BLUETOOTH 24-BIT HEADPHONE ADAPTER

By

Noctis Z.

Sang Baek Han

Wangming Yuan

Final Report for ECE 445, Senior Design, Fall 2017

TA: Zhen Qin

13 December 2017

Project No. 47

Abstract

Wireless technology has been replacing all the cables in our life. However, the quality that wired technology has been providing is difficult for a wireless device to mimic. This also applies to Bluetooth headphone adapter on the market nowadays. It allows us to have our phone or computer away from our body and play the music from our pocket. Yet, the audio quality provided is not satisfying compared to the wired-device. Therefore, we explore to design a Bluetooth headphone adapter with better audio quality by upgrading it to transmit 24-bit audio data using aptX™ HD codec while maintaining the pocket-size. In addition to this, we aimed to make it to play music longer by having a longer battery life. We successfully achieved these goals, but there are improvements that we can make for a better product. All the lessons learned and the future works will be discussed throughout the report.

Contents

1. Introduction

1.1 Objective

With the progress of the semiconductor industry, people started to cut down the cord, ditch the heavy equipment, to pursue the freedom of portability. This is refining the way of how people consume music contents in daily life. Since we stepped into the era of LTE, the rich bandwidth of mobile network, the music industries have shifted their concentration onto a music streaming service. Since streaming cost is aggressively cheaper than purchasing songs stand alone, people tend to compromise on the audio quality. Major streaming services are providing the users with 16-bit, 96 kbps for mobile low tier standard, 128 kbps for desktop standard. However, the base of the standard is raising up to 160 to 320 kbps for a premium audience [1]. Also, some Hi-Fi oriented services started streaming in 24-bit or even higher. It waves the flag of the next generation service in the market.

Yet, when we look at the popular peripherals, we know that for music enthusiasts, the top headphone sets are all using analog signal as input. Also, they play at a high impedance level. However, normal mobile devices won't be able to drive them. Now, the number of mobile devices, which are taking the music quality into consideration, is increasing. The Bluetooth standard is about to update to version 5.0. With the push from chipsets manufacturers, we can foresee a trend of public to have the investment onto the newer peripherals. 16-bit exists for a reason, and became a symbol of an era. However, innovators are not satisfied.

To follow this innovative movement, our goal is to make a Bluetooth headphone adapter that can transmit 24-bit audio data with a portable size and a longer battery life than the ones available on the market. The adapters on the market generally provide 16-bit audio data and last only for 9-15 hours [2]. We aim to make ours to transmit 24-bit and 192 kbps audio data while the battery last 40 to 50 hours.

1.2 Background

As an evidence of this innovative movement to provide better audio quality, LG came up with LG V20 last year with the support of playing 24-bit audio data [3]. The major music streaming services such as Spotify, Apple Music, Amazon Music Unlimited, Tidal, and Google Play Music started to provide the audio files with maximum of 256 to 1411 kbps [1]. As the trend moves to the wireless audio devices, the necessity of a portable wireless device that can play the audio files as provided for those music streaming services will eventually increase.

1.3 High-Level Requirements

- The adapter must be able to produce audio quality of 24-bit and 192 kbps above; better than that of Bluetooth headphone adapters on the market (16-bit and 128 kbps)
- The adapter must operate longer time; ideally 40~50 hours while playing music
- The adapter must be small size as possible; keeping the size portable

2. Design

The adapter requires total of four modules to make it a successful product: a Bluetooth module, a Digital-to-Analog module, a control module, and a power supply as shown in *Figure 1* below. The Bluetooth module will wirelessly receive audio data through Bluetooth receiver. Bluetooth A2DP protocol will send aptX™ HD audio codec data with 24-bit and 192 kbps. aptX/AAC/SBC codecs are also available for a device that can only send 16-bit. The Digital-to-Analog module will receive this 24-bit and 192 kbps digital audio data and convert it to analog audio data. Then, the analog audio data get amplified before outputting through 3.5mm audio jack. The control module will allow the user to increase/decrease volume, play/pause music, move to previous/next song, and power ON/OFF. Status LEDs will indicate whether the device is ON/OFF, pairing connection, and Bluetooth discoverable mode. Along with buttons and status LEDs, the microprocessor controls the power management to lithium-ion battery. The power supply module will provide 3.3V for Bluetooth module, Digital-to-Analog module, and control module from the lithium-ion battery.

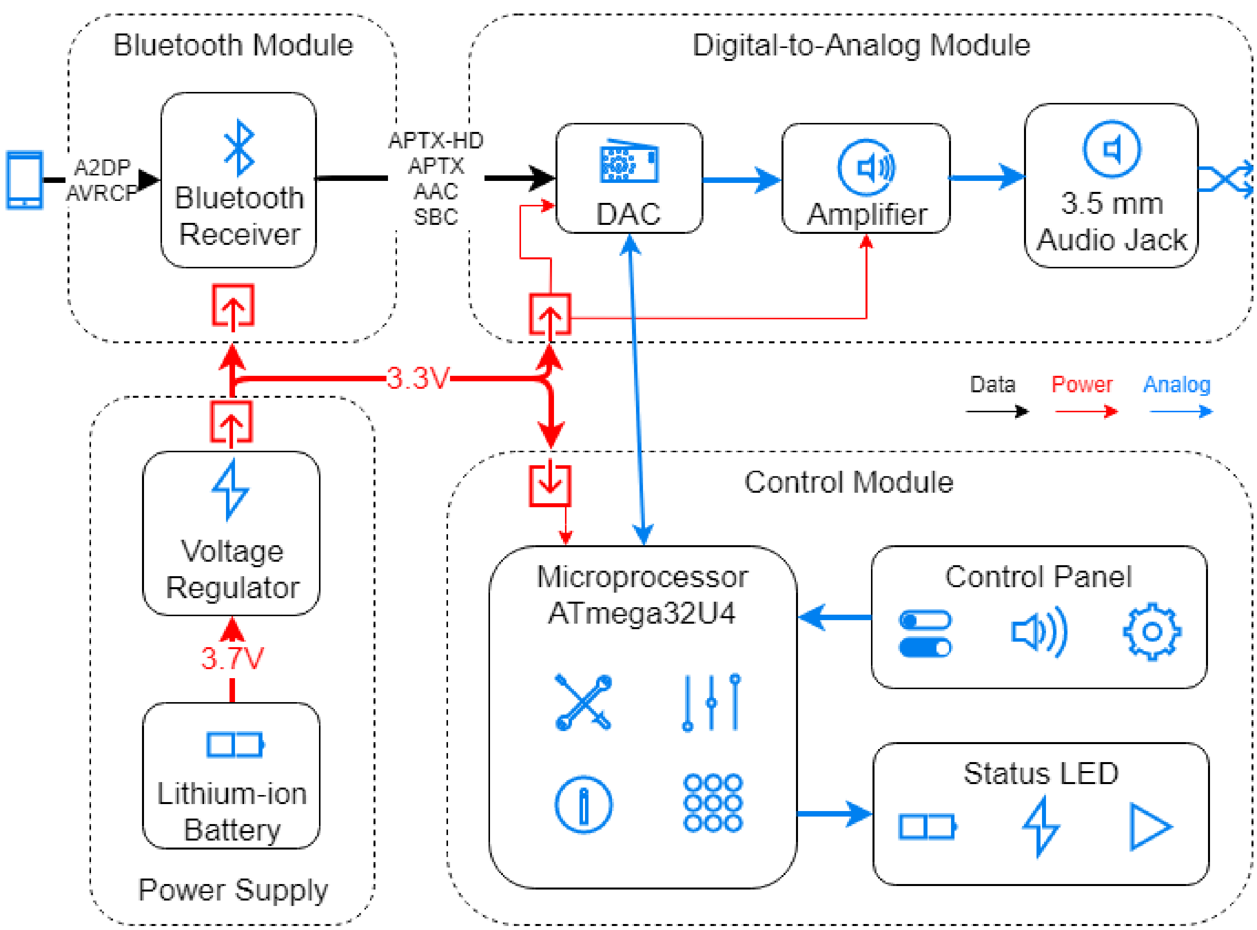

Figure 1 – Block Diagram

2.1 Bluetooth Module

A Bluetooth module receives the audio data through Bluetooth 4.2 protocol with Advanced Audio Distribution Profile (A2DP) and Audio/Video Remote Control Profile (AVRCP). A2DP will receive audio data using aptX™ HD codec for 24-bit audio data, aptX/AAC/SBC codec for 16-bit audio data. The detailed schematic is shown in Appendix D.

2.1.1 Bluetooth Receiver

This module takes the role of receiving incoming wireless signal via Bluetooth 4.2 protocol from the digital audio signal output device. This module pairs up with the output device and create a dual-way communication channel with continuous handshaking. The Bluetooth radio wave in between two devices operates with a bandwidth of 83.5 MHz [4], in a section of 2.4GHz frequency band which is for industrial, scientific, and medical use. This has been widely applied in daily life and commercial companies including Logitech have shown the stability of this wireless transmission. Hence, we have decided to choose the chipset, which can meet up with our design aim.

The Bluetooth Receiver will perform self-initialization with the respect of the user configuration from the microcontroller once the device is turned on. Next, if the device has been paired up with any devices before, by simply tapping the 'Pair' button on the board. If not, the module will stay on discoverable mode and wait for the incoming pairing token. Once the device is paired up, the receiver sends back the designed configuration to the digital audio signal output device.

A2DP protocol is used for our music streaming needs. A2DP stands for Advanced Audio Distribution Profile [5], it defines how high the stereo audio can be wirelessly streamed from one device to another. Moreover, Bluetooth Receiver uses AVRCP, which is Audio/Video Remote Control Profile. It is designed to provide a standard interface control for varieties of peripherals [6]. This adds up the compatibility for our module to control multi-system based output devices.

Once initiation is done, the user can use the on-board buttons to control music playback on the output device. When playback is enabled, the Bluetooth module switches to the best codec format which is available for the output devices. The priority for the codecs from high to low is aptX™ HD, aptX™, AAC, SBC. It starts the detection with the codec which has the highest priority, if it is not available, it will move to the next available codec sorted by the audio quality. Hence, we are able to provide the best available audio streaming to users.

Depending on the codec which is being chosen for the playback, aptX™ HD from Qualcomm is chosen by default. It has the compatibility to encode and decode audio data with 24-bit bit depth. Compared to the conventional 16-bit bit depth, which is the current common standard for audio capturing, the 24-bit audio is able to have 16,777,216 possible levels rather than 65,536 levels [7]. The more levels an audio signal can have, the more accurate the impulse signal can match with the source analog signal being captured. Also, it provides a lower noise floor for the audio. So that the 24-bit audio has a significantly better clarity and noise level control. In addition, the higher the audio sampling rate is, the more genuine the audio signal will be compared to the analog source. The Bluetooth audio sampling rate of aptX™ HD codec clocks is 48.0 kHz [8]. This is higher than the 44.1 kHz standard for CD recordings [9]. Meanwhile, when aptX™ HD is not available, the rest of the codecs aptX/AAC/SBC are able to process audio signals with 16-bit.

2.2 Digital-to-Analog Module

A Digital-to-Analog module receives 24-bit digital audio data from Bluetooth module. This digital audio data will be in aptX™ HD codec with 24-bit. 16-bit digital audio data can be in aptX/AAC/SBC codec. The analog audio signal will go through an amplifier to increase the gain, then output through 3.5mm audio jack.

2.2.1 Digital-to-Analog Converter

Our initial design in design document was to use PCM1794A 24-bit Digital-to-Analog Converter chip to receive 24-bit digital audio data and convert to analog audio signal. Yet, we found out that this chip has supply voltage of 5V and it outputs analog audio signal as differential signal. This will require us to have an op-amp in our PCB and 5V boost converter. Since we aimed to make this device pocket-size and we have an internal DAC on our Bluetooth Receiver, we decided to not use it on the basic design to reduce the size. If we had time, we could have externally attached PCM1794A on different PCB since we have external digital audio data output pins as shown in *Figure 2* below.

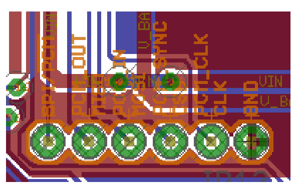

Figure 2 – External Digital Audio Data Output pins

2.2.2 Amplifier

For the best audio quality, Class-A amplifier will be the best because Class-A amplifier is conducting all the time for the output stage to maintain high linearity and gain. This means that the output amplified signal is not distorted as much from the input signal. However, the amplifier is conducting all times and constantly carries current; resulting a continuous loss of power [10].

But one of our design goal is to have a longer battery life, and our main focus on making the high audio quality is on usage of aptX™ HD that uses 24-bit codec. So, we decided to use Class-D amplifier that operates in non-linear switching. Class-D amplifier can theoretically reach 100% power efficiency since there is no voltage and current waveforms overlapping [10]. Also, there were no Class-A amplifier that uses 3.3V voltage supply for SOIC package. Class-A amplifiers available on the market are using mostly BGA, which is impossible for us to solder on PCB.

Our Bluetooth module outputs the analog audio signal as differential balanced audio output through internal DAC. Each channel is generated as a differential pair of signals so total of four connections are made with the amplifier. We used TPA6138A2 Class-D amplifier with 3.3V supply voltage to keep the size of PCB small by not using a 5V boost converter. Also, we can make the battery last longer than Class-A amplifier as discussed above. The amplifier schematic is shown in *Figure 3* below. The gain setting is 2V/V (6dB).

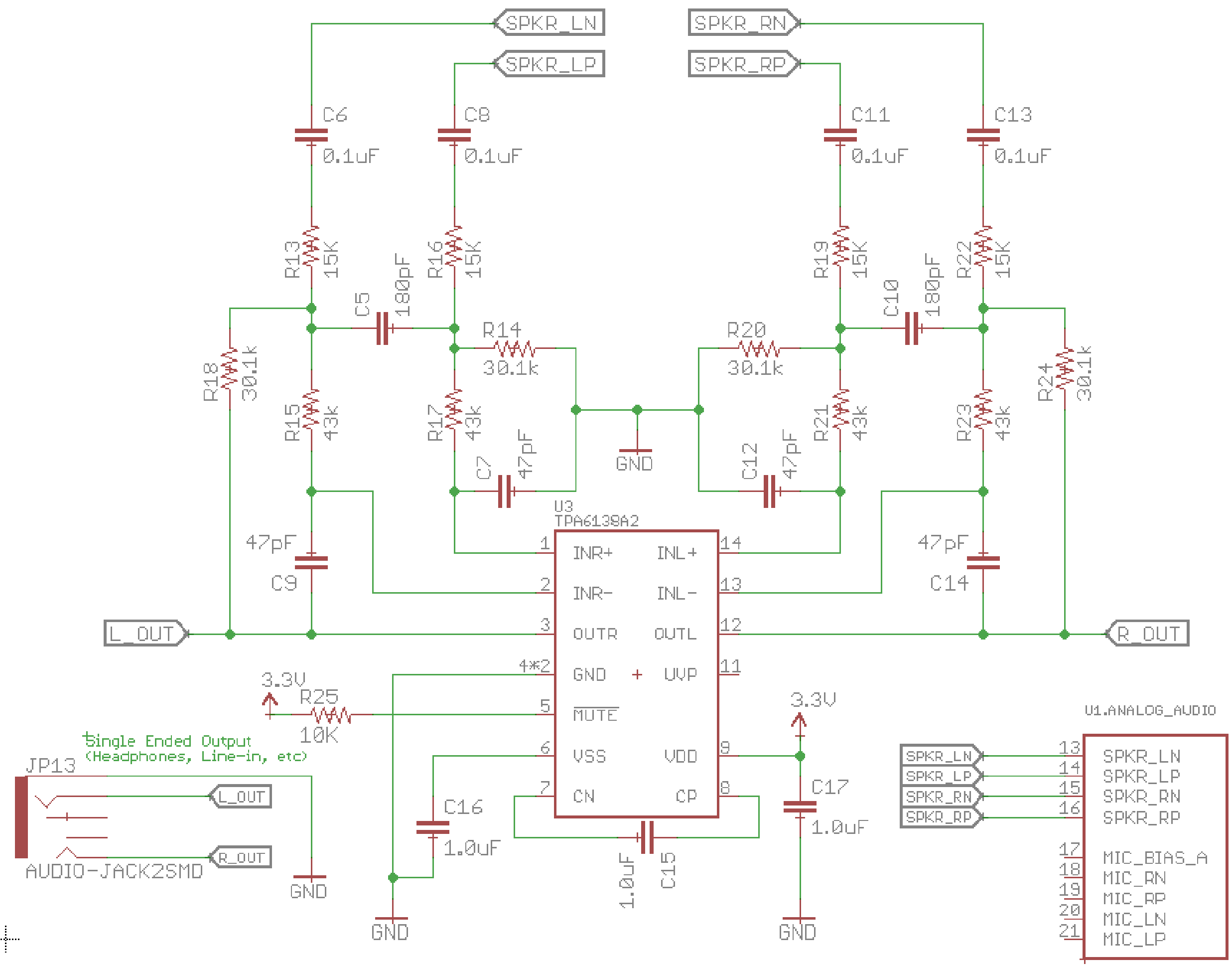

Figure 3 – TPA6138A2 Class-D Amplifier and 3.5mm Audio Jack Schematic

2.2.3 Audio Jack (3.5mm)

The output of the amplifier is connected to 3.5mm female audio jack to output the analog audio signal to headphone/earphone/speaker. 3.5mm was used as this is the most common peripherals for consumers as this is the standard jack of headphone/earphone/speaker. The audio jack schematics are shown in *Figure 3* above.

2.3 Power Supply Module

A power supply module provides constant DC voltage of 3.3V to keep the entire chips running. The detailed schematics are shown in Appendix E. The Lithium-ion battery will be charged using micro USB-B port. We wanted to make the device to run at least 40~50hrs continuously playing music. We are using calculations shown below to verify if we have enough capacity for Li-on battery.

$$P = I * V \tag{1}$$

Table 1: Power Consumption of Major Components

Part	Voltage (V)	Current (mA)	Power Consumption (mW)
Bluetooth Receiver (BC127-HD)	3.3	15	49.50
Class-D Amplifier (TPA6138A2)	3.3	6.4	21.12
Microprocessor (ATmega328P)	3.3	12.2	40.26

$$Total\ Power\ Consumption = 49.50 + 21.12 + 40.26 = 110.88mW \tag{2}$$

$$Total\ Current = \frac{P_{total}}{V_{supply}} = \frac{110.88mW}{3.3V} = 33.6mA \tag{3}$$

$$Battery\ Life = \frac{Battery\ Capacity}{I_{total}} = \frac{2000mAh}{33.6mA} = 59.52hr \tag{4}$$

Hence, we expect our device to play music continuously for at least 59hours.

2.3.1 Lithium-ion Battery

We initially thought of using Li-on battery with capacity of 1000mAh. We changed the capacity to 2000mAh because the device will play music continuously less than 30hrs with 1000mAh as shown in Equation 5 below. We aimed for 40~50hrs so increasing the capacity of the battery is the simplest solution

$$Battery\ Life = \frac{Battery\ Capacity}{I_{total}} = \frac{1000mAh}{33.6mA} = 29.76hr \tag{5}$$

Li-on battery outputs at average of 3.7V. The maximum voltage is 4.2V and the minimum voltage is 2.7V. Li-on battery can be recharged with micro USB-B port on our device through the power management circuit.

2.3.2 Voltage Regulator

MIC5219 Voltage Regulator 3.3V is used to supply 3.3V to all chips on our device. We initially thought of using LM3671 3.3V step-down DC-DC converter and TPS6122DC 5V boost converter, but we removed them because we are not using any chip that requires a 5V voltage supply. This reduces the size of our PCB by great quantity.

2.4 Control Module

Bluetooth Receiver can automatically receive the audio codec and send it to DAC to output the analog audio signal. Yet, Bluetooth pairing, power management, button control, and status LEDs need to be controlled by the control module. The system flowchart of the control module is shown in *Figure 4* below.

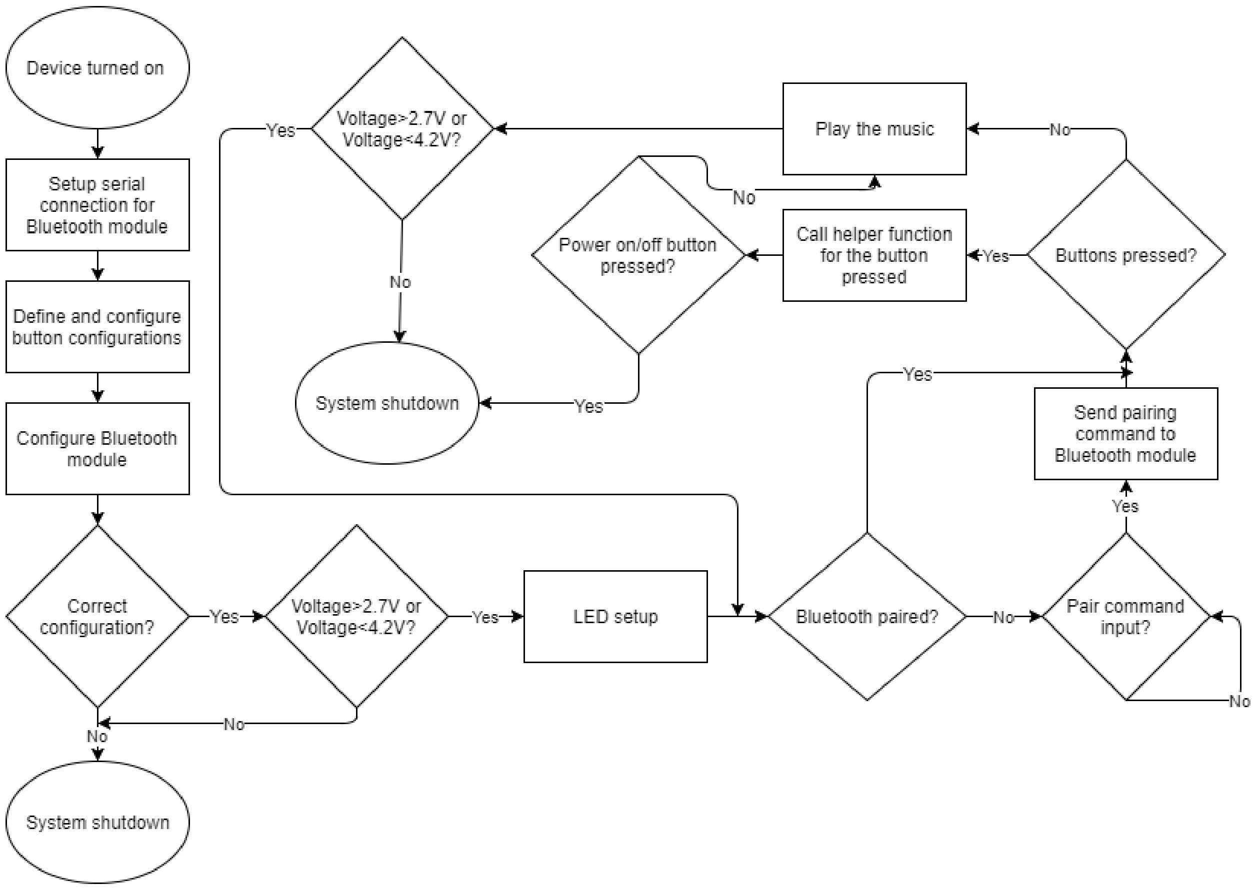

Figure 4 – System Flowchart of Control Module

When the MCU is turned on, it first sets up the communication with the Bluetooth module via UART serial communication. It then configures Bluetooth Receiver and the buttons. Then, it checks if the configuration is correct. If the configuration is wrong, it shuts down the whole system. If the configuration is correct, it checks if the system is operating between the safe voltage 2.7V and 4.2V, if not, it shuts down the whole system to protect other modules. If the voltage is safe, it sets up the LEDs to display and checks if the Bluetooth Receiver is paired. If it is not paired, it waits to be paired in a loop. If the Bluetooth module is paired, it checks if there are any buttons pressed and call the firmware functions to process the functionalities of the buttons. It keeps Bluetooth Receiver to send out the audio data it receives. When the power down button is pressed, it shuts the whole system down. It also constantly checks if the voltage is safe, if not, it immediately shuts the whole system down.

2.4.1 Microprocessor

Since we are not using microprocessor for any heavy math calculation, we used ATmega328 for the simplest usage. The firmware is already uploaded on Bluetooth Receiver and ATmega328 will communicate with Bluetooth Receiver through UART serial communication. Along with this, ATmega328 will control the GPIO button configuration and status LEDs. Also, the power management will be handled by ATmega328 to charge Li-on battery at 200mAh, which takes approximately 10hrs to charge our device. Microprocessor schematics are shown in Appendix F.

2.4.2 Control Buttons

Seven control buttons are used in the adapter. Six buttons are active-high when pressed, and one button is active-low when pressed. VREGEN button makes the adapter to DISCOVERABLE MODE, allowing the mobile device or computer to find the adapter. When VREGEN is pressed long, it will power ON/OFF the adapter. Remaining five buttons are: Volume DOWN, Volume UP, PREVIOUS song, PAUSE/PLAY song, NEXT song. RESET button is active-low and will reset the adapter status when pressed shortly. When pressed long, the adapter will force shut down. The control buttons schematic for active-HIGH is shown in *Figure 5* below.

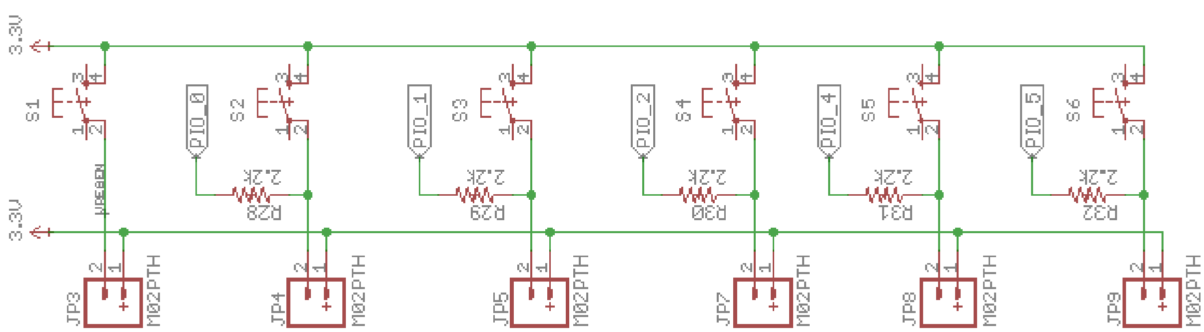

Figure 5 – Control Buttons Schematic for Active-HIGH

2.4.3 Status LEDs

Status LEDs indicate several statuses using three LEDs. When the adapter powers ON/OFF, all three LEDs will be on for one second and turn off. No LEDs flicking indicates that the adapter is in sleep mode. When the adapter is on, it will be flicking between LED0 (Green) and LED1 (Yellow) to indicate it is waiting for Bluetooth pairing. When the device connects to the adapter, and successfully paired up through Bluetooth, only LED1 (Yellow) will flick every one second to indicate the adapter is paired to the device. When the device disconnects the pairing, the adapter will be not paired to any device and LED0 (Green) will flick every one second to indicate DISCOVERABLE MODE.

3. Verification

3.1 Bluetooth Module

3.1.1 Bluetooth Receiver

As we pre-set the Bluetooth codec to be aptX™ HD, when it pairs up with aptX™ HD compatible Google Pixel XL, which runs on Android 8.1.0 Developer Beta, we start the music streaming to the CSR 8675. We are able to monitor the Bluetooth codec status under the 'Developer Settings' tab. As shown in Appendix G, with the compatible device, our radio frequency module is able to receive the signal with bit depth of 24-bit. In addition, the Bluetooth sampling rate clocks at 48.0 kHz, which meets up with the maximum data transmission capacity of our design.

In addition, we are also able to switch the codec to AAC, aptX™, and SBC as shown in Appendix G. All of them have a bit depth of 16-bit and sampling rate clocks at 44.1 kHz, except the aptX™ clocks at 48.0 kHz. This also meets up with our codec compatibility of our design.

3.2 Digital-to-Analog Module

3.2.1 Digital-to-Analog Converter

We made the communication to internal DAC inside Bluetooth Receiver through UART serial communication. Putty from the computer can send 24-bit 1kHz pure sine wave directly to internal DAC. For the ease of our usage, Noctis made the firmware so that we just need to type "TONE 24 SINE 1000 100". "TONE" is the general command function to produce the digital audio sound. "24" indicates the bit, which can be any of 8/16/24 bit. "SINE" indicates what type of wave. We can choose from SINE/SQRE/TRIA/SAW. "1000" indicate the frequency and "100" indicates peak-to-peak voltage. We were not able to make the oscilloscope to measure the wave output due to how internal DAC is built, but we were able to successfully hear pure 1kHz. 1kHz is approximately C6 in musical notation.

3.2.2 Amplifier

The gain setting for Class-D amplifier was set to 2V/V (6dB). The capacitors and resistors with the right values are chosen to properly pass 2V/V gain to the amplifier. The verification data are shown in Appendix H. *Figure 17* shows the pure 1kHz sine wave generated by wave generator and inputted to the amplifier. The peak-to-peak voltage is 100mV for the input sine wave. *Figure 18* shows the output from the amplifier, with the pure 1kHz sine wave amplifies to 200mV peak-to-peak. The amplifier is properly amplifying the analog audio signal by 2V/V.

3.2.3 Audio Jack (3.5mm)

Continuing from the amplifier verification, we plugged in the headphone with 32 ohms and hear the resulted pure sine wave. Both left channel and right channel worked with the same amount of gain.

3.3 Power Supply Module

As for the actual testing of how long our adapter will last, we did simple and time-consuming way to measure it. We charged our adapter for more than 10 hours to make the battery fully-charged. Then, we turned on our adapter and connect to the computer through Bluetooth. We plugged 32 ohms headphone and played the music constantly at medium volume setting for first 48 hours straight. After that, we set up the video camera and keep recording it until our adapter turns off. When we noticed that the adapter got turned off, we check the video to verify when it got turned off. From the recording, it showed that the adapter turned off after 7 hours and 34 minutes. Adding this to 48hours that has been already playing, our adapter had been playing music constantly for 55 hours and 34 minutes. The actual battery life was lesser than what we calculated in Equation 4 above. However, it still satisfied our design goal of having the battery life of 40~50hours.

3.3.1 Lithium-ion Battery

We used 2000mAh Li-on battery for our adapter. When it was fully charged, the output voltage was 4.2V. We used 400ohm load open circuit to drain out the battery for 1hour and measured the voltage. The voltage was still 3.7V, indicating that it does have more than 1000mAh.

3.3.2 Voltage Regulator

Voltage regulator MIC5219 was used to supply 3.3V for amplifier and Bluetooth receiver. *Figure 6* below shows the output from the voltage regulator, with the percentage difference within 1%.

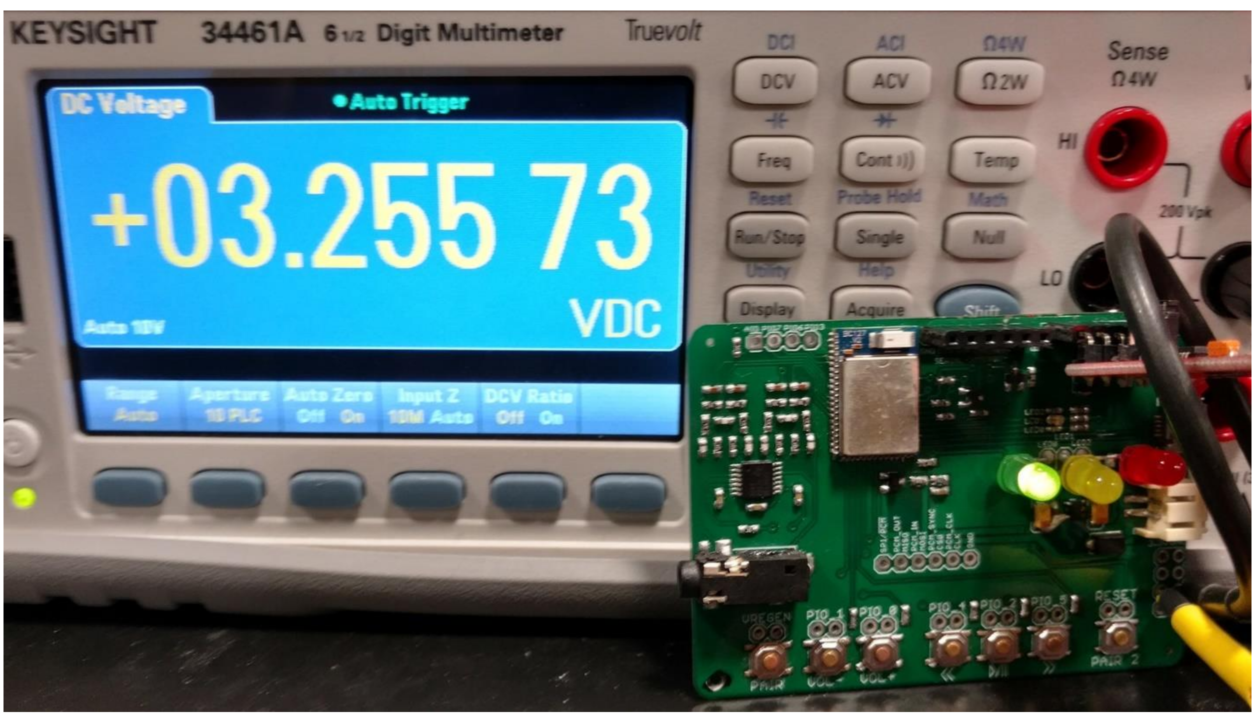

Figure 6 – Voltage Regulator Output

3.4 Control Module

3.4.1 Microprocessor

First of all, UART serial communication in between Bluetooth Receiver and MCU was successfully done. We were able to successfully connect to our adapter from mobile device and computer within the range of 30m. Power management from MCU is working perfectly as the dead 2000mAh Li-on battery got fully charged after 10hrs by showing 4.2V output when measured through voltmeter. All the buttons and status LEDs worked according to how they are programmed on MCU.

3.4.2 Control Buttons

The control buttons had no issue from software side, but we had lots of issue when it comes to hardware side. This button we bought was small and thin that it is very sensitive to the flux. Soldering flux left on the button had been causing lots of issue electrically. We thought our Bluetooth Receiver chip got broken thanks to the broken buttons. We found out that our buttons are the issue and fixed them by replacing with the spare buttons. Six buttons are active-high and only RESET button is active-low.

3.4.3 Status LEDs

Status LEDs were perfectly visible from 3m away. We could even see it from 10m away and status LEDs are working as how we intended it to be. We initially wanted to use 0603 size SMD LED, but we burned our small SMD LEDs while we were soldering so we replaced that with pin-hole LEDs. Yet, it works as how we intended to do so while keeping the size small.

4. Costs

4.1 Parts

Table 2: Parts Costs

Part	Manufacturer	Quantity	Unit Cost ($)	Item Total ($)
Bluetooth Receiver (BC127-HD)	Blue Creation	1	35.09	35.09
Class-D Amplifier (TPA6138A2)	Mouser	1	0.91	0.91
Schottky Diodes 40V 3A (B340A)	Mouser	3	0.43	1.29
Voltage Regulator 3.3V (MIC5219)	Mouser	1	0.96	0.96
n-MOSFET (BSS138TA)	Mouser	2	0.45	0.90
PNP transistor (MMBT4403)	Mouser	1	0.22	0.22
Microprocessor ATmega328P	Mouser	1	2.07	2.07
Micro-B USB female connector	SparkFun	1	0.95	0.95
Mini Push-button Switch	SparkFun	7	0.95	6.65
Audio Jack 3.5mm	SparkFun	1	0.95	0.95
JST Right Angle Connector	SparkFun	1	0.95	0.95
Lithium-ion Battery – 2Ah	SparkFun	1	12.95	12.95
Resistor 0.22 ohm 0603	DigiKey	1	0.02	0.02
Resistor 0.75 ohm 0603	DigiKey	1	0.02	0.02
Resistor 240 ohm 0603	DigiKey	3	0.02	0.06
Resistor 2.2k ohm 0603	DigiKey	5	0.02	0.10
Resistor 10k ohm 0603	DigiKey	8	0.02	0.16
Resistor 15k ohm 0603	DigiKey	4	0.02	0.08
Resistor 30.1k ohm 0603	DigiKey	4	0.02	0.08
Resistor 43k ohm 0603	DigiKey	4	0.02	0.08
Capacitor Ceramic 47pF 0603	DigiKey	4	0.03	0.12
Capacitor Ceramic 180pF 0603	DigiKey	2	0.03	0.06
Capacitor Ceramic 0.1uF 0603	DigiKey	4	0.02	0.08
Capacitor Ceramic 1uF 16V 0603	DigiKey	3	0.05	0.15
Capacitor Ceramic 1uF 25V 0805	DigiKey	1	0.11	0.11
Capacitor Ceramic 4.7uF 0603	DigiKey	1	0.12	0.12
Capacitor Tantalum 10uF 1206	DigiKey	2	0.42	0.84
			Total ($):	**65.97**

4.2 Labor

We expect the development cost to be $40 per hour, 8 hours per week for three people. We assumed that we work about 10 weeks out of the entire semester for this project.

$$3 \times \frac{\$40}{hr} \times \frac{8\ hrs}{week} \times 10\ weeks \times 2.5 = \$24{,}000 \tag{6}$$

5. Conclusion

5.1 Accomplishments

The project was successful as we were able to make all the requirements working with verifications. There were some minor changes made after the design review because our design review was not in a great condition when we submitted. Those changes had drastically made our project better than how we initially designed.

By changing Li-on battery capacity from 1Ah to 2Ah, we were able to reach our goal of making the adapter to run for more than 50hours without any noticeable sacrifice on portability. As we mentioned above, the adapter can run while playing music continuously for approximately 55hours. If we had not changed our design and stick with 1Ah, we would have a battery life of only about 25~30hours. Yet, it would still be outstanding in nowadays commercial market.

The size requirement was also possibly due to removal of any chips that uses 5V supply voltage and the 5V boost converter. By using the internal DAC instead of external DAC as we planned originally in design review, about 2cm x 1cm of PCB has been reduced to meet our goal of making device portable. The adapter would not have a comfortable size for holding if we had both 5V boost converter and external DAC with 5V supply voltage.

The most important goal of our project, Hi-Fi audio quality with 24-bit Bluetooth transmission was successfully implemented. In addition, our product is able to have the Bluetooth sampling rate clocked at 44.1 kHz or at 48.0 kHz depending on the codec selection. As the radio frequency module and the digital-to-analog converter module can handle the Hi-Fi audio signal, we are expected to enjoy the music at Hi-Fi standard. Given the fact that we are only using a Class-D amplifier, our product is not the best to drive any peripherals with extremely high impedances. Yet, the audio quality from the usage of aptX™ HD had pleasantly satisfied our ears. Moreover, as most common peripherals have impedances in a range of 5 ohms to 100 ohms, our device is good enough to drive these peripherals.

5.2 Uncertainties

With all the successes we have with our project, only uncertainty we had was the Bluetooth transmission noise. This was the first thing we noticed when we made everything working and listen to the music through our adapter. Bluetooth transmission noise does not exist when the adapter is not paired with any device. It is evident only when paired up with the device. In the beginning, we had this noise with peak-to-peak voltage of 400~500mV. We thought our amplifier was the issue but it turned out that the pins popped out on the bottom of our PCB is touching the table and causing the amplification of the noise.

We fixed this issue by making a wood case so there is no electrical disturbance to any of our pins on PCB. By doing so, we were able to minimize Bluetooth transmission noise to 200mV as shown in the figures on Appendix I.

It turned out that the fault of having this Bluetooth transmission noise was due to having no separation of analog ground and digital ground [11]. Amplifier and audio jack circuits should have separate analog ground while Bluetooth Receiver and the power supply circuits are connected to digital ground. By doing so, we could have reduced this noise much more.

5.3 Ethical considerations

In our project, one of the most important safety concern is the usage of lithium-ion battery. The lithium-ion battery contains both cathode and fuel in a container. These two parts can react and cause fire or explode. An explosion can also be caused by overcharging or overheating the battery. Inbuilt temperature sensor circuit on lithium-ion battery will insure that the battery stays within 0 °C to 45 °C during charging and disconnect the battery from the entire circuit if the temperature is abnormal to avoid further chemical reaction and potential fire hazard or explosion. Power management circuit controlled by control module will stop charging the battery if the voltage is over 4.2V.

In the beginning of our project, we made a statement to not make false claims about our project, according to the IEEE Code of Ethics, number 3: “to be honest and realistic in stating claims or estimates based on available data” [12]. We made sure that we provide 24-bit audio data through Bluetooth connection and user can easily verify this on the android developer mode.

5.4 Future works

We strongly believe that when it comes to develop a product, the development should take a “tick-tock” cycle. As we have successfully designed and produced the base of our product, and it is market-ready. The next thing we need to do is the optimization and market adaptation for varieties of customer groups. The further we investigate into the possibility of our project, there are eight certain aspect we can do further development on.

First, the security is a top-concerned point when it comes to choosing a product. Customers would like the data transmission to be as secure as possible. The possible solution which may work out the best would be create a mobile application which will encrypt the audio streaming packet by adding extra mask bits or thoroughly encrypt the data packet with a certain dictionary key. Then the product can decrypt the signal through the on-board micro-processor unit and pass the decrypted data packet into the digital-to-audio module. This is important if the company decides to add calling functionality into the product so that it will make it hard for other people to attack and obtain your personal information.

Second, we would like to concentrate on the shielding for the product. Due to the fact that the radio frequency module is really sensitive to the change of the electrical and magnetic field changes, any unintentional changes may introduce a large amount noise signal during the audio playback. So, we would like to suggest developers to design a proper shielding layer to not only the Bluetooth chipset, but also the rest of the on-board unit. This will help to eliminate the electro-magnetic interferences and the electro-magnetic discharges.

Third, we need to understand the importance of the user control interface. The user control interface plays an important role in the user's judgement of the quality of the product. As for our product, we included simple on-board buttons and LEDs for the user to understand the current operating state of the machine and to control the product directly. However, these are not the best-in-class solution for our product. As our product is designed with the consideration of modulization, we have tried to execute a solution with simple ink-screen to display the music meta data, volume level, and the battery percentage. It is doable with the micro-control unit. However, developers may be able to come up with a more elegant and user-friendly solution for the product.

Fourth, we would like to talk about the ground separation. Currently, as we have mentioned in the section 5.2, we conducted a Bluetooth noise analysis on our product. We found out that the major noise is from the Bluetooth transmission. This can be voided by separating the grounds for the analog signal and digital signal components. Therefore, developers will be able to lower down the static noise during the playback.

Fifth, we want to address the possibility of modular design. Modulus design is a new rising design language. There are several advantages when it comes to modular hardware design. First, it can dramatically reduce the process system of building the product. Since it can shorten the hardware debugging time. Moreover, it can reduce the costs of building the product. The building project timeline will be shortened so that the labor and operational costs will be reduced. Moreover, when it comes to quality control, if any part of the product fails, it is easy to be replaced without the waste of the entire product unit. In addition, it is easy for users to discover the possibility of the product, which makes the product self-serviceable. Users can simply alter the chipset selection. Hence, it will make it easy for the product to cater people with different needs.

Sixth, we would like to talk about the on-board storage. Even though our product is heavily relied on the Bluetooth RF module, our product is still able to process the audio data from wired connection or pin connections. It is possible to add SD card module or enable the micro-USB data transmission. The micro-processor unit can fetch the data and pass it to the digital-to-analog converter module on the board. Hence, product can be useful with the absence of the Bluetooth output peripherals, or use it for other purposes, such as the sound card replacement for computers or mobile devices which may not have an audio jack or Bluetooth module.

Seventh, we strongly believe that the size of our product can be further trimmed for extra portability. If we look close to our PCB layout, you may find out that we have extra ports allocated for debugging purposes. It is safe to remove these extra ports without the interruption of the functionality.

Lastly, we think it is easy and preferable to have a radio frequency protocol upgrade. The CSR 8675 chipset supports Bluetooth 5.0 transmission standard [13]. With the proper software update, it is possible to enable the Bluetooth 5.0 communication with compatible devices. This will bring down the power consumption for the wireless communication as well as the range of communication [14].

References

[1] X. Blanco, “Spotify, Apple Music, Tidal, Amazon Music Unlimited and Google Play Music: Which music streaming app is right for you?”, *CNET*, 21-Oct-2016. [Online]. Available: https://www.cnet.com/how-to/best-music-streaming-service/. [Accessed: 12-Dec-2017].

[2] “iTrip Clip Bluetooth Headphone Adapter”, *Griffin Technology*, 2017. [Online]. Available: https://griffintechnology.com/us/itrip-clip-bluetooth-headphone-adapter. [Accessed: 12-Dec-2017].

[3] *LG V20™: Superior Video, Photography, & Next-Level Audio | LG USA*, 2016. [Online]. Available: http://www.lg.com/us/mobile-phones/v20. [Accessed: 12-Dec-2017].

[4] M. Shoemake, "Wi-Fi (IEEE 802.11b) and Bluetooth Coexistence Issues and Solutions for the 2.4 GHz ISM Band", *Texas Instruments*, 2001. [Online]. Available: http://focus.ti.com/pdfs/vf/bband/coexistence.pdf. [Accessed: 12- Dec- 2017].

[5] "What does the A2DP Profile mean?”, *Motorola*, 2017. [Online]. Available: https://motorola-global-en-uk.custhelp.com/app/answers/detail/a_id/5579. [Accessed: 12- Dec- 2017].

[6] "Traditional Profile Specifications | Bluetooth Technology Website", *Bluetooth*, 2017. [Online]. Available: https://www.bluetooth.com/specifications/profiles-overview. [Accessed: 12- Dec- 2017].

[7] "16 Bit vs. 24 Bit Audio", *Tweakheadz*, 2017. [Online]. Available: http://tweakheadz.com/16-bit-vs-24-bit-audio/. [Accessed: 12- Dec- 2017].

[8] "Qualcomm® aptX™ HD Audio", *Qualcomm® aptX™*, 2017. [Online]. Available: https://www.aptx.com/aptX™ HD. [Accessed: 12- Dec- 2017].

[9] "Sound Quality Comparison of Hi-Res Audio vs. CD vs. MP3 | Sony US", *Sony US*, 2017. [Online]. Available: https://www.sony.com/electronics/hi-res-audio-mp3-cd-sound-quality-comparison. [Accessed: 12- Dec- 2017].

[10] “Amplifier Classes and the Classification of Amplifiers”, *Electronics Tutorials*, 2017. [Online]. Available: http://www.electronics-tutorials.ws/amplifier/amplifier-classes.html. [Accessed: 12-Dec-2017].

[11] H. Zumbahlen, “Staying Well Grounded”, *Staying Well Grounded | Analog Devices*, Jun-2012. [Online]. Available: http://www.analog.com/en/analog-dialogue/articles/staying-well-grounded.html. [Accessed: 12-Dec-2017].

[12] IEEE.org, "IEEE Code of Ethics", *IEEE – Advancing Technology for Humanity*, 2017. [Online]. Available: http://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed: 12- Dec- 2017].

[13] "CSR8675 | Qualcomm", *Qualcomm*, 2017. [Online]. Available: https://www.qualcomm.com/products/csr8675. [Accessed: 12- Dec- 2017].

[14] "Bluetooth 5 | Bluetooth Technology Website", *Bluetooth*, 2017. [Online]. Available: https://www.bluetooth.com/specifications/bluetooth-core-specification/bluetooth5. [Accessed: 12- Dec- 2017].

Appendix A: Requirement and Verification Table

Table 3: Bluetooth Module - Requirements and Verifications

Requirement	Verification	Verification status (Y or N)
1. Bluetooth Receiver must successfully pair up via Bluetooth 4.1 protocol or above.	A) Connect to Windows laptop/LG V20/ Google Pixel XL, under the device manager (either via system report, or ADB over terminal). Check the LMP version. See if the value is 7 or 0x7 or above.	Y
2. Bluetooth Receiver must successfully transmit audio signal under aptX™ HD/AAC codec.	A) Under the Windows environment, launch Bluetooth Driver Stack provided by Microsoft, then pair up devices. Manually toggle in between AAC and aptX™ HD and see if it will be okay to execute. B) Under Ubuntu or Debian environment, install the package bluez-hcidump, then run the command as sudo hcidump avdtp, and check the codec report. C) Under Android environment, with some devices that aptX™ HD enabled by default, once the device is paired up, there will be popped up notification indicating that "aptX™ HD device is connected". Then toggle the developer option, under the Networking→Bluetooth AVRCP, make sure the version is above 1.4, and check if able to toggle and register aptX™ HD and AAC under Networking→Bluetooth Audio codec.	Y
3. Bluetooth Receiver must successfully receive 24-bit signal with a sampling rate higher than 44.1kHz.	A) Make a test audio sample file which is 24-bit coded with a sampling rate at 44.1 kHz. B) Under Windows environment, while streaming the sample file, check under the Bluetooth Driver Stack and check if the connection is under 24-bit mode and the maximum sampling rate is above 44.1kHz for the connection. C) Under Linux environment, check the AVDTP report while streaming, see if the connection is under 24-bit and Bluetooth maximum sampling rate is above 44.1 kHz. D) Under the Android environment, either check via ADB mode or under developer options, see if the Bluetooth codec is logging as 24-bit while receiving the audio, and the Bluetooth maximum sampling rate is above 44.1 kHz.	Y

Table 4: Digital-to-Analog Module - Requirements and Verifications

Requirement	Verification	Verification status (Y or N)
1. DAC must successfully receive 24-bit audio data and output through both left and right channel.	A) Input a digital 24-bit pure 1kHz sine wave to DAC through UART connection from Putty. B) Use oscilloscope at the output of DAC to verify the pure 1kHz sine wave. C) Verify this can be separated into both left and right channel properly by sending the digital 24-bit sine wave to left channel first, then right channel next.	Y
2. Amplifier must successfully amplify audio signal with a gain setting of 2V/V (6dB) for each channel.	A) Input a pure 1kHz sine wave with 100mV peak-to-peak voltage to amplifier using wave generator. B) Use oscilloscope at the output of amplifier to verify that peak-to-peak voltage has been amplified to 200mV.	Y
3. 3.5mm audio jack must successfully output the amplified audio signal to both right and left channel of headphone/speaker.	A) Input a pure 1kHz sine wave with 100mV peak-to-peak voltage to audio jack using wave generator. B) Plug headphone to audio jack and listen to 1kHz sine wave. C) Verify that each channel work by sending into left channel first, then right channel next.	Y

Table 5: Power Supply Module - Requirements and Verifications

Requirement	Verification	Verification status (Y or N)
1. Li-on battery must successfully provide 3.7V supply voltage to voltage regulator. 2. Li-on battery must have a capacity of 1000mAh or more.	1) A) Measure the fully charged battery with voltmeter, ensure that the output is 3.7V or above. 2) A) Discharge the fully charged Li-on battery with 400ohm resistor for 1 hour. B) Measure the voltage of Li-on battery to see if it is still above 2.7V.	Y
3. Voltage regulator must successfully convert 3.7V from Li-on battery and output 3.3V.	A) Measure the output voltage from LDO using voltmeter. Ensure that the output voltage is within 5% of 3.3V.	Y

Table 6: Control Module - Requirements and Verifications

Requirement	Verification	Verification status (Y or N)
1. MCU must successfully activate Bluetooth pairing between the device and Bluetooth Receiver. 2. MCU must successfully configure the button setting through GPIO control. 3. MCU must successfully control the status LEDs.	1. A) Connect MCU to Bluetooth Receiver through UART serial connection. B) Using Putty to monitor the serial communication is done properly. C) MCU sends PAIR command to Bluetooth Receiver when the device request for Bluetooth pairing. D) Verify Bluetooth Receiver is connected through Bluetooth to the device using Putty. 2. A) Connect button signals to digital pin of MCU. B) Detect the voltage input to digital pin when the button is pressed. C) Depends on which button is connected to which digital pin, different functions will occur. D) Major functions are: Power ON/OFF, Volume UP/DOWN, move to PREVIOUS/NEXT song, and PAUSE/PLAY. 3. A) Connect status LEDs to analog pin of MCU. B) Pull-down to light up the status LEDs. Stay pull-up to light off. C) Verify the functions on software is working as accordingly.	Y
1. Buttons must successfully activate low/high signal and feed into GPIO control.	A) Connect one end to ground (not pressed) and one end to supply voltage (pressed). B) Verify the voltage is ground when not pressed using voltmeter. C) Verify the voltage is equal to supply voltage when pressed using voltmeter.	Y
1. Status LEDs must successfully light up to be visible from at least 3m away.	A) Place LEDs with 2.2k ohm resistors to supply voltage (pull-up). B) Observe LEDs at least 3m away.	Y

Appendix B: Printed Circuit Board Layout

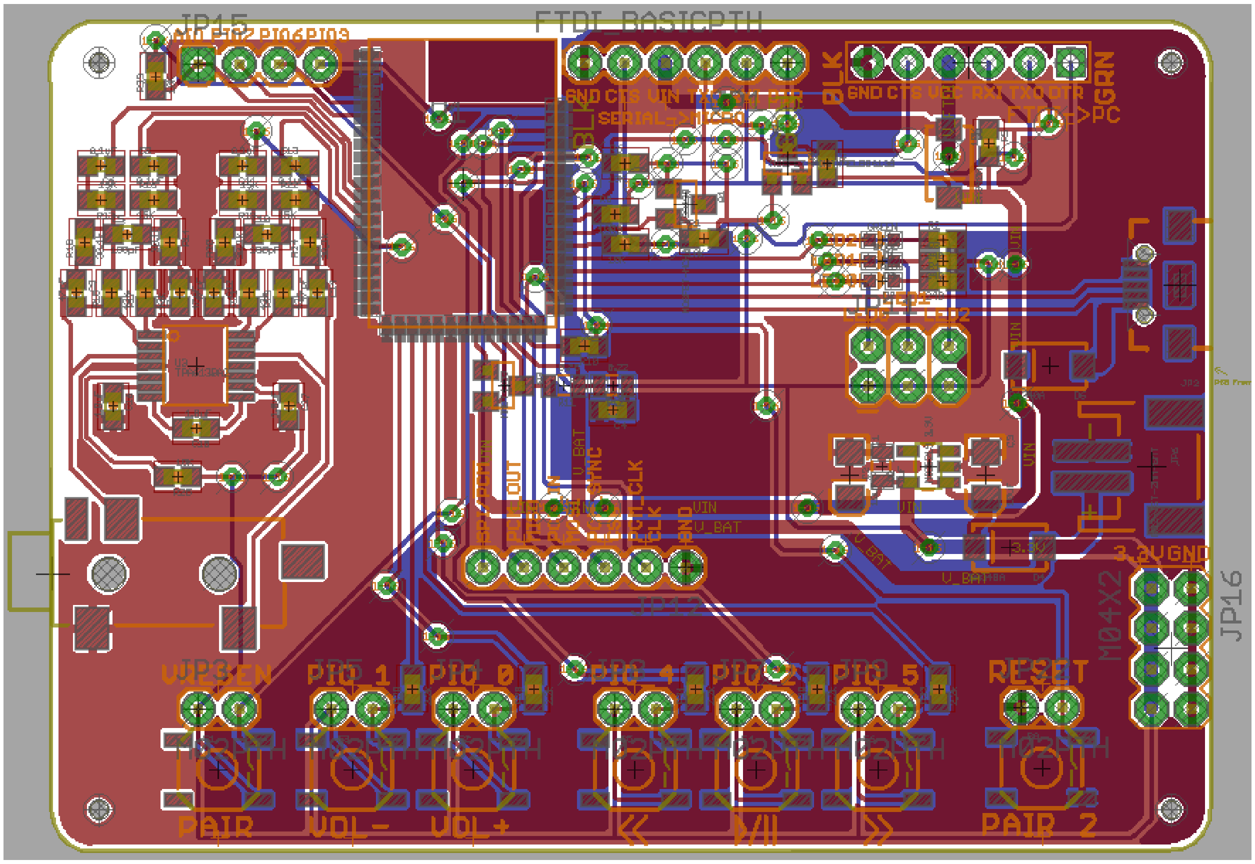

Figure 7 – Printed Circuit Board Layout

Appendix C: Final Product

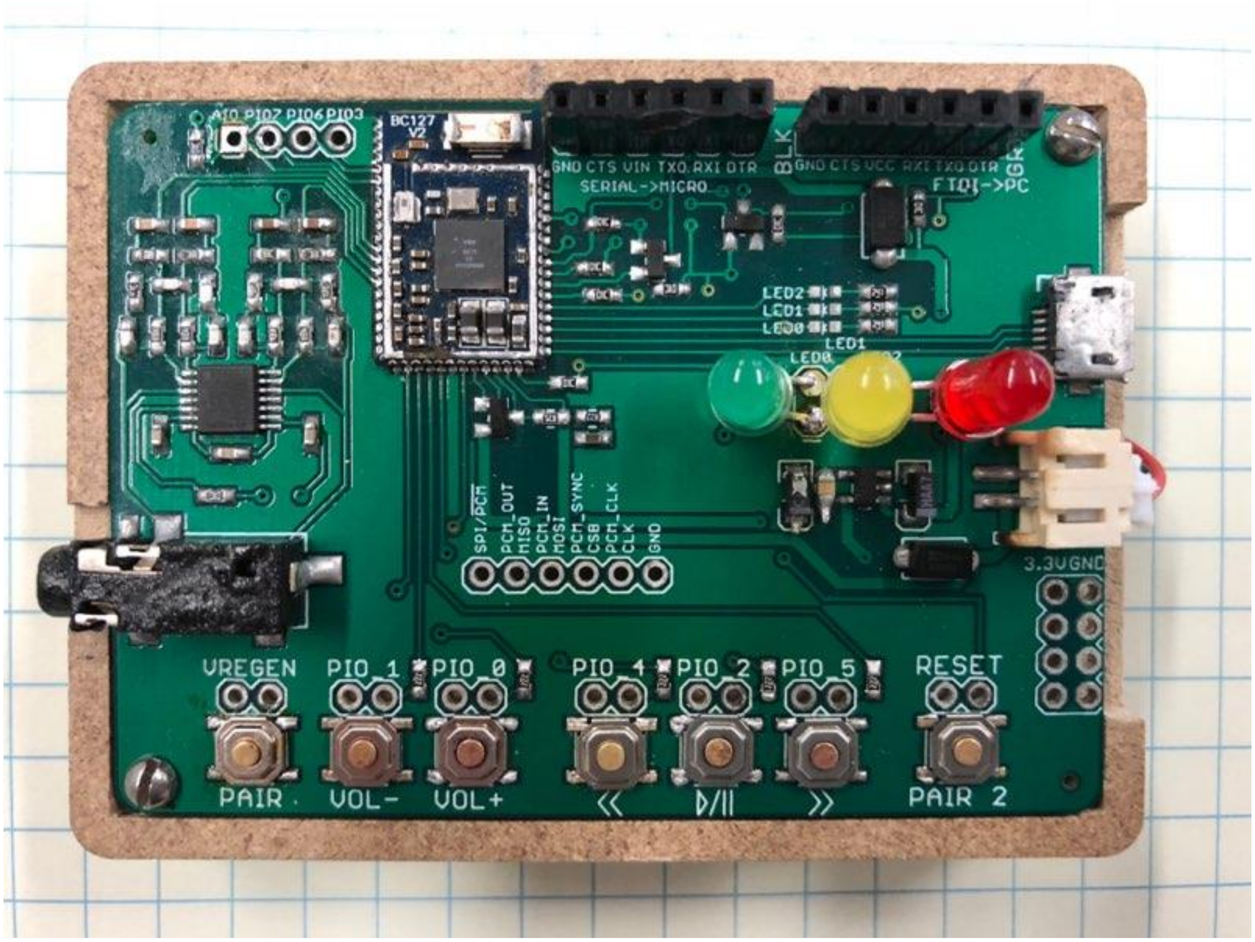

Figure 8 – Final Product Image

Table 7: Width, Height, and Length of Device

Width (cm)	Length (cm)	Height(cm)
7.6	5.7	1.5

Appendix D: Bluetooth Module Schematic

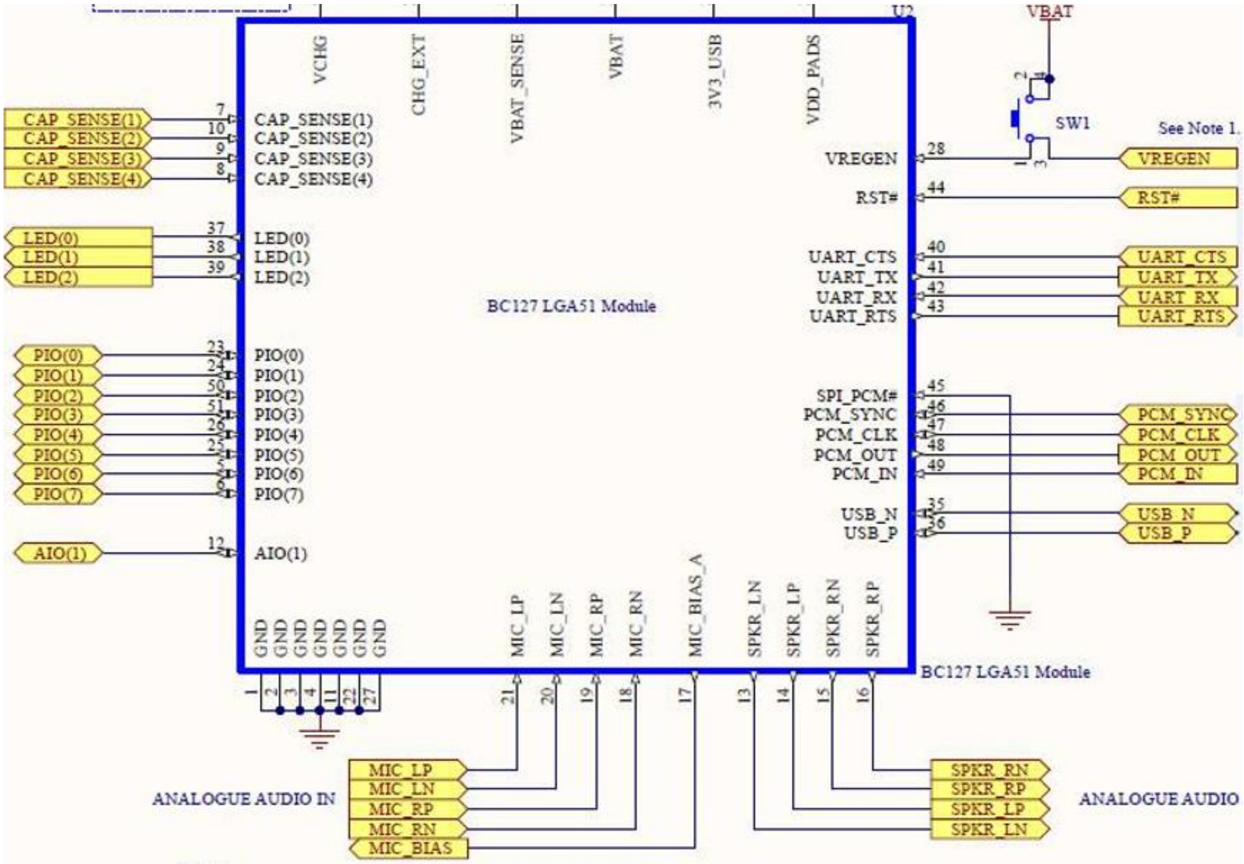

Figure 9 – Bluetooth Module Schematic

Appendix E: Power Supply Module Schematic

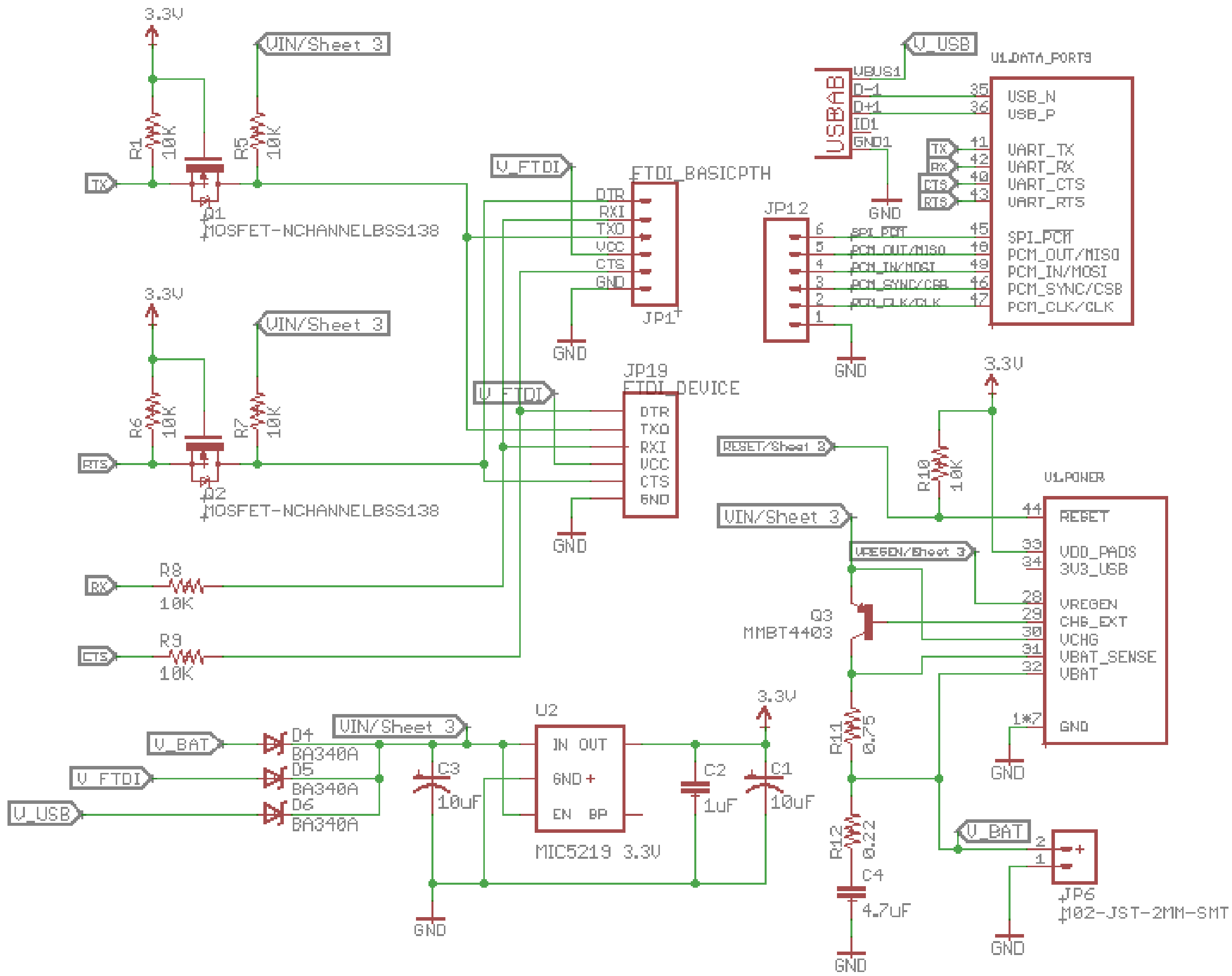

Figure 10 – Power Supply Schematic

Top-left: UART serial connection

Top-right: micro USB-B port

Bottom-left: Voltage Regulator 3.3V

Bottom-right: Power Management Circuit

Appendix F: Microprocessor Schematic

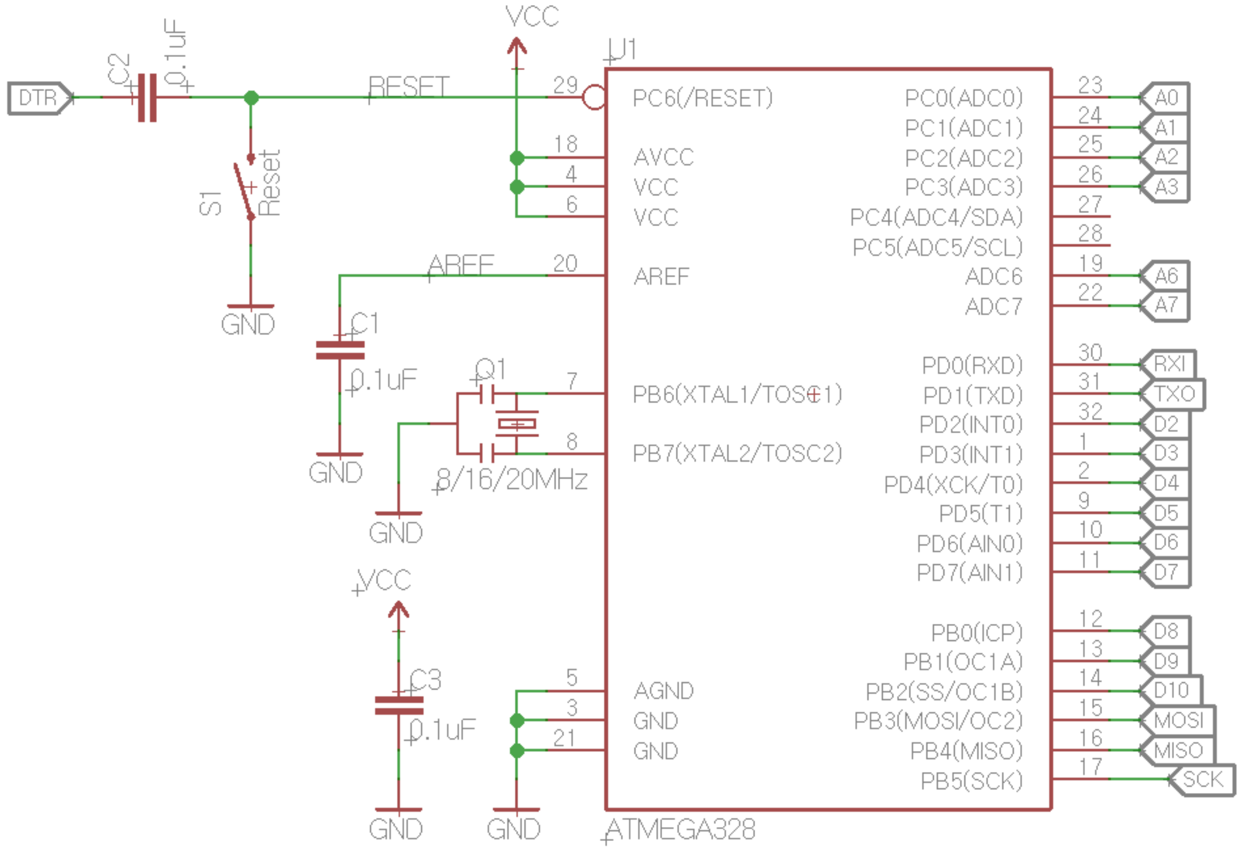

Figure 11 – Microprocessor Schematic

Appendix G: Bluetooth Verification

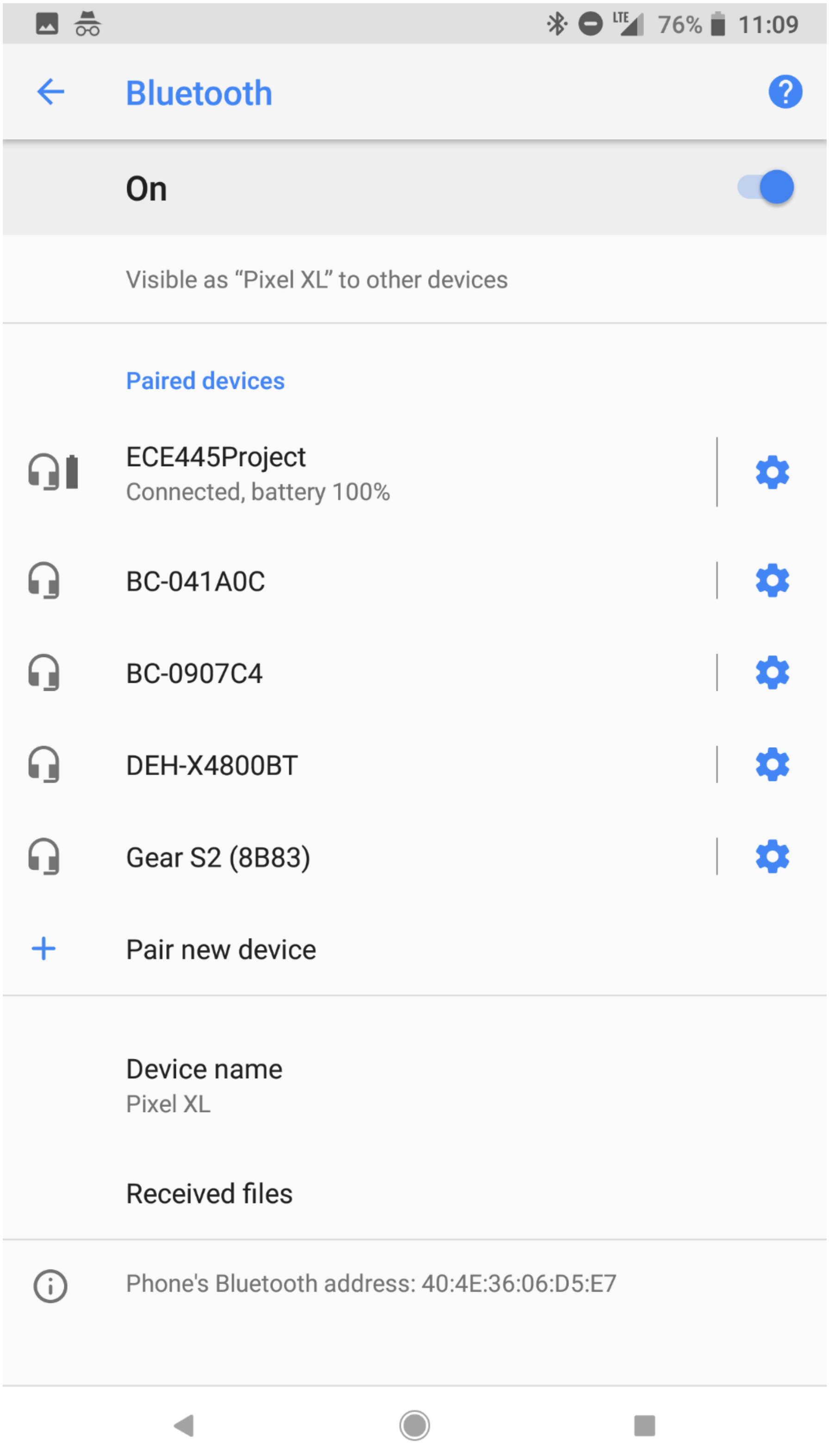

Figure 12 – Bluetooth Connection through Mobile device

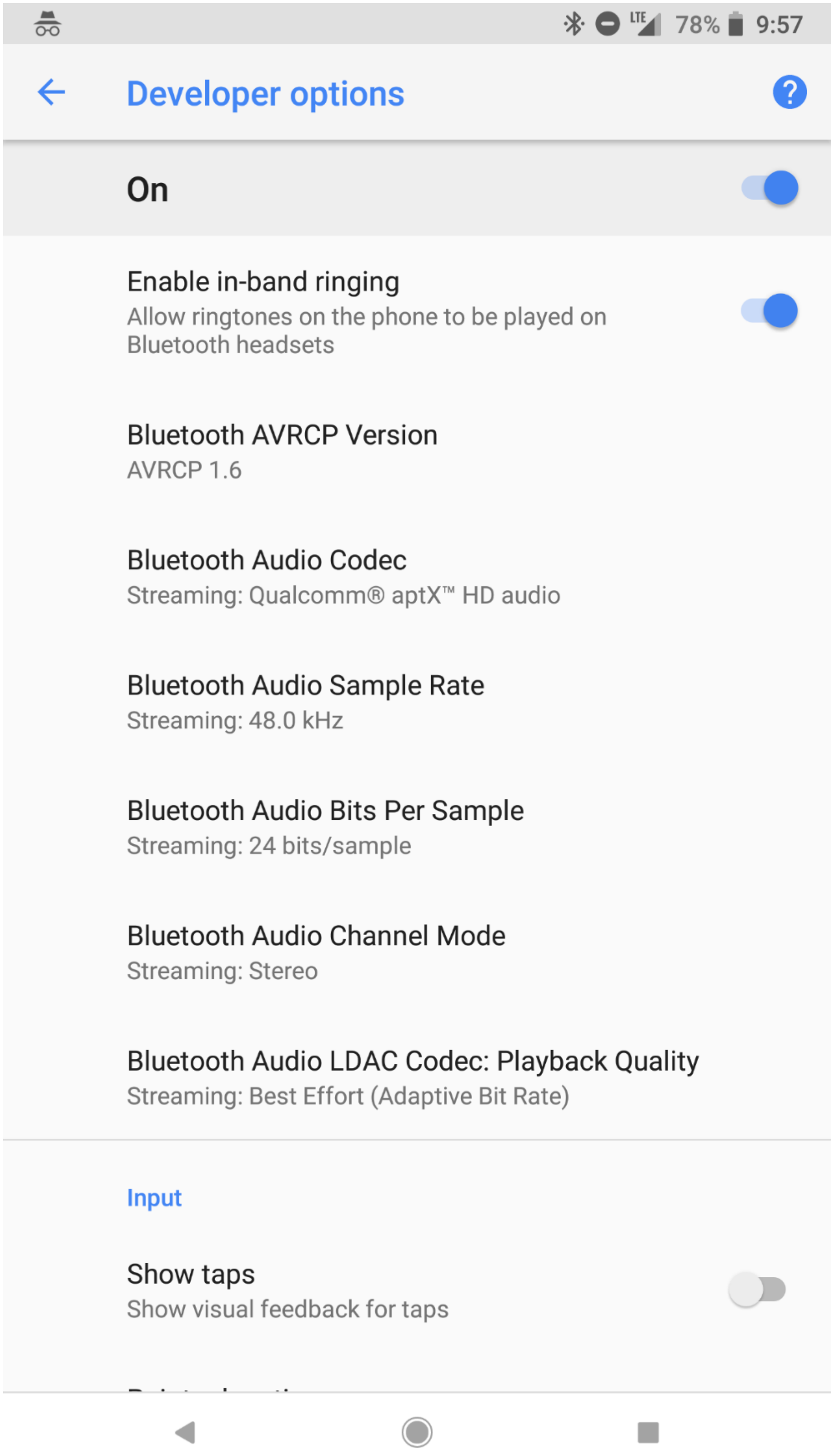

Figure 13 – Bluetooth Verification through Android Developer Mode (24-bit aptX™ HD)

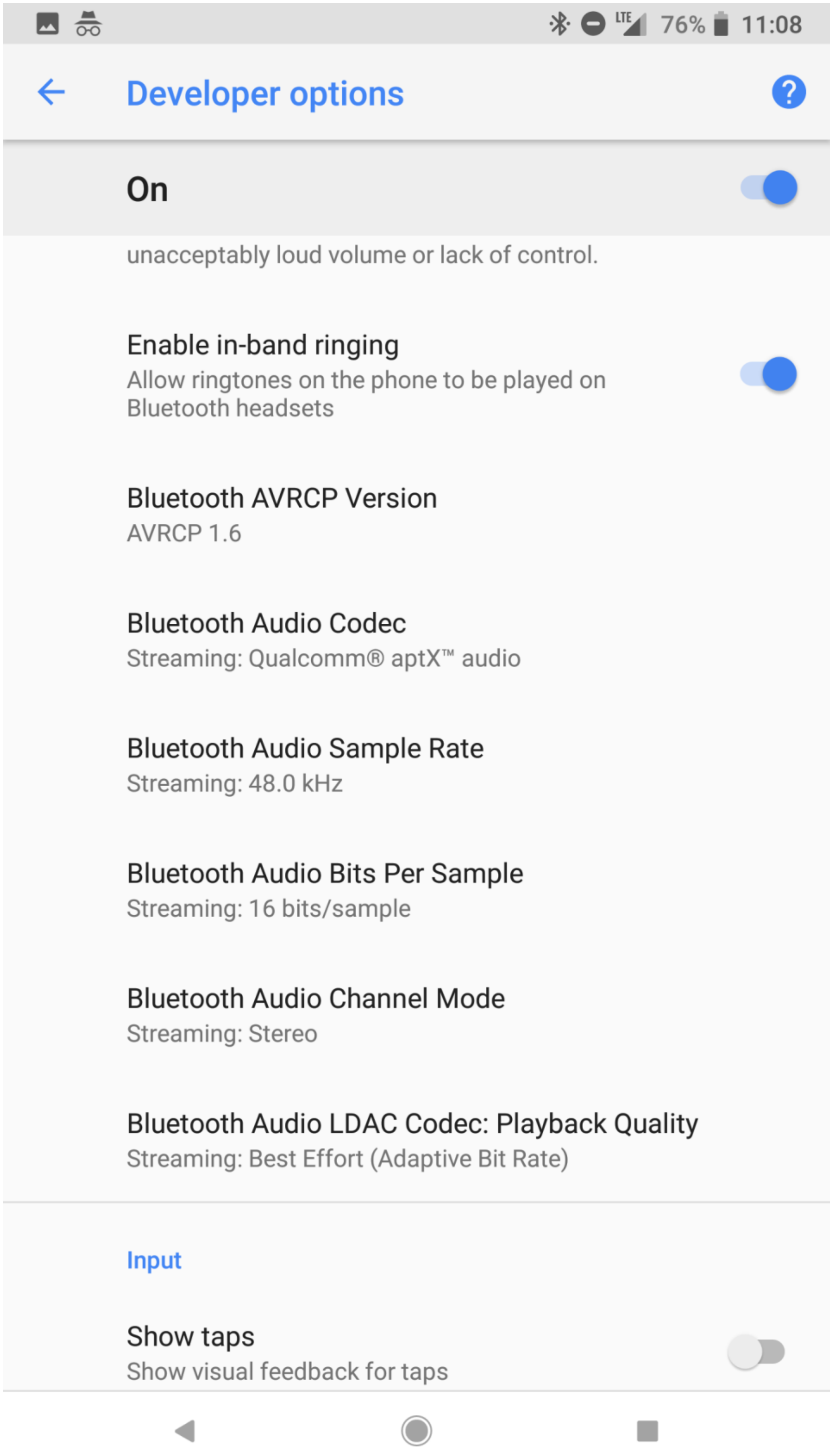

Figure 14 – Bluetooth Verification through Android Developer Mode (16-bit aptX)

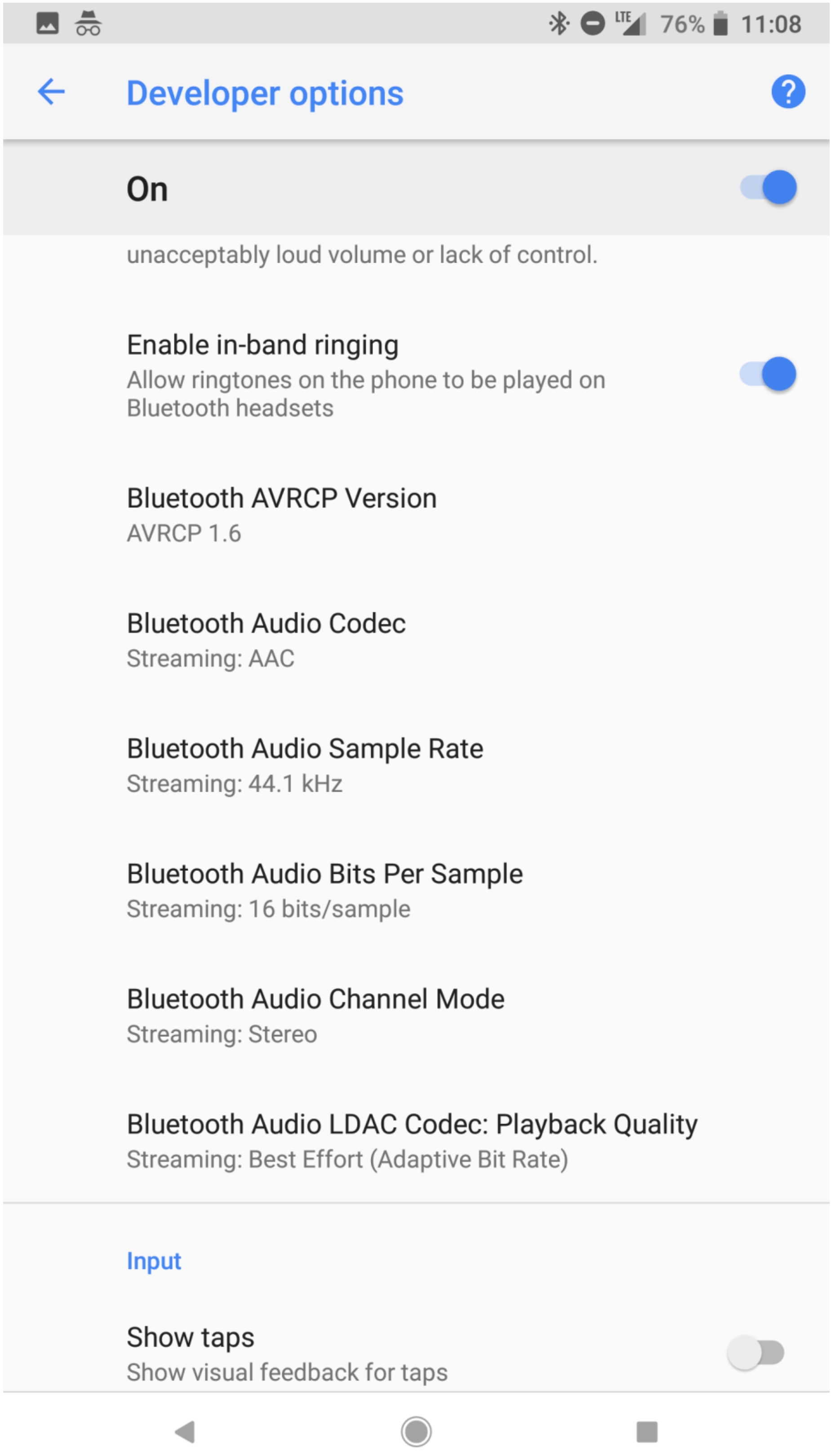

Figure 15 – Bluetooth Verification through Android Developer Mode (16-bit AAC)

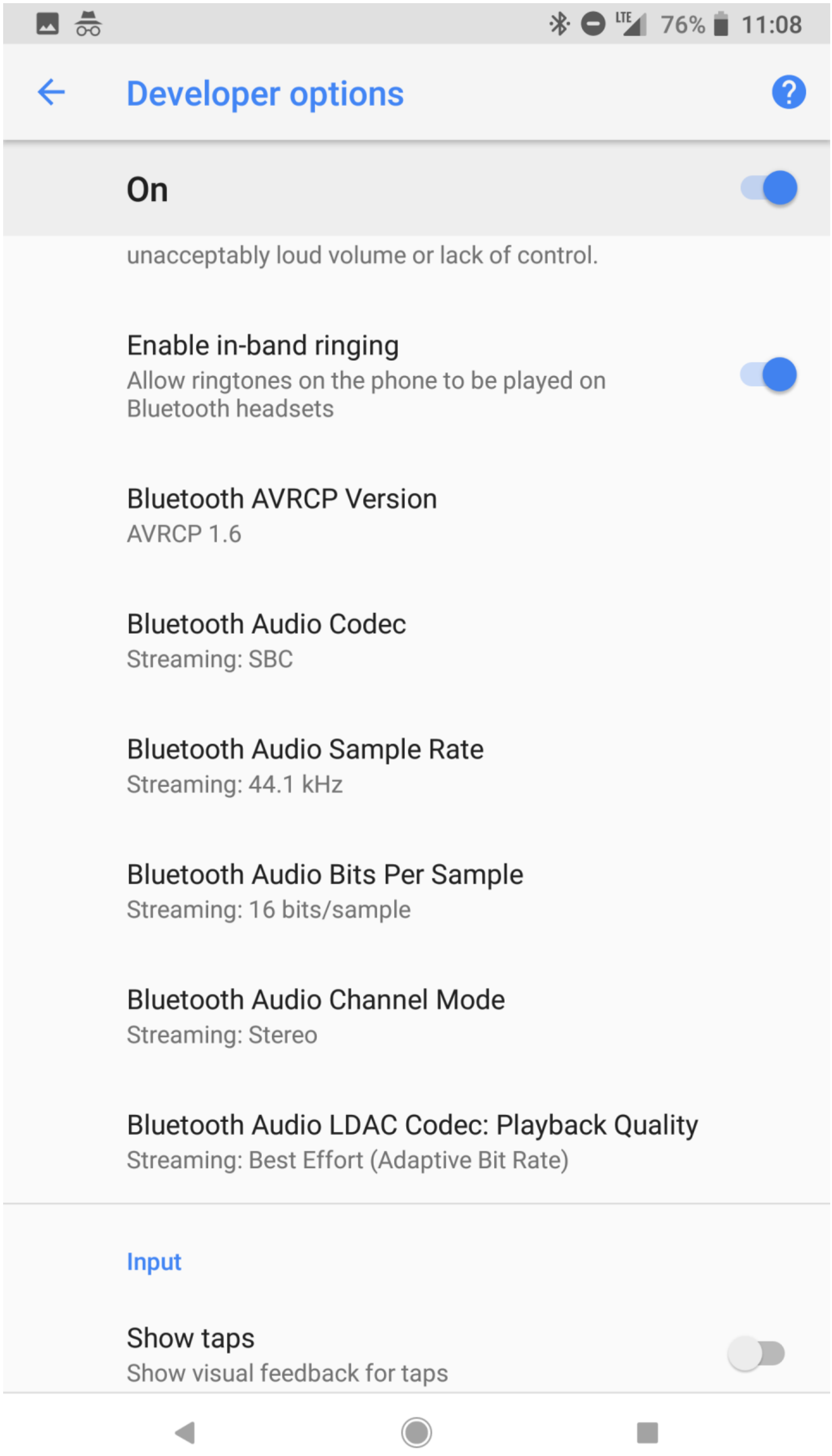

Figure 16 – Bluetooth Verification through Android Developer Mode (16-bit SBC)

Appendix H: Class-D Amplifier Verification

Figure 17 – Pure 1kHz Sine Wave Before Amplifier

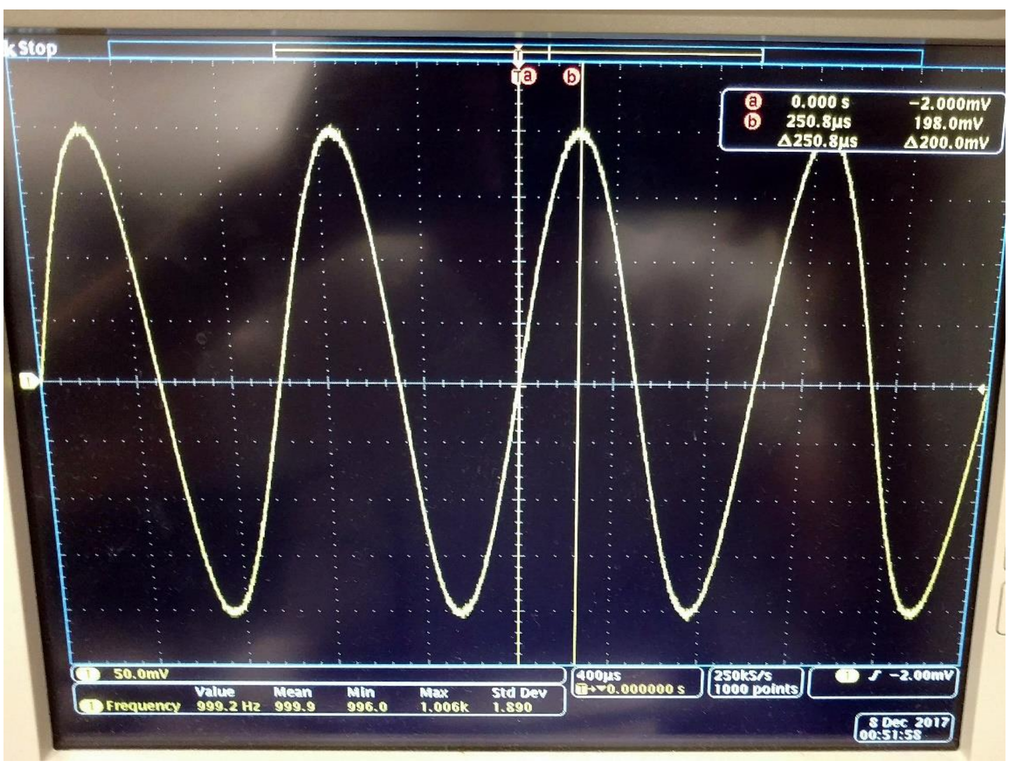

Figure 18 – Pure 1kHz Sine Wave After Amplifier

Appendix I: Bluetooth Transmission Noise Analysis

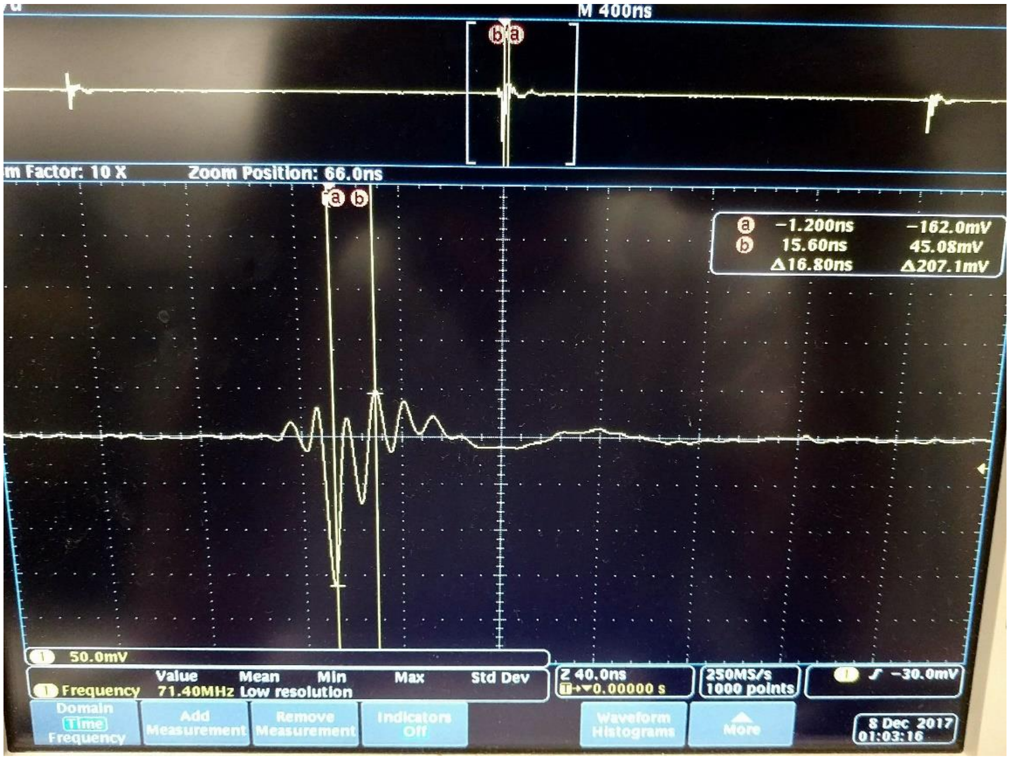

Figure 19 – Bluetooth Transmission Noise Peak-to-peak Voltage

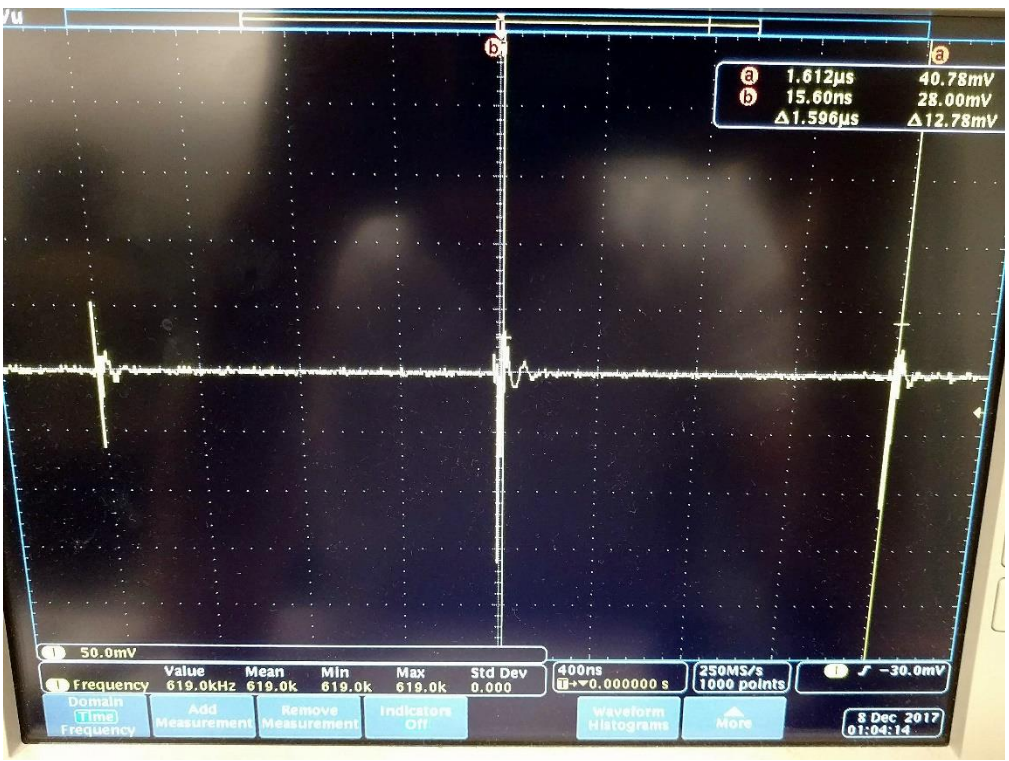

Figure 20 – Bluetooth Transmission Noise Frequency