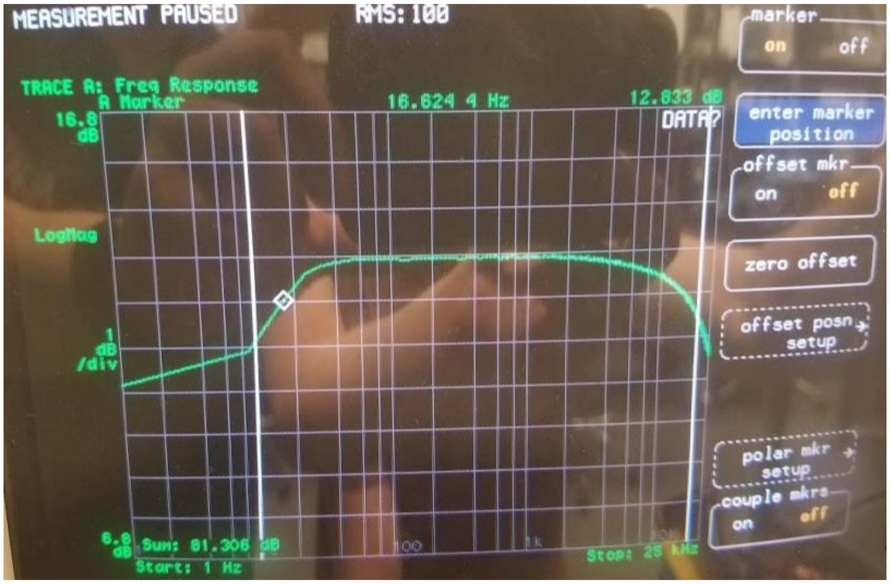

Figure 10. Frequency Response of Amplifier

3.3.2 Digital Analog Converter

3.3.2.1 PCM to Analog Signals

To verify the DAC, we must connect the word-clock, bit-clock, and data-out pints from the Bluetooth module into the DAC. We must then program the appropriate registers to enable the correct signal path from the DAC to the output. For example, powering on the DAC, closing switches, and selecting the clock rate. The code that we used to program that DAC is shown in Appendix C Figure 15. Figure 11 shows the output of the DAC which matches the Bluetooth input from our phone.

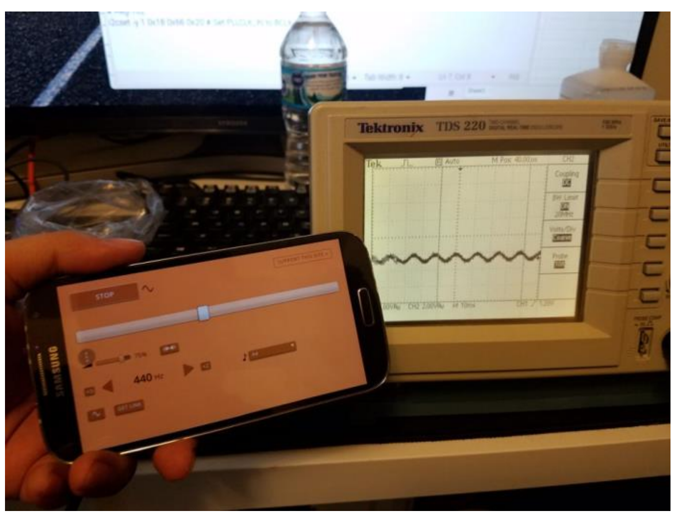

Figure 11. DAC Bluetooth Verification

4. Cost and Schedule

4.1 Parts

Part Name	Quantity	Unit Price	Total Cost
Texas Instruments MSP430F5659 Microcontroller	1	$6.02	$6.02
Texas Instruments TPA3112D1 Analog Amp	1	$2.54	$2.54
Texas Instruments CC2564MODA Bluetooth CS	1	$11.27	$11.27
Texas Instruments 6PAIC3109-Q1 DAC	1	$4.25	$4.25
CUI INC SJ1-352xN 3.5mm Jack	1	$0.76	$0.76
SparkFun PRT-09739 Binding Post Red	1	$0.35	$0.35
SparkFun PRT-09740 Binding Post Black	1	$0.35	$0.35
Texas Instruments BQ24610 Li-Polymer Battery Charger	2	$5.76	$11.52
PCBWay 18 in^2 PCB and Stencils	1	$50.00	$50.00
Texas Instruments LM1117 1.8V Linear Regulator	1	$1.14	$1.14
Texas Instruments LM2576SX-3.3 3-A Step-Down Voltage Regulator	1	$2.86	$2.86
Mouser Electronics CAPACITOR TANT 10UF 6.3V 20% 1206	5	$0.34	$1.70
Multistar 8000mAH Battery	1	$49.99	$49.99
Miscellaneous Capacitors, Resistors, ferrite beads, and Wires (estimate)	N/A	$10	$10
Generic 24V DC AC Power Supply	1	$22.99	$22.99
Chassis Print Filament	1	$3.00	$3.00
PCBWay PCB Shipping	1	$40.00	$40.00
HobbyKing Battery Shipping	1	$8.55	$8.55
Digi-Key Shipping	1	$7.49	$7.49
		Total	$234.78
		Single Unit Price (Without Shipping)	$99.99

4.2 Labor

Using the average salary of an EE graduate at the University of Illinois at Champaign-Urbana [15], we can determine a reasonable hourly rate for labor to be $32.21 an hour.

$$(\$67{,}000 / 52 \text{ weeks} \cdot 40 \text{ hours}) = \$32.21 \text{ an hour}$$

$$\$32.21 \cdot (3 \text{ Members}) \cdot (10 \text{ hours/week}) \cdot 16 \text{ weeks} = \$15{,}460.80$$

4.3 Schedule

Week	***Anthony***	***Nicholas***	***Austin***
Feb. 19	Compose Design Document, Breakout Amp Pins, Verify calculations for capacitors/inductors/resistors	Compose Design Document, calculate resistors/capacitors/inductor values for charge circuit, create initial chassis design	Compose Design Document, create DLU Eagle packages
Feb 26	Assemble Amplifier	Update components needed for charge and voltage regulation circuits	Troubleshoot MSP430F5234 debugging issues
Mar. 5	Troubleshoot Amplifier - No audio output	Finalize all components based on power requirements	Troubleshoot MSP430F5234 debugging issues
Mar. 12	Troubleshoot Amplifier - Chips being damaged	Finalize schematic of the charge and voltage regulation circuits	Design PCB consisting of MSP430F5529 and CC2564MODA
Mar. 19	Spring Break [Begin Final Report]		
Mar. 26	Assemble DAC, Troubleshoot Amplifier - Chips being damaged	Finalize PCB board layout, order PCB and all components	Read Bluetooth stack and modify it to fit on MSP430F5529 (doesn't fit)
Apr. 2	Contact Texas Instruments for Troubleshooting, Design PCB for Amplifier	Finalize verification steps for testing the Power Unit Board, finish design of 3D-printed chassis	Design PCB consisting of MSP430F5659 and CC2564MODA
Apr. 9	Test PCB for amplifier, verify amplifier requirements, Program DAC, integrate with battery unit	Assemble full Power Unit PCB, complete all verification testing, 3D print chassis	Configure the stack on the MSP430F5659 - Build prototype using MSP432P401R Launchpad and BOOST-CC2564MODA
Apr. 16	Write Final Papers and prepare for Mock Demo Presentation		
Apr. 23	Continue Final Papers and prepare for Presentation		
Apr. 30	Complete Final Papers and Present		

5. Conclusion

5.1 Accomplishments

Although we were not able to fully assemble the chassis with all our components, we were able to validate all our requirements for each module separately and have a working device that we can use. The DAC was programmed successfully and was able to decode the Bluetooth signals it received from a mobile phone. We were able to demo the amplifier and power unit in a portable enclosure and test it with our own external speaker. This device was able to play audio at over 20 watts continuous output, thus exceeding our requirements. We also were able to design multiple custom PCBs with tight component layouts. This resulted in small PCBs like the Amplifier, with dimensions less than a 3 in. square.

5.2 Uncertainties

During development, we ran into issues with configuring the CC2564MODA stack and reducing the size so that it fit onto an MSP430F5529. After contacting Texas Instruments, and receiving no assistance on the matter, we decided that it was not plausible to prototype with the stack on an MSP430F5529 or any other chip with less than 256 kB of flash storage. This is because we would not be able to run a full audio distribution program on 128 kB of flash storage, and especially not with debugging capabilities.

Due to time constraints, we were not able to completely integrate the MSP430F5659/MSP432P401R with the DAC via I2C, even though we verified the capability to do so. We ended up having to initialize the DAC with an external microcontroller and serial command line to achieve the high-level requirements.

5.3 Ethical considerations

User safety is of the utmost concern when developing a product. Rule #1 of the IEEE Code of Ethics [5] was followed by assuring that the boosted audio signal cannot damage the user's property by limiting the amount of achievable gain, and more importantly by implementing a fuse between the battery the rest of the circuit to minimize the damage of a short in the battery.

Working with lithium batteries is inherently dangerous, especially during prototyping. We had to keep in mind the danger of batteries and the high possibility of shorting. While not in use, the battery was kept in a safe storage location away from people to prevent any injury of others [5]. When the battery was in use, we ensured that it was protected against damage and kept at a safe distance away from any person.

Being a Bluetooth device, there are several regulations that it must adhere to to maintain safety and legality. The Bluetooth module and stack must be qualified with Bluetooth SIG; however, a qualified, unmodified module can be used, which is what we used [4]. Also, the Bluetooth module (and hence the device as a whole), used the correct "Bluetooth SIG"-approved profiles, namely the advanced audio distribution profile and the audio/video remote control profile.

All engineers should remember to cite all sources and credit contributors properly [5], and since this project required some software, we made sure to check if any code used or modified had been trademarked or licensed. The Bluetooth module uses a royalty-free stack software for specific microcontrollers, so no special licensing was needed.

5.4 Future work

As the product is something that is unique to today's market, there are two different devices that can branch out of this that can cater to more specific user needs. As our battery of over-spec for our requirements, we can develop a device that supports either a higher wattage amplifier or support stereo output. For those looking for more power and do not mind the hassle of carrying around two amplifiers, they can incorporate the technologies of Bluetooth 5.0 to connect to multiple amplifiers at once.

Our final product met the requirements of being portable, however it still had a lot of space inside the chassis. Therefore, we can slim down the design in future revisions.

An alternative design would be to incorporate the DAC into the Digital Logic Unit, rather than the Audio Output Unit. As the DAC requires many handshakes between the microcontroller and itself it would be easier to have these connections traced on a PCB rather than through a dozen header cables into the DAC.

References

[1] Jesse Maida, "Global Bluetooth Speaker Market to Grow at a CAGR of 33%". Available: https://www.businesswire.com/news/home/20160812005201/en/Global-Bluetooth-Speaker-Market-Grow-CAGR-33. [Accessed 8-Feb-2018]

[2] Bose, "SoundLink Revolve+ Bluetooth speaker". Available: https://www.bose.com/en_us/products/speakers/portable_speakers/soundlink-revolve-plus-bundle.html. [Accessed 8-Feb-2018]

[3] JBL, "JBL Boombox". Available: https://www.jbl.com/bluetooth-speakers/JBL+BOOMBOX.html?cgid=bluetooth-speakers&dwvar_JBL%20BOOMBOX_color=Black-USA-Current#start=1. [Accessed 8-Feb-2018]

[4] Bluetooth SIG, "Product Qualification and Declaration". Available: https://www.bluetooth.com/develop-with-bluetooth/qualification-listing. [Accessed 8-Feb-2018]

[5] IEEE, "IEEE Code of Ethics". Available: https://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed 8-Feb-2018]

[6] Texas Instruments, "bq24610 Datasheet". Available: http://www.ti.com/lit/ds/symlink/bq24610.pdf. [Accessed 13-Feb-2018]

[7] All About Circuits, "The Importance of Total Harmonic Distortion", Available: https://www.allaboutcircuits.com/technical-articles/the-importance-of-total-harmonic-distortion. [Accessed 21-Feb-2018]

[8] Texas Instruments, "BQ24610 Datasheet". Available: http://www.ti.com/lit/ds/symlink/bq24610.pdf. [Accessed 21-Feb-2018]

[9] Texas Instruments, "LM2576 Datasheet". Available: http://www.ti.com/lit/ds/symlink/lm2576.pdf. [Accessed 21-Feb-2018]

[10]Texas Instruments, "LM1117 Datasheet". Available: http://www.ti.com/lit/ds/symlink/lm1117.pdf. [Accessed 21-Feb-2018]

[11]Texas Instruments, "TPA3112D1 Datasheet". Available: http://www.ti.com/lit/ds/symlink/tpa3112d1.pdf. [Accessed 21-Feb-2018]

[12]Texas Instruments, "TLV320AIC3109-Q1 Datasheet". Available: http://www.ti.com/lit/ds/symlink/tlv320aic3109-q1.pdf. [Accessed 21-Feb-2018]

[13]Texas Instruments, "CC2564MODA Datasheet". Available: http://www.ti.com/lit/ds/symlink/cc2564moda.pdf. [Accessed 21-Feb-2018]

[14]Texas Instruments, "MSP430F5418A Datasheet". Available: http://www.ti.com/lit/ds/symlink/msp430f5418a.pdf. [Accessed 21-Feb-2018]

[15]University of Illinois, "Salary Averages – ECE Graduate Starting Salaries". Available: https://ece.illinois.edu/admissions/why-ece/salary-averages.asp. [Accessed 22-Feb-2018]

[16]Lifewire, "Car Amplifier Classes". Available: https://www.lifewire.com/car-amplifier-classes-534555. [Accessed 29-Apr-2018]

[17]Texas Instruments, "Chip permanently faults after a certain input/output threshold". Available: http://e2e.ti.com/support/amplifiers/audio_amplifiers/f/6/p/676902/2494136#2494136 [Accessed 29-Apr-2018]

Appendix A Requirement and Verification Table

Table 8. System Requirements and Verifications

Requirement	Verification	Verification status (Y or N)
External DC Input		
1. Must be able to supply 26 V +0.2 V / -1 V at 6 A current draw to the charge circuit and to the voltage regulation circuit.	1. Connect the supply to a DC Electronic Load in constant current mode and increase the current to 6.0 A. Measure the voltage to confirm it is between 25.0 V and 26.2 V.	Y
2. Must be removable from the system for portability.	2. Confirm that the output plug of the external supply is a common DC barrel jack connector.	Y
Internal Battery		
1. Must have a fuse between the battery and the rest of the circuit to prevent damage to the battery in the case of a short-circuit.	1. Place a fuse with a voltage rating above 26 V and a current rating less than 20 A immediately after the battery in the circuit.	Y
2. Must supply 20-26 V at 3 A to the voltage regulators and amplifier.	2. Connect the battery to a DC Electronic Load in constant current mode and increase the current to 3.0 A. Measure the voltage to confirm it is between 20.0 V and 26.0 V.	Y
3. Must have a capacity large enough for at least 3 hours of runtime.	3. Connect the battery to a lithium battery charger/discharger set to discharge at 2.0 A. Stop the discharge when the battery voltage drops to 21.5 V ±0.5 V and verify that the recorded time elapsed is greater than 3.0 hours.	Y
Charge and Power Selection Circuit		
1. Must be able to stop the battery from charging once the battery has reached full charge.	1. Connect the battery to the charge circuit and measure the current applied to the battery with a multimeter. Confirm that the current going into the battery is below 10 mA when battery voltage is greater than the external supply voltage.	Y

Voltage Regulation Circuit		
1. Must be able to supply 3.3 V ±0.2 V at 2 A to the Digital logic and Audio Output Units.	1. Connect the 3.3 V output to a DC electronic load in constant current mode and increase the current to 2.0 A. Measure the voltage to confirm that it is between 3.1 V and 3.5 V.	Y
2. Must be able to supply 1.8 V ±0.2 V at 500 mA to the Bluetooth module.	2. Connect the 1.8 V output to a DC electronic load in constant current mode and increase the current to 500 mA. Measure the voltage to confirm that it is between 1.6 V and 2.0 V.	Y
Bluetooth Module		
1. Bluetooth module/stack must be Bluetooth qualified and must be compatible with the advanced audio distribution profile (A2DP) and audio/video remote control profile (AVRCP) [4].	1. Find the necessary documentation to prove that the module we use is Bluetooth qualified and compatible with both profiles. Run the Bluetooth stack in both profiles, verifying they work separately and simultaneously.	Y
2. Bluetooth module must be able to transmit PCM data via I^2S with: a. a sampling frequency of at least 44.1 kHz b. and a 16-bit frame width.	2. Verification steps: a. Verify under an oscilloscope that the bit clock signal has a frequency of 705.6 kHz mono or 1.4112 MHz stereo b. View the I^2S signal with an oscilloscope and verify that the frame is 16 bits wide.	Y
3. Device should support a range of at least 20 feet in unobstructed sight. Passed maximum distance means constant interference/drop-out occurs; finite/unpredictable interference/drop-out is acceptable.	3. Connect to the device and play audio. Increase distance from the device until consistent interference or drop-out occurs. Measure distance.	Y
Microcontroller		
1. The controller must transmit a 3-wire UART signal with speeds >1 Mbps for fast control responsiveness.	1. View the UART signal in an oscilloscope, locate a start and stop sequence, and calculate the transfer speed of the sample packet.	Y

2. The controller must be able to initialize and shutdown the device from a switch.	2. Initialize device and measure with multimeter current drawn from power source to each module. Shutdown device with switch and verify with multimeter that current to all modules is 0.	Y
3. The controller must be able to adjust the gain of the amp via Bluetooth UART and the physical volume buttons.	3. Probe the i/o pins to the gain control pins on the microcontroller. When the volume button is pressed, the oscilloscope should show a spike for increasing the gain.	Y
4. Must be able to communicate via I^2C at 400 kbps.	4. View the I^2C signal in an oscilloscope, locate the start and stop sequence, and calculate the transfer speed of the sample packet.	Y
Input Control Buttons		
1. Must be tactile for the user to input functions.	1. Press the button and confirm that there is physical feedback when the button is pressed.	Y
2. Must be momentary type buttons.	2. Check the resistance across the button terminals when pressed and not pressed to confirm that the resistance is low (less than 10 Ω) only when the button is pressed.	Y
Digital Analog Converter (DAC)		
1. The DAC must be able to receive and convert PCM data via I^2S to an analog signal.	1. Connect the word clock, bit clock, and data output from the Bluetooth module into the DAC and play a 1 kHz sine wave from your audio device. Measure the outputs of the DAC and verify that the frequency of the wave it outputs is 1 kHz.	Y

2. The DAC must be able to switch between an analog input and a digital input (PCM data from I^2S) from a switch.	2. Play a 100 Hz sine wave through the Bluetooth device, and a 1 kHz sine wave through the analog input of the DAC. Probe the outputs of the DAC with an oscilloscope. Flip the pin associated to the switch on the DAC from i/o level high (1.8 V) to GND and verify that the output wave frequency changes from 100 Hz to 1 kHz.	Y
Analog Amplifier (Amp)		
1. The amp must be able to playback an analog signal.	1. Connect the amp's output into the resistive load and probe the output of the amp with an oscilloscope. Use a signal generator and generate a 1 kHz sine wave into the amp's audio input. Verify that the amp outputs a 1 kHz sine wave through the oscilloscope.	Y
2. The amp must be able to reproduce audio between the frequency ranges of 20 Hz and 20 kHz with response variation 4 dB or less.	2. Connect the resistive load to the amp's output and probe the outputs with a vector signal analyzer. Connect the source output of the vector signal analyzer to the input of the amp. Perform a sweep from 20 Hz to 20 kHz. Measure the maximum and minimum amplitudes within the 20 Hz to 20 kHz range and verify that it is within 4 dB of each other.	Y
3. The amp must be able to output 20 watts ±2 watts continuous signal with an 8 Ω load using a 1 kHz sine wave input.	3. Connect the resistive load to the amp's output and probe the outputs with an oscilloscope. Generate a 1 kHz sine wave to the amp's input. Measure the cycle RMS voltage of the output wave and record this value as voltage. Input the voltage into Power = $V^2/8$ and increase the input level and gain until the power is between 18 and 22 watts.	Y

4. The amp must be able to keep a THD of less than 1% with an 8 Ω load using a 1 kHz sine wave input.	4. Connect the amp's output into the resistive load and probe the output of the amp with a spectrum analyzer. Use a signal generator and generate a 1 kHz sine wave into the amp's audio input. Follow the steps found in verification 3 of this section to set the output power to 18 to 22 watts. Enable the oscilloscope FFT function to view the frequency spectrum. Record the amplitudes as volts RMS of the distorted sine wave.	Y
External Auxiliary (Aux) Audio Input		
1. Input jack must be a 3.5 mm (⅛ inch) headphone port.	1. Connect a 3.5 mm connector from the device to a phone. Ensure that the ring and tip pins on the jack are receiving a signal by probing the pins with an oscilloscope.	Y
2. Input jack must have a switch that detects if a 3.5 mm jack is inserted.	2. Probe the GND and switch pins on the input jack with a multimeter and start a continuity check. Insert and remove a 3.5 mm jack into the input. Continuity should switch.	Y
Chassis (Enclosure)		
1. Cannot decrease the functional Bluetooth signal range below 20 feet.	1. Enclose the device in the chassis. Power it on, connect to the device via Bluetooth, and play audio. Increase distance from the device until consistent interference/drop-out occurs. Measure the distance.	Y
2. The dimensions of the chassis must be equal to or less than 10 in. x 5 in. x 5 in.	2. Measure the chassis with a ruler/caliper and verify that the dimensions are equal or less than 10 in. x 5 in. x 5 in.	Y
3. The weight of the device must be under 10 lbs.	3. Measure the weight of the chassis with all its internal components and verify that the device is under 10 lbs.	Y

Appendix B Circuit Schematics

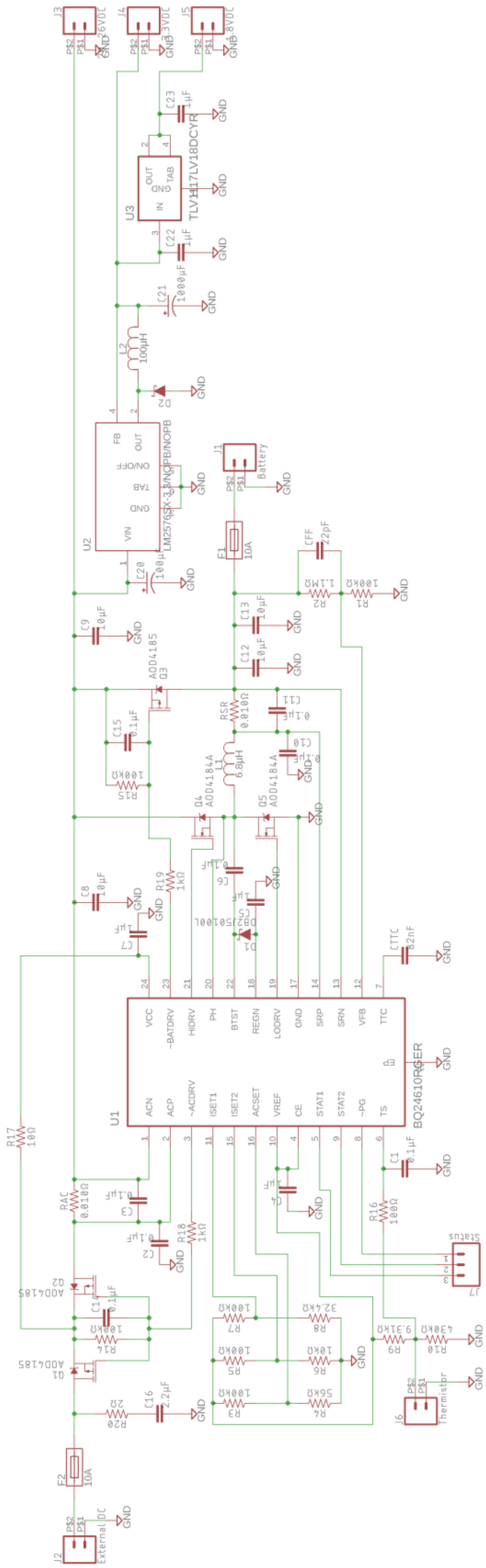

Figure 12. Full Circuit Schematic of the Power Unit

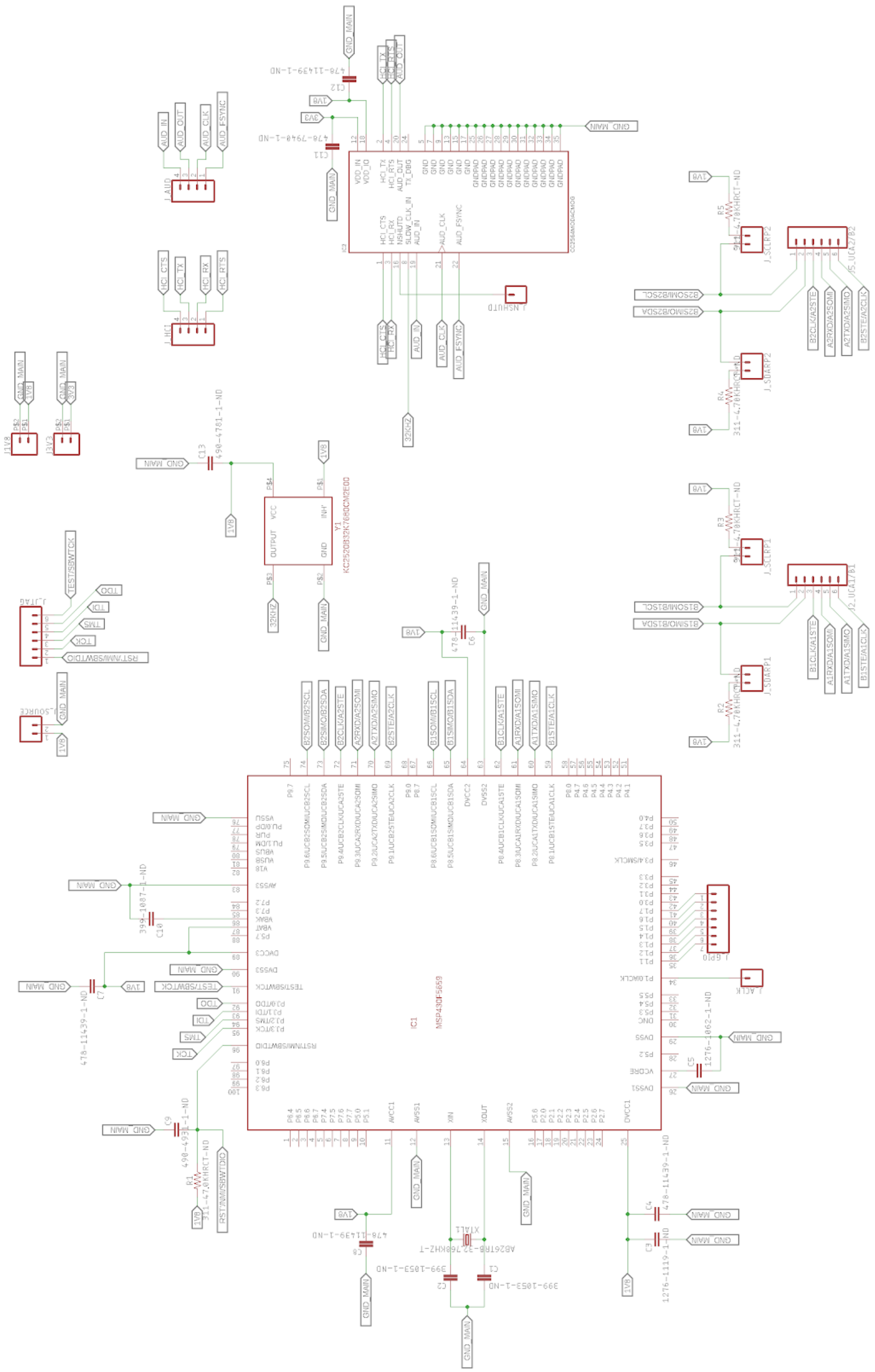

Figure 13. Full Circuit Schematic of the Digital Logic Unit

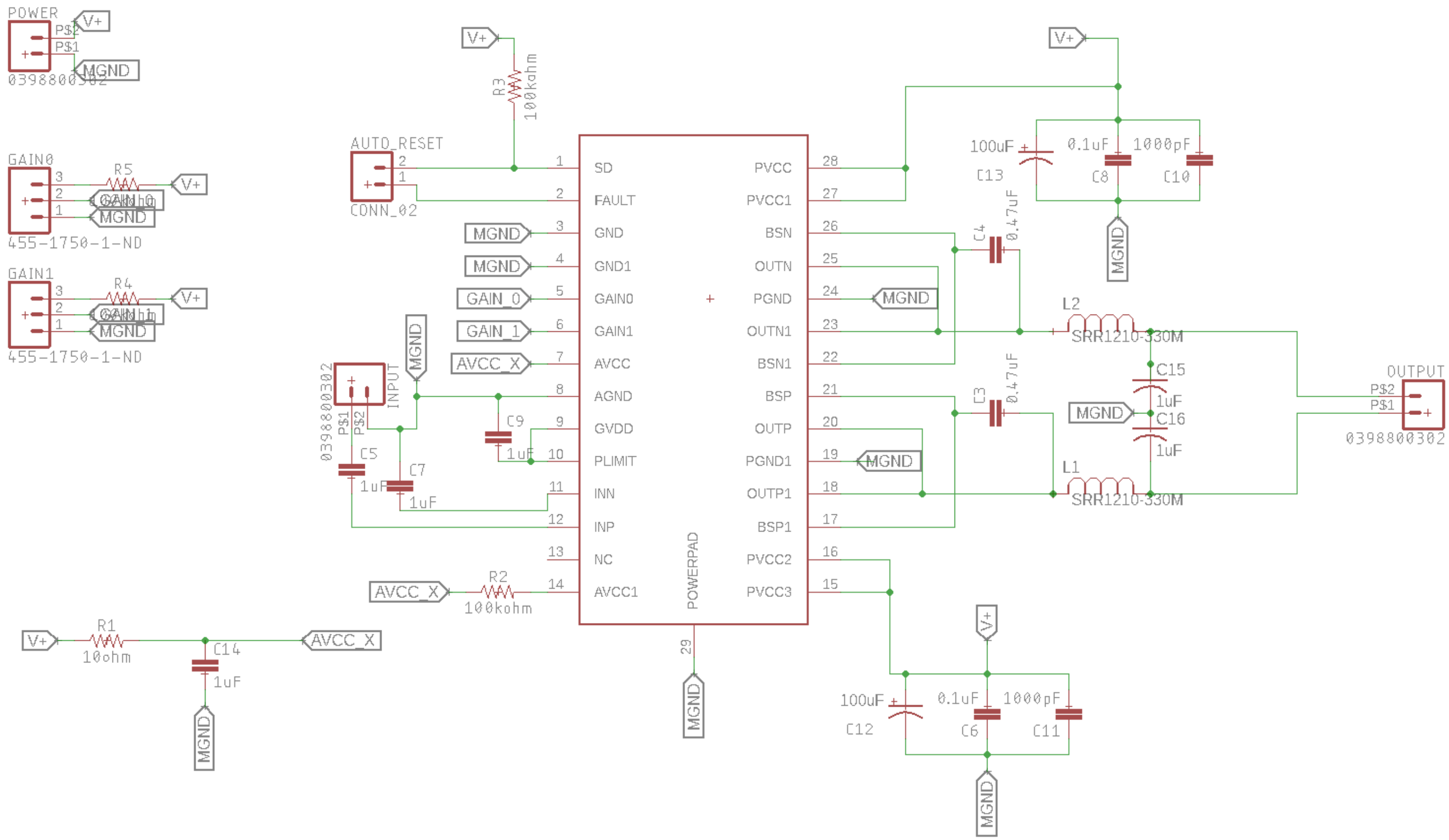

Figure 14. Full Circuit Schematic of the Audio Output Unit

Appendix C Code Snippets

```
# Reg-2
i2cset -y 1 0x18 0x02 0xAA # Fs divide by 6
# Reg-3
i2cset -y 1 0x18 0x03 0x85 # PLL On, Q=16, P=5
# Reg-4
i2cset -y 1 0x18 0x04 0xC0 # J=48
# Reg-7
i2cset -y 1 0x18 0x07 0x18 # unmute DAC
# Reg-8
i2cset -y 1 0x18 0x08 0x00 # Slave Mode
# Reg-9
i2cset -y 1 0x18 0x09 0x40 # DSP Mode, 16 bit word
# Reg-10
i2cset -y 1 0x18 0x0A 0x11 # 17 bit offset
# Reg-11
i2cset -y 1 0x18 0x0B 0x00 # pll R=16
# Reg-37
i2cset -y 1 0x18 0x25 0x80 # Power Dac On
# Reg-41
i2cset -y 1 0x18 0x29 0x40 # Set Dac output to LOP
# Reg-43
i2cset -y 1 0x18 0x2B 0x00 # UnMute Dac
# Reg-82
i2cset -y 1 0x18 0x52 0x80 # Dac to LOP Mux
# Reg-86
i2cset -y 1 0x18 0x56 0x08 # Un-Mute LOP, maybe 0x09
# Reg-102
i2cset -y 1 0x18 0x66 0x20 # Set PLLCLK_IN to BCLK
```

Figure 15. DAC Initialization Code

Modification 1 - A3DP_Demo.C beginning at line 3992

```
                                                  /* Name the bluetooth stack */
                    Parameter_t defaultStackName;
                    defaultStackName.strParam = "Bluetooth Amp";
                    defaultStackName.intParam = 0;
                    ParameterList_t defNameList;
                    defNameList.NumberofParameters = 1;
                    defNameList.Params[0] = (defaultStackName);

                    SetLocalName(&defNameList);
```

Modification 2 - HAL.C beginning at line 572

```
void HAL_EnableAudioCodec(unsigned int BluetoothStackID, HAL_Audio_Use_Case_t AudioUseCase, unsigned long
SamplingFrequency, unsigned int NumChannels)
{
   unsigned char InputLine;
   unsigned char OutputLine;

   const eUSCI_I2C_MasterConfig I2CConfig =
   {
      EUSCI_B_I2C_CLOCKSOURCE_SMCLK,      /* SMCLK Clock Source           */
      SMCLK_FREQUENCY,                    /* SMCLK Frequency              */
      EUSCI_B_I2C_SET_DATA_RATE_400KBPS,  /* I2C Clock Rate               */
      0,                                  /* No byte counter threshold    */
      EUSCI_B_I2C_NO_AUTO_STOP            /* No Autostop                  */
   };
```

Figure 16. Modifications to Sample Audio Sink Program and Bluetopia Stack

```
 /* Configure the I2C SDA and SCL pins.                               */
   GPIO_setAsPeripheralModuleFunctionInputPin(HRDWCFG_I2C_SDA_PORT_NUM, HRDWCFG_I2C_SDA_PIN_NUM,
GPIO_PRIMARY_MODULE_FUNCTION);
   GPIO_setAsPeripheralModuleFunctionInputPin(HRDWCFG_I2C_SCL_PORT_NUM, HRDWCFG_I2C_SCL_PIN_NUM,
GPIO_PRIMARY_MODULE_FUNCTION);

   /* Initialize I2C as the master.                                      */
   I2C_initMaster(HRDWCFG_I2C_MODULE, &I2CConfig);
   I2C_setSlaveAddress(HRDWCFG_I2C_MODULE, SLAVE_ADDRESS);
   I2C_setMode(HRDWCFG_I2C_MODULE, EUSCI_B_I2C_TRANSMIT_MODE);
   I2C_enableModule(HRDWCFG_I2C_MODULE);

   switch(AudioUseCase)
   {
      case aucA3DPSink:
         InputLine  = NO_INPUT;
         OutputLine = CODEC_LINE_OUT;
         break;
      case aucA3DPSource:
         InputLine  = CODEC_LINE_IN;
         OutputLine = NO_OUTPUT;
         break;
      case aucHFP_HSP:
         InputLine  = CODEC_ONBOARD_MIC;
         OutputLine = CODEC_LINE_OUT;
         break;
      case aucLoopbackTest:
         InputLine  = CODEC_LINE_IN;
         OutputLine = CODEC_LINE_OUT;
         break;
   }

   /* Initialize the local audio codec.                                  */
   //CodecInit(InputLine, OutputLine);

   /* Flag that the local audio codec is enabled.                        */
   AudioCodecEnabled = TRUE;

   /* Configure the controller's audio codec.                            */
   ConfigureControllerAudioCodec(BluetoothStackID, SamplingFrequency, NumChannels);
}
```

```
Modification 3 - HAL.C beginning at line 127

static void ConfigureControllerAudioCodec(unsigned int BluetoothStackID, unsigned long SamplingFrequency,
unsigned int NumChannels)
{
   Word_t Channel1Offset;
   Word_t Channel2Offset;

   union
   {
      VS_Write_Codec_Config_Params_t          WriteCodecConfigParams;
      VS_Write_Codec_Config_Enhanced_Params_t WriteCodecConfigEnhancedParams;
   } u;

   /* Set the codec config parameters.  The PCM clock rate is set to 80 */
   /* times faster than the frame sync clock frequency in order to match*/
   /* the BCLK/WCLK ratio expected by the CC3200AUDBOOST's audio codec  */
   /* (the TLV320AIC3254).                                               */
   Channel1Offset = (NumChannels == 1) ? 17 : 1;
   Channel2Offset = Channel1Offset + 16;
   BTPS_MemInitialize(&u.WriteCodecConfigParams, 0, sizeof(u.WriteCodecConfigParams));
   u.WriteCodecConfigParams.PCMClockRate_KHz      = (((SamplingFrequency) * 128) / 1000);
   u.WriteCodecConfigParams.FrameSyncFrequency_Hz = (DWord_t)SamplingFrequency;
   u.WriteCodecConfigParams.FrameSyncDutyCycle    = 0x0001;
   u.WriteCodecConfigParams.CH1DataOutSize        = 16;
   u.WriteCodecConfigParams.CH1DataOutOffset      = Channel1Offset;
   u.WriteCodecConfigParams.CH1DataInSize         = 16;
   u.WriteCodecConfigParams.CH1DataInOffset       = Channel1Offset;
   u.WriteCodecConfigParams.CH1InEdge             = 1;
   u.WriteCodecConfigParams.CH2DataOutSize        = 16;
   u.WriteCodecConfigParams.CH2DataOutOffset      = Channel2Offset;
   u.WriteCodecConfigParams.CH2DataInSize         = 16;
```

Figure 17. (Continued) Modifications to Sample Audio Sink Program and Bluetopia Stack

```
u.WriteCodecConfigParams.CH2DataInOffset       = Channel2Offset;
   u.WriteCodecConfigParams.CH2InEdge             = 1;
   VS_Write_Codec_Config(BluetoothStackID, &u.WriteCodecConfigParams);

   /* Set the codec config enhanced parameters.                         */
   BTPS_MemInitialize(&u.WriteCodecConfigEnhancedParams, 0, sizeof(u.WriteCodecConfigEnhancedParams));
   u.WriteCodecConfigEnhancedParams.CH1DataOutMode = 0x01;
   u.WriteCodecConfigEnhancedParams.CH2DataOutMode = 0x01;
   VS_Write_Codec_Config_Enhanced(BluetoothStackID, &u.WriteCodecConfigEnhancedParams);
}
```

Figure 18. (Continued) Modifications to Sample Audio Sink Program and Bluetopia Stack