DRUM SMART

By

Collin Haney

John Miller

Kevin Kovathana

Final Report for ECE 445, Senior Design, Spring 2020

TA: Ruhao Xia

8 May 2020

Project No. 56

Abstract

A problem in America is the gap in possibilities among different communities. We planned to partner with the Hip Hop Xpress to help shrink this divide by educating people on music and technology to show them their possible opportunities. Our original solution attempted to solve this problem by showing kids and adults a simplified version of a DJ drum board. This drum board would allow them to make personalized beats to combine them in a constantly repeating loop. Our new solution also addresses the problem; this device is an electronic marching band drum that behaves similar to a regular drum. However, this drum alone is capable of sounding like several other types of drums and offers the user some pitch variation in these sounds. Our first solution was more digital and less instrument themed whereas our new solution is less digital and more instrument themed.

Contents

1. Second Project Motivation

Our team sought to bridge the opportunity gap among American communities by educating people on music and technology. Our original solution, Beat Starter, was a simple DJ drum board that allowed the user or users to create beats that would continuously play in a loop. Due to the current circumstances, we were unable to finish Beat Starter but instead came up with another solution for this problem. Our new solution, Drum Smart, is an electronic marching band drum that behaves similar to a regular drum. This drum alone is capable of sounding like several other types of drums and offers the user some pitch variation in these sounds. By making an electronic drum, we are still educating people on technology and music.

1.1 Updated Problem Statement

An issue in today's society deals with the amount of opportunities each community throughout America has to offer. For instance, one area might have thousands of available options and possibilities whereas another town could have only a small fraction of the previous community's opportunities. To reduce this difference in opportunity, the Hip Hop Xpress is a bus that will travel to areas and neighborhoods throughout the US to educate people on both music and technology to demonstrate the resources that are available outside of their area [1]. We want to bring each visited community together in an educational way. By adding an interactive device that people could pick up and use, we would have these people physically working with some of the resources their environment does not currently offer.

For many people in America today, they are immensely impacted by their surrounding environment. For example, some individuals have a lot fewer options than others purely due to their communities. Some areas do not present all the possibilities or opportunities that exist around the world. If we are able to demonstrate something that they find very cool or interesting, we have a chance at changing those individuals' futures, hopefully for the better.

1.2 Updated Solution

To meet our objective, we plan to transform a regular drum, given to us from the Hip Hop Xpress, into an electronic drum capable of sounding like several other types of drums. Just like a regular drum, the user strikes our drum with drumsticks to create a sound. The drum pad consists of a layer of rubber to absorb most of the noise coming from the physical striking of the pad to minimize the sound coming from the strikes. Since our drum can sound like multiple kinds of drums, we need the created drum sounds to be output through a connected speaker by an AUX cable. Our electronic drum will be similar to an actual snare drum in the aspect of force-to-sound ratio; the harder the strike, the louder the sound. To select the type of drum the user desires, we will implement a control knob the user can twist to select the type of sound the drum will make when the pad is struck. We also plan to allow the user to control the pitch of every selected drum sound by flipping a switch to either raise or lower the pitch to add more variation to their sounds.

To make our drum portable, all of our electrical components will be powered by a rechargeable battery. With the help of Hip Hop Xpress, our idea presents some simple yet cultivating aspects of music and technology to people that have never seen or even knew of technology like this existing. We want to spark interest in musical and technological industries.

The use of technology in music today is very prevalent. Besides gifting musicians with the ability to record their music, technology is also used to develop many of their skills such as rhythmic accuracy, memorization, and expression [2]. Since technology has become such a large part of music, we knew we had to find a creative way to utilize technology with an instrument to achieve our goal of teaching people about both music and technology. However, Drum Smart is not the first solution for addressing our objective. Some techniques that try to solve our issue involve incorporating engagement, utilizing music the user enjoys, and inspiring creativity [3]. We believe Drum Smart demonstrates all the previous methods in one device. Since the user is physically playing the drum, we are having the user engage with our device. The user can play any music they desire, so they have the option to play music they enjoy or not, the choice is theirs. Drum Smart inspires creativity because there is no electronic drum that allows the user to switch the sound it emits. The most similar product that resembles Drum Smart would be the Rock Band drum set. However, Drum Smart has only one pad to be struck whereas the Rock Band drum set has four separate pads along with a pedal [4]. Our device is also able to be used without having to play a video game, however, the drum set from Rock Band only functions when the game is running. This limits the user to playing only the songs on the game whereas Drum Smart allows the user to play anything they desire. Since our device incorporates both technology and music, we hope that someone visited by the Hip Hop Xpress finds their true passion from interacting with the resources that were not previously offered to them.

Our previous idea, Beat Starter, was a simple DJ drum board capable of being used by children, teens, and adults. The board offered a variety of sounds to select from and combine in a loop to create personal beats. To receive user input, the face of the board would have buttons and control knobs. The buttons were for the different instrument sounds, power, and clear function. The turn knobs would adjust volume and BPM. The user’s created beats would run in a loop in memory. Beat Starter would also have a simple LCD display panel to indicate the current settings provided by the microcontroller. These settings include volume level and beats per minute (BPM) count. Our new idea, Drum Smart, is not a board at all but a marching band drum. Instead of a variety of different instrument sounds, our drum would provide different types of drums and their sounds. Instead of pressing buttons to create sounds, the user strikes the drum pad to create the sound. Instead of a knob to control volume, we are now having the volume controlled by the amount of force the user hits the drum pad. Beat Starter needed a display for the user to see the volume level and BPM. Now, we no longer have a need for a display because we will not be using BPM and will not need to show the volume level of each strike. Beat Starter also kept the created beats repeatedly playing in a loop. Our new device will not play any beats in a loop but rather emit that certain sound once per strike.

1.3 Updated High-Level Requirements

- ❖ One drum sound is emitted for every strike on the drum pad. We do not want two sounds outputted when the pad is only struck once. We want our electronic drum to behave like a real drum. Since a real drum does not make multiple sounds from one strike, our device should not either. We also want our drum to output louder sounds when the drum is struck with more force; the same behavior of a real drum.
- ❖ The user has the choice to pick between four different types of drums. The types of drums we wish to include are snare, tenor, bass, and cymbals. These are the main instruments that makeup a marching band percussion section giving the user a variety of drums to pick from which increases the chances the user finds something they enjoy.
- ❖ All of the components in our device are powered by only one rechargeable battery that does not need to be removed from the device to be recharged. We do not want the user needing to take apart the device just to recharge the battery. The single battery is able to power everything for at least four hours when the drum is under continuous use without needing a recharge.

1.4 Updated Visual Aid

Figure 1 shows the physical design of our drum. Our electronic drum looks very similar to a marching band drum. Just like a regular drum, the user hits the drum with drumsticks to create a sound. Our drum pad consists of a layer of rubber to absorb most of the noise coming from the physical striking of the pad to minimize the sound coming from the strikes. Since our drum can sound like multiple kinds of drums, we need the created drum sounds to be output through a connected speaker through an AUX cable. On the drum shell, there will be a control knob for drum type selection and a switch for pitch level. Additionally on the shell, there will be an AUX port to connect an AUX cable to play the created sounds through a speaker and a Micro-USB port to recharge the battery.

1.5 Updated Block Diagram

For an overview on our subsystems and components necessary for this drum, Figure 2 presents a block diagram that shows the general interconnections. A rechargeable battery will power our device. The user will be able to control volume by the accelerometer, type of drum by a control knob, and pitch level by a switch. The data collected by the accelerometer, knob, and switch will be sent to the microcontroller. Our microcontroller will access memory on the SD card based on the user input data to pull the correct sound from memory. The microcontroller will send this data from memory to the Sound Card to improve the sound quality. This improved sound is lastly outputted through AUX.

1.6 Figures

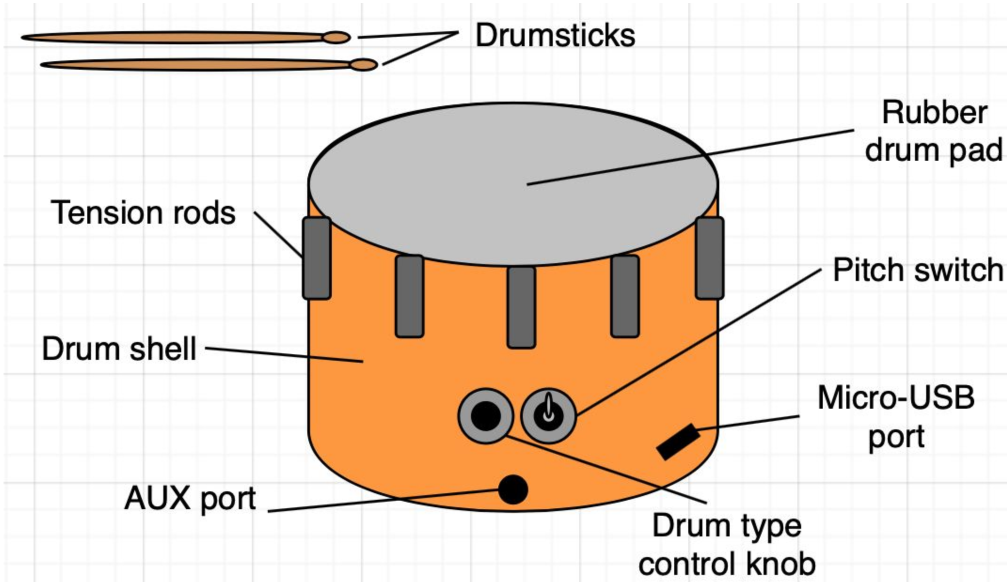

Figure 1 Drum Smart Physical Design

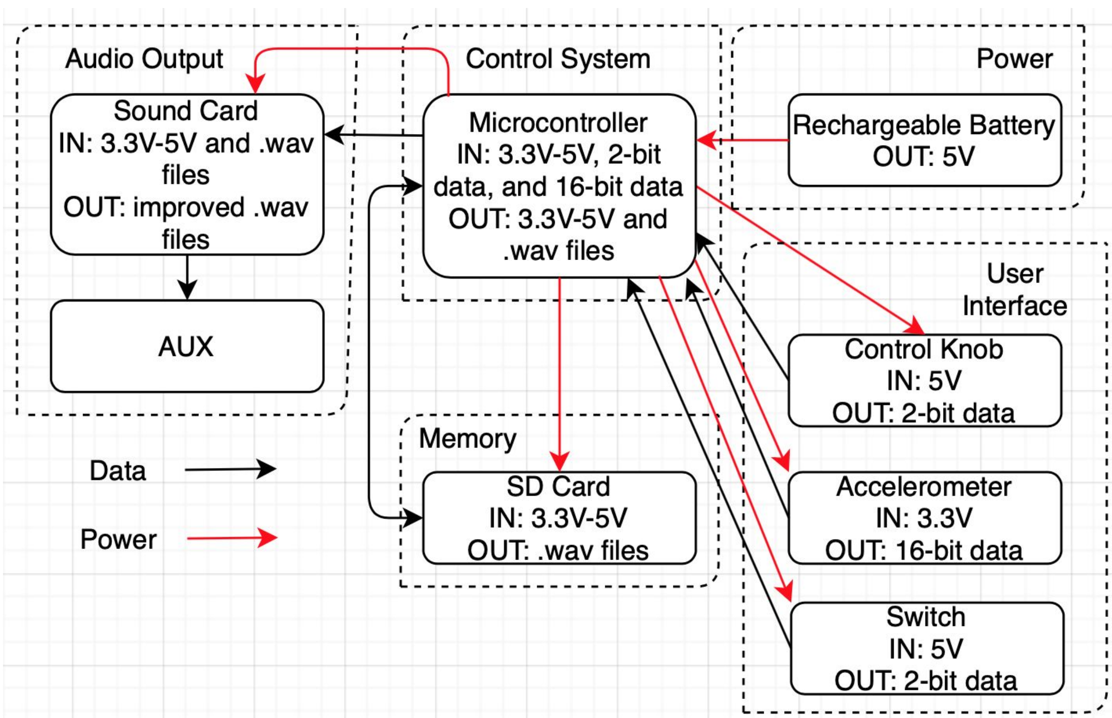

Figure 2 Drum Smart Block Diagram

2. Second Project Implementation

Once the general idea of Drum Smart was well thought out, we started working on the details to bring our electronic drum idea into reality. Through extensive research, the necessary hardware components to build Drum Smart were chosen by considering their function, tolerance, and cost. When our choices were finalized, we began designing an overall circuit for our device. Once the hardware was largely figured out, we then looked at the software. A common software problem for creating more realistic sounds was also understood and resolved.

2.1 Implementation and Details

Our plan is to design a drum that behaves like a regular drum and is capable of sounding similar to several other kinds of drums. Our device will look like a marching band drum with a drum pad as a layer of rubber to absorb most of the noise coming from the strikes on the pad. Underneath the rubber, an accelerometer will measure the amount of force used to hit the pad. This data will be sent to our microcontroller. Our microcontroller will use this user input data to access memory to output the desired drum sound. This sound data from memory will be sent to a Sound Card which will significantly improve the sound quality. These created drum sounds will be outputted through a connected speaker via AUX. We will implement a control knob the user can twist to select the type of drum sound our drum will emit when the pad is struck. We also plan to allow the user to control the pitch of every selected drum by flipping a switch to either raise or lower the sound by one octave to add more variation to their drum sounds. To make our drum portable, all our electrical components will be powered by a rechargeable battery. For pricing, Table 1 shows ordering information about each component including their cost. The total cost of parts is $183.54. Table 2 presents the labor cost due to time spent and hourly wage. Table 3 demonstrates the total cost of our project; this table sums up the parts total from Table 1 with the labor total from Table 2 to have the grand total of this project to be $18,933.54.

2.1.1 Power

The power supply will ensure all our components in Drum Smart remain functional. Appendix A contains two power requirements and their methods of verification to guarantee all our components are operational. We wanted our device to be portable, so we decided to use a rechargeable battery as our power source instead of using a plug-in which is limiting where the user can take the device.

2.1.1.1 Rechargeable Battery

This component needed to be capable of recharging without requiring the user to dismantle our device. We decided this component needed to be rechargeable through a Micro-USB port. To achieve one of our high-level requirements, we also needed a single battery to power all of our components for at least four hours. To solve our power problems, we found the PiJuice HAT. This component consists of a 5000 mAh Lithium-Ion battery and battery HAT that recharge via

Micro-USB [5]. They attach to the microcontroller like a hat and transfer data through two GPIO pins, the Serial Data (SDA) and Serial Communications Clock (SCL) pins [6]. This component is also operational while recharging.

2.1.2 Memory

All of the sounds the device will output are stored as .wav files. Drum Smart must store these .wav samples and play back when needed. The memory system must allow for quick access so that the drum's hit to sound output delay is not noticeable.

2.1.2.1 SD Card

We needed a component that would be capable of storing all of our .wav files for sound. These files include the regular sounds for our four different types of drums along with their sounds; one octave higher and lower. To store all of our .wav files, we decided to implement a 32 GB Micro SD Card. This SD Card connects to our microcontroller via the Micro SD Card Slot. This component enables us to store higher quality samples without worrying about limitations of memory. Up to 8 samples can be played at one time. The data from the SD Card is accessed by the microcontroller which sends the file to the Sound Card for improvement and output.

2.1.3 Control System

The control system of the unit must take inputs from the accelerometer, control knob, and switch. Based on those inputs, the microcontroller needs to find the correct .wav file on the SD Card and send that data to the Sound Card for improvement and output. The created drum sound has to be emitted very quickly; otherwise, if there is a very noticeable latency, our drum will not be meeting our high-level requirement of behaving like a real drum. Appendix A goes into further detail about latency and voltage level requirements and their techniques for verification.

2.1.3.1 Microcontroller

This component needed to work with all of the components in our device. It needed to be operational from one rechargeable battery. The microcontroller had to take in digital data from the accelerometer, control knob, and switch. It was necessary that our microcontroller contained a Micro SD Card slot for our Micro SD Card; this would give us the ability to access all of our sound files based on user input. This component also needed to have a USB port for our Sound Card. We needed to send files to the Sound Card for file improvement and sound output for better user experience. The component that fulfilled all of the previous requirements was the Raspberry Pi Model 3 A+. This Raspberry Pi model fits all our criteria and more.

2.1.4 User Interface

For user interface, our desire is to make Drum Smart behave like an average drum while including parameters that a user can change. The user will strike the drum pad like a normal drum. The amount of force used to strike the drum pad will correlate to the volume of the sound. The user will also be able to change the drum type and pitch level.

2.1.4.1 Accelerometer

This component needed to measure changes in force on the drum pad. It had to be capable of detecting a wide range of force exerted on the drum pad to output the proper volume of the sound. This component allows us to incorporate the force-to-sound ratio which helps us achieve another high-level requirement of real drum behavior. We found the ADXL375 Accelerometer as a good solution for this problem. This accelerometer converts the force on the drum head into three digital outputs for the X, Y, and Z axis measurements. This component has a wide force detection range of ±200 g which corresponds to ±1,960 m/s^2. This data is sent to the microcontroller as a 16-bit digital output which means no additional Analog to Digital Converter (ADC) is needed to make coding feasible [7]. This accelerometer is fully functional at 3.3 V. Its digital data is sent through the SDA and SCL pins to the microcontroller.

2.1.4.2 Control Knob

This component had to be easy-to-use for any user to select a drum type. To satisfy our high-level requirement for drum type selection, we need to give the user the choice to pick among four different drum types. We decided that turning dials, or control knobs, to select the desired drum was the easiest method for any user to understand. For the drum type control knob, we will be using a rotary encoder. This encoder operates at 5 V like a knob and sends 2-bit digital data to our microcontroller through two GPIO pins [8].

2.1.4.3 Switch

This component needed to provide three different positions for pitch selection. The piece that met this condition was the SPDT ON-OFF-ON Switch. This switch can be moved to three different positions [9]. When the switch is flipped up, our drum will make sounds one octave higher. When the switch is flipped down, the drum will make sounds one octave lower. When the switch is in the center, our drum will make normal drum sounds at a regular octave. This switch functions normally at 5 V and sends 2-bit digital data to the microcontroller through two GPIO pins [9].

2.1.5 Audio Output

After the drum has been activated and a hit has been detected, our device plays the desired sound using the output of a Sound Card. The audio must be high quality and have negligible delay from hit to sound output. Appendix A goes into further detail on our sound quality requirement and its method for verification. To create drums that sound realistic, the same exact .wav file should not be played over and over again. To counter this, we will sample each unique drum sound multiple times, with random changes in velocity and EQ values. The velocity values range from 1-127, but in our implementation we randomly adjust the value in each sample to values ranging from

40-80. Specifically, the velocity of every other hit is in a different velocity range. The even number hits will have from velocity 60-80 and the odd number hits will have velocity from 40-60. We chose this method because a drummer will usually hit harder on the first hit than the second hit during a snare roll. The EQ changes are subtle but add some differences in each drum strike, such as taking off part of the low-end or high-end. All together, each drum hit will be slightly different than the last. This is especially important during a drumroll. During a drumroll, the user repeatedly strikes the drum. In our previous design, the same .wav file would play over and over, creating a non-realistic effect. Our new randomized hit sample allows for realistic sounding drums. The volume of each sample is decided by the force of each hit. Each drum sample can be pitched up or down an octave depending on what the user selects.

A common problem is the machine gun effect where the exact same sound is constantly repeated. To avoid this issue, subtle differences need to be added to the original sound and played at random to be more realistic [10]. To implement this solution for our drum, every other hit has a random velocity within ranges 60-80 and 40-60, respectively. Each hit is pitch shifted within ±50 cents of its original pitch. Then, we added random EQ spikes of a 20 Hz range from frequencies 200-2000 by 0-10 dB to add more acoustic realism.

2.1.5.2 Sound Card

Although our microcontroller already provides an AUX port, the sound quality through this port is very low. We needed a component that improved the quality of our sounds. One piece that is capable of connecting to our microcontroller and contains an AUX port is a Sound Card. The Sound Card connects to our microcontroller through the microcontroller's USB port. Through this port, the microcontroller sends the Sound Card the necessary .wav files based on the input from the user. This component improves the resolution of the .wav files resulting in a higher quality sound. The signal-to-noise ratio now becomes 92 dB [11]. This improved sound is output through any speaker connected to the Sound Card's AUX port.

2.1.6 Overall Schematic

Figure 3 demonstrates the overall schematic of Drum Smart. The PiJuice HAT with battery powers the Raspberry Pi which supplies power to all of our other components. The HAT and battery not only supply power but also can be programmed which requires using the SCL and SDA pins to send and receive data. The accelerometer will also be connected to the SCL and SDA pins for transferring 16-bit digital force data from the drum pad to the Raspberry Pi. Since multiple components are on the same I^2C bus, two pull-up resistors are needed to prevent any communication errors on the bus. The accelerometer also has two capacitors near its input pins to reduce noise from the power supply. The rotary encoder and switch are both powered by 5 V and output 2-bit digital data requiring them each to be connected to two GPIO pins on the Raspberry Pi. The rotary encoder sends data on drum selection, and the switch transmits data on pitch level.

The switch also has two pull-down resistors to prevent any accidental shorts when the switch is moved to different positions. The Micro SD Card contains all of our sound files. It plugs into the Raspberry Pi through its built in Micro SD Card slot. The Sound Card will improve the sound files on the Micro SD Card. The Sound Card connects to the Raspberry Pi through the USB slot.

2.1.7 Software Overview

The software for Drum Smart can be broken down into four components: input, processing, sound driver, and sound module. The first component, input, is the digital data sent in by the accelerometer for the amount of force, the control knob for the drum type selection, and the switch for the pitch selection. The second component of the software is processing. This component uses the input data to select the volume level and the desired sound. The data from the accelerometer is converted to gainIncrease for volume while the drum selection knob and pitch switch choose which filename system to use for the sound. The third component, sound driver, implements the ALSA Audio Driver library. The sound driver module switches the default sound output from the on-chip system to the Sound Card, allowing a powerful, lossless sound conversion from software to AUX. The last component is the sound module. Each sound module consists of a filename, gainLevel, and isConsecutiveHit boolean value. The filename is the name of the specific sample file to be used. The gainLevel is based upon how hard the drum is struck and changes based on a $\log_{10}$ scale of the accelerometer force. Using a $\log_{10}$ scale sounds better than a linear change in volume. The smallest discrepancies in the force of a hit will be unnoticeable unless the user specifically taps the drum, creating a softer drum sound. The isConsecutiveHit value acts as a counter that alternates values between each hit. This variable helps create the random acoustic change heard by a physical drum, while alleviating the machine gun drum effect that is possible with electronic drum machines.

2.1.8 Pull-Up Resistor Calculations

Since we will have multiple devices on the same bus, the operating voltage level of these devices cannot exceed a certain value to stop communication issues. The maximum operating voltage is

$$V_{max} = V_{cc} + 0.3 \tag{1}$$

where V_{cc} is the power supply voltage [7]. Since our V_{cc} has a value of 3.3 V, Equation (1) takes 3.3 and adds it with 0.3 to find our maximum operating voltage to be 3.6 V. To keep the voltage levels from exceeding 3.6 V, we will need to implement several pull-up resistors [7]. These resistors will prevent any I^2C communication errors between the Raspberry Pi, PiJuice Hat, and accelerometer. For many cases, there are a wide range of different resistor values to choose that will still satisfy the circuit. The minimum value of a pull-up resistor is

$$R_p(min) = \frac{V_{cc} - V_{OL}(max)}{I_{OL}} \tag{2}$$

where V_{cc} is the power supply voltage, $V_{OL}(max)$ is the maximum low-level output voltage, and I_{OL} is the low-level output current based on the input voltage [12]. After reviewing the datasheet for the accelerometer, the power supply voltage is 3.3 V, the maximum low-level output voltage is 0.4 V, and the low-level output current is 3 mA [7]. Plugging these values into Equation (2), we find our minimum pull-up resistor to have a value of 967 Ω. To find the other limit to this range, we need to calculate the maximum pull-up resistor value. The maximum value of a pull-up resistor is

$$R_p(max) = \frac{t_r}{(0.8473)C_b} \tag{3}$$

where t_r is the rise time of both SDA and SCL signals and C_b is the capacitive load for each bus line [12]. Both these unknown variables can be found without calculation on the datasheet for the accelerometer. The rise time is 300 ns and the capacitive load for each bus line 200 pF [7]. From Equation (3), we find the value of the maximum pull-up resistor to be 1.77 kΩ. With a minimum of 967 Ω and a maximum of 1.77 kΩ, choosing a pull-up resistor of 1.5 kΩ will be a safe choice for the project.

2.1.9 Rechargeable Battery Calculations

To achieve one of our high-level requirements, we needed our device to function off of a single charge for at least four hours. This required us to find the right rechargeable battery that can support the power drawn from all of our components under continual use. The amount of stored energy in the battery is

$$E = V * Q \tag{4}$$

where V is the battery voltage and Q is the battery capacity [13]. Our Lithium-Ion battery is currently 5 V with 5000 mAh [5]. Plugging these values into Equation (4), we find our battery to have a 25 Wh energy capacity. Dividing our battery's energy capacity by our high-level requirement of at least four hours of operation, we calculate that the power consumption can not exceed 6.25 W. To make sure we do not exceed this limit, we need to find the power of all of our components in our device. Power is calculated by

$$P = V * I \tag{5}$$

where V is voltage and I is current [14]. Our Raspberry Pi operates at 5 V with a maximum of 470 mA [15]. Using these values in Equation (5), we find out Raspberry Pi to draw 2.35 W. Looking at the datasheet of our 32 GB Micro SD Card, we find this component to function at 3.3 V with 200 mA [16]. Plugging these numbers into Equation (5), we calculate the SD Card to consume 0.66 W. The specifications sheet for the Sound Card provided the maximum power consumption of this piece to be 0.03 W [11]. The power from the accelerometer, knob, and switch is extremely minimal. These values are so small, we can assume the power for these components to be negligible. Adding up the power from each component, we find the power consumed by these pieces to be 3.04 W. Since the power consumed by our components is a lot smaller than the power capacity of our battery, this Lithium-Ion battery will be a good choice for our project.

2.1.10 Tolerance Analysis

To accomplish another one of our high-level requirements that involves the force-to-sound ratio, we had to find a part that could measure the amount of force on the drum pad. This piece needed to be both sensitive and be able to measure a wide range of forces. We first considered the flex sensor. This sensor works by changing its resistance when it bends. Our idea was to attach this piece to the underneath side of the rubber drum pad, so when the pad gets struck, it will bend the sensor detecting a force. After some research, we found that the rubber drum pad will not be bending as much as we imagined, and these kinds of sensors are not very accurate. They give a reading in the right area but are not very precise [17]. If the user kept hitting the drum with the same amount of force, it would be possible that the volume would be changing noticeably. Since we expect our drum to always have the correct volume for the amount of force applied, we had to look for a more precise sensor. We then looked into a type of vibration sensor, the accelerometer. These components work by measuring changes in acceleration on the part. If these are attached to an object, they will be detecting the acceleration of that object. They come in many different variations covering both a wide range of sensitivity and acceleration. We believe the accelerometer is the correct component for this task; we just need to find a version that is capable of being quite sensitive and detecting a wide range of forces.

The sensitivity range of our accelerometer was a very crucial aspect to our drum. Since we wanted our drum to be similar to a real drum, we wanted the force-to-sound ratio to be similar as well. This means the harder the strike on the drum pad, the louder the sound that is created. To make this feature a reality, we needed to implement an accelerometer that would be able to withstand the maximum amount of force our drum pad would feel [18]. Most accelerometer data sheets provided ranges of acceleration each component could handle. We knew we had to do some force calculations and research to find the best accelerometer for our project. The force on any object can be found by

$$F = ma \tag{6}$$

where m is the mass of the object and a is the acceleration of the object. We started with Equation (6) to find the maximum acceleration our accelerometer would be put under [19]. For this equation, the variable, m, will be the mass of one drumstick. Since our drum will look like a marching band drum and used by a variety of different users, we wanted our drumsticks to be the marching band type and very durable to withstand the style of every user. We found some durable drumsticks each having a mass of 0.080 kg [20]. Next, we had to find some data on the amount of force, F, on the drum pad. One study found that raising the drumstick 0.6 m resulted in a force of about -140 N on the drum pad [21]. This was the maximum force that was recorded, so we will use this to calculate our maximum acceleration, a. Now, plugging our values for m and F into Equation (6), we find the maximum amount of acceleration our drum pad would feel is -1,750 m/s^2. Since this is the maximum amount of acceleration our drum will be experiencing, we need to find an accelerometer that has a range larger than the calculated acceleration value to account for the difference in strength among users. The ADXL375 Accelerometer can measure accelerations up to -1,960 m/s^2 providing us an extra 210 m/s^2 for the different users. Since this accelerometer can measure accelerations a little larger than our calculated maximum, it seems like a good component for the project.

2.2 Figures and Tables

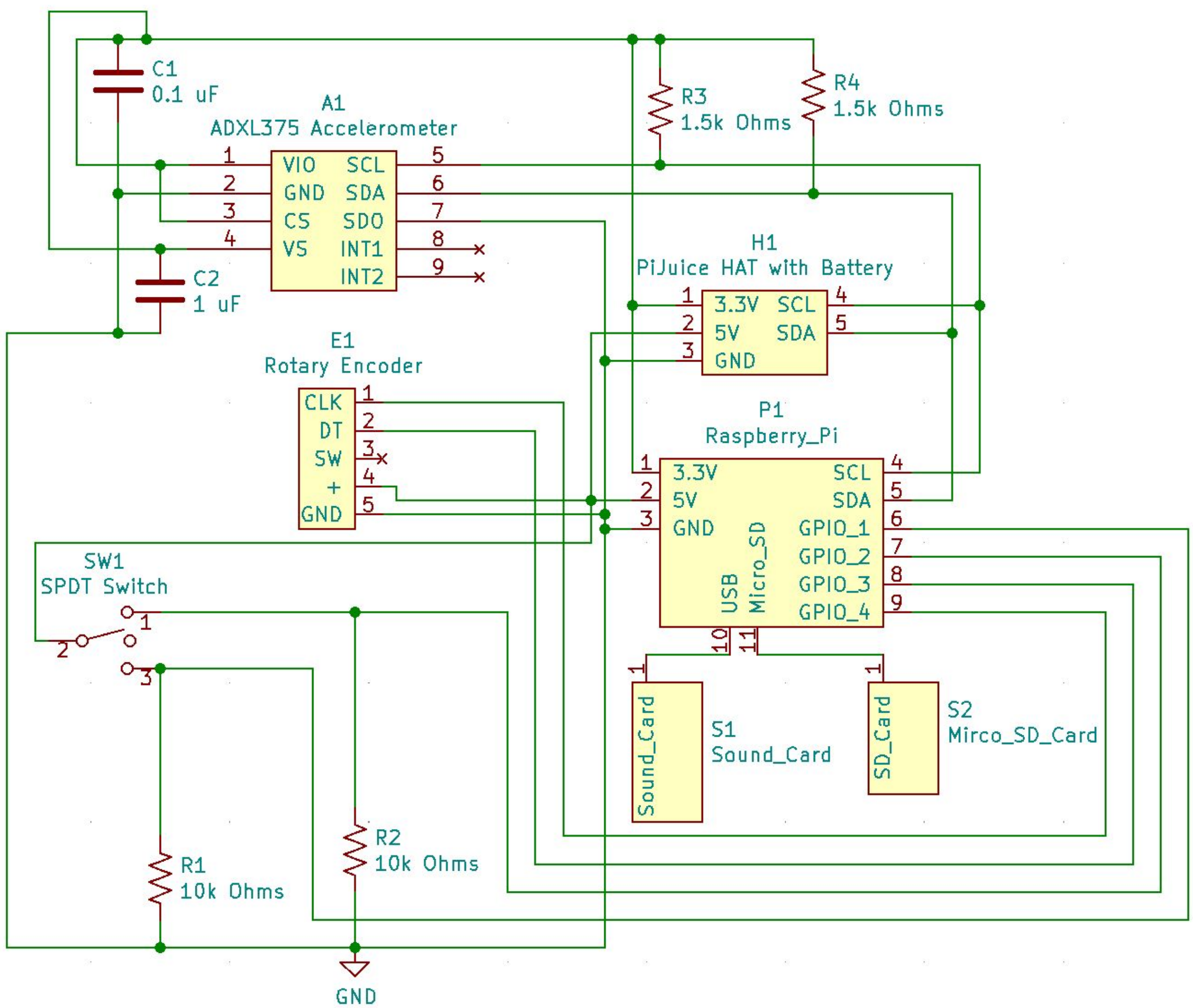

Figure 3 Drum Smart Overall Schematic

Table 1 Parts Cost

Description	Manufacturer	Part Number	Quantity	Cost ($)
PiJuice HAT and Battery	Pi Supply	PIS-0212	1	82.43
SanDisk 32GB Ultra microSDHC Card Class 10	SanDisk	SDSDQUA-032G-A11A	1	8.00
Raspberry Pi Model 3 A+	Premier Farnell	RPI3-MODAP	1	25.00
ADXL375 - Accelerometer, 3 Axis Sensor Evaluation Board	Analog Devices Inc.	EVAL-ADXL375Z	1	30.00
KY-040 Rotary Encoder Brick Sensor Module Development	Frentaly	B07VKF9S3W	1	8.60
Toggle Switches 3-6A 125-250VAC SPDT ON-OFF-ON SLDR IP68	NKK Switches	M2013WBW01	1	12.92
UGREEN USB External Stereo Sound Card Audio Adapter with 3.5mm Aux and 2RCA Converter	Ugreen Group Limited	30521	1	16.59
TOTAL PARTS COST				183.54

Table 2 Labor Cost

Number of Partners	Completion Time (Hours)	Hourly Rate ($)
3	125	50.00
TOTAL LABOR COST		18,750.00

Table 3 Grand Total Cost

Section	Total ($)
Labor	18,750.00
Parts	183.54
GRAND TOTAL COST	18,933.54

3. Second Project Conclusions

After careful design and implementation, we found promising solutions to bring Drum Smart into reality and achieve our overall objective of teaching people about music and technology. Throughout these processes, safety and ethics were considered in every step. While the current version of this device satisfies all our high-level requirements, the future versions of Drum Smart have room for improvement. Due to the current circumstances, some additional testing will be needed in the future for verification of the subsystem requirements.

3.1 Implementation Summary

We accomplished a design of an electronic drum that behaves like a regular drum and is capable of sounding similar to several other kinds of drums. It has the ability to measure the amount of force used to hit the drum pad. This data is sent to our microcontroller which will process the user input to access memory and output the desired drum sound. This sound data from memory will drastically be improved before the actual sound is emitted. These enhanced drum sounds will be played through AUX speaker connection. The user can twist the knob to select the drum they want to hear. The pitch switch allows the user to raise or lower the drum sounds by one octave for more variety. All our electrical components will be powered by a rechargeable battery giving this drum the ability to be portable. With all these different aspects and areas of our device, we decided to divide this project into smaller, manageable tasks for each individual group member. Collin worked mainly on the hardware and reports. This included the problem, solution, background, project differences, high-level requirements, design, component information, schematic, calculations, tolerance analysis, writing style, and formatting. Kevin also aided in hardware work. He came up with the drum solution, worked on high-level requirements, and implemented requirement and verification tables. John spent a lot of time on the audio and software portions of this project. He worked on memory, several components, audio improvements, project differences, and ethics and safety. He also discovered a big sound effect issue and found a working solution to fix this problem. Without this division of labor, it would have taken our group a lot longer to design and implement Drum Smart; however, our key divisions and organization helped make this process go a lot smoother.

3.2 Unknowns, Uncertainties, Testing Needed

Due to the current circumstances, our group was unable to access the lab. Not having access to the lab played a big role in preventing us from moving on to the building, testing, and demonstrating phases. We could not go any further than the design phase due to several issues. The time constraint and lack of equipment are the main factors that prevented us from moving past the design phase. If we had more time, we would have ordered all of our parts and the printed circuit board (PCB). At this time, we would also be working with the machine shop to find a suitable physical design for the inside and outside of the drum. If we had an oscilloscope or multimeter, we would be verifying the operation of each individual component before we used it in the circuit. Once verified, we would need a soldering iron to solder the capacitors and resistors to the PCB. Next, we would connect one component at a time by soldering wires to both the component and PCB to add it to the circuit. Once soldered, we would, again, verify that it is functional within the circuit. By adding one piece at a time, this would make finding the source of verification problems a lot easier. Once the circuit is completed, the physical design of the inside and outside of the drum should also be finished. Then, we would place the circuit and all the components in their designated areas on the inside of the drum. Before sealing the inside off from the user, we would test our drum to make sure everything is working as planned. Once this quality check is complete, we would seal off the inside of the drum. Once sealed, we would test it one more time to make sure our finished project is fully operational.

3.3 Ethics and Safety

Our new solution, Drum Smart, adheres to the specific ethic and safety guidelines put forth by IEEE and ACM. It is important to practice safety throughout development and use of Drum Smart. We plan on accomplishing this by limiting dangers and electrical hazards. Code 1 of the IEEE Code of Ethics states “to hold paramount the safety, health, and welfare of the public” [22]. The safety of the public is our number one priority. Drum Smart is used recreationally and part of a good experience in ensuring the correct connections of electrical current and creating a very reliable device.

An aspect of our device that we were concerned about that could cause hazards is the rechargeable battery. The battery we have selected is a 5000 mAh Lithium-Ion battery. We will place a warning upon the battery cautioning the user to not store the device above 60 ℃, the temperature at which the battery could become hazardous. Adhering to Code 9 of the IEEE Code of Ethics, we will “avoid injuring others” by offering a proper disclaimer of what could cause our system to malfunction, and in turn, harm the user [22].

Drum Smart is a device that can be used by any able-bodied individual that would like to use the product. According to the ACM Code of Ethics, 1.4 emphasizes the value of equality and fairness [23]. Our device encourages use by people of any age or background. Since the Hip Hop Xpress bus will be travelling to lower income communities, our product specifically targets the people living in these areas but our device is not limited to these people; Drum Smart can be used by anyone who enjoys music and technology.

According to the CDC, damage to the hearing can be caused by prolonged exposure to sounds over 80 dB [24]. Following the advice of the CDC, we want to limit the danger of hearing loss, while allowing for the possibility of loud audio output. In order to be safe and still produce loud audio, we will set extremely hard drum pad strikes to have a maximum sound output of 80 dB.

3.4 Project Improvements

If we had a year to complete this project instead of just a few weeks, there are three beneficial changes we would make to Drum Smart. The first improvement in the design of the device would be to incorporate a microcontroller that uses less power. The recent versions of the Raspberry Pi draw a lot of power when they are idle. This unnecessary power consumption will greatly affect the amount of time the battery can operate without recharging. Since we want our drum to be operated off of a single battery for as long as possible, implementing a Raspberry Pi with less power consumption would greatly improve this drum. The second improvement in Drum Smart would be to include speakers on the drum shell. Currently, the only aspect that prevents this device from being played anywhere on land is the audio output; Drum Smart needs to be connected to an external speaker via AUX to output the drum sounds. Including speakers on the drum frees the user of the AUX cable requirement and makes our device even more portable. This will allow the user to move around anywhere they desire while playing this drum. The third improvement to benefit Drum Smart would be to find a cheaper rechargeable battery. For this project, we went quite a bit over budget due to one pricey component. The HAT and battery component alone cost more than $80, consuming more than 80 % of our $100 budget. Finding a cheaper rechargeable battery that works well with the Raspberry Pi would drastically help reduce the cost of this device.

4. Progress Made on First Project

After the first design document review, there were several changes that were made to our first project, Beat Starter. The most important improvements were in our component information and block diagram. For our parts information, we found a few different options for each of our necessary pieces. Each component option included operational voltages, output data, pin configurations, and cost. After looking at all the choices, we compared them all and chose the right parts for each required task. Once these pieces were finalized, we updated the block diagram to include these new details.

4.1 Updated Block Diagram

Figure 4 shows the updated block diagram for Beat Starter. The bus battery will be connected to a voltage regulator to turn the voltage down to a lower voltage to be used for our other components. The push button switches and control knobs will send digital data to the microcontroller. Based on the data from the switches, the microcontroller will read from the SD Card and pull the desired sound samples from the library. The microcontroller will write to the RAM to update it with the new sound samples from the SD Card. These new sound samples will be continuously played in a loop. The sound samples in the loop will be sent to AUX in order to play the newly created beat. The Audio Bonnet improves the sound output from AUX. Based on the data from the knobs, the microcontroller will change the BPM and volume while at the same time update the LCD display panel with the adjustments to BPM and volume.

4.2 Figure

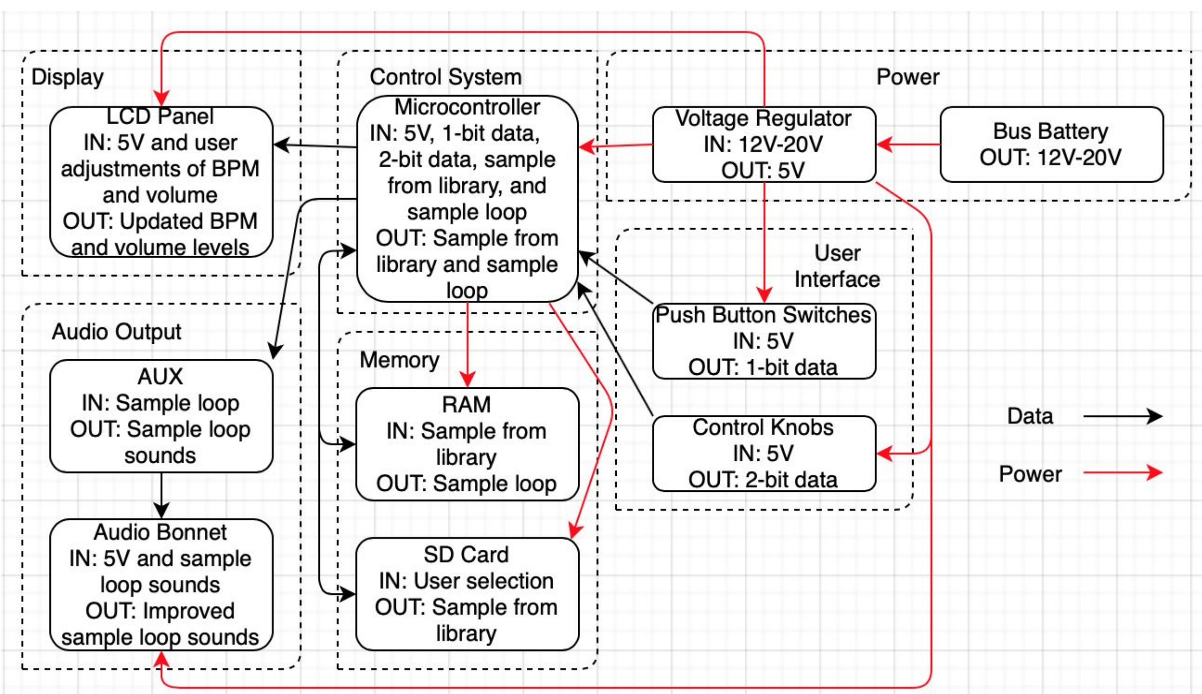

Figure 4 Beat Starter Updated Block Diagram

5. References

[1] The Hip Hop Xpress. Available at: www.publish.illinois.edu/hiphopxpress/

[2] Waddell, et al. “Technology Use and Attitudes in Music Learning.” Frontiers, Frontiers, 29 Apr. 2019. Available at: www.frontiersin.org/articles/10.3389/fict.2019.00011/full#T2

[3] “10 Tips For Teaching Music to Kids More Effectively.” Solfeg.io, 11 Dec. 2019. Available at: www.solfeg.io/teaching-kids-music/

[4] “Rock Band Drum Set.” Rock Band Wiki. Available at: www.rockband.fandom.com/wiki/Rock_Band_Drum_Set

[5] Pi Supply. 2020. *Pijuice 5000Mah Battery*. Available at: https://uk.pi-supply.com/products/pijuice-5000mah-battery

[6] “PiJuice HAT – A Portable Power Platform For Every Raspberry Pi.” *Pi Supply*. Available at: https://media.digikey.com/pdf/Data%20Sheets/Pi%20Supply%20PDFs/PIS-0212_Web.pdf

[7] “ADXL375.” Analog Devices , Analog Devices Inc., 2014. Available at: www.analog.com/media/en/technical-documentation/data-sheets/ADXL375.PDF

[8] Codelectron. “Rotary Encoder with Raspberry Pi - OLED Menu.” Codelectron, 28 Apr. 2019. Available at: www.codelectron.com/rotary-encoder-with-raspberry-pi/

[9] “Waterproof Miniature Toggles.” Series M. Available at: www.mouser.com/datasheet/2/295/Mtoggle_Waterproof-1508280.pdf

[10] Peter Ginneberge. Ableton Tutorial: How to Avoid Machine Gun Effect. (Jun. 9, 2012). Accessed: Apr. 22, 2020. [Online Video]. Available: https://www.youtube.com/watch?v=VaSqSieQ4gQ

[11] “USB External Stereo Sound Card with 3.5mm Aux 2RCA Converter.” Ugreen. Available at: https://www.ugreen.com/product/UGREEN_USB_External_Stereo_Sound_Card_Aux_and_2RCA_Converter-en.html

[12] R. Arora. “I2C Bus Pullup Resistor Calculation.” Feb. 2015. Available at: http://www.ti.com/lit/an/slva689/slva689.pdf

[13] Omnicalculator.com. 2020. *Battery Capacity Calculator - Omni*. Available at: https://www.omnicalculator.com/other/battery-capacity

[14] "Ohms Law Tutorial and Power in Electrical Circuits", *Basic Electronics Tutorials*, 2020.

Available at:
https://www.electronics-tutorials.ws/dccircuits/dcp_2.html?utm_referrer=https%3A%2F%2Fwww.google.com%2F

[15] "How Much Power does Raspberry Pi 3A+ plus Use?", *RasPi.TV*, 2020. Available at: https://raspi.tv/2018/how-much-power-does-raspberry-pi-3a-plus-use

[16] *Alliedelec.com*, 2020. Available at: https://www.alliedelec.com/m/d/04db416b291011446889dbd6129e2644.pdf

[17] Electrical Engineering Stack Exchange. *Blend or Flex Sensors*. 2020. Available at: https://electronics.stackexchange.com/questions/34066/blend-or-flex-sensors

[18] “Measuring Vibration with Accelerometers.” Measuring Vibration with Accelerometers - National Instruments. Available at: www.ni.com/en-us/innovations/white-papers/06/measuring-vibration-with-accelerometers.html

[19] VectorNav. “Accelerometer - VectorNav Library.” VectorNav Embedded Navigations Solutions. Available at: www.vectornav.com/support/library/accelerometer?gclid=CjwKCAjwvtX0BRAFEiwAWJyZBY5x2az9E_jO52aPJPCVI_y6WSKsyUR4UgBRCDT-FypMrAyHCzj7xoCvoMQAv_BwE

[20] “Percussion Information.” Properties and Shapes. Available at: www.marcz1.home.xs4all.nl/percussioninformation/Sticks/Sticks/properties.html

[21] Dahl, S. “ARM MOTION AND STRIKING FORCE IN DRUMMING.” Dept. Speech, Music and Hearing, Royal Institute of Technology, Stockholm/Sweden. Available at: https://pdfs.semanticscholar.org/4ea6/0679d6443a2bff1a2a7a9c9f04d9c3190de6.pdf

[22] Ieee.org, "IEEE IEEE Code of Ethics", 2016. Available at: http://www.ieee.org/about/corporate/governance/p7-8.html

[23] “The Code Affirms an Obligation of Computing Professionals to Use Their Skills for the Benefit of Society.” Code of Ethics. Available at: www.acm.org/code-of-ethics

[24] “What Noises Cause Hearing Loss?” Centers for Disease Control and Prevention, Centers for Disease Control and Prevention, 7 Oct. 2019. Available at: www.cdc.gov/nceh/hearing_loss/what_noises_cause_hearing_loss.html

Appendix A Requirement and Verification Table

Table 4 System Requirements and Verifications

Requirement	Verification
1. Power - Must output a minimum 5 V.	1a. Probe the battery with an oscilloscope. 1b. Ensure the battery is outputting our minimum voltage of 5 V.
2. Power - Must operate for at least 4 hours.	2a. Test the system at various stress levels until the battery can no longer power the system. 2b. Time each session to ensure that even at the highest stress levels the system receives power for at least 4 hours.
3. Control System - Accelerometer, knob, and switch must have a maximum latency of 0.2 seconds.	3a. Use an oscilloscope to measure response time after an individual pad strike. We will do this by probing the signal from the accelerometer and compare that time to the signal time sent by the microcontroller due to the action. 3b. The difference between signal times must be at most 0.2 seconds. 3c. Repeat steps 4a and 4b for a knob turn and a switch flip instead of the pad strike.
4. Control System - Operates within range of 5 V +/- 5 %.	4a. While running the microcontroller, probe input and output voltage with DMM/oscilloscope. 4b. Confirm the difference between Vin and Vout is within 4.75 V - 5.25 V.
5. Audio is outputted with a signal to noise ratio of at least 25 dB.	5a. Using an oscilloscope, measure the fourier transform of the audio output signal and observe the output. 5b. Calculate the signal-to-noise ratio (SNR) from the frequency spectrum to ensure it is at least 25 dB.