# Portable In-Line Audio Equalizer

Design Document
Team Number 8

**Ankit Jayant (ajayant2)**
**Avinash Subramaniam (avinash6)**
**Ji Yeon In (jiyeoni2)**

ECE 445
TA: Prashant Shankar
March 1, 2021

# 1 Introduction

## 1.1 Objective

There are varying preferences to equalization (EQ) in audio, whether it is through personal preference or a need, such as helping with a hearing impairment. Some media players do not have a built-in equalizer nor do they allow for downloading EQ mobile apps. Therefore, users are unable to adjust the sound signature of what they are listening to. Also, many pre-existing EQ devices are too large or heavy to be portable.

The solution is the Portable In-Line Audio Equalizer (PIAE). Using the data from a desired media player, the PIAE uses signal processing algorithms to output audio data with a boost or attenuation at certain frequency ranges. This device allows for equalization, and has the advantages of low cost and convenient everyday use.

## 1.2 Background

Hearing loss can come in different ranges. One form of hearing loss to consider is a "notch" hearing loss, which is hearing loss at a certain frequency range [1]. In order to help with this type of issue, any desired frequency range can be boosted by an audio equalizer when using a media player. There are also people with personal preferences with sound signatures who use equalizers.

Some devices have built-in equalizers, like in computers and MP3 players, but that is dependent on the specific version and brand. Equalizer mobile apps can also be downloaded, but that is not possible for older devices, such as CD players.

There are also portable audio equalizers that exist. Typically, the more portable an audio equalizer is, the fewer operating ranges, or bands, it will have. Larger operating ranges allow for more options for the user, as well as a greater ability for the user to fine-tune the emphasis on the desired frequencies. This is especially important for users suffering from hearing loss. Commercial equalizers can have eight band filters, but those devices are not usable in a casual setting [2]. Portable devices are more convenient, but sacrifice performance by using less operating ranges [3]. The goal for the PIAE is to maintain the performance provided by commercial equalizers while also providing usability by everyday people.

The performance of an audio equalizer is not only restricted to operating frequencies, but also latency. The limit for sound latency to be imperceptible by a performer hearing the sound that they are creating through their instrument is 10 to 12 milliseconds [4]. We can reason that this limit is also applicable to a user changing songs or resuming their music, which are situations where they would not want to perceive much lag between action and audio.

## 1.3 High-Level Requirements List

- The PIAE must have a low latency of less than 12 milliseconds.
- The PIAE should use eight frequency bands when constructing its filters, instead of the typical three frequency bands.
- The PIAE must have a size of less than 11.4 x 8.9 x 5.1 cm for the device to be sufficiently portable.

# 2 Design

## 2.1 Block Diagram

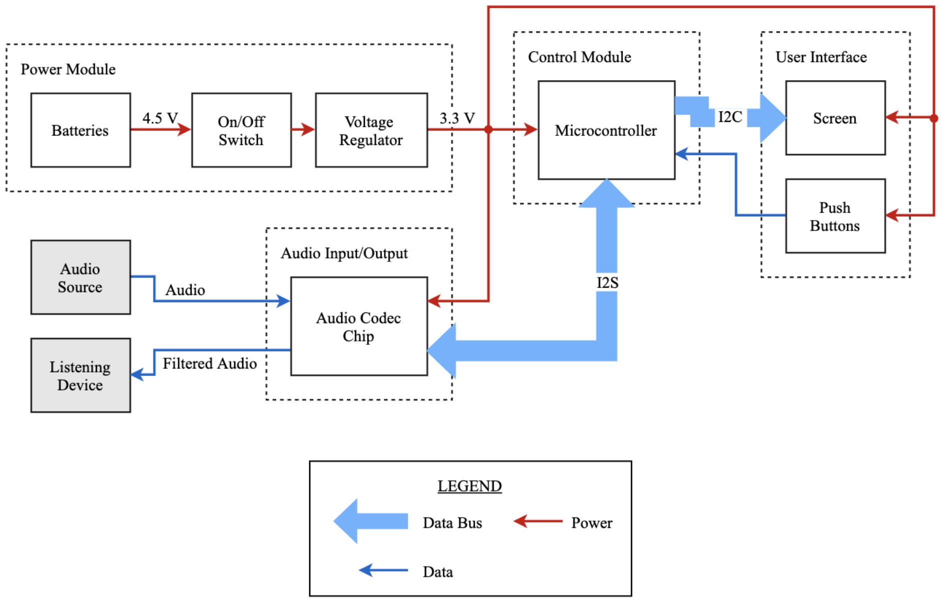


Figure 1: A Block Diagram of the Portable Audio Equalizer

The PIAE design has power, control, user interface, and audio input/output as the primary units. The power module generates an adequate amount of voltage for the other modules to use. The audio input/output module translates the audio data accordingly, which then allows the other components to understand the data. Using the data and desired filters that the user interface decides, the control module generates filtered data. This filtered data returns to the audio input/output module which is then outputted.

## 2.2 Physical Design

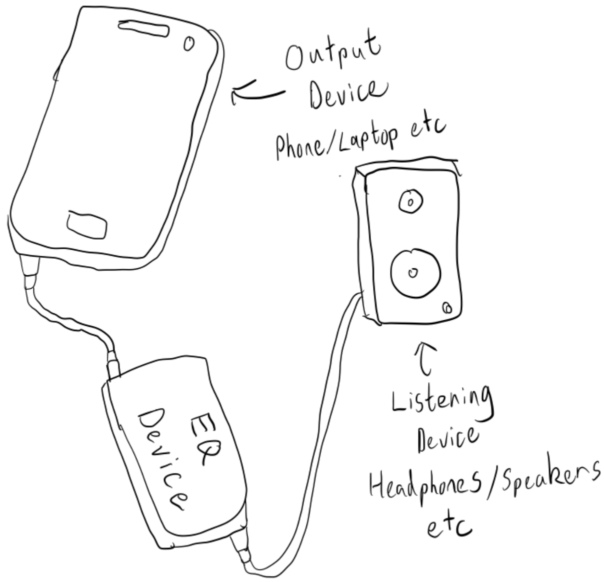


Figure 2: A Diagram of a Possible Use of the Portable Audio Equalizer

# 3 Requirements and Verification Tables

Table 1: Requirements and Verification for the Voltage Regulator

| Requirements | Verifications |
|---|---|
| 1. The voltage regulator should regulate the output voltage of the batteries to exactly 3.3 V. | 1. We will use a 3.3 Voltage regulator IC chip to ensure that regardless of the battery output voltage, exactly 3.3 V will be supplied to the STM32 microcontroller so it can operate safely. We will verify the chip as follows :<br>(a) Connect a variable voltage source in series with our IC chip input pin.<br>(b) Connect the ground pin to the appropriate ground in the circuit.<br>(c) Connect a resistor in series with the output of our IC chip.<br>(d) Connect probes across the resistor to check the voltage drop across the resistor.<br>(e) Start the variable voltage source at 3.3 V. Increase the voltage and check if the voltage across the resistor. If it is 3.3 V consistently, we have been successful. |

Table 2: Requirements and Verification for the Microcontroller

| Requirements | Verifications |
|---|---|
| 1. The microcontroller must be able to receive audio data incoming from the audio codec chip. | 1. To check if the microcontroller receives audio data from the audio codec chip, we will do the following :<br>(a) Load the audio codec driver into the microcontroller flash memory<br>(b) The Power module supplies an appropriate amount of power so the microcontroller is operational.<br>(c) Plug the audio source into the line-in audio jack and start sending the data.<br>(d) Verify that the data is accessible in memory stored in the appropriate address specified by the audio codec driver. |
| 2. The microcontroller must be able to output modified audio data through the audio codec. | 2. To check the microcontroller is able to modify and output audio data, we will do the following :<br>(a) Power on each device and transmit audio data to the microcontroller as specified by Process 1.<br>(b) Access audio data stored in memory address specified by audio codec driver.<br>(c) For each frequency band, shift the decibel value by precisely +1 dB.<br>(d) Plot the FFT of both the original signal and the modified signal and compare. If the plots match a +1 dB shift, we have modified the signal correctly.<br>(e) Transmit the data through the headphone jack in the PCB to a speaker to verify that the audio is being output through the microcontroller. |

## 4 Circuit Schematics

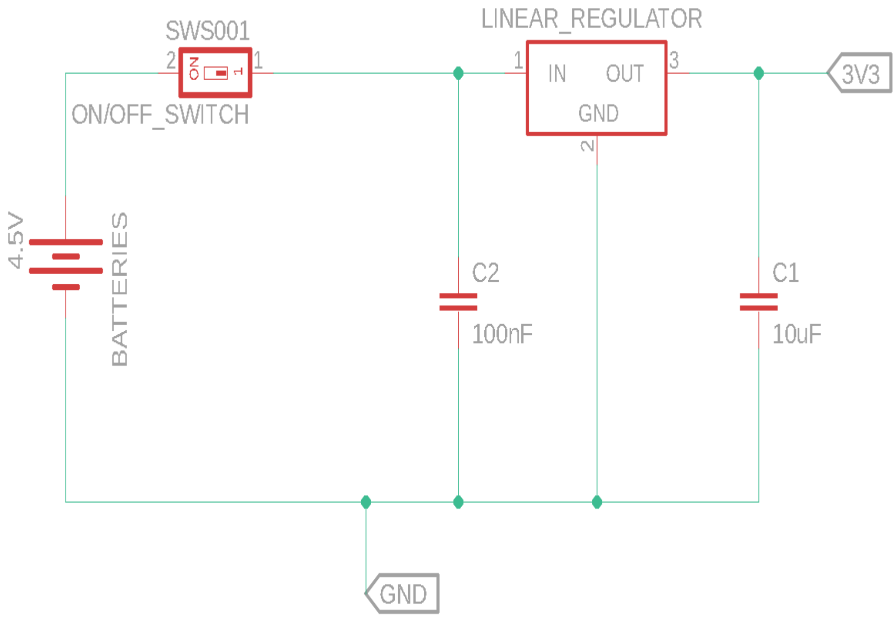


Figure 3: Circuit Schematic for the Power Unit

## 5 Tolerance Analysis

The microcontroller poses the greatest challenge to implement in our project, both from a hardware and a software perspective. However, from a quantitative perspective, we concern ourselves with the latency the microcontroller operations introduce, the amount of memory needed for these operations, and finally the trade-offs involved in designing the equalization filters.

We intend to execute equalization in the frequency range [20, 20000] Hz, and use the Fast Fourier Transform (FFT) to transform and then filter the input sound in the frequency domain. We use eight frequency bands for the equalization, centered on the frequencies 100, 250, 500, 1000, 2000, 4000, 8000, and 16000 Hz, and our FFT bin size will be 512. However, using all 512 bins will not be necessary, because the latter half of the FFT is merely the complex conjugate of the first half, as the input data is real [5]. Therefore, we only use 257 bins, in accordance with the below equation [6].

$$\text{FFT Len} = (N/2) + 1 \text{ if N is even else. } ((N+1)/2) \tag{1}$$

Where N is the FFT bin size. The FFT of N=512 requires a number of operations as specified below

$$Nlog_2N \text{ additions and } (N/2)log_2N \text{ multiplications} \tag{2}$$

Therefore, the latency of the FFT operation scales accordingly

$$\text{t\_add} * Nlog_2N + \text{t\_mult} * (N/2)log_2N \tag{3}$$

where t_add refers to the amount of time the microprocessor needs for a single complex addition, and t_mult refers to the amount of time for a single complex multiplication [7]. The total latency of our PIAE must not exceed 12 milliseconds, so it is critical that the FFT does not take up a significant part of that.
Next, the number of operations required for the sum and multiplication of the filters with the original audio signal is shown below.

$$8 * 257 = 2056 \text{ multiplications and additions} \tag{4}$$

where 8 is the number of frequency bands and therefore equalization filters, and 257 is the number of FFT bins that the audio signal and equalization filters contain in the frequency domain. Therefore, the total latency expected by the microcontroller processing is

$$\text{t\_add} * (Nlog_2N + 8 * (N/2+1)) + \text{t\_mult} * ((N/2)log_2 + 8 * (N/2+1)) \tag{5}$$

Now we proceed to analyze the memory constraints imposed by these calculations. The PIAE contains all 8 filters in memory, each of which is

$$8 * N * (\text{16 bytes per complex number}) = 33\,kB \tag{6}$$

Furthermore, the PIAE contains several frames or chunks of audio data, both for input and output. This number will need to be determined, but will accordingly increase the amount of memory used by the microcontroller in the following manner.

$$\#\text{chunks} * 2 * N * (\text{8 bytes per float}) \tag{7}$$

The number of chunks is multiplied by two because the input buffer (receiving ADC data) and the output buffer (sending data to the DAC) should be equally large. Overall, we are certain that our chosen microcontroller can handle the amount of memory required for the project, as with 10 chunks, the total memory in RAM would then be 80 kB + 30 kB = 110 kB, and the model of STM-32 we are likely going to choose has a RAM of 512 kB.

# 6 Ethics and Safety

## 6.1 Development Issues

Our ethical considerations extend primarily to issues that could arise during the development of our project. Because the PIAE filters audio that is designed for listening through headphones and speakers, we need to ensure that our product does not make the audio too loud. Audio at extremely loud volumes damages human hearing over time [8]. The IEEE Code of Ethics requires us "to hold paramount the... health and welfare of the public" [9], and therefore, the PIAE should not damage our user base's hearing without their knowledge. To this effect, the volume of the PIAE's output audio must be clipped at 100 decibels and it must warn users that listening to sound louder than 75 decibels could damage their hearing [8].

We may also encounter issues relating to the power unit. Alkaline batteries, which are used for the power unit, may leak or explode when used incorrectly [10]. Therefore, we intend to build a housing compartment for the power unit that allows enough temperature dissipation so that there is no threat of the batteries overheating. Furthermore, we must use a voltage regulator so that the batteries are not overloaded, and ensure that no short circuit or charging goes on during the equalizer's operation. Finally, the housing compartment should be designed to minimize damage to the user should the batteries malfunction.

## 6.2 Accidental or Intentional Misuse

The concern with accidental or intentional misuse is the scenario where a user increases the volume of the audio they are listening to by unsafe amounts using the PIAE. As stated before, we intend to mitigate this by clipping the volume of the PIAE's output, and by warning users if the audio output of the PIAE is greater than 75 decibels.

# References


[1] "Vital Signs: Noise-Induced Hearing Loss Among Adults — United States 2011–2012". CDC, https://www.cdc.gov/mmwr/volumes/66/wr/mm6605e3.htm. Accessed 14 Feb. 2021.

[2] Large commercial equalizer. https://www.cheapham.com/8-band-eq-by-w2ihy-special/. Accessed 14 Feb. 2021.

[3] Small Portable equalizer.https://www.amazon.com/Syba-SD-DAC63106-Control-Headphone-Amplifier/dp/B00TYYMHQS. Accessed 14 Feb. 2021.

[4] Aug 06, Joe Albano on, et al. Monitoring Latency (How Low Can You Go?). http://ask.audio/articles/monitoring-latency-how-low-can-you-go. Accessed 17 Feb. 2021.

[5] Scipy.Fftpack.Fft — SciPy v1.6.1 Reference Guide. https://docs.scipy.org/doc/scipy/reference/generated/scipy.fftpack.fft.html. Accessed 1 Mar. 2021.

[6] Scipy.Fft.Rfft — SciPy v1.6.1 Reference Guide. https://docs.scipy.org/doc/scipy/reference/generated/scipy.fft.rfft.html#scipy.fft.rfft. Accessed 1 Mar. 2021.

[7] Fast Fourier Transform (FFT). http://www.cmlab.csie.ntu.edu.tw/cml/dsp/training/coding/transform/fft.html. Accessed 1 Mar. 2021.

[8] "Hearing Loss - Symptoms and Causes." Mayo Clinic, https://www.mayoclinic.org/diseases-conditions/hearing-loss/symptoms-causes/syc-20373072. Accessed 14 Feb. 2021.

[9] IEEE Code of Ethics. https://www.ieee.org/about/corporate/governance/p7-8.html. Accessed 14 Feb. 2021.

[10] "Alkaline Batteries Handling Precautions for Safe Use." FDK, https://www.fdk.com/battery/alkaline_e/caution/. Accessed 15 Feb. 2021.