The City College of New York
The Grove School of Engineering

Noise Dosimeter

Group B: Power Beats

Andrew Birnbaum, Kristian Jacome, Daniel Khalil,

Mohigul Nasimova, and Sebastian Sauco-DeLima

BME 45000: Senior Design II Spring 2023

Department of Biomedical Engineering, The City College of New York

Instructors: Professor Carriero, and Professor Ahmed

Sponsor: Claus-Peter Richter, Northwestern University

Table Of Contents

Abstract

Human auditory frequency falls in the range of 20Hz – 20kHz. Within those frequencies, permanent damage, such as sensorineural hearing loss, can occur because of accumulating noise exposure. However, not everyone is aware of this accumulation. To reduce the likelihood of damage, there is a need for an affordable, wearable, easy-to-use device for continuous feedback on noise exposure experienced by a person.

In order to create a device that would fulfill this need, we researched current devices and patents of sound dosimeters and created our own design specification that would entail the goals to be fulfilled by our device. Our proposed device is a noise dosimeter that follows OSHA regulations for daily use. The layout of our noise dosimeter centers around using both the Electric Condenser microphone (ECM) and Micro-Electro-Mechanical system (MEMS) microphone in order to cover the frequency range heard by a human ear. From the concept list, the "DoseClip" device met most of the requirements using the Pugh method. Feasibility testing for this device involves evaluating the functionality of every component used in making the noise dosimeter and ensuring the proper integration of them together.

Background

Physiology of the Ear

Sound waves are the mechanical vibrations traveling through the air and are perceived by the human ear which leads to the hearing of different sounds (1). The human ear consists of three regions; the external ear, the middle ear, and the inner ear (1). The outer ear consists of the pinna, and the external auditory canal (1,2). The outer ear helps in collecting and channeling the sound waves to the middle ear from sources in front of the person more than behind the person (1,2). The middle ear consists of multiple parts including the tympanic membrane (also known as the eardrum), the middle ear cavity, and the ossicular chain (1,2). The ossicular chain consists of three small bone structures called; malleus, incus, and stapes(2). When sound is perceived by the ear these bones vibrate against the tympanic membrane which causes the sound waves to transfer to the inner ear (1,2). Also, these structures ensure that the sound waves are impeded to match the environment of the inner ear's fluid (1). The inner ear is mainly composed of the cochlea, which is a bone that looks like a snail shell, fluid compartments (containing perilymph and endolymph), and the hair cell (1,2). The hair cells are located at the bottom of the cochlea and contain stereocilia (1,2). When the sound waves are transferred to the fluid inside of the cochlea these hair cells start to vibrate according to the different characteristics of the sound wave vibrations and then the vibrations are transduced to electrical signals to the brain (1,2).

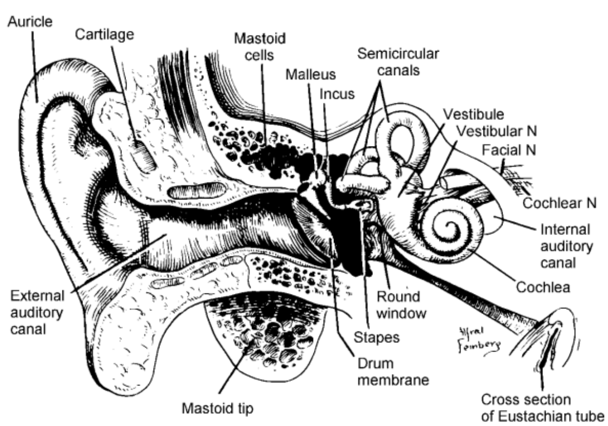


Fig. 1: Anatomy and physiology of the ear (2)

Types of Hearing Loss

As of now, we know that hearing loss can be caused by both environmental and genetic factors. The classifications are made based on the auditory area damaged. The four types of hearing loss include conductive, sensorineural, mixed, and auditory neuropathy spectrum disorder. Conductive hearing loss (CHL) is damage or any prevention of sound to the outer or middle ear. This can include the buildup of earwax of fluid, damage to the eardrum or ossicles, or a birth defect. Sensorineural hearing loss (SHL) is occupied by the inner ear and the acoustic nerve which focuses on the firing of the sound waves to the brain rather than the mechanical aspect of the ear. This form can be caused by loud noise exposure, ototoxic medication, illness, aging, or genetic causes. Unlike CHL, SHL is permanent and would require the assistance of implantation. Mixed hearing loss is a combination of both CHL and SHL, where all of the ears are affected. Lastly, Auditory Neuropathy Spectrum Disorder (ANSD) is a special classification since it can detect sound but due to the damage in the inner ear or the nerve, the sound doesn't fire properly for the brain to comprehend. This can be due to damage to the inner ear that hinders its ability to transmit sound information, mutation of genes, or general damage to the auditory nerve (3).

Aside from the classifications, the degree of one's hearing loss is essential in the treatment and prevention method. The degree of hearing loss can range from mild, at 26 to 40 dB, to profound, above 91 dB. This shows that for mild hearing loss, sounds have to be in the range of 26 to 40 dB to be able to hear them. With profound hearing loss, the only sounds one can transmit are above the regular OSHA threshold of 90 dB (4).

Human Hearing Range

The Frequency range for human hearing is 20Hz-20kHz; the upper limit in average adults is often closer to 15–17 kHz (5). Human speech falls into the 500Hz-4kHz range. The threshold of hearing, the quietest audible sound, is 0 dB, which is equal to 20 micropascals of pressure at 1kHz.

The threshold of pain, the loudest audible sound before pain, is 140 dB, which is equal to 200 pascals of pressure at 1kHz. This is nearly 10 million times louder than the quietest sound. Most other common sounds fall between this range (Table 1). Sensorineural hearing loss gives off an audiogram with a notch at 3kHz-6kHz with 3kHz-4kHz being the earliest affected (Fig. 2). Hearing loss in this range, the audible speech frequency range, is debilitating for many adults.

The A-weighting network was designed to account for the ear's natural sensitivity to middling frequencies. It attenuates the low-frequency noises as they do not have as much of an effect on the ear. (Fig. 3) This filter gives a better approximation of the consequences of injurious frequencies on the ear and may help in protecting adults from hearing loss. The curve of the A-weighted filter is designed to be directly opposite the hearing loss "notch" seen side by side (Fig. 2 & Fig. 3). The A-Weighting filter is a mandatory ANSI requirement for any approved noise-measuring device. The different frequencies will be attenuated or amplified based on the A-Weighting curve (Fig. 3).

A-Weighting Specifications:

- Below 1000 Hz, the sound levels are attenuated
- At 1000 Hz, there is no gain or attenuation

- Between 1000 and about 6000 Hz, the levels are amplified by a few decibels
- At about 6000 Hz and higher, the sound levels are attenuated (6)

Table 1: Common Sound Environments' Decibel and Pressure Levels (4)		
Common Sound	**dB Level**	**Pressure Level**
Threshold of Hearing	0 dB	0.00002 Pa
Business Office	60 dB	0.02 Pa
Shop Noise	80 dB	.2 Pa
Large Truck	94 dB	1 Pa
Jackhammer	100 dB	2 Pa
Airplane Take-Off	120 dB	20 Pa
Threshold of Pain	140 dB	200 Pa

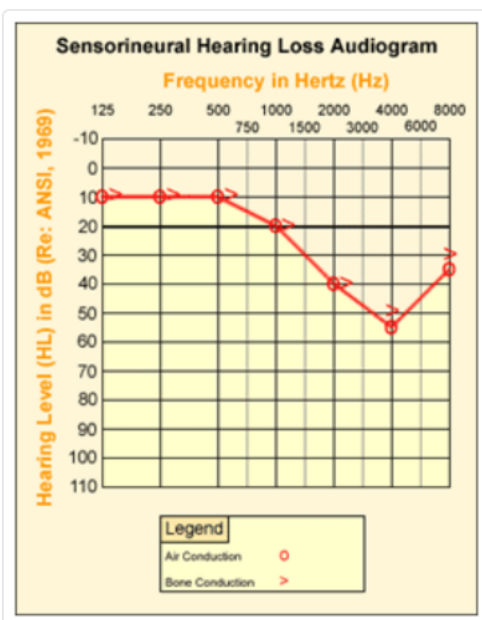


Fig. 2: Sensorineural Hearing Loss Audiogram that displays a "Notch" at from 1kHz-8kHz (4)

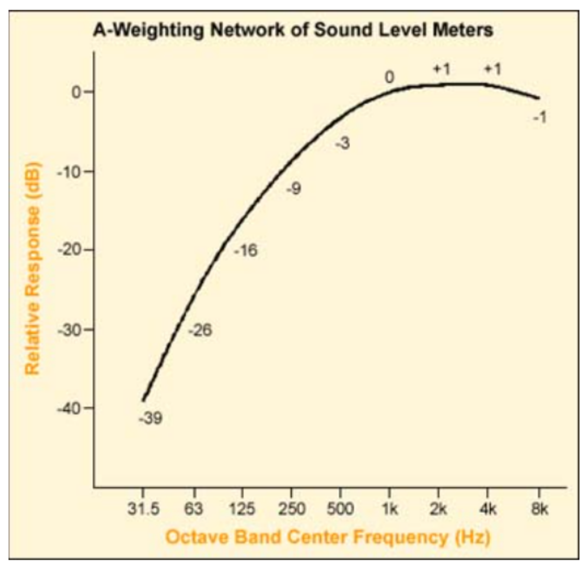


Fig. 3: A-Weighting Curve that attenuates frequencies below 1kHz and amplifies above 1kHz - 8kHz (4)

Side Effects

There are many side effects of hearing loss, such as problems with communication, speech, and cognition leading to social isolation and loneliness. Due to these side effects, many elderly people lack lucidity and coherence. This, in turn, can lead to depression for the affected and their family.

Also, there are many non-elderly people who suffer from hearing loss, yet still, need to find a mode of employment. This becomes difficult as communication is a core element of workplace compliance, which then leads to a higher rate of unemployment, despite there being available jobs.

As a deaf person does not use the parts of their brain allocated for hearing, essential neuronal pathways become neglected causing these parts of the brain to atrophy which may cause dementia (7). Symptoms of dementia include forgetfulness, limited social skills, and thinking abilities so impaired that they interfere with daily functioning (8). This further promotes social isolation and loneliness.

In order to prevent hearing loss and the altercations that results from it, noise dosimeters are used to assist the user in monitoring their hearing habits and to warn them about dangerous levels of sound that they are being exposed to.

Noise Dosimeter

A noise dosimeter is a device that detects, processes, and determines outputs of sound data for the purpose of informing users of noise exposure over time in order to protect them from hearing loss. The dosimeter uses a pressure sensor or microphone to detect the sound waves, often in all directions (omnidirectional). The microphone has an exposed diaphragm that oscillates as a result from a change in sound pressure causing a fluctuation in capacitance and current. The output that results from it is an alternating voltage. Sound waves have a very minimal amplitude when measured and therefore output small voltage readings which require amplification to allow computers to register sound levels. In addition, the circuit will include filters to minimize the DC currents' impact on the signal and reduce unwanted low and high frequencies based on microphone specifications as well as A-weighting the signal. A microcontroller is usually used to process the amplified signal into a digital signal through its analog-to-digital converter (ADC) component pins. The microprocessor will use implemented code to calculate noise exposure and give feedback to the user (Fig. 4 & Fig. 5). Most ADCs have an 8-bit per sample (1-byte/sample) conversion rate. Using a minimum sampling rate of 44.1 kHz, the size of the digital audio data that the device will register for 10 hours is around 1.56 GB. A noise dosimeter that saves audio recordings before processing the data will have memory storage larger than the mentioned value.

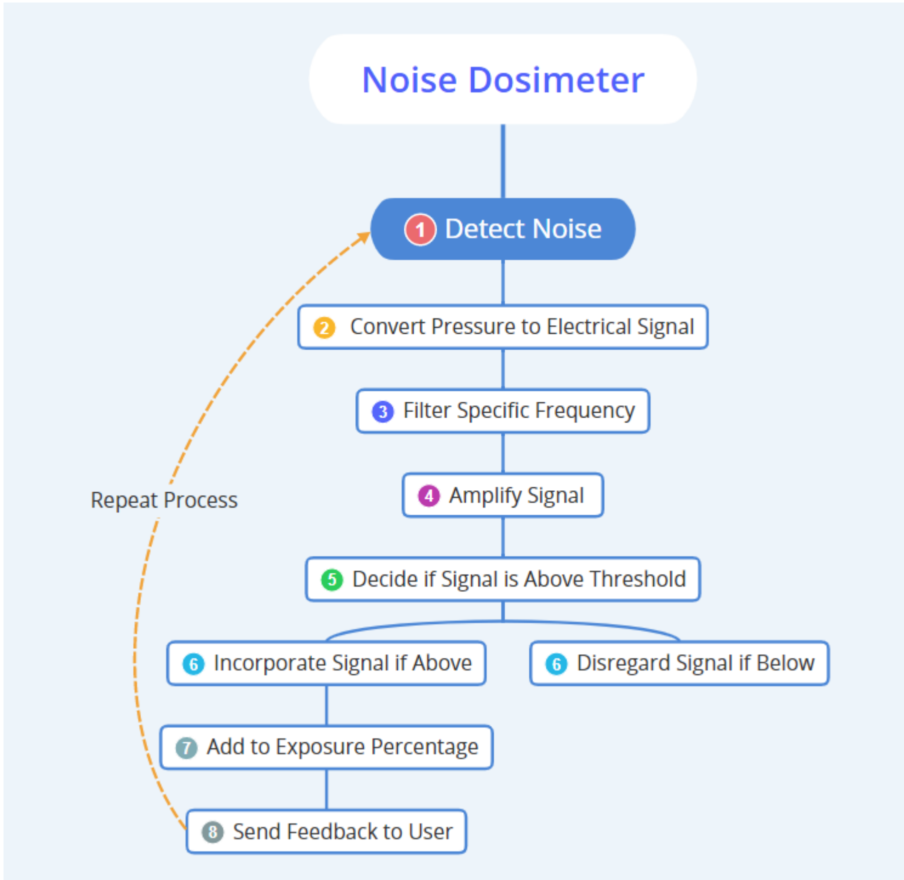


Fig. 4: Flowchart of Noise Dosimeter Process from Detection to Feedback

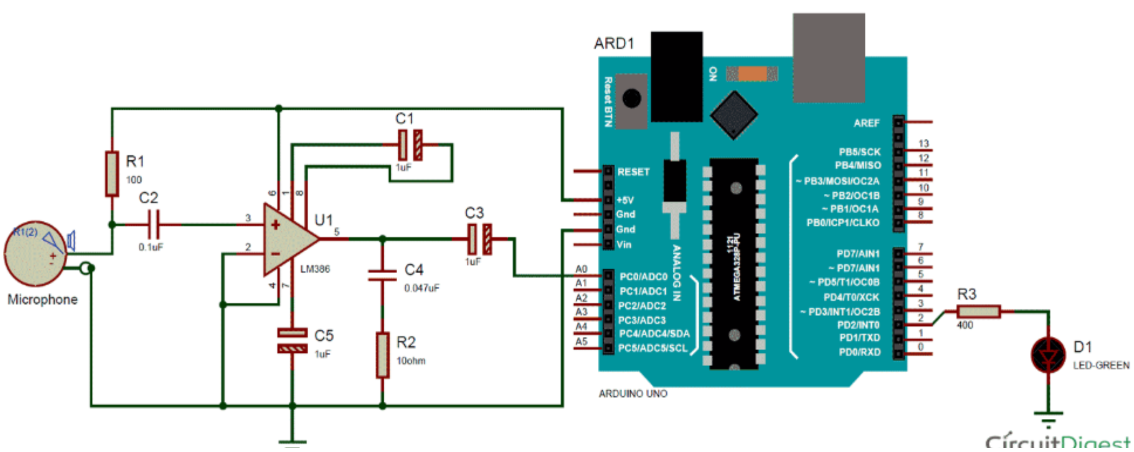


Fig. 5: Schematic of a Sample Noise Dosimeter Circuit Connected to an Arduino

OSHA Noise Threshold Standards

Our noise dosimeter will be following guidelines put in place by OSHA to protect the hearing of adults. These devices must be worn to determine their daily personal noise dose to ensure that safety regulations are followed. The OSHA standard has an exchange rate of 5 dB which means that with an increase of 5 dB to the sound the person is being exposed to the exposure time allowed is cut in half. Also, it follows the A-frequency weighting in which the frequencies that the ear is less sensitive to will be attenuated, and the frequencies that the ear is more sensitive to will be amplified (weighted more heavily) (Fig. 3). Furthermore, it has a response of either slow or average, which means how often the measurements of the sound are going to be taken. Slow indicates 1-second intervals, while average is a mean of measurements taken over time. In addition, the OSHA standard follows a criterion level of 90 dBA and threshold of 80 dBA. This means that the noise volume at which 8 hours exposure will be 100% of the daily dose is 90 dBA, and the quietest noise volume that is integrated into the daily exposure is 80 dBA while any noise under 80 dBA is disregarded. The maximum acceptable continuous noise level allowed is 115 dBA and the maximum acceptable instantaneous (impulse) noise level allowed is 140 dBA. Lastly, the accuracy and precision is considered to be ±2 dB for noise measurements, this is called type 2.

Other Noise Threshold Standards

As discussed above, OSHA has a 90 dBA criterion level for an 8-hour noise exposure period with a 5 dB exchange rate. There are two other standards that are worth mentioning; NIOSH and EPA. NIOSH has an 85 dBA criterion level for an 8-hour noise exposure period with a 3 dB exchange rate while EPA has a 70 dBA criterion level for a 24-hour noise exposure period or a 75 dBA criterion level for an 8-hour noise exposure period with a 3 dB exchange rate. The OSHA and NIOSH standards divide their allowed noise exposure around a work-based time frame: 8 hours a day for 5 days a week which amounts to 261 days of the year. These standards assume that the individual spends the other 16 hours in the day, as well as their weekends, in quieter conditions. The EPA standard, however, uses 24-hours in day for a 7-day week time frame. This amounts to 365 days a year. Therefore, the EPA dB range limitation is set at a lower dB to offset the longer periods considered of continuous noise exposure (9). The OSHA standards were chosen as they are a requirement from our sponsor, therefore it will be used in our device.

All of these standards apply to both men and women. There is no specific age for the standards but it should be noted that the NYS department of labor's guidelines is for workers 18 years and older. 16 & 17-year-olds can work at a limited capacity with proper training but most standards are for full-time workers (10). There is research that shows minimal higher damage in children than adults in the high dB range. As a precaution, the standards for children are often lowered to 80 dB to protect their developing ears (11).

Problem Statement

Throughout the day, people are exposed to dangerous sound pressure without taking into account how long they are being exposed. They are able to perceive the loudness of a sound as an indicator of pain, but not perceive the damaging effect of repeated exposure on hearing. Although people are not aware of it, these damages accumulate. An informed individual who receives feedback on the sound dose received daily is able to change their behavior to minimize the sound dose and reduce the risk of hearing damage.

Need Statement

There is a need for an affordable, wearable, easy-to-use, device to provide continuous feedback on a daily noise dose based on OSHA standards to adults.

Product Design Specification

Performance

- The device will be a noise dosimeter that will measure 60 – 115 dB SPL environmental sound in all directions and will follow the Occupational Safety and Health Administration's (OSHA) noise standard (Table 2)

Table 1: Decibel Level vs Exposure Time Allowed (12)

Sound level (dBA, slow response)	Duration per day (hours)
85	16
90	8
92	6
95	4
100	2
105	1
110	1/2
115	1/4

- The device will follow OSHA Noise Dosimeter Specifications:
    - Exchange rate: 5 dB
    - Frequency weighting: A
    - Response: slow or averaging
    - Criterion level: 90 dBA
    - Threshold: 80 dBA

- The Accuracy and Precision for the microphone is ± 2 dB SPL as Type 2 SLM which is used for general purposes
- The range of recorded sound data uses 10 to 18 hour time frame per day (13)
- The recorded sound data is stored in at least 2 GB of memory prior to processing the data
    - Daily recordings are erased after data is processed
- The device will be wearable such that it can be attached to the user while simultaneously being comfortable and easy to place and remove from oneself
- For the most accurate measure possible, the device will be placed within 10-30 cm of the user's ear
- The device will send back visual or auditory feedback to the user that will be useful to the average adult
- The product will calculate the noise level based on sound pressure readings. As certain sound dose thresholds are reached (ex. 50%, 75%, 90%), warnings will be implemented to inform the user of their current noise dose

Durability

- The device will be durable for the entirety of its life cycle, which will be between 2 to 5 years

Environment

- The device will have an enclosure to be properly protected from external environmental factors for everyday use
- The casing will have an IP65 rating indicating protection against jets of water, and completely preventing dust/dirt from entering the inside of the device

Dimensions

- The size of the device should be no larger than 8cm x 8cm x 4cm
- The device should weigh less than 200 grams

Usability

- The device will be easy to use with feedback being presented in an easy-to-understand manner.

- The user will be able to access the processed data through the use of a user interface. Included data will be sound over time, noise dose percentage available to the user, and information about a recovery period
- It will be provided with a user manual to instruct the user on how to use the device
- The comfort and usability of the device will be assessed through the use of a modified SUS test

Safety

- The device will have a casing component to prevent each component from coming into contact with the user
- All the components must be secured properly inside the housing to avoid damage to the individual parts
- The product is a surface device possibly in contact with the skin and the individual components must satisfy a series of standards:
    1) IEC-61010 on product safety for products working at a set temperature range to see if the product will not cause burns or fires to the user (14)
    2) ISO 10993-1 standard on risk evaluation and biocompatibility testing to prevent adverse risks on the body, such as cytotoxicity, sensitization, and irritation
    3) IEC 63203-101 standard for wearable technology shall also be followed, dictating that the finished product needs to be resistive and durable when worn by the user and during maintenance (15)

Cost

- The market value of the device will be less than 250 USD

Intellectual Property Claims

Our device is a wearable noise dosimeter monitoring system with built-in indications in proximity to the ear. Since our device focuses on collecting environmental sounds, having a 10-30 cm proximity to the ear will give us a better representation of the sound going into the uncovered ear. The further away from the ear the larger the dB discrepancy (16)

Additionally, our device is made specifically for commercial to be sold at a reasonable price for an average human being to get daily monitoring of sound exposure. Since many noise dosimeters are used for occupational purposes only, e.g. construction sites, for the safety of workers' auditory systems, and sold at a higher cost, making them non-affordable to the average middle class.

The algorithm uses an allowance system triggered by a predetermined sound range based on a set threshold of noise exposure and duration. Our software will have the unique ability to set an allowable noise exposure to the user for a full day and warn them when exposed to a set threshold and at pre-set percentages. Having an allowance system keeps the user aware of the overall damage to their ear in a given day rather than instantaneous data given by most devices on the market.

Prior Art

Devices on Market

The **dBadge2** is a personal noise exposure meter used mainly for construction workers (17). The device measures all noise dose parameters simultaneously, not just a single standard. Standards include OSHA, ISO, MSHA, and ACGIH guidelines. The device clips to the shoulder and has an LCD display (Fig. 6A, 6B). The device connects to the Airwave software on a mobile device using Bluetooth to remotely monitor multiple dosimeters, without disturbing the wearer. (Fig. 6A) This application is for the project manager to be able to monitor all of his workers simultaneously. The device also used Noise Safe software to create data reports of all workers at the end of a shift/week (Fig. 6C).

Product Specifications:

- Linear Operating Range 55 to 140.3 dB(A) RMS (average is A-weighted)
- Dynamic Range 96 dB
- Peak Measurement Range 90 to 143.3 dB (C or Z weighted)
- Time Weightings (Fast, Slow, and Impulse)
- Exchange Rate (3 dB or 5 dB)
- Threshold (70 to 90 dB in 1 dB steps)
- Criterion Level (70 to 90 dB in 1 dB steps)
- Criterion Time (8hr)

The dBadge2 costs $2,506 each and $4,917 for 3. The dBagde2 Pro version with added features like audio recording and real-time octave band filters costs $3,425 each and $7,167 for 3.

A

B

C

Fig. 6: dBadge2 (17). (**A**) AirWave application receives data from device via Bluetooth. (**B**) dBadge2 device measures all noise dose parameters using a multitude of different standards. (**C**) NoiseSafe software allows project managers to monitor all workers simultaneously without disturbing the wearer.

The **doseBadge** is a wireless personal noise dosimeter mainly for construction and industrial workers (18). The doseBadge will measure, store and calculate the parameters essential for compliance with either OSHA, NIOSH, or ISO regulations, and must be set before use. Important measurements made by the doseBadge include the time-weighted average, exposure as a percentage, A-weighted average sound level, maximum peak pressure, and average noise level every minute. The doseBadge will store a Time History, or Noise Profile, throughout the measurement using the NoiseTools software also creating data reports to summarize workday use for the project manager. The interface will have threshold values and graphs of average dB exposure over time (Fig. 7A).

Product Specifications:

- Decibel Range
    - 70 dB(A) to 130 dB(A) RMS (average is A-weighted)
    - 120 dB(C) to 140 dB(C) Peak (impulse is C-weighted)
    - 115 dB(A) Maximum Sound Level Exceedance
- Exchange Rate (3dB, 4dB, or 5dB)
- Criterion Level (80dB, 85dB, 87dB, 90dB)
- Criterion Time (8hrs, 12hrs, 16hrs, 18hrs)
- Threshold (None, 80dB, 85dB, 90dB)
- Time Weighting (None, Slow)

The doseBadge has been designed to survive use in the toughest and harshest environments. There are no cables, controls, or displays to damage, and the microphone, battery, and electronics are all housed in a robust and lightweight metal case that is mounted on the shoulder of the user (Fig. 7B). The doseBadge can be set to indicate when the noise dose has exceeded 100%. Under normal conditions, the blue LED flashes once every second. When the noise dose limit is exceeded, it flashes twice every second (Fig. 7C).

The doseBadge costs $1,776 each, and 5 for $4,276, usually a few are purchased to track multiple workers. The doseBadge Pro costs $2,220 each and 5 for $6,353. This comes with an additional smartphone application that displays exposure values for easier use.

A

Cirrus Research plc

Measurement Summary Report

Name	2						
Time	11/5/2006 12:36:00 PM	**Person**		**Place**		**Project**	
Duration	07:08:09						
Instrument	5117, CR:110A						

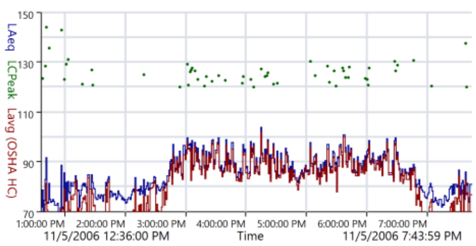
Calibration

Before	11/5/2006 12:35 PM	Offset	0.10 dB	**After**	Offset

Peak & Max Values	
LCPeak	143.7 dB

Custom Integrator	
LAeq	90.1 dB
Dose	290.0 %
LAE	134.0 dB

OSHA HC	
Lavg	86.2 dB
TWA	85.4 dB
Dose	52.0 %
Est. Dose	58.0 %



B

C

Fig. 7: doseBadge (18). (**A**) The doseBadge will store a Noise Profile, using the NoiseTools software and create data reports to summarize, including averaging measurements and exposure percentage. (**B**) The microphone, battery, and electronics are all housed in a robust and lightweight metal case that is mounted on the shoulder of the user. (**C**) The indicator has a blue LED that can be set to indicate when the noise dose has exceeded 100%. When the noise dose limit is exceeded, it flashes twice every second.

The **MiNUT** device is an at-home-based system that measures the environmental sound within the home. It uses an interface that can be accessed through an app on a phone. It has a customizable threshold and alarm time before warning the user. It monitors and alerts the user when the sound reaches above the threshold at 10-minute intervals. The final protocol sends an automatic responder to the user's home (preferably a common area like a living room) to let them know of the continuous exposure. The standard they use is customizable, hence, the user has the power to control the threshold of sound in their home before getting alerts. The duration of which the notifications are sent is also customizable with a range from 5 to 15 minutes.

Aside from the noise monitoring service, they also provide temperature and occupancy tracking. These values can be seen in the app. The target of these services is to control the user's environment in case of neighbor complaints or accusations and to have controlled parties/events.

The pricing on the system is charged on a monthly subscription. The 'Standard' package includes a $50 sensor and a monthly charge of $10, bringing to a total of $170 for the first year. The 'Pro' package provides a free sensor with a monthly charge of $15, bringing to a total of $180 per year (19).

Device Patents

"The ultra-low power dosimeter" patent entails that the circuitry of the device would be small and draw minimal power which will enable it to be integrated within smart devices or headphones. The device would respond to sound from an audio device or a microphone by producing an input voltage that gets amplified by a logarithmic amplifier which triggers a second voltage that represents a noise dose. If the second voltage reached a certain predetermined threshold then the device would produce an audio signal interrupting the audio being played by the device through an earphone or utilize a blinking LCD to warn the user about their exposure (Fig. 8). This dosimeter device is designed to use a battery of 1.4 volts to function (20).

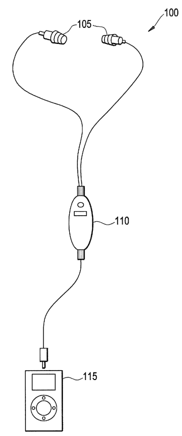


Fig. 8: The low Power Device (20)

"A Method and system for noise dosimeter with quick-check mode and earphone adapter" the patent states that their noise dosimeter will be able to determine the long-term noise exposure based on a measurement of the sound taken from a short period of the sound exposure for a sound that would have the same nature over an extended period of time. Also, this sound signal would be from a sound-playing device into earphones or through a mic that is acoustically sealed around the earphones. The dosimeter is directly connected to the sound source (Fig. 9A). This connection is achieved through the utilization of an electric or acoustic adaptor. The processing is done through the obtaining of an audio signal as AC (analog signal) which is converted using an RMS detector to a digital signal and relayed to 2 amplifiers (Fig. 9B). The first amplifier will amplify the whole input signal from the RMS while the second amplifier will further amplify the signal coming from the first amplifier but focusing on the lower level signals. This amplified signal is then processed by a processor and the information is displayed on the device to the user (21).

A **B**

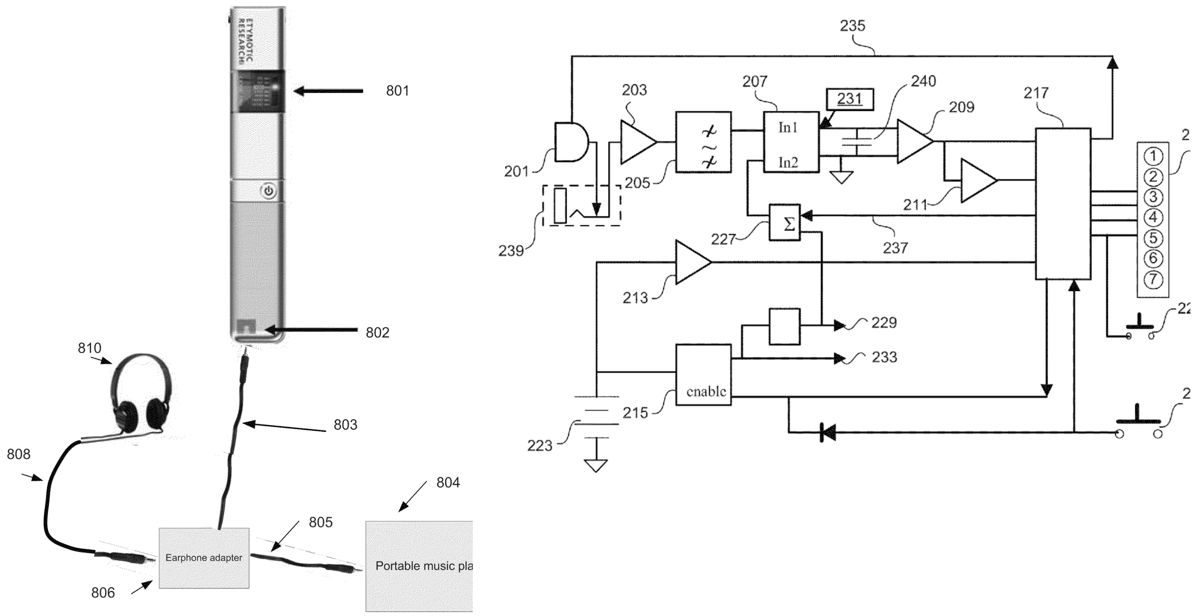


Fig. 9: A Method and system for noise dosimeter with quick-check mode and earphone adapter (21). (**A**) The dosimeter is directly connected to the sound source. (**B**) The processing is done through the obtaining of an audio signal as an analog signal which is converted using an RMS detector to a digital signal and relayed to 2 amplifiers.

The "Integrated wearable noise dosimeter" patent involves a device that collects sound information and another type of biometric data which correlates between both sets of data in order to observe and study the physiological information. The audio information is collected via a microphone and then the data is amplified using a logarithmic amplifier which is then converted to digital data using an ADC converter. The device collects the data at different intervals of time and the output digital information is then transmitted to a computing device for the data to be correlated with the other biometric data. The biometric data is collected using a different type of sensor that is appropriate for the intended biometric information (Fig. 10).

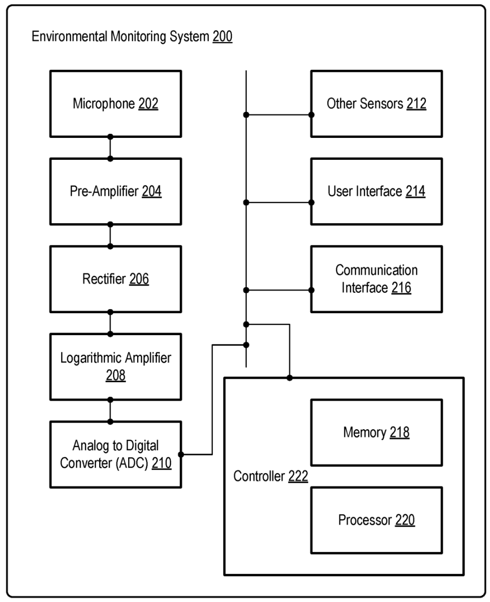


Fig. 10: The logic of the Noise Dosimeter device (22)

"In Ear Noise Dosimeter" is a patent for a device that will record in-ear measurements of sound pressure levels for humans. The earpiece will be made to fit inside the ear canal and coupled with a microphone which will receive sound pressure signals from within the ear canal (Fig. 11A). The earpiece will be connected to an external dosimeter via a flexible wired connection (Fig. 11B). The sound pressure signals from inside the ear will be communicated to the sound monitor which will collect the data. The device will only measure the in-ear noise by isolating it from the ambient noise. The in-ear component helps get more accurate data. There was a difference of approximately 1-13 decibels of sound pressure level measured between a shoulder-measured sound dose and sound dose in the ear when an earpiece is worn (23).

A **B**

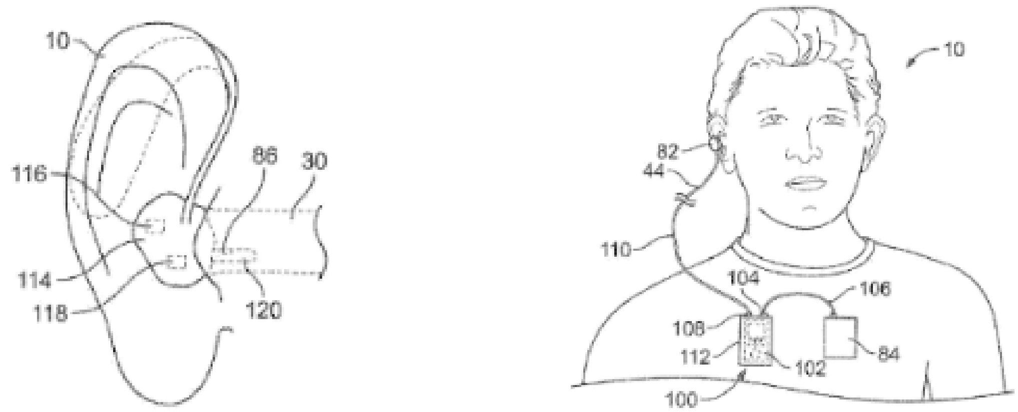


Fig. 11: In Ear Noise Dosimeter (23). (**A**) The earpiece fits inside the ear canal and is coupled with a microphone which receives sound pressure signals from within the ear canal. (**B**) The earpiece is connected to an external dosimeter via a flexible wired connection and the sound pressure signals from inside the ear are communicated to the sound monitor which collects the data.

Software Patents

Apple Inc. is creating an application that will monitor noise exposure called the Noise app (24). The app will monitor a user's noise data over time using the microphones on the iPhone and Apple Watch for environmental noise and audio exposure from paired devices such as AirPods and headphones.

Their app will provide the user with the environmental noise and audio exposure data separately. The data will display a summary of the week’s listening patterns including a decibel range for each day (Fig. 12A).

Their app will have specific settings that can help avoid high noise exposure which must be activated for use. The settings include receiving notifications when certain noise thresholds are reached (Fig. 12B), automatically lowering headphone volume, and a real-time sound meter that will give notifications about the noise level (Fig. 12C).

Their set thresholds for noise monitoring will follow WHO-ITU standards for safe listening devices. This standard, 80 dBA for 40 hrs/per week, is lower than OSHA regulations which are 85 dBA for 40 hours per week. Their app will give daily exposure limits at the following values: 5 ½ hours at 80 dB, 1 ¾ hours at 85 dB, 34 min at 90 dB, 11 min at 95 dB, and 3 min at 100 dB, and will notify the user when the daily limit is reached (Fig. 12A). The Apple Watch’s price range is \$250-\$800 and needs an iPhone to function for another \$450-\$1600.

A **B** **C**

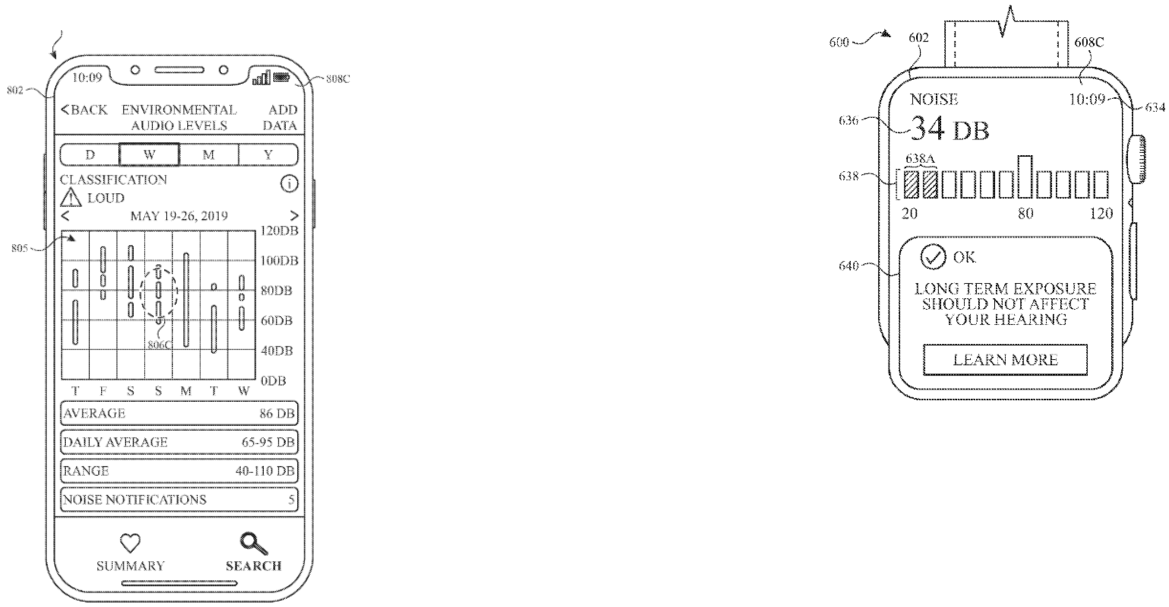


Fig. 12: Apple Noise App (24). (**A**) The app displays a summary of the week's listening patterns including a decibel range for each day (25). (**B**) The app's settings include receiving notifications when certain noise thresholds are reached (26). (**C**) The app can automatically lower headphone volume, and a real-time sound meter that will give notifications about the noise level (27).

Motorola Mobility LLC is designing a mapping software that detects noise levels in rooms, floors, etc… Once the floor model is uploaded to their server, their smartphone application will be able to measure and locate sources of noise using microphones from users' phones and uploading it to a cloud server (Fig. 13a). Their application will notify users about noisy areas and give tips on where to relocate the room/floor for quiet (Fig. 13b).

A **B**

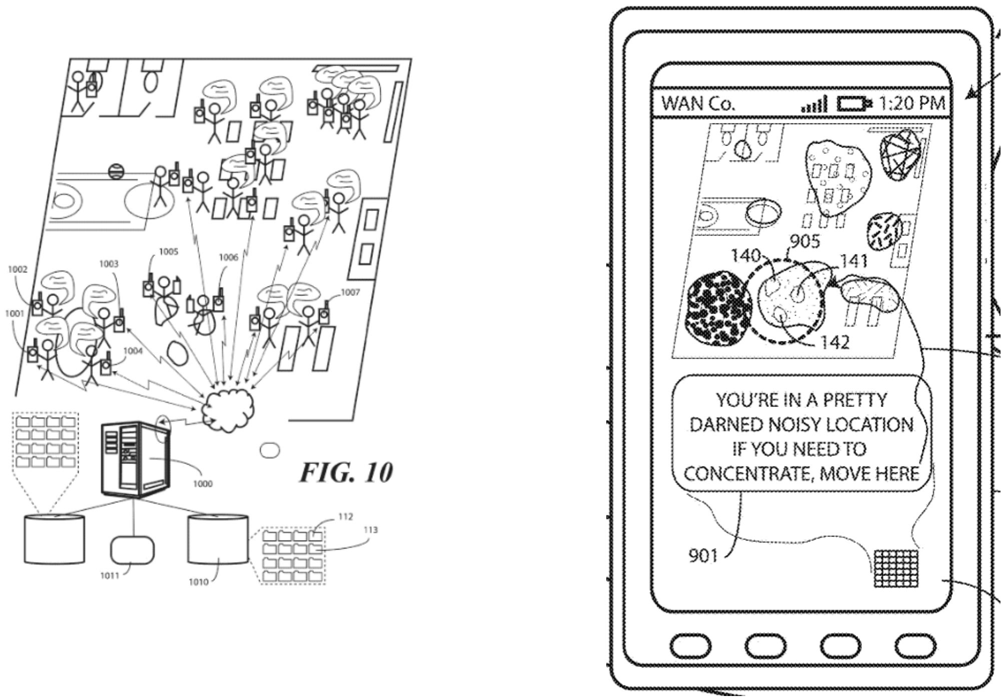


Fig. 13: Motorola App (28). (**A**) The smartphone application is able to measure and locate sources of noise using microphones from users' phones and upload it to a cloud server. (**B**) The application notifies users about noisy areas and gives tips on where to relocate the room/floor for quiet (29).

Table 3. Comparison of Our Device Against Prior Patents and Available Devices on the Market

Feature	Our Device	Dbadge2	Dosebadge 5	MiNUT	Ultra-Low Power Dosimeter	Quick Check Dosimeter	Integrated Wearable Noise Dosimeter	In Ear Noise Dosimeter	Apple Noise App
Type		Market Device	Market Device	Market Device	Device Patent	Device Patent	Device Patent	Device Patent	Software Patent
Indication based on the allowance model to the user	✓	✗	✓	✗	✗	✗	✗	✗	✗
Indication based on the noise measurement	✓	✓	✗	✓	✓	✓	✓	✓	✓
Intended for personal use	✓	✗	✗	✓	N/A*	N/A*	N/A*	N/A*	✓
Follow OSHA standards	✓	✓	✓	✓	N/A*	N/A*	N/A*	N/A*	✗
Evironmental noise	✓	✓	✓	✓	✗	✗	✗	✓	✓
Wearable	✓	✓	✓	✗	✓	✓	✓	✓	✗

*Circumstances of the feature is not provided due to its status as a patent

Workflow

The workflow process considered multiple possible approaches to building a noise dosimeter. The device has different aspects of engineering involved: mechanical, electrical, and computer/software components all bundled into this noise dosimeter. The main focuses of the workflow were data acquisition, filtering method, processing, programming, danger identification, and user experience. The concepts for the noise dosimeter design were then formed by combining these approaches into devices that fulfill the need statement and product design specifications (Fig. 14). Data acquisition is made possible with the use of different microphones. The type of microphone affects how the rest of the device will work. There are two major microphones: Electret Condenser Microphones (ECMs) and Micro-Electro-Mechanical System (MEMS) microphones. ECMs stick out of the device and give off an analog signal. ECM microphones can detect sound omnidirectionally or unidirectionally and give off an analog signal. MEMS can only detect sound omnidirectionally and can give off either an analog signal or a digital signal.

The filtering method depends on the kind of signal received. There can be an analog signal with an analog filter which will comprise electrical components (resistor, capacitors, etc.…). This signal can also be converted to digital prior to filtering using an ADC. Once the signal is digitized, filtering can be performed by the software. There are two main filters that must be implemented: Bandpass and Weighting. The bandpass is used to make frequency cutoffs that will remove frequencies outside the audible human hearing range (20Hz-20kHz). The weighting filter is used to attenuate frequencies that are not particularly sensitive to the human ear (outside 500Hz-4kHz). There are three kinds, A, C, and Z. A is used for low dB levels. C is used for high dB levels. Z is unweighted, the signal in is the signal processed. Processing will be performed on the newly filtered signal. The mode of processing depends on the signal being received. An analog processor or comparator can be used with filtered analog signals. The filtered analog filter can also be converted to digital at this point to be used in a microcontroller. This is where the main programming and processing take place. The noise thresholds and algorithms will be included in this section. The kind of processing can be fast (125ms), slow (1s), or average-based. This is how often measurements will be taken for the noise exposure.

Danger identification is the kind of noise that will be considered for the algorithm. The danger identification depends on the standards that are being used. We will be using OSHA which requires a 90 dBA threshold for 8 hours and a 5 dBA exchange rate.

Once thresholds are reached, indications must be sent to the user. The type of feedback is the user experience. It may be done in a variety of ways including LEDs, LCD, haptic feedback, and audible cues.

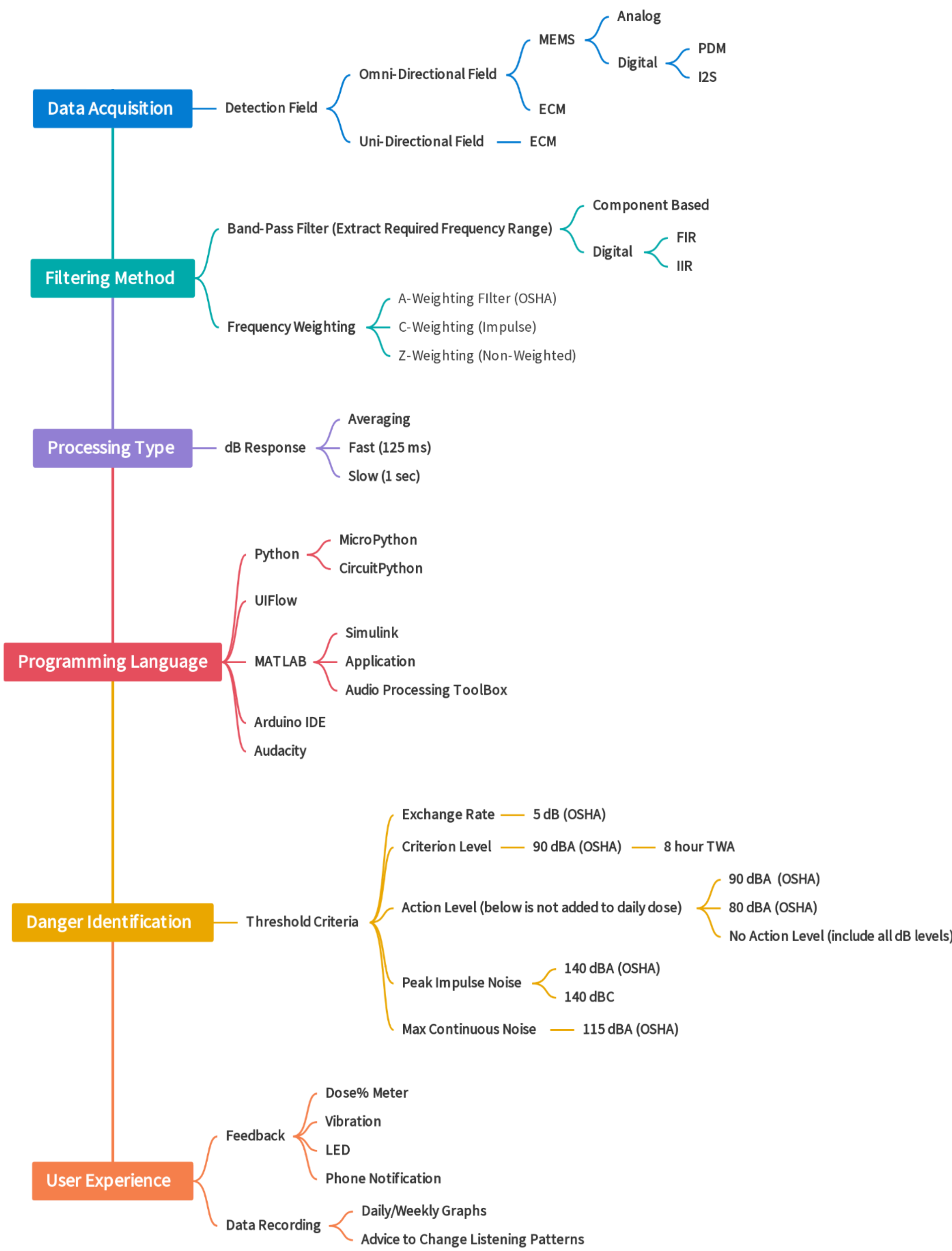


Fig. 14: Breakdown of Potential Approaches for Noise Dosimeter Functionality

Concept Map

The concept map is composed of multiple clustering levels (Fig. 15). The first level of clustering is the sound detection type. The three are digital omnidirectional, analog omnidirectional, and analog unidirectional. The next level of branching further divided the devices into software approaches, electrical approaches, or a combination of the two. The concepts are described in greater detail below. Each concept has a component list and description. The concepts chosen for further evaluation have additional justifications, specifications of components, and cost analysis.

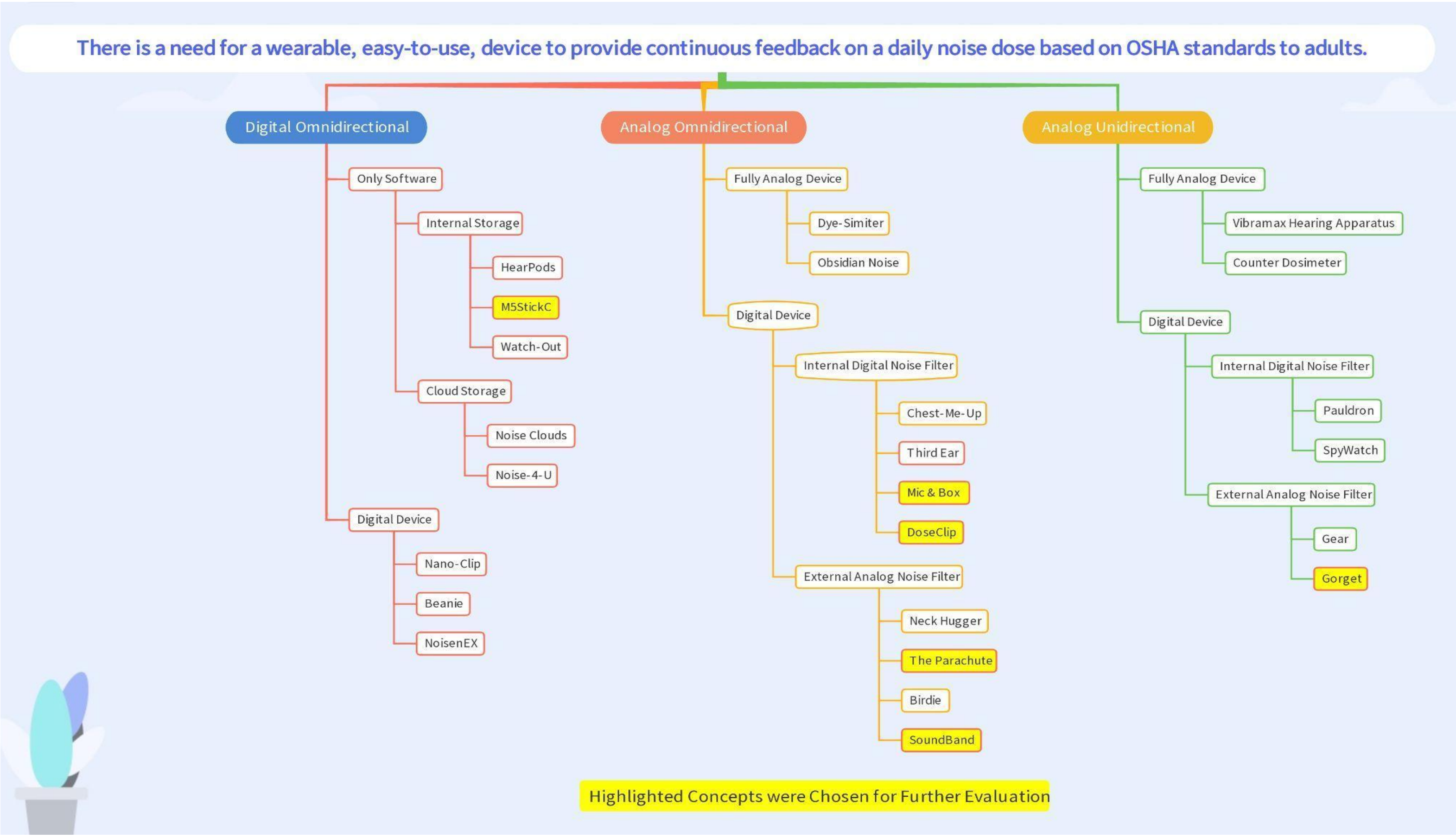


Fig. 15: Concept Map of Solutions for the Need Statement

Pseudocode

The following pseudocode is a simplification of the coding steps necessary for the creation of a noise dosimeter. The outline of the pseudocode is presented below as a flowchart (Fig. 16) and will be expounded upon in succeeding figures (Fig. 17-25). The coding is similar for all concepts described below.

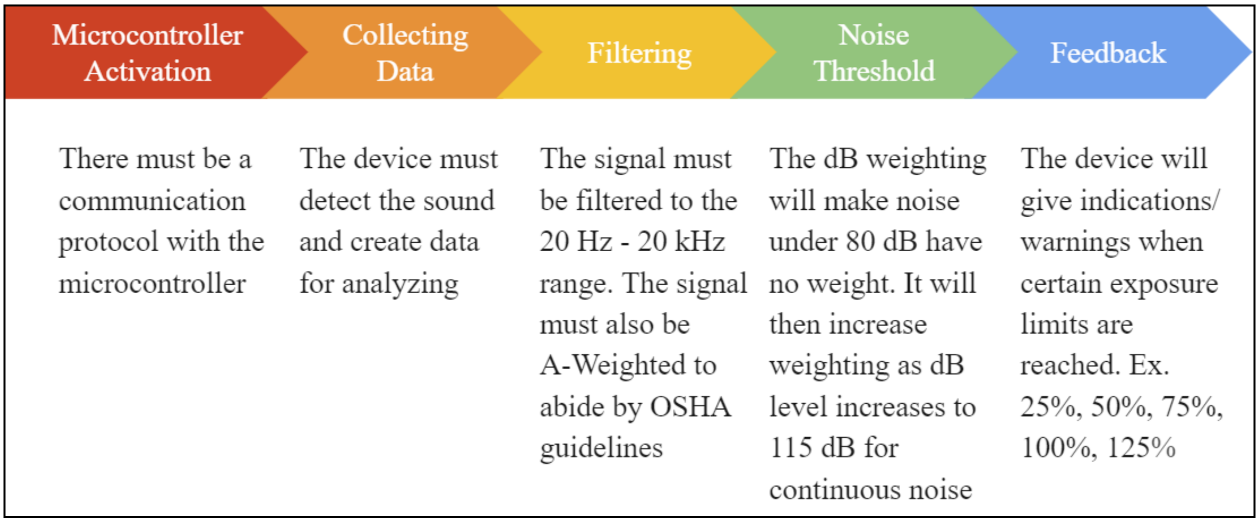


Fig. 16: Pseudocode Flowchart

1. Microcontroller Activation

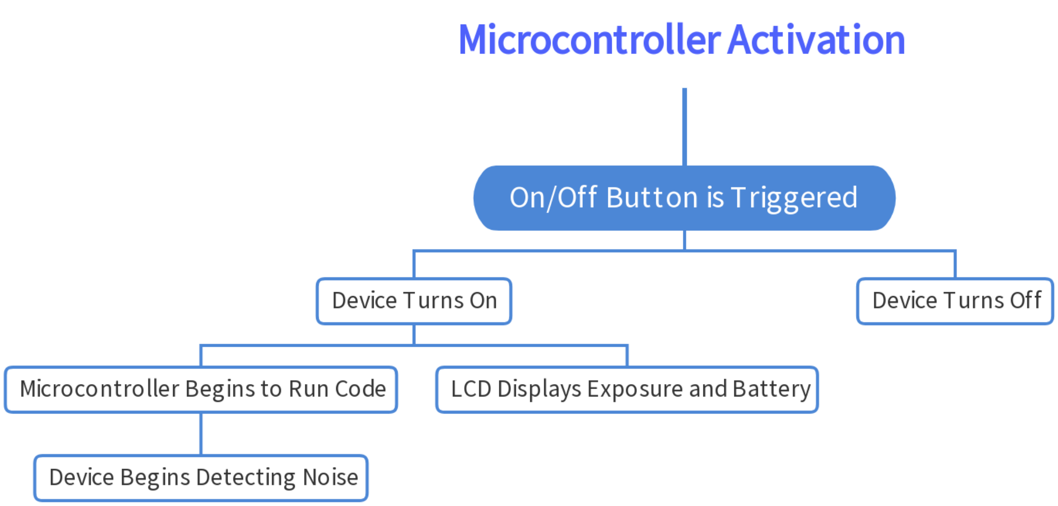


Fig. 17: Flowchart of Steps Needed to Activate the Microcontroller and Turn on/off Device

2. Bandpass Filtering

A bandpass filter will be used to remove unwanted frequencies below 20 Hz and above 20 kHz. Frequencies in-between will pass through. Filtering will be done on MATLAB using either a FIR or an IIR bandpass filter which locates the needed frequencies and suppresses the remainder of the signal (Fig. 19).

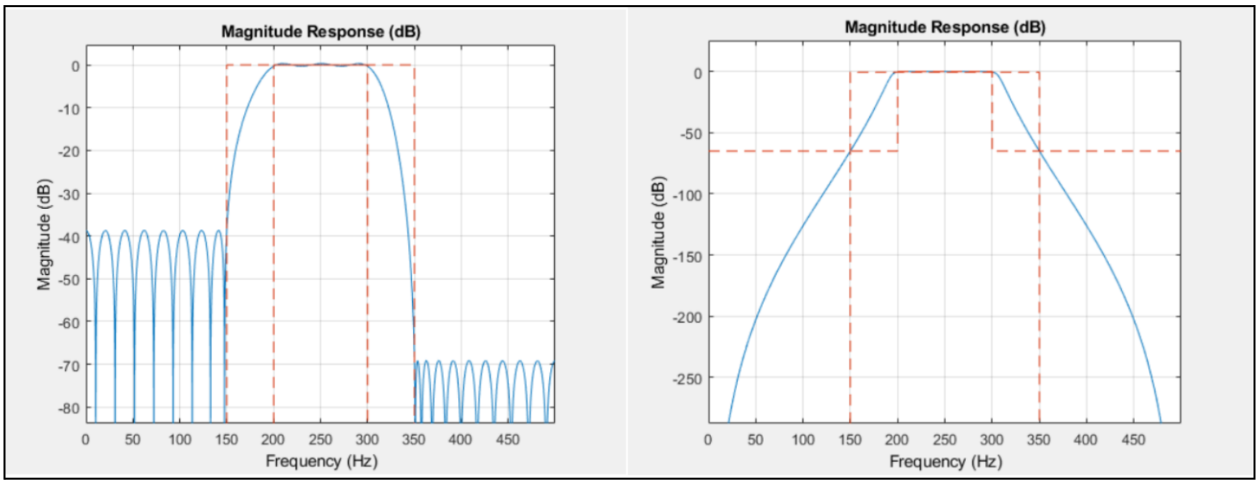


Fig. 18: Sample MATLAB FIR (Left) & IIR (Right) Bandpass Filters

3. A-Weighting Frequency (Filter)

The band-pass signal is further filtered using an A-Weighting filter which will provide the proper amplifications and attenuations necessary to abide by OSHA standards (Fig. 19). Frequencies at 1 kHz will pass through, below 1 kHz or above 6 kHz will be attenuated, and between 1 kHz and 6 kHz will be amplified. In order to accomplish A-Weighting, the signal will be broken down into octaves and then recombined after applying the filter. The below equation (Fig. 20) is the formula for combining frequencies to receive an A-Weighted dBA level. This measurement will be used for calculating threshold values. There is MATLAB software that can display an A-Weighting filter in real time which can be used to visualize the filtered signal (Fig. 21) and will be useful in designing the user interface.

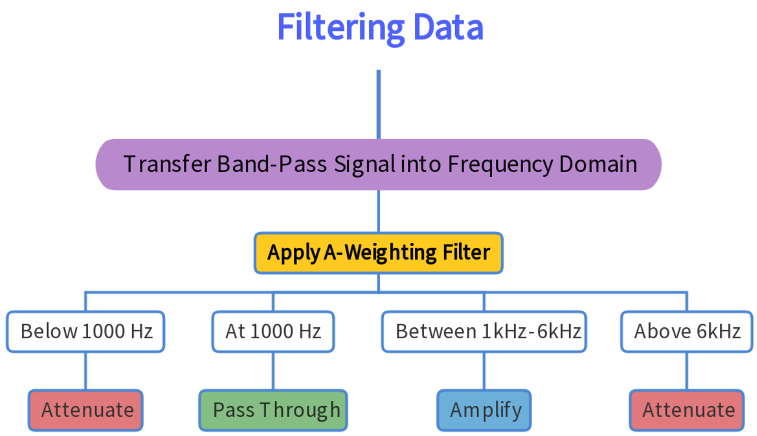


Fig. 19: Flowchart of Steps Needed to Apply an A-Weighted Filter to the Signal

$$L_A = 10\,x\log_{10}\left(\sum_1^n 10^{\frac{L_i}{10}}\right) = 10\,x\log\left(10^{\frac{L_1}{10}} + 10^{\frac{L_2}{10}} + 10^{\frac{L_3}{10}} + \cdots + 10^{\frac{L_n}{10}}\right) = A - Weighted\ dBA\ Level$$

Fig. 20: Equation used to Calculate dBA Level Using A-Weighted dB Values

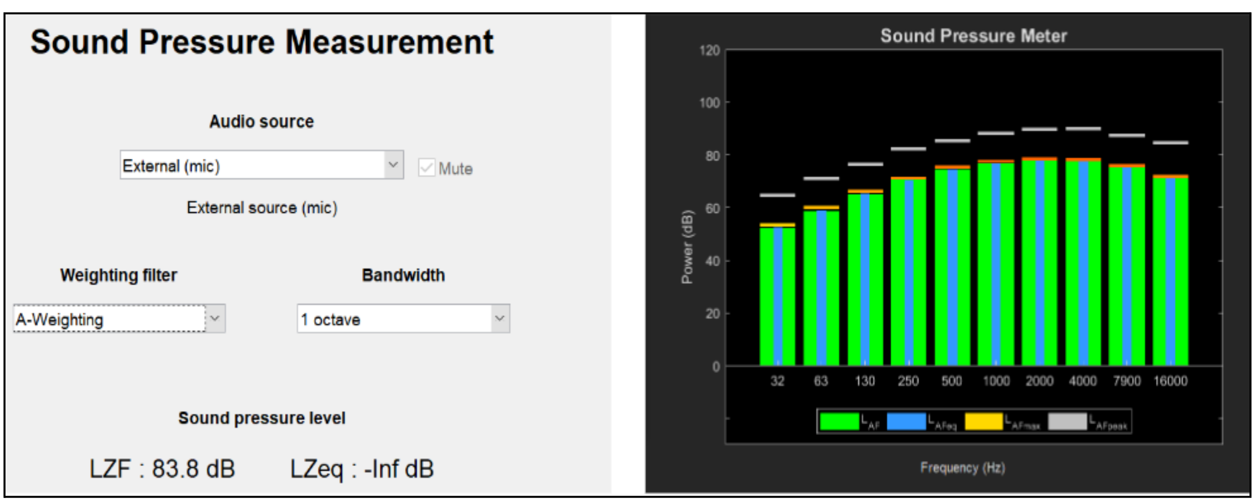


Fig. 21: Sample MATLAB Graphic User Interface (GUI) for A-Weighting Sound Level Meter

4. Weighting Specific dBA Range Data

The device will need to determine which dBA values to integrate into the exposure percentage. To accomplish this, the filtered noise measurements will be weighted by decibel level. Measurements will be added to the allowance based on sound volume. Values below 80 dBA will be disregarded (weighted zero) as they are below the OSHA's threshold level, values above 80 dBA will be integrated into the exposure percentage. For each 5 dBA increase above 80 dBA the weight of the measurement will be doubled. For example: a 90 dBA noise will be allowed for an 8-hour period, this will be 100% of the daily exposure, a 95 dBA noise will only be allowed for 4 hours, therefore, an 8-hour period will be 200% of the daily exposure (Fig. 22) because its weight is doubled. Measurements will be taken from 80 dBA to 115 dBA.

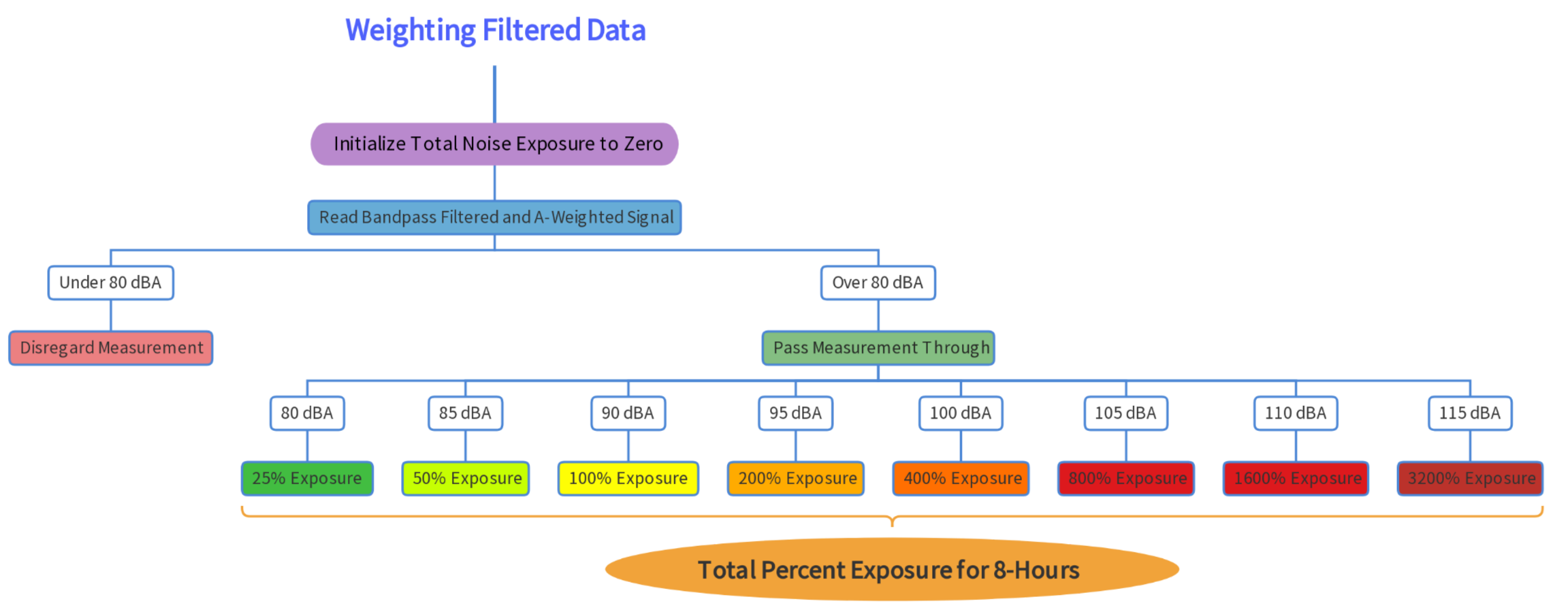


Fig. 22: Flowchart of Steps Needed to Determine the Weightings of Specific dBA Levels

5. Calculating Allowance

Once all of the filtering and weightings have been completed, the next step is to calculate the allowance. The device will start to add up noise measurements to assess the noise exposure percentage that is still available. The calculations will be taken as an average each minute. Using the noise exposure equation (Fig. 23) we are able to determine remaining exposure. The device will remove percentages from the allowance based on the weightings (Fig. 22). The heavier the weight (dBA level) of the measured noise, the larger percentage that will be removed and the less time allowed at that level (Fig. 24).

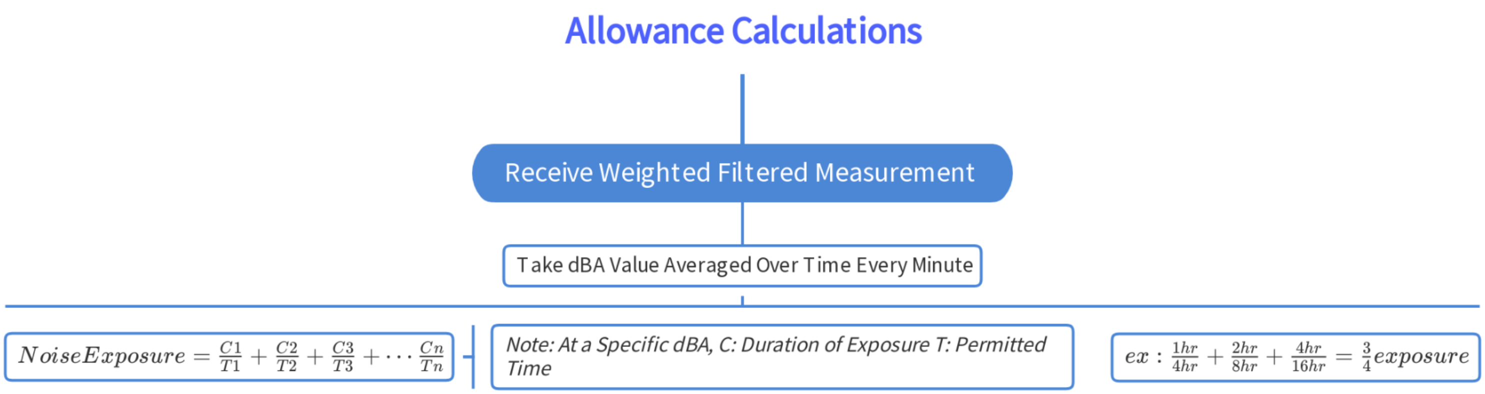


Fig. 23: Flowchart of Steps and Equations Needed to make Noise Allowance Measurements

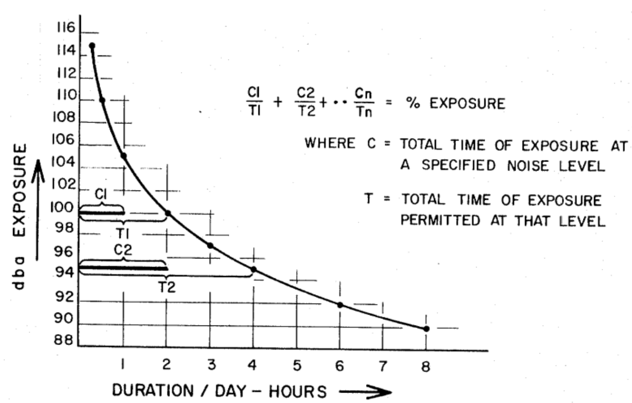


Fig. 24: Graph Plotting the dBA Exposure Weightings vs Time. (26)

6. Feedback

The final step of the pseudocode is the feedback to the user. The goal of the device is to inform continuously therefore the noise percentage allowance will be displayed at all times (Fig. 25). There is also a need to indicate threshold information to the user, this will be displayed as a pop-up notification. Notifications will appear as exposure thresholds are reached (Fig. 25).

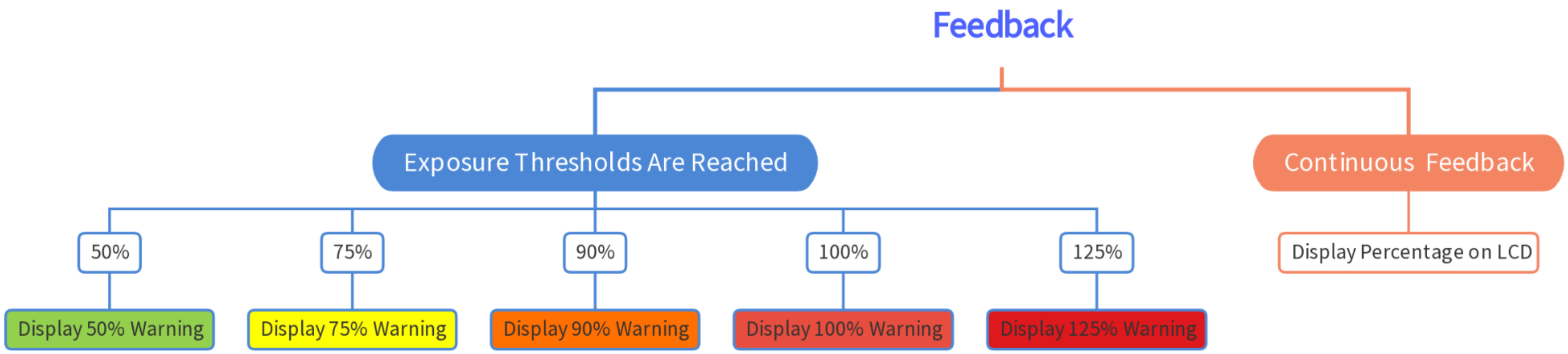


Fig. 25: Flowchart Showing the Feedback Mechanisms: Continuous and at Specific Thresholds

Concept Designs and Selection:

Justifications for Chosen Devices

1. The device has to be completed in the time allotted during the second semester of the class. With our abilities combined as a group, we considered the manufacturing process to be achievable through our collaboration. Hence, Ease of manufacturing was a criterion that had to filter out the designs.
2. The type of microphone and directionality is important as they will eventually affect the rest of the circuitry. These factors were considered when creating concepts. It is a criterion because the frequency the microphone detects, the frequency response it outputs, and the sensitivity toward environmental noise all play a role in the accuracy of the device.
3. A major contribution of our device is the allowance system which collects and calculates daily the noise percentage left for the user to be exposed to sound above the threshold. This information will be translated to the user through some feedback type or display. The display is crucial for our device, as it will give the necessary feedback. LCDs are the best option as they give the most information to the user.
4. The mobility of the device allows the user to change the locations of the device/microphone. Since direction and area of placement affects the sound being picked up by the microphone, we wanted to get a clear/accurate exposure of noise to the ear. Hence, the device's ability to move is an imperative factor.

Chosen Devices

Final Devices

We were able to use the above justification criteria to deduce the following 6 devices:

1. **M5StickC**

   Components:

   - M5StickC Plus with PDM MEMS mic and ESP-32 Microcontroller

Description:

The **M5StickC** is a very useful microcontroller that can be programmed similarly to an Arduino but it has the added benefit that it comes with a variety of sensors, an LCD, already built in batteries and a microphone. The device will detect sound using an on-board PDM MEMS mic. After the sound wave is detected, it will be digitally converted using the built in ADC and sent directly to the speedy ESP-32 microcontroller chip for analyzing and processing. The LCD will give the user notifications regarding thresholds. This is a completely **software solution** because the M5StickC is an already built device, only the coding portion remains. MATLAB would be used for the programming as it has Arduino IDE capabilities which is compatible with this device. The user will have to buy the device on their own since we do not have permission to sell it. They will then purchase our noise dosimeter software for a few dollars or as a monthly subscription. There are two options for purchasing, with/without a watch band, but we will advise to put it around the neck for best results.

Justification:

This is an already built device that would only need a software design to work. There is no need to order many different parts and rely on the specifications for compatibility. This is a cheap solution, that is lightweight and small that can be programmed just as easily as any microcontroller. The device and built-in sensors fulfill most, if not all, of the PDS requirements. The M5StickC can always be a back-up plan since it is already built. The other designs might end up being too difficult because building a device from components might not be feasible.

Specifications:

- Mic-SPM1423HM4H-B
  - PDM-MEMS
  - Frequency Range: 10Hz- 10kHz (with 3 dB or less deviation) (Fig 26)
  - Max: 140 dB
  - Sensitivity:-22dBfs
  - SNR: 61.5 dBA
  - Acoustic Overload: 110 dB
- Processor
  - 240 MHz speed
  - Bluetooth
  - Wifi
  - 4 MB flash and 520 kB RAM
  - Operating Voltage: 3.3 V
  - Input Voltage: 5V
- Software
  - MATLAB using Arduino IDE
- Size
  - 48.2x25.5 x 13.7 mm
  - LCD - 1.14 inch

 - ○ Weight - 21g
- Sensors
    - ○ Buzzer
    - ○ Gyroscope
    - ○ IR
    - ○ LED
    - ○ 2x Buttons

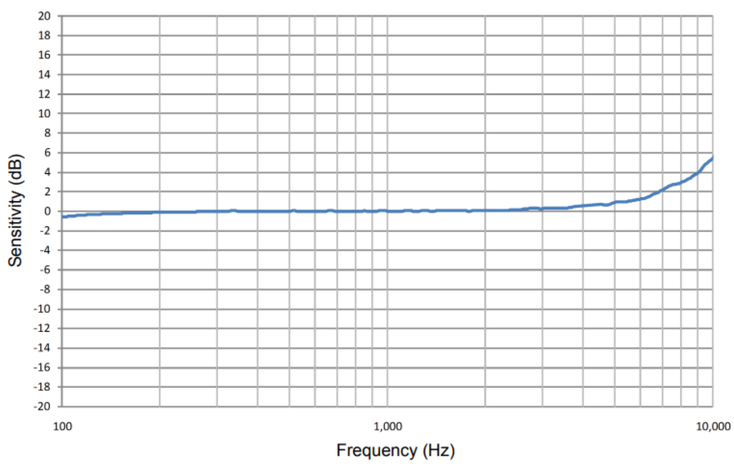


Fig. 26: Frequency Response of Mic-SPM1423HM4H-B (30)

Table 4: Cost Analysis of M5StickC		
Part	**Price**	**Picture**
M5StickC Plus	$19.95	

M5StickC Plus with watch	$22.90	

2. **Gorget**

Components:

- ECM
- Bandpass filter
- Arduino Every Microcontroller
- Vibration Disc and LEDs used for feedback
- Nylon strap and necklace clasps used to secure device

Description:

The **Gorget** is a noise dosimeter that is placed around the user's neck and pointed perpendicular to the opening of the user's ear. Due to its short distance from the ear, the noise data obtained will be similar to what is heard by the user. It uses a unidirectional ECM microphone with an analog signal output to detect noise. The signal would be passed through an amplifier as well as a low and high pass filter to get rid of any unwanted noise. The signal would then be transmitted to the Arduino microcontroller where the signal will be converted to digital by using a program written in Arduino IDE. This data would then be run through an algorithm that averages the data to determine whether or not the user has passed a preset hearing threshold. When the user reaches a certain amount of exposure a set of LEDs will begin to light up one by one until the exposure limit is reached.

Justification:

The Gorget was chosen above the rest of the unidirectional microphones for several reasons. The most important aspect was its location. By its nature a unidirectional microphone is only capable of detecting sounds in front of it and to its sides, so placing the microphone on the neck allows the microphone to detect sounds from the most important directions. The sound coming from behind a person is less relevant as the outer ear naturally muffles noise at a 180-degree angle behind the user. By having the device on the neck, there is no concern of sound being muffled by the wearer's clothes. Also, due to its compact size, it is less noticeable than the rest of the devices.

Specifications:

- Unidirectional ECM
  - Frequency Range: 100 Hz ~ 10 kHz
  - Sensitivity: -47dBV ± 4dBV
- Filters
  - Preamplifier with bandpass filter
  - Digital A-weighted filter
- Arduino Every
  - Operating Voltage: 5V
  - Clock Speed: 20MHz
  - Analog to digital converter
- Feedback
  - Vibration disk and LEDs

Table 5: Cost Analysis of Gorget	
Component	**Price**
Unidirectional ECM	$2.03
Gain Amplifier	$0.44
Arduino	$12.00
Vibration Disk	$1.95
Lithium-ion Battery	$14.95
Bandpass Filter/LEDs	$6.485
Nylon Necklace Strap	$5.89
Necklace Clasps	$4.09
Total	**$47.83**

3. **Mic & Box**

Components:

- Omnidirectional ECM
- Analog Signal
- Low-Voltage Amplification

- Analog band pass filter for noise
- Arduino
- ADC
- Digital A-weight
- LCD

Description:

ECM would be used to collect the environmental sound signal 3 times per half a minute (every 10 seconds). The output would be a voltage change, which then is amplified using an **analog low-voltage amplifier circuit**. The amplified signal would then be filtered using an analog filter that filters out the DC noise (high pass) and filters out the higher frequency range that is higher than the hearing range (20kHz). Then the output signal would be fed into the microcontroller (**Arduino**), and the voltage would be transferred to digital which is then weighted according to the A weight system after accounting for the frequency response of the mic. After the weighting of the signal, it would then be processed by the algorithm developed. The algorithm would average the 3 measurements for that half a minute and that would be the dbA value for this half a minute. Then if the measurement of that half a minute is above 80 dBA it would be taken into account depending on the OSHA guidelines. Then the output would be displayed using the LCD in the form of a percentage exposure allowed. The device would include a mic that's attached to the clothes (like a mic piece). Then the circuit components and the display are going to be housed in a separate compartment, with the mic and the housing connected via a wired connection (Fig. 27).

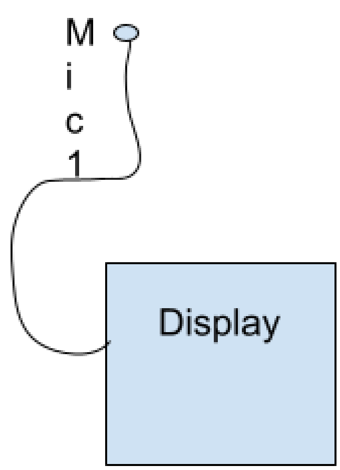


Fig. 27: Drawing of the Mic&Box

Justification:

ECM outputs an electrical voltage, the voltage from the capacitor proportional to the original pressure oscillation corresponding to the individual microphone's sensitivity. An omnidirectional ECM perceives the sound from every direction the same. This design utilizes low voltage components for lower voltage needs. Also, the microphone's movability would give the user the freedom of placing the microphone around the neck without having an article of clothing interfering with it. Furthermore, having an on device display is great for informing the user about their noise exposure allowance.

Specifications:

- Mic CMB-6544PF
  - Freq range:20 Hz ~ 20 kHz (Fig. 28)
  - Operating Voltage: 4.5 V ~ 10 V
  - Sensitivity: -44dB ±3dB @ 94dB SPL
  - Method of connecting to circuit: PC Pins
- Low Voltage Amp.
  - 200X gain
- Band Pass analog filter
  - Cut Off at lower than 20 Hz and greater than 20 kHz
- Arduino Micro
  - 10 bit ADC
  - 32 KB Flash memory
  - SRAM 2.5 KB
  - 5V I/O Voltage
  - Low power processing
- Software
  - Arduino IDE
- LCD

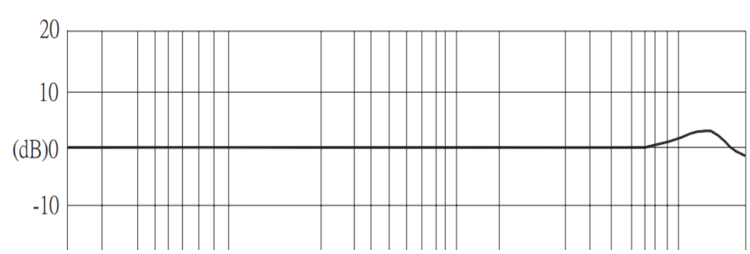


Fig. 28: Frequency Response of CMB-6544PF (31)

Table 6: Cost Analysis of the Mic & Box	
Component	**Price**
ECM CMB-6544PF	$0.85
Low-Voltage Amp.	10/$10.99
Band Pass	Built with components supplied from the Makers Space
Arduino Micro	$30
LCD	$9.00
Total	**$40.19**

4. **The Parachute**

Components:

- Omnidirectional MEMS
- Analog A-weighted Filter
- Seeed Microcontroller
- USB Storage
- Graphic LCD

Description:

For **The Parachute**, we follow with a MEMS device and a secondary microphone for better signal (Fig. 29). This will pick up noise and amplify. An A-weighted filter will pick up the information and attenuate as needed to model the impact onto the human ear (Fig. 30). This will be inputted into a microcontroller that will convert the analog to digital data, hence, we don't need a converter. The data will be processed using an algorithm that will collect all information from a 1 minute range and display the results onto the LCD. The code will also account for the removal of storage every day so that space is available for use for new data. This will be placed as a clip onto the targeted area on the chest.

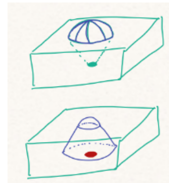


Fig. 29: The Parachute Device

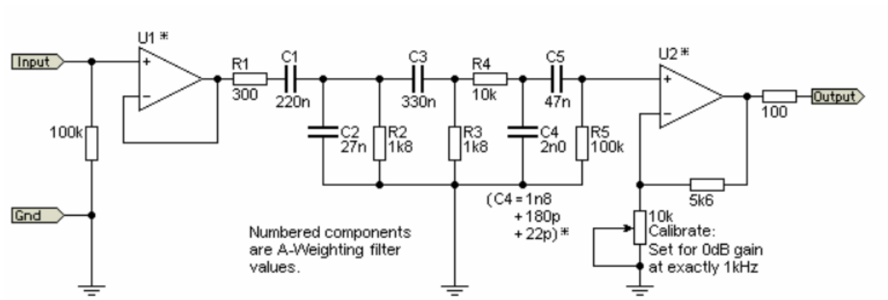


Fig. 30: The analog A-weighting filter circuit (32)

Table 7: Cost Analysis of The Parachute	
Part	**Price**
MEMS	$1.37
A-Weight Filter	Analog components given
Microcontroller	$4.99
Storage	$18.99 (for 5)
LCD	$34.25
TOTAL	**$44.41**

Justification:

The device contains a dome shape that picks up sound from all directions. The cone-shaped housing directs the sound toward the microphone for better collection and amplification. It has a great frequency response from the MEMS, hence, it will be able to pick up distinct sound ranges. It has an LCD with three primary colors, hence, the user will be able to get great visuals of their daily reminders. Lastly, it’s placed on a clip which allows for a change in location based on comfort and noise direction.

Specifications:

- Omnidirectional MEMS collects and amplifies sound
  - 3.35mm x 2.5mm x 0.98mm
  - 80Hz - 20kHz Analog Microphone
  - 1.52V - 3.6V
  - Sensitivity: -38dBV
  - Microcontroller
  - 400KB SRAM & 4MB Flash
  - Bluetooth 5.0
  - Charging current: 50mA/100mA
  - 5V
- Storage
  - Compatible with most PC, Mac laptops and desktop computers.
  - Made of zinc alloy and ABS material, not easy to get bent or broken off
  - water proof, shock proof, X-ray proof, magnetic proof, temperature proof, dust proof
- LCD
  - TFT - Color
  - Screen: 43.20mm W x 57.60mm H
  - 18-Bit

5. **SoundBand**

Components:

- ECM Omnidirectional Microphone
- External Amplification
- A-weighted filter
- Microcontroller

- SD Storage
- Vibration Exposure limit Indicator

Description:

The SoundBand utilizes a single ECM omnidirectional microphone that is fixed on the surface of the device with a cotton headband to trap moisture away from the internal circuitry. The housing of the dosimeter will be sandwiched between two additional layers of polyester to further prevent liquids from getting into the circuit. Regarding the circuit of the device, the device will be powered by a 9V battery. The SoundBand will have a mircoUSB port to connect it with a computer interface for a full report of the different dBA reading throughout the day and an SD card reader. This report includes a graph of the sound reading over time and a breakdown of the different pitch heard (frequency range).

For the entire circuitry of the dosimeter, the device will be powered using a 9V battery. The preamplifier for the microphone will have a 100x gain to bring the analog signal within the 0.5V minimum working range of a 10-bits microcontroller processor. This is necessary for digital processing and data storage. From pre-amplification, we incorporated an A-weighted network filter that captures analog signals in the 40-20kHz range. This signal is converted digitally with an 8-bit ADC to save storage space and finally picked up by the microcontroller. The microcontroller is pre-programmed through MATLAB to take the converted signal and keep a counter of the exposure limit of the user. Though this will not be displayed on a screen, the user will still have access to their exposure limit by pressing a button that is placed on the surface of the dosimeter's case. When pressed, the microcontroller will activate a vibrating motor that will pulse up to five times, each pulse indicating twenty percent of exposure left (Fig. 31).

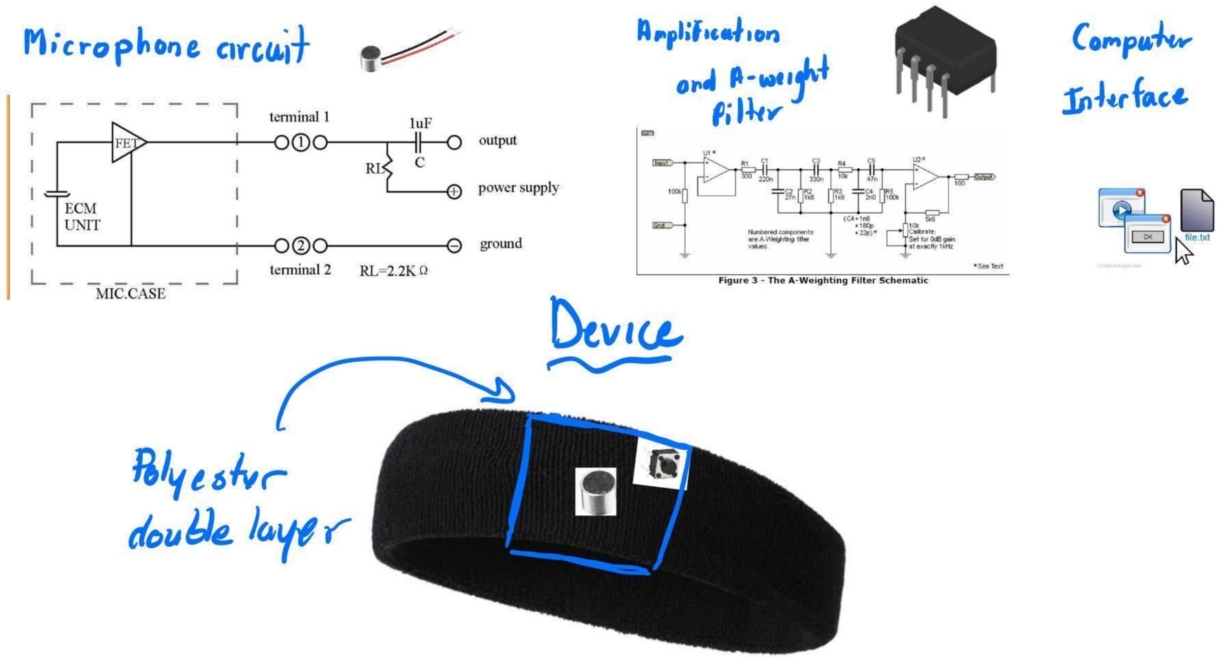


Fig. 31: Workflow of the SoundBand

Justification:

The cotton material absorbs liquids well which will prevent damage caused to the circuit by a leakage. The polyester double layers act like a waterproof mechanism in case the whole headband gets completely soaked in water or sweat. A 9V battery is used to power multiple stages of the circuitry. 100x gain is used because a typical open-circuit microphone gives a voltage output in the hundredths range. The gain will satisfy the 0.5V input requirement of the microcontroller. We only care about the human auditory range which frequency spans 40-20kHz and is the reason why the filter selects these frequencies for the analog output. 8-bit conversion is to keep the storage as low as possible and a 10-bit processor is used to draw less current from the battery when compared to a higher bit processor such as a 32-bit processor. The choice of a vibrating indicator, as opposed to a screen display, would save more battery.

Table 8: Cost Analysis of SoundBand	
Part	**Price**
CMC-2742WBL-25L: ECM MIC	$2.00
OP AMP	$6.00
Assortment of Electrical elements	$14.00

Microcontroller	$25.00
Cotton Headband	$7.00
Polyester Material	$10.00
Total	**$64.00**

Specifications:

- **Microphone:**
  - Model**:** CMC-2742WBL-25L
  - Type: ECM
  - Freq range: 100 Hz - 20kHz
  - Sensitivity: - 42 dBV
  - Operating Voltage: 2V
- **Amplifier:**
  - Type: LM386N
  - Supply Voltage: 15V Max
  - Gain: 100
- **Microcontroller:**
  - ADC: 10-bit conversion
  - 32KB flash storage
  - Input: 5V Max
  - Output: 5V Max
- **Fabric**
  - Polyester: 8.5 cm x 8.5 cm
  - Cotton: 21 cm x 6.1 cm

6. **DoseClip**

Components:

- Omni-Directional Microphones
- 1 ECM
- 1 MEMS
- Low-Voltage Amplificator for the ECM
- 2x Digital filters (band-pass)
- Digital A-weighting Filter

- Microcontroller (Seeed Xiao)
- LCD for feedback

Description:

This device will utilize the use of two different microphones in order to cover the whole human audible frequency range (20Hz-20kHz). Both ECM and MEMS microphones will obtain the environmental noise data at 12 measurements per minute (5 second intervals) that will be averaged separately. Both signals coming from each microphone are then inputted separately into a Seeed Xiao microcontroller. The two signals will be filtered to the desired frequencies using bandpass filters. The MEMS microphone will measure the low frequencies, it will have a pass range of 20 Hz-10 kHz. The ECM will measure the high frequencies, it will have a pass range of 10 kHz-20 kHz. The two resulting signals will be combined into one signal and attenuated by the A-weighted filter. The filtered signal will then be processed using MATLAB code. The average measurements from each minute will be taken into account only if it is above 80 dBA, in order to comply with OSHA's threshold guidelines. The feedback measurements will be displayed using an LCD that shows the allowable noise exposure percentage remaining. The storage of data will be cleared on a daily basis to allow for the following day's information to have room to be stored. The device will be in a rectangular housing with both microphones on the same level surface to eliminate any difference in sound measuring (Fig. 32). The housing will be attached to the user via a clip. The housing and the clip will be 3D printed.

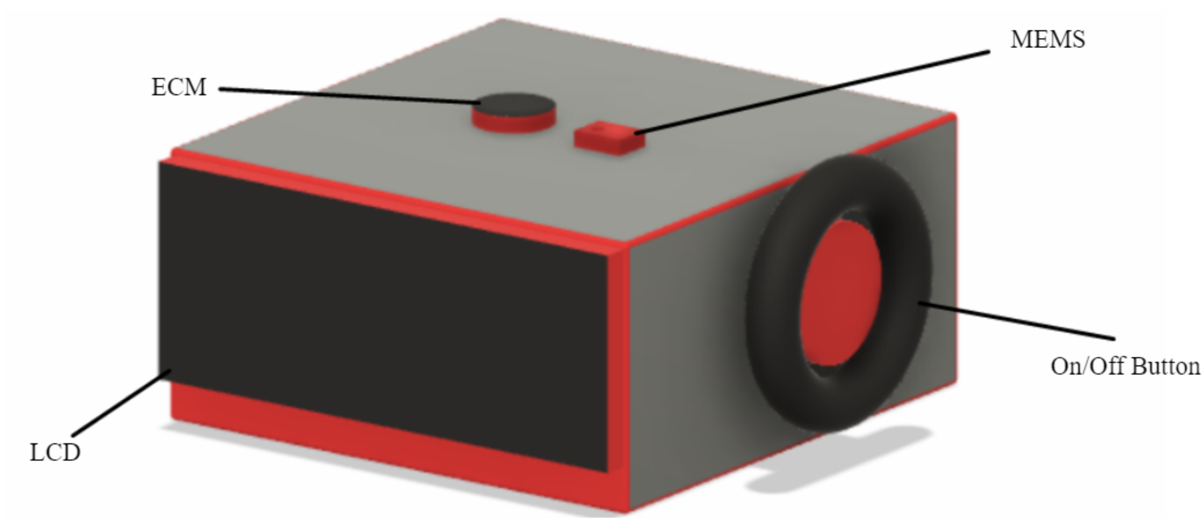


Fig. 32: Design Concept for DoseClip Made in Fusion 360 Including: LCD, Microphones, and On/Off Switch

Justification:

This device harnesses the power of 2 microphones to make accurate measurements of the environmental noise. Also, it utilizes low power design which will prolong its life on a single charge. The device will be able to cover the full frequency range by using a low frequency microphone and a high-frequency microphone and combining their signals. The LCD will properly notify the user of potentially harmful noise dosages. The clip will allow the user to have the freedom to place the device comfortably. The device will be small thanks to the Seeed Xiao's small size. The device will also be very cheap to produce, under twenty dollars.

Specifications:

- ECM Mic CMB-6544PF
  - Freq range: 20 Hz ~ 20 kHz (Fig. 33)
  - Operating Voltage: 4.5 V ~ 10 V
  - Signal to Noise ratio 60 dbA
  - Sensitivity: -44dB ±3dB @ 94dB SPL
  - Method of connecting to circuit: PC Pins
- MEMS Mic SPH1878LR5H-C
  - Freq range: 7 Hz ~ 36 kHz (Fig. 34)
  - Operating Voltages 2.3 ~ 3.6 V
  - Signal to Noise: 67 dBA
  - Sensitivity: -44dB ±0.5dB SPL
- Low Voltage Amp.
  - 200X gain
- Seeed Xiao
  - 160 MHz speed
  - Low Energy Bluetooth
  - Wifi connectivity
  - 4 MB flash and 400 kB RAM
  - Operating Voltage: 3.3 V
  - Input Voltage: 5V

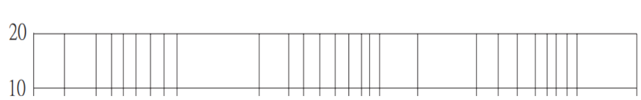


Fig. 33: Frequency Response of CMB-6544PF ECM Mic (31)

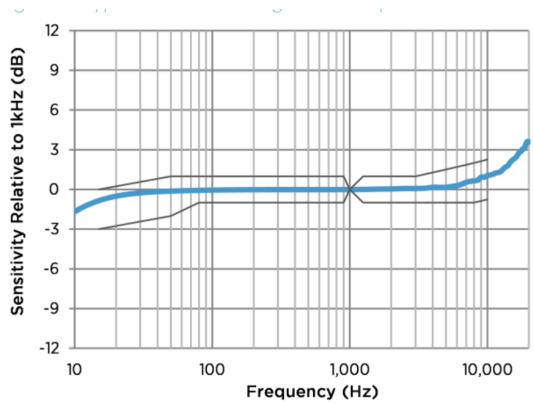


Fig. 34: Frequency Response of SPH1878LR5H-C MEMS Mic (33)

Table 9: Cost Analysis of DoseClip	
Component	**Price**
ECM CMB-6544PF	$0.85
SPH1878LR5H-C	$1.73
Low-Voltage Amp.	10/$10.99
Band Pass	Digital
SEEED STUDIO XIAO ESP32C3 WIFI+B	$4.99
LCD	$9.00
Total	**$17.67**

Branch 1: Digital Omni-Directional

1. HearPods

Software Components:

- iPhone Application
- Swift development
- Apple Compatible

- UI will allow them to see statistics and graphs
- Using Airpods

Description:

The **HearPods** phone application will make use of the AirPods' built-in external microphone system to measure noise (Fig. 35). People already have their AirPods in their ears all day, it will be a good microphone system for measuring in-ear exposure. This app will be made in Swift to be Apple compatible and will be **exclusive** for Apple users with AirPods. The iPhone will make measurements based on the AirPods' sound measurements and give warnings about thresholds that are being reached. The user won't need to use any special equipment, just make sure that the AirPods are in use. The AirPods also block noise from coming in to protect the ear. It can also measure music exposure from the AirPods.

Fig. 35: Image of Apple Airpod (34)

2. **Watch-Out**

   Software Components:

   - [Smart Watch]
   - Watch application
   - Java app development
   - Send Notifications to User
   - UI will allow them to see statistics and graphs
   - Using watch microphone

Description:

The **Watch-Out** watch application will make use of the smartwatch' built-in external microphone system to measure the environmental noise (Fig. 36). The watch is convenient to wear all day. The app will record the measurement of sound throughout the day and analyze the thresholds of the noise exposure. The noise exposure will then be deducted from the noise allowance. This app will be made in Java and will be compatible with the smartwatch. The watch will make measurements and give warnings about thresholds that are being reached. The user won't need to use any special equipment. It may be able to connect to headphones to measure audio exposure.

Fig. 36: Image of Smartwatch (35)

3. **Noise-4-U**

   Software Components:

   - Phone Application
   - Python development
   - Android Compatible
   - Send Notifications to User
   - UI will allow them to see statistics and graphs
   - NextCloud Raspberry Pi cloud server

Description:

The **Noise-4-U** phone application will make use of the phones' built-in microphone system to measure noise. It will be convenient as people already have their phones on their person at all times. This app will be made in Python to be android compatible to hit a larger market than an apple device. The phone will make measurements and give warnings about thresholds that are being reached. The user won't need to use any special equipment, just make sure that the microphone is available to the environment. This app will make use of the Raspberry Pi cloud system, NextCloud, for data storage.

4. **Noise Clouds**

   Components:

   - MEMs I2S
   - Arduino Nano 33 BLE
   - Band-Pass Filter
   - Send data to computer using Bluetooth
   - Processing will be done in the cloud

- UI will allow send notifications to user’s phone

Description:

MEMS microphone (using I2S interface) will measure surrounding sound and output a digital signal. The signal is received by the Arduino Nano 33 BLE microcontroller and will apply a band-pass filter in order to filter the noise of the microphone internal and frequencies above 20kHz. The Arduino will then send this data to a computer software or smartphone app via its Bluetooth capability. The software would send this data to a cloud server. The cloud server will then be able to process this data via our code, which will include the A-weighting and the dBA thresholding. The server then when reaching a certain threshold will send a message to the user's phone in order to warn them about the sound allowance consumption.

Omni-Directional Digital MEMS microphones:

Note: Microcontroller must have a PDM/I2S interface to use PDM/I2S microphone (Fig. 37)

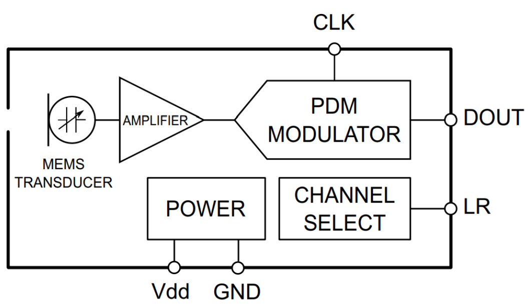


Fig. 37: MEMs PDM microphone schematic (36)

5. **Nano Clip**

   Components:

   - Omni-Directional Microphone
   - MEMS
   - Self Amplification (on PCB board)
   - Digital Output
   - Pulse Density Modulation
   - Using [Seeed Studio XIAO Sense microcontroller](#) with PDM
   - Digital A-Weighted Filter

Description:

The **Nano Clip** will be a noise dosimeter that is either clipped on or around one ear to allow for readings very similar to the environmental noise entering the ear. It will detect noise with a Digital MEMs microphone, which is omnidirectional. It will use a Pulse Density Modulation (PDM) interface which will need a microcontroller that is compatible. This includes having a clock with a similar rate (usually 1-3 MHz). This specific device will make use of the Seeed Seeed Studio XIAO Sense microcontroller which has the PDM microphone already installed and is therefore compatible (Fig. 38). The device has preamplification and ADC in the MEMS microphone. The filtering of the signal and threshold algorithm will be made on either CircuitPython or Arduino IDE. The threshold will make use of an averaging system to calculate decibel range for the duration of use. This microcontroller has Bluetooth connectivity and it will send the exposure data to a computer application rather than having notifications on the device. This will keep the device as small as possible.

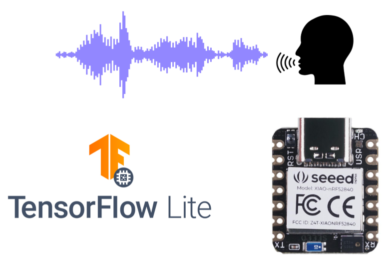


Fig. 38: Image of Seeed Xiao with Speech Recognition (37)

6. **Beanie**

   Components:

   - Omni-Directional Microphone (can be used with PDM or I2S)
   - 2x MEMS [Adafruit PDM](#) *(1MHz-3MHz clock rate)*
   - Self Amplification (on PCB board)
   - Digital Output
   - Pulse Density Modulation (can be connected with one clock)
   - Using [Adafruit Feather M4 Express](#) microcontroller with PDM interface
   - Digital A-Weighted Filter
   - CircuitPython/Arduino Code
   - This device has 2 mics so it can be used with one of the 2 microphone designs
   - The output can be vibrations and you can check data by connecting to a computer

Description:

The **Beanie** will be a noise dosimeter placed around the head with 2 microphones at each ear to make readings very close to the environmental noise entering each ear. It will detect noise with a Digital MEMs microphone, which is omnidirectional. It will use a Pulse Density Modulation (PDM) interface which will need a microcontroller that is compatible (Fig. 39). This includes having a clock with a similar rate (usually 1-3 MHz). This specific device will make use of the Adafruit PDM microphone and Adafruit Feather M4 Express microcontroller which is compatible with both PDM and I2S. The device has preamplification and ADC in the MEMS microphone. The filtering of the signal and threshold algorithm will be made on either CircuitPython or Arduino IDE. The threshold will make use of an averaging system to calculate the decibel range for the duration of use. This device will make use of vibrations to warn the user, it may also have lights on the beanie to indicate exposure level. To check data, plug the device into a computer to receive graphs and useful info.

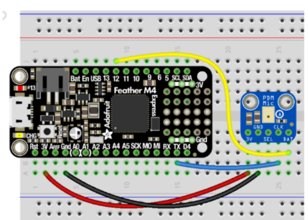


Fig. 39: Sample of PDM Mic attached to Microcontroller (38)

7. **NoisenEx**

   Components:

   - Omni-Directional Microphone
   - MEMS
   - Self Amplification (on PCB board)
   - Digital Output
   - I2S interface
   - Using AtomU ESP32 Development Kit with USB-A with I2S Interface
   - Digital A-Weighted Filter
   - UIFlow/MicroPython/Arduino Code
   - Device is bluetooth
   - Can be connected to Computer Program

Description:

The **NoisenEx** will be a noise dosimeter clipped to the chest or shoulder. The clip will give the user freedom to place it comfortably. This will be close enough to give environmental noise data similar to the ear. It will detect noise with a Digital MEMs microphone, which is omnidirectional. It will use an I2S interface which will need a microcontroller that is compatible. This includes having a clock with a similar rate (usually 1-3 MHz). This specific device will make use of the AtomU ESP32 Development Kit with USB-A microcontroller which is a single device that has an I2S microphone and compatible circuitry (Fig. 40). The device has preamplification and ADC in the MEMS microphone.The filtering of the signal and threshold algorithm will be made on either CircuitPython, Arduino IDE, or UIFlow. The threshold will make use of an averaging system to calculate the decibel range for the duration of use. This microcontroller has Bluetooth connectivity and it will send the exposure data to a computer application rather than having notifications on the device. This will keep the device as small as possible. To check data, plug the device into a computer to receive graphs and useful info.

Fig. 40: Image of M5AtomU microcontroller (39)

Branch 2: Analog Omni-Directional

8. **Dye-simeter**

   Components:

   - Omni-Directional Microphone
   - ECM
   - Amplification Needed
   - Analog Output
   - Analog Filter
   - Analog Comparator
   - Chemical Dye

Description:

The **Dye-simeter** will be a noise dosimeter strapped to the chest. The strap will give the user freedom to place it comfortably. This will be close enough to give environmental noise data similar to the ear. It will detect noise with an ECM microphone, which is omnidirectional. It will output an analog signal. The signal will have to be amplified and filtered using analog components, such as resistors and capacitors. The threshold algorithm will be made of various switches that will be triggered when differing decibel ranges are detected. Each specific dB threshold will be assigned a gain so it is easier to distinguish. The threshold will make use of an averaging system to calculate the decibel range for the duration of use. Each tick measured will add to the total exposure. This will make the electrically charged dye pack change colors. As the dye becomes darker, more exposure is measured. The limit will be reached when the color no longer becomes darker. There will be a color chart to indicate the level of exposure reached. The dye pack will have to be switched between uses.

9. **Obsidian Noise**

   Components:

   - Omni-Directional Microphone
   - ECM
   - Amplification Needed
   - Analog Output
   - Analog Filter
   - Analog Comparator

Description:

The **Obsidian Noise** will be a noise dosimeter clipped to the shoulder for accurate readings. It will detect noise with an ECM microphone, which is omnidirectional. It will output an analog signal and be amplified and filtered using analog components. The threshold algorithm will be made of various switches that will be triggered when differing decibel ranges are detected. Each specific dB threshold will be assigned a gain so it is easier to distinguish. At specific exposure thresholds, the bioluminescent oil will drop into water similar to the test tube dy process (Fig. 41). The more exposure the brighter the tube will be and the user will be more aware of the high noise exposure.

Fig. 41: Image of Sample Test Tube Dye Process (40)

Digital Device: Internal Digital Noise Filter

10. **Chest Me Up**

    Components:

    - Omni-Directional Microphone
    - ECM
    - Non-Inverting Amplification
    - Analog Output
    - Arduino (ADC and Processing)
    - Average dB 1-minute (3 measurements)
    - LED Ring

Description:

ECM would take in the measurement of the environmental noise. The signal output of the ECM will be connected to a non-inverting amplifier. The amplified signal would then be connected to an Arduino, the bandpass filter and the A-weighting are both applied digitally. After the weighting of the signal, it would then be processed by the algorithm developed. The algorithm would average the 3 measurements for that minute and that would be the dbA value for this minute. Then if the measurement of that minute is above 80 dBA it would be taken into account depending on the OSHA guidelines. Then the output is a ring of 10 LEDs with each lED presenting 10% percent and when the 10% is consumed the LED will turn off. The device will be a chest strap that is located at the sternum (Fig. 42).

A **B**

Fig. 42: Chest Me Up (A) Shows how the device housing will look like with the display (B) Shows the chest strap with the device

11. **Third Ear**

    Components:

    - Omni-Directional Microphone
    - ECM
    - Analog Output
    - Amplifier
    - Raspberry Pi
    - Digital Filter and Threshold

Description:

The **Third Ear** will be a noise dosimeter that will be designed to behave similarly to the ear.It will be either clipped or on a necklace. This will be close enough to give environmental noise data similar to the ear. It will detect noise with an ECM microphone, which is omnidirectional. The size of the diaphragm should be similar to the tympanic membrane Eardrum) 8-10mm. This added size will also give a large SNR. There will be a cone around the ECM to help collect sound similar to the method of the outer ear. The length will be around 20 mm with a diameter of 6mm. The end of the cone will have flaps facing outwards to block sound waves coming from the users' own voice. The sound detected will be amplified before being connected to the analog pins of Raspberry Pi Pico for ADC and processing. The filtering of the signal and threshold algorithm will be made on either Python. The threshold will make use of an averaging system to calculate the decibel range for the duration of use. This microcontroller has Bluetooth connectivity and it will send the exposure data to a computer application rather than having notifications on the device. This will keep the device as small as possible. To check data, plug the device into a computer to receive graphs and useful info.

The **Third Ear** will be a noise dosimeter that will be designed to behave similarly to the ear. All of the specs that are being measured are specific to the ear, the design should be too. The frequency range is 20Hz-20kHz so is the ECM chosen. The diameter is 9.7 mm which is similar to the 8-10 mm of the tympanic membrane. This also gives a high SNR. The cone will help focus the noise into the microphone. It will also block some of the noise coming from the user speaking (which we weren't sure how much it would affect the measurements). This device will mimic sounds going into the ear and will be a good design for measuring noise.

12. **Neck Hugger**

Components:

- Omni-Directional Microphone
- 2x ECM
- Low-Voltage Amplification
- Analog Output
- Analog Filter (band-pass)
- Arduino

Description:

This device will utilize the use of 2 ECMs in order to capture the environmental noise. Each ECM would be connected to a low voltage amplifier then both outputs will be averaged using a voltage divider (get average of both) circuit layout. The signal is then filtered using an analog band-pass filter and the signal is then fed to an Arduino which will do the A-weighting and the processing of the signal. The device will capture the environmental noise 3 times per minute. The algorithm would average the 3 measurements for that minute and that would be the dbA value for this minute. Then if the measurement of that minute is above 80 dBA it would be taken into account depending on the OSHA guidelines. The output will be vibration for each 10% decrease. The device will be resting around the neck (Fig. 43). The data will be sent to an external display that will show the data of the allowance quota per day.

Fig. 43: Shows the Head Hugger with the mic sensor and the vibration motor (41)

13. **Birdie**

    Components:

    - Omni-Directional MEMS Analog
    - Microphone
    - mems
    - Self-amplify
    - A weight Filter
    - ADC
    - Process data —>Second to second.
    - Software storage
    - LCD

Description:

For the **Birdie**, we are focusing on an omnidirectional field of detection where it receives sound from all directions. This would be found in the MEMS (micro-electromechanical system) which will pick up and measure the sound frequency ranging from 100Hz to 80kHz. The amplifier will output analog data with a max distortion of 0.2% into the A-weighted filter (Fig. 44). The filter would take in the data and attenuate as needed into an analog-to-digital converter. A 16-bit resolution would suffice for a sampling rate of 44.1kHz. The voltage would be processed in the algorithm that outputs an averaged noise of each second as a daily percentage onto an LCD screen. This would be strapped to the shoulder.

Fig. 44: Birdie Device

Branch 3: Analog Unidirectional

14. Vibramax Hearing Apparatus (VHA)

Components:

- Uni-Directional Microphone
- ECM
- Amplification Needed
- Analog Output
- Analog Filter
- Analog Comparator

Description

The **Vibramax Hearing Apparatus (VHA)** will be a noise dosimeter with an external microphone clipped to shoulder and vibrating mechanism placed in a pocket of choice to get best results. The two part device will give the user freedom to place it comfortably. It will detect noise with an ECM microphone, which is unidirectional. It will output an analog signal and be amplified and filtered using analog components. The threshold algorithm will be made of various switches that will be triggered when differing decibel ranges are detected. Each specific dB threshold will be assigned a gain so it is easier to distinguish. At specific exposure thresholds, the vibration mechanism will begin to vibrate at various intensities as the exposure dosage reaches the exposure limit.

15. Counter Dosimeter

Components:

- Uni-Directional Microphone
- ECM
- Amplification Needed
- Analog Output
- Analog Filter
- Analog Comparator

Description:

The **Counter Dosimeter** will be a noise dosimeter strapped to the shoulder. The strap will give the user freedom to place it comfortably. This will be close enough to give environmental noise data similar to the ear. It will detect noise with an ECM microphone, which is unidirectional. It will output an analog signal. The signal will have to be amplified and filtered using analog components, such as resistors and capacitors. The threshold algorithm will be made of various switches that will be triggered when differing decibel ranges are detected. Each specific dB threshold will be assigned a gain so it is easier to distinguish. The threshold will make use of an averaging system to calculate the decibel range for the duration of use. Each tick measured will add to the total exposure. When the counter hits 50,75,100 it will make a noise to notify.

16. Pauldron

Components:

- Unidirectional Microphone
- ECM
- Gain Amplification
- Arduino Nano Microcontroller
- Analog to Digital conversion via Arduino IDE
- Bandpass filter via Arduino IDE
- LCD TFT screen
- UI will allow the user to see data about noise allowance
- The device is strapped to the user's shoulder

Description:

The **Pauldron** is a noise dosimeter that is clipped onto the user's shoulder and pointed at a 90-degree angle from the user's ear. Due to its short distance from the ear, the noise data obtained will be similar to what is heard by the user. It uses a unidirectional ECM microphone with an analog signal output to detect noise. The signal would be passed through an amplifier and then transmitted to the Arduino microcontroller where the signal would be converted from analog to digital and passed through a high pass filter using a program written in Arduino IDE. This data would then be inputted into an algorithm that averages the data to determine whether or not the user has passed a preset hearing threshold. The specific threshold information can be viewed through the use of a TFT screen. When the user reaches a certain amount of exposure the screen will begin to alert the user.

17. Spy Watch

Components:

- Unidirectional Microphone
- ECM
- Analog Output
- Gain Amplification
- Bandpass filter
- Arduino Every Microcontroller
- LCD TFT screen
- UI will allow the user to see data about noise allowance
- The device is located on the user's wrist

Description:

The **Spy Watch** is a watch that functions as a noise dosimeter with a built-in microphone. It uses a unidirectional ECM microphone with an analog signal output to detect noise. The signal would be passed through an amplifier as well as a bandpass filter to get rid of any unwanted noise. The signal is then passed through an Arduino microcontroller where the data is run through an algorithm that averages the data to determine whether or not the user has passed a preset hearing threshold. The data can then be viewed on the watch's LCD screen.

18. Gear

Components:

- Unidirectional Microphone
- ECM
- Analog to Digital converter
- Gain Amplification
- Bandpass filter
- Arduino Every Microcontroller
- Vibration Disk for feedback
- Device can be plugged into a computer to view information about noise allowance
- The device is placed behind the user's ear

Description:

The **Gear** is a noise dosimeter that is placed behind the user's ear with the microphone on the antihelix of the ear perpendicular to the opening of the ear. Due to its short distance from the ear, the noise data obtained will be similar to what is heard by the user. It uses a unidirectional ECM microphone with an analog signal output to detect noise. The signal would be passed through an amplifier as well as a low and high pass filter to get rid of any unwanted noise. The signal would then be passed through an analog-to-digital converter. The output is then transmitted to the Arduino microcontroller where the data is run through an algorithm that averages the data to determine whether or not the user has passed a preset hearing threshold. A vibration disk will periodically alert the user whenever they reach a certain amount of exposure until they reach the exposure limit.

Pugh Method

In order to determine the final device from the selected 6 devices, we conducted a Pugh method assessment. The criteria was based on the PDS, the need statement, and the sponsor’s needs. The criteria included the pricing and OSHA requirements which were heavily weighted because of the sponsor's need to make the device cost-effective and comply with the industry’s standards. The frequency and sound pressure ranges were important to include in the criteria as they would determine the capability of the device in collecting the data. The size and weight of the device are one of the criteria because it allows for comfortable personal use. According to the PDS, the device should be able to function in typical environmental conditions. The temperature was the main focus of this criterion as it is one of the most important factors considered for the electric component's functioning. The final selected device is the DoseClip (Table 10).

Table 10: Pugh Method for Final Concept Selection								
Category	**Weightings**	**Baseline (dBadge)**	**DoseClip**	**M5stick C Plus**	**Mic & Box**	**The Gorget**	**The Parachute**	**SoundBand**
Pricing < $200 Cheap solution	**5**	**0**	**1**	**1**	**1**	**1**	**1**	**1**
Frequency range (20Hz-20 kHz)	**2**	**0**	**1**	**-1**	**0**	**-1**	**-1**	**-1**
Sound pressure range (60-115 dB SPL)	**3**	**0**	**1**	**-1**	**-1**	**-1**	**-1**	**-1**
Size & Weight (8cm x 5.4cm x 5.5cm & weight max 200 grams)	**1**	**0**	**1**	**1**	**1**	**1**	**-1**	**1**
Environmental factors (Temperature: 43 C)	**2**	**0**	**1**	**1**	**1**	**-1**	**1**	**-1**
OSHA Compliant	**5**	**0**	**1**	**1**	**1**	**1**	**1**	**1**
Total	**18**	**0**	**18**	**8**	**10**	**4**	**7**	**0**

Feasibility Plan

Feasibility planning is used to determine if a specific concept should be investigated in greater detail by examining a prototype. This is done by performing a series of trials made to confirm the validity of the concept. In general, the noise dosimeter functionality is based on the circuit components and how they are assembled. For this reason, the individual components must be tested separately at each stage of the manufacturing for the DoseClip. The outline below highlights the different observations that must be completed at each section before declaring the DoseClip to be a fully functional device. This is the feasibility plan; no actual feasibility testing will be performed until the ordered parts arrive as they are necessary for testing.

1. **Microphone Test**
    a. Ranges of frequencies collected
    b. Distance from sound source
    c. Accuracy compared to a current product device as a standard
2. **Pre-Amplification**
    a. Gain verification
3. **Noise Filtering**
    a. Frequency cutoff verification
4. **Programming**
    a. Acquires dBA sound reading successfully
    b. Run time of the programming
5. **Feedback**
    a. Displays the data correctly and when requested
6. **Battery Life**
    a. Test to check the battery meeting our specification of operating for 10-18 hours

1) Testing the selected microphone:

We will generate several pure tones of various frequencies for the microphone to capture. The audio signal collected from the microphone should be able to identify the corresponding frequency used to create audio sound. This can be verified by mapping all the collected signals after performing a Fourier transformation to look at only the frequency content. Once we see that only the frequency generated for pure tone was present for the range of frequency we are interested in, we should look into the accuracy of the microphone.

The accuracy of the microphone will be based on a commercial sound meter since the device would already be confirmed to give accurate dBA readings before distribution to the public. The microphone and the sound meter would be placed horizontally across the ear to standardize how the sound is collected. This is to look at only the sounds collected from the microphone and sound meter without looking at how distance affects the reading. However, once the dBA reading from the microphone of choice reflects the same reading as the commercial sound meter with a maximum deviation of two dbA, we can start looking at the placement of the microphone relative to the ear.

The DoseClip is designed to reflect the sound pressure entering the ear but also to be placed a distance away from the ear. This distance away from the ear can affect the dBA reading as the distance from a sound source tends to affect the reading from a microphone. To check for any changes in reading, we will place the microphone in different locations below the ear. If there is a difference in the reading from the microphone, then we include a correction factor during the programming process. This is so that the microphone can have the same reading as when placed at the same height as the ear.

2) Testing the selected amplifier:

Without amplification, most electronic devices cannot connect to a microphone due to the microphone's low output voltage. However, if the amplification is too high, the audio signal will get distorted, and our device will give a false dBA reading. We will test the gain of our amplifier by using a waveform generator to create a voltage signal to input the amplifier. The output coming from the amplifier would contain the same shape as the generated input signal but with higher peaks due to the gain from the amplifier. The gain will be determined as the ratio of the output signal to the input signal. When we determine that the gain and shape from the output signal are exactly as we predicted, we can connect the amplifier to the microphone since we know that the amplifier would not cause any distortion.

3) Testing the filters:

Creating filters, analogically or digitally, is needed so that the DoseClip only captures the human frequency range of 20 – 20kHz. This is done by creating an A-weighted network filter and a bandpass filter. The bandpass filter would ensure that the frequencies captured are in the range mentioned earlier. The A-weighted network filter ensures that each sound input with frequencies in the 20 – 20kHz range is appropriately suppressed. As mentioned before, we will generate a voltage signal at 20 – 20kHz frequencies and outside that range using a waveform generator. If the generated signal is not detected for frequencies outside the range, we can conclude that the bandpass filter is working as expected. Similarly, we can conclude that the A-weighted network filter is working as expected by taking the ratio of the output voltage signal after filtering and the generated voltage signal before filtering. The ratio here represents the suppression factor which varies depending on the signal's frequency and is needed to compare with the theoretical suppression factor at that frequency.

4) Testing the code:

Programming the microcontroller is important for displaying the dBA reading back to the user and updating the exposure limit for the user. The time to display the data is important and is dependent on the runtime of the code. We will test if our program returns the corrected value by providing an audio file of a known dBA reading in the code. The program will convert the audio file into its frequency content by a Fourier transform and by summing the magnitude of all the frequencies present in the audio file, giving us the calculated dBA reading. If done correctly, the computed value will reflect the known reading of the audio file.

To determine the program's runtime, we let the code return the time it took to process the entire program. If we see that the program was stuck processing the code, we will consider picking other audio files to see how the input size affects the runtime. The file size will let us determine the extent to which the input signal size from the microphone should be during the collection period to reduce the program's runtime while preserving the information needed to get the dBA reading.

5) Testing the Feedback:

The display allows the user to understand what changes were made in the exposure limit without explicitly looking at the program used in computing those changes. In the end, the display will reflect only the dBA reading and the exposure limit. To test the functionality of the display chosen for the DoseClip, we will create a separate program to output text onto the display. This test checks for any defects the display may have and to see if we need to use a different display. Once this test is done, we test the output of the display using the program used in computing the dBA reading, as mentioned in the previous section.

6) Testing the battery:

The battery capacity of the DoseClip is supposed to work within 10 – 18 hours of the typical time that a person is awake. We will test for this capacity by keeping the DoseClip on until the battery is depleted and checking the battery's voltage every 30 minutes using a multimeter. This would let us know how fast the battery is being depleted in the DoseClip and whether we need to consider using a battery with a higher voltage.

Hardware Manufacturing, Validation, and Testing

Hardware Manufacturing

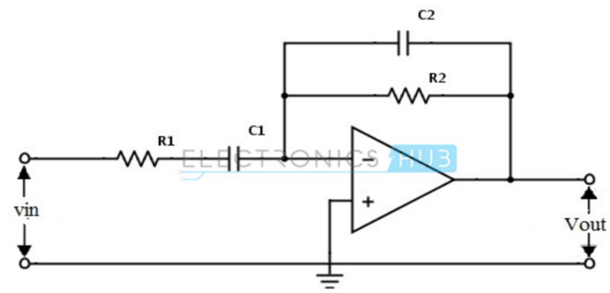
In the manufacturing process of the noise dosimeter, the first step involves creating a circuit simulation to compare results such as magnitude response, voltage output, and gain. Once the ideal schematic is chosen, the next step is to assemble a circuit prototype of the simulation to compare the results mentioned above of the stimulation to the prototype design. Once that was done, the circuit was scaled down and manually assembled in a pinhole printed circuit board (PCB) to avoid outsourcing and to reduce cost of manufacturing. To design the layout of the electrical components (such as resistors, capacitors, and op-amps) their physical location relative to the PCB must first be determined as to prevent shorting and electrical failure. In addition to the individual electrical components, the location of other device parts, including the microphone output and microcontroller must also be taken into account regarding their connection to the PCB.

Soldering was done independently, with assistance from the Makerspace and at the Center for Discovery and Innovation (CDI) neural engineering department on the third floor. To calibrate the microphone, a Sound Pressure Level (SPL) meter was provided from Dr. Luis Cardoso's research lab.

An important part of the manufacturing was the creation of filters to consider only the human auditory range of 20Hz - 20kHz and reject all other frequencies. OSHA We initially used an active band pass filter due to its feasibility in assembling the circuit but disregarded this design since the weighting at each frequency was about the same inside the passband was about the same (Fig. 45)

Our design changed from using analog active band pass filtering (Fig. 45) to using an analog A-Weighting filter. This was done due to the fact that the cutoffs for the active bandpass were not ideal (Fig. 46) meaning that the signal beyond the cutoffs was not suppressed as needed and would require more processing power to adequately adjust it. The A-weighted filter suppresses the signal with proper cut-offs according to the OSHA standard which would prevent any further processing by the microcontroller (Fig. 46). In order to build this circuitry we prototyped the schematic of the circuit on a DigiliantKey Studio breadboard. Regarding the A-Weighting filter, we initially adopted the schematic in Fig. 47 but with further testing, used the ANSI A-weighting filter was used (Fig. 48) with an induced positive voltage bias to work with the microcontroller (Fig. 49).

After testing all individual components, we focused on reducing the size of the A-weighted ANSI filter. The majority of changes were made by decreasing the number of wires connecting each electrical component, changing the capacitors and resistors values to reduce the number of electrical components, and reducing the size of electrical component contact points using a wire clipper. The changes in size can be observed in Figures 50a and 50b. Furthermore, we used the components mapping of the second prototype of the ANSI A-weighted filter (Fig. 50b) in order to manufacture the final device A-weighting filter by soldering the components to a through-hole perfboard which has a platform to connect all the other components with the microcontroller (Fig. 51).



R1= 2.2 k, C1= 3.3 uf/ C2= 0.01 uf, R2= 1.5k

Figure 45. Active Band Pass Filter Schematic showing the arrangement of the different capacitors and resistors and their connection to the op amp leading to the output

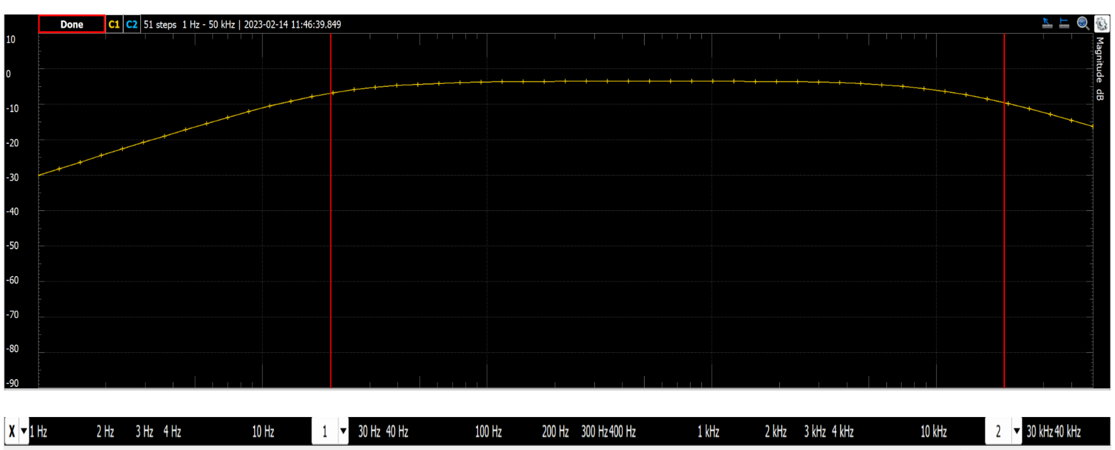


Figure 46. Active BandPass Filter Magnitude Response with the red lines referring to 20 Hz and 20 kHz. The graph shows the suppression of the signal beyond the red lines

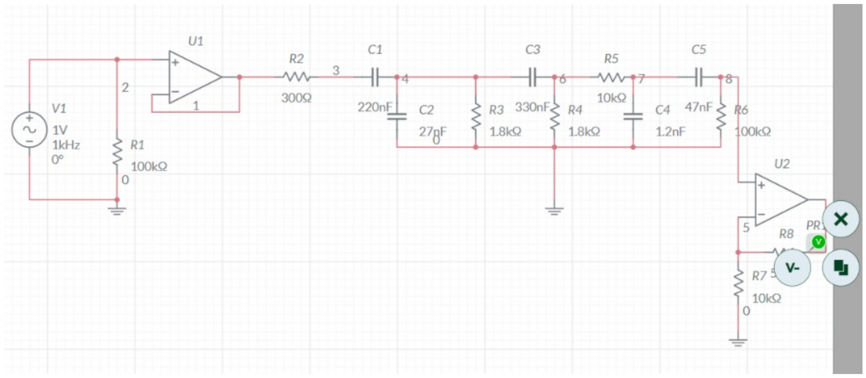


Figure 47. A-Weighting Band Pass Filter Schematic showing the cascaded of the capacitors and resistors in connection with 2 op-amps (follower and noninverting) in series to form the A-Weighting

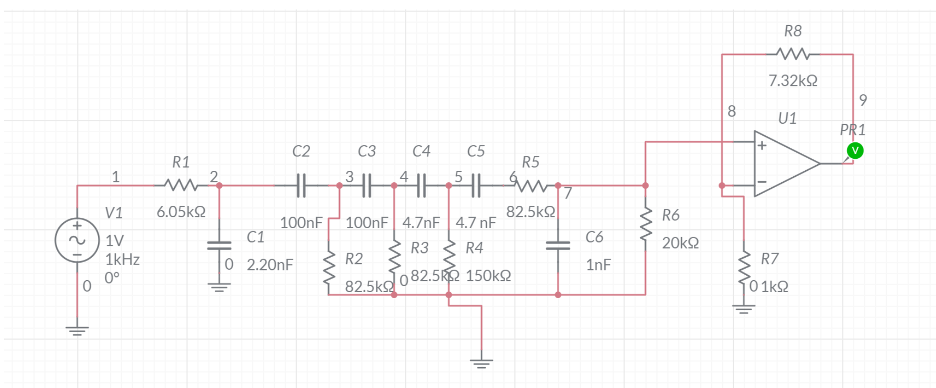


Figure 48: ANSI A-Weighting filter Schematic showing the cascaded resistors and capacitors connected to a noninverting amplifier in series to form the A-Weighting filter

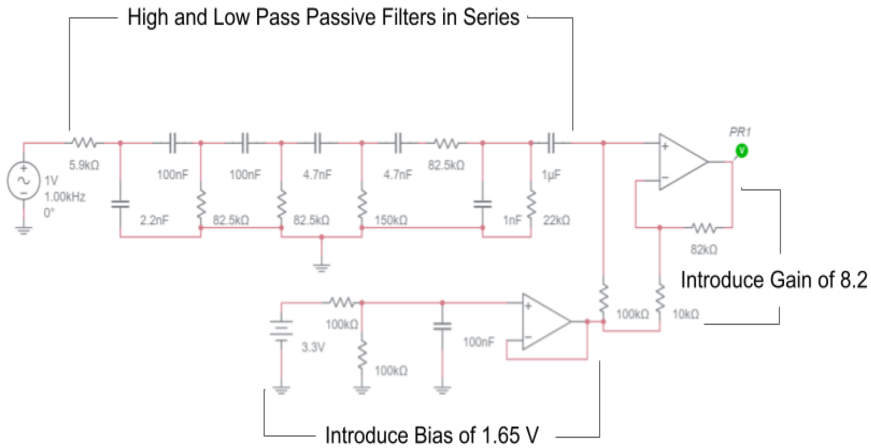
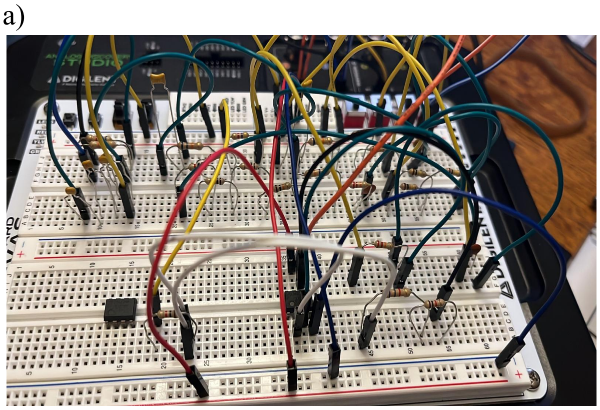


Figure 49: ANSI A-Weighting filter Schematic with modification to include a bias of 1.65V.

a)



b)

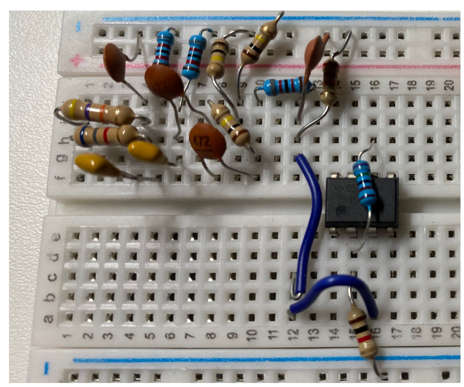


Figure 50. a) Initial prototype of ANSI A-weighted filter b) Second prototype of the same filter after reducing the number of wires used in assembling the filter and using fewer components

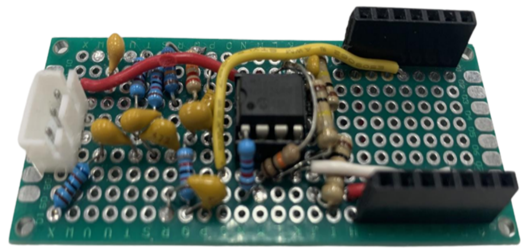


Figure 51. Final A-weighted filter on PrefBoard with Platform (black female pins) to connect with the microcontroller and other components

For hardware manufacturing, the housing and circuitry were the two individual parts we focused on. The aim of the housing is to protect and contain all the circuitry parts while making the device wearable and easy to interact with. Regarding the circuitry, its aim is to make accurate measurements of the environmental noise and integrate the software with the device while making the appropriate adjustments to the signal being processed.

Using Solidworks, a housing unit for the device was designed. Initially, the housing was designed with port holes for USB-C power cables, so that the internal microcontroller can be charged without having to remove it from the housing, a port for the microphone so that it can be placed on the exterior of the housing, and a socket for a clip on the bottom of the housing so that it can be attached to the user's clothes (Fig. 52).

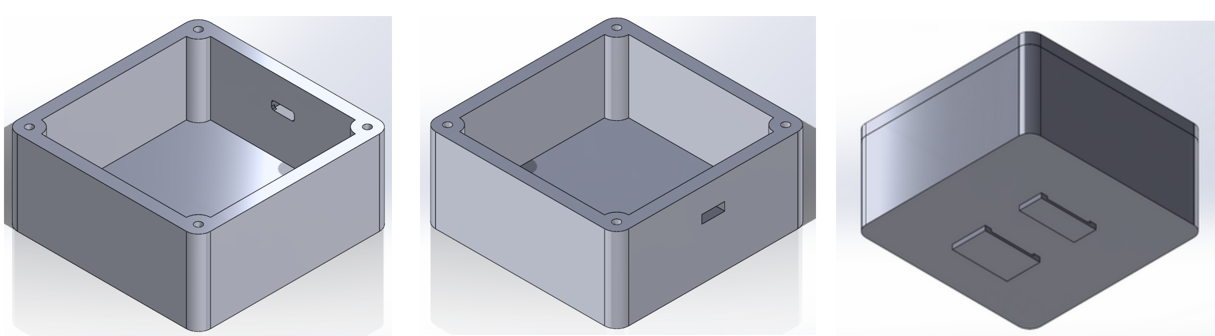


Figure 52: a) Front view of the housing with the port for USB-C b) Back view of resized housing with a port for USB-C c) Bottom view of resized housing with a port for USB-C

As the design of the housing continued to be improved upon, the socket for the clip was removed in favor of gluing the clip on so that the housing could be printed without supports. Two ports were also created, one for the LCD screen that would be inserted through the exterior, and one for a sheet of plexiglass that would hold the screen in place as well as protect it (Fig. 53).

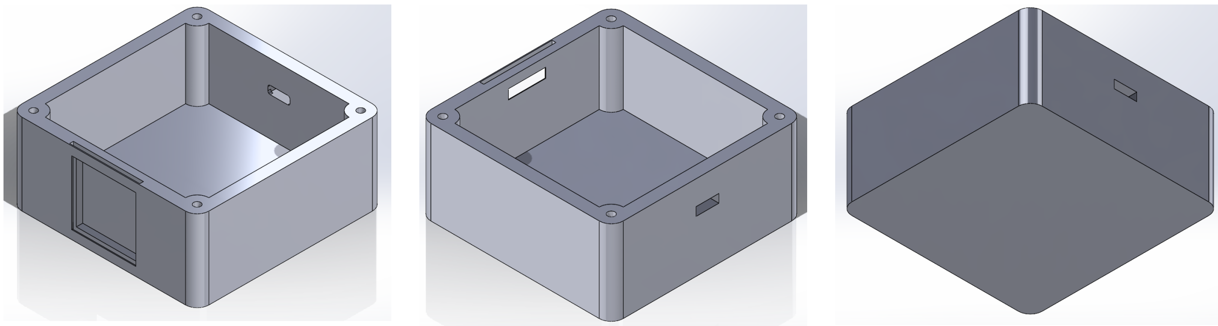


Figure 53: a) Front view of the housing with the port for LCD screen and plexiglass protector b) Back view of resized housing with port for LCD screen and plexiglass protector c) Bottom view of resized housing with port for LCD screen and plexiglass protector

By the time of the midterm, the dimension of the housing was changed from 8x8x4cm to 5.5x7.5x4cm and the walls were made thinner to make the device less bulky and easier to wear (Fig. 54).

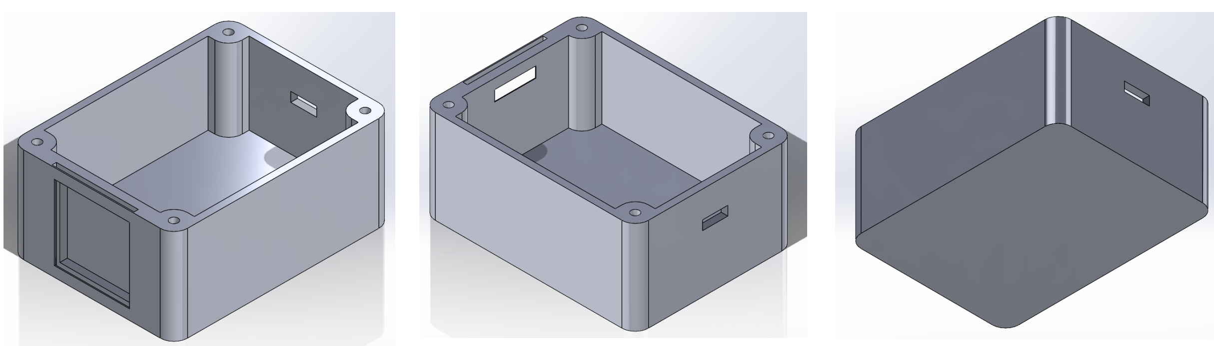


Figure 54: a) Front view of resized housing b) Back view of resized housing c) Bottom view of resized housing

By the time the current design of the housing was finalized, the dimension of the housing had to increase from the previous 5.5x7.5x4cm to 5.6x8x3.75cm, due to the increased size of the new battery. A port was also added for the power switch, the slot for the LCD screen was extended slightly to make more room for the acrylic screen protector, compartments and platforms were added to hold the individual components in place, and the design of the corners was changed so that it could better hold the roof in place. (Fig. 55).

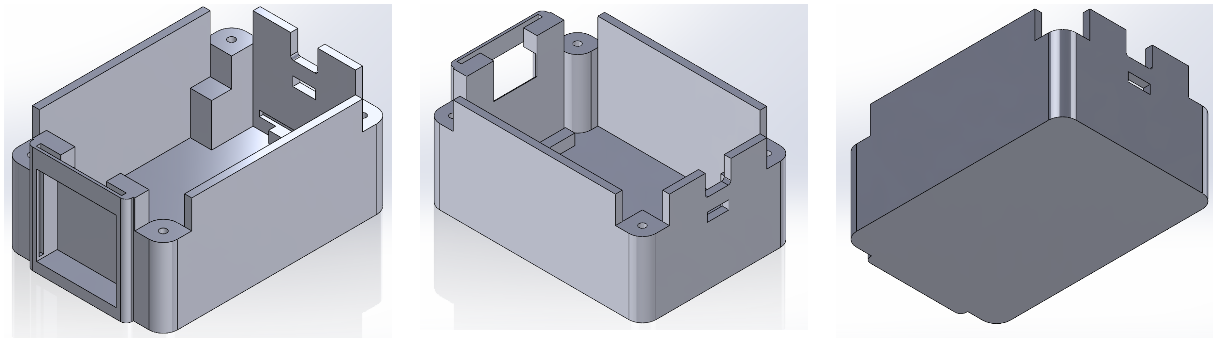


Figure 55: a) Front view of final housing b) Back view of final housing c) Bottom view of final housing

Initially, the roof of the housing used pegs to attach itself to the base and had an exterior port for the microphone. However, as the design changed, the pegs were replaced with screw holes to nullify the risk of the pegs breaking inside of the base and to make it more secure. The microphone port was also changed from exterior to interior and a hole was made so that the measurements would still be accurate (Fig. 56).

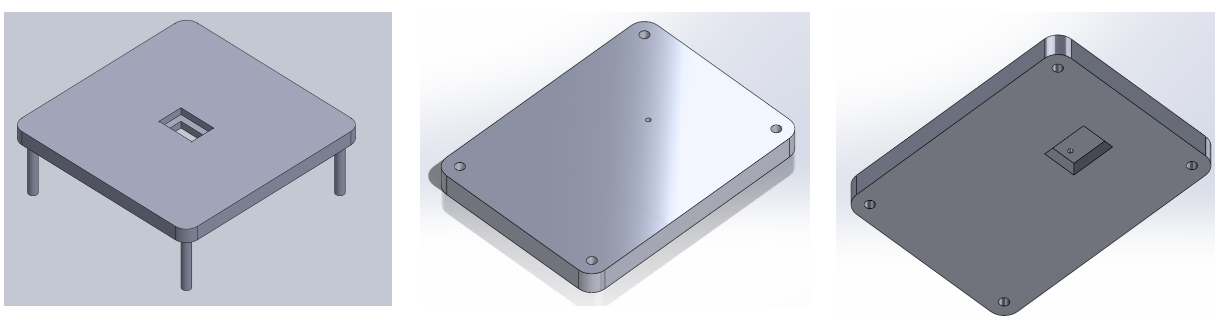


Figure 56: a) Roof component with pegs and an exterior microphone port b) Roof component with holes for screws and resized for new housing c) Roof component with socket for microphone

Currently, the final design has been completely overhauled. The roof now consists of two pieces, so that the microphone can be held in between them. Multiple holes were also added so that the LED light could be visible and for that the user would be able to use a pin to press the buttons on the microcontroller should the need ever arise (Fig. 57).

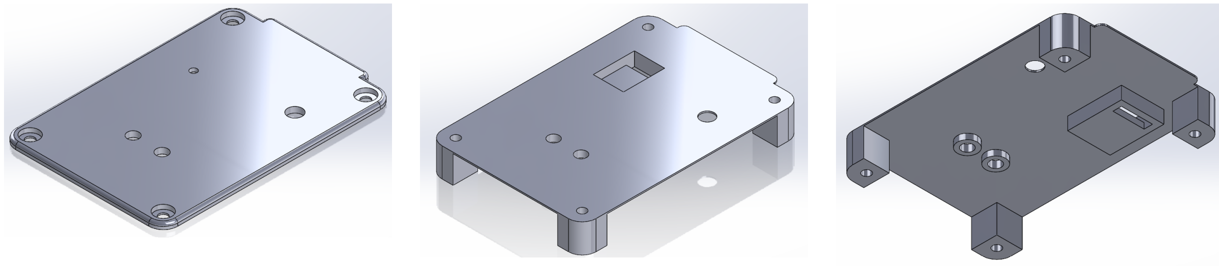


Figure 57: a) Top layer of the roof with holes for access to the microphone, LED, and buttons b) Bottom layer of the roof featuring a compartment for the microphone c) Underside of the bottom layer of the roof

Using the various features Solidworks offers, the housing can be manufactured as multiple components so that they can be assembled through a simulation. To further this, replicas of each of the other components can be designed and placed within the housing to be able to accurately test whether or not the design of the housing works. Small replicas of the ports featured in the housing were also replicated to reduce the number of times the housing would have to be printed for testing, thus saving both time and materials (Fig 58.). After the design of the housing is completed, it is 3D printed using materials and equipment provided by the Makerspace, such as the Ultimaker 2+ Connect and the Polylactic Acid printing filament. Upon completion, the housing will be assembled and inspected for any modifications that might need to be made. Any differing versions of the housing will also be compared to determine which one has the greatest effectiveness.

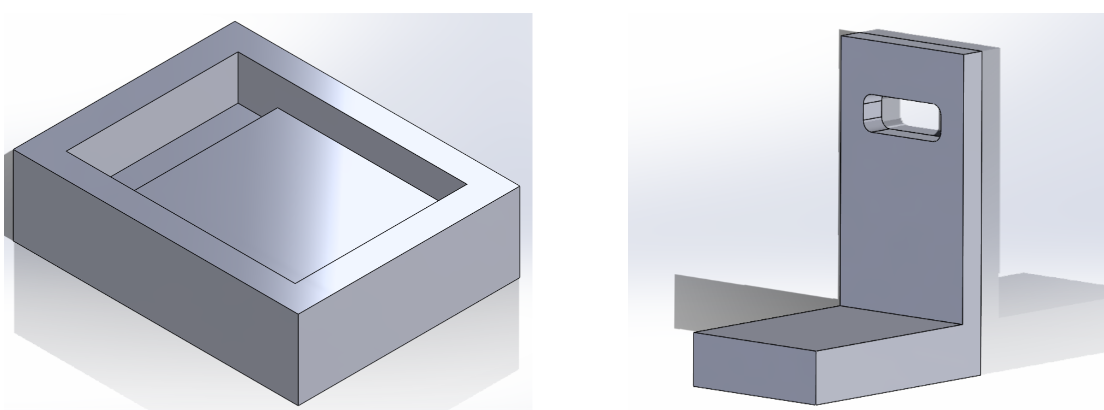


Figure 58: a) Cross section of roof for testing how the microphone fits b) Cross section of wall for testing how the USB-C port fits

In the manufacturing process of the noise dosimeter, the first step involves creating a circuit simulation to compare results such as magnitude response, voltage output, and gain. Once the ideal schematic is chosen, the next step is to design a surface-mounted printed circuit board (PCB) using a free open-source program such as KiCad EDA. However, acquiring a custom-designed PCB in small quantities can be quite expensive. Instead, a cost-effective alternative was to use a two-layer pinhole PCB. One tradeoff of using a pinhole PCB is that it offers better flexibility for assembling a permanent connection between each electrical component at the cost of limiting the choices of PCB sizes. To design the layout of the electrical components (such as resistors, capacitors, and op-amps) their physical location relative to the PCB must first be determined. Next, the components are soldered onto the PCB to create a permanent connection. During the soldering process, ensuring that the soldering wire is applied to the appropriate location is crucial to prevent electrical shorting, which may result in the entire circuit being ruined. In addition to the individual electrical components, the location of other parts, including the microphone output and microcontroller must also be taken into account regarding their connection to the PCB.

Soldering can be done independently, with assistance available at the Makerspace or at the CDI building's neural engineering department on the third floor. To calibrate the microphone, a Sound Pressure Level (SPL) meter is needed, which can be accessed at Dr. Luis Cardoso's research lab.

Testing Methods

We tested the magnitude response of three different A-weighted filters at the time of prototyping using an online simulation and compared it to the OSHA magnitude response (Fig 59). According to the online simulation we expected to have the graphs shown in figure 59a when we perform a magnitude response analysis for each of the three different schematics of the different A-Weighting filters. Thus after building the circuit, we used the Analog Discovery Studio tools to analyze the magnitude response and compare it to the expected graphs shown in Figure 59.

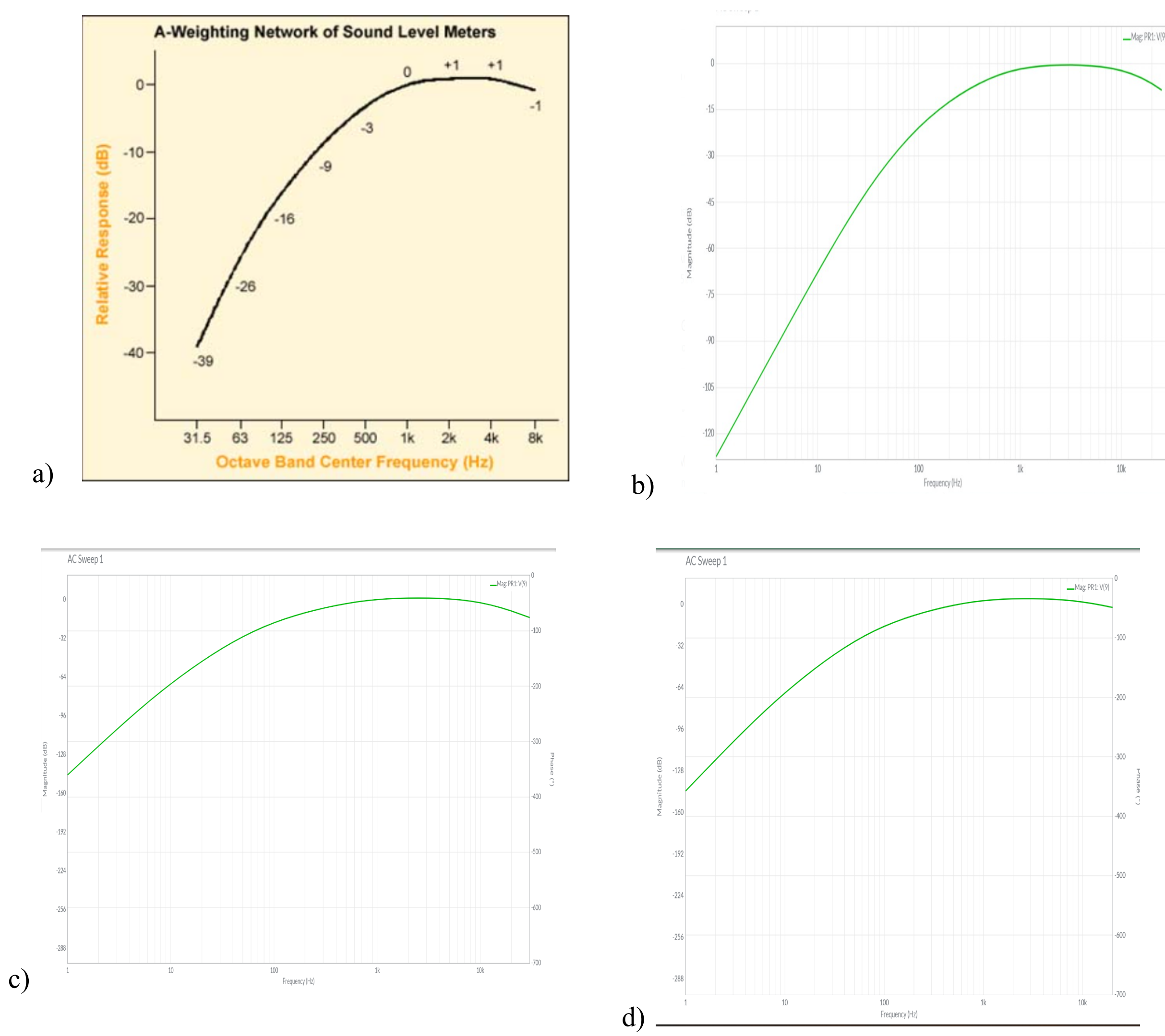


Figure 59. Comparison of (a) OSHA Magnitude Response to prototype (b) A-Weighting Filter , (c) ANSI A-Weighting Filter and (d) Adjusted ANSI A-Weighting Filter shown sequentially from left to right.

We tested the compatibility of the MEMs SPH8878LR5H-1 microphone, OLED display, and Xiao Seeed ESP32C2 microcontroller. When all three device components are connected together, the microphone should pick up the sound pressure reading while the microcontroller send a command to the display to show the voltage input equivalent of the sound pressure reading. If a numerical output was observed, it confirmed their proper functionality, as shown in Figure 60.

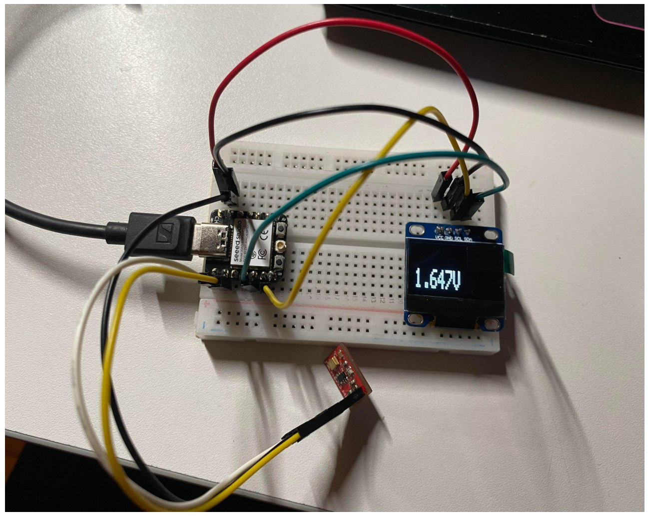


Figure 60. Successful connection of Xiao Microcontroller, MEMs microphone, and OLED screen displaying voltage output from the microphone.

After successfully testing the compatibility of the microphone with the microcontroller and display, we proceeded to incorporate the A-weighted filter, the external storage card for saving the microphone sensor data points, and an LED indicator to improve the user interaction with the device. To test the compatibility of the storage card, we compiled an initialization test for the microcontroller using the Arduino programming language, which is designed specifically for compatible microcontrollers (Fig. 62). This test helped to determine whether the microcontroller was properly connected to the storage component and whether it could access the file inside the storage card. To test the compatibility of the LED with the other components, we compiled a command into the microcontroller to turn on the light after the sensor reading was above the predetermined sound threshold set on the microcontroller.

In order to test the hardware compatibility with the code section, we use the current circuit in Figure 60, and an Arduino Uno board instead of the Xiao Seeed since it is easier and faster to make changes in the code and test it on the spot without having to reboot the microcontroller which is shown in Figure 61.

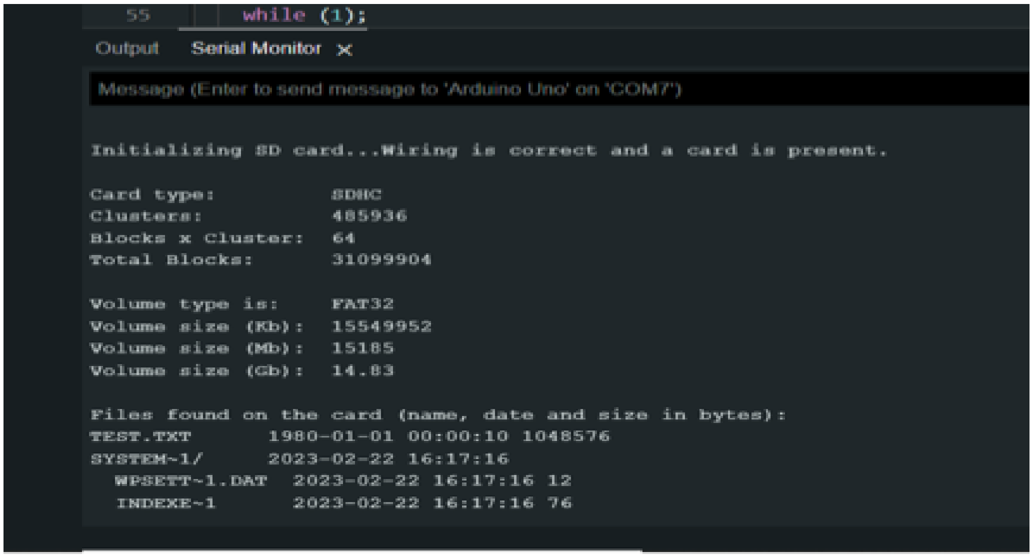


Figure 62. Initialization test for the storage card. The initialization test includes the type of storage card connected to the microcontroller and the file that is located inside the storage card.

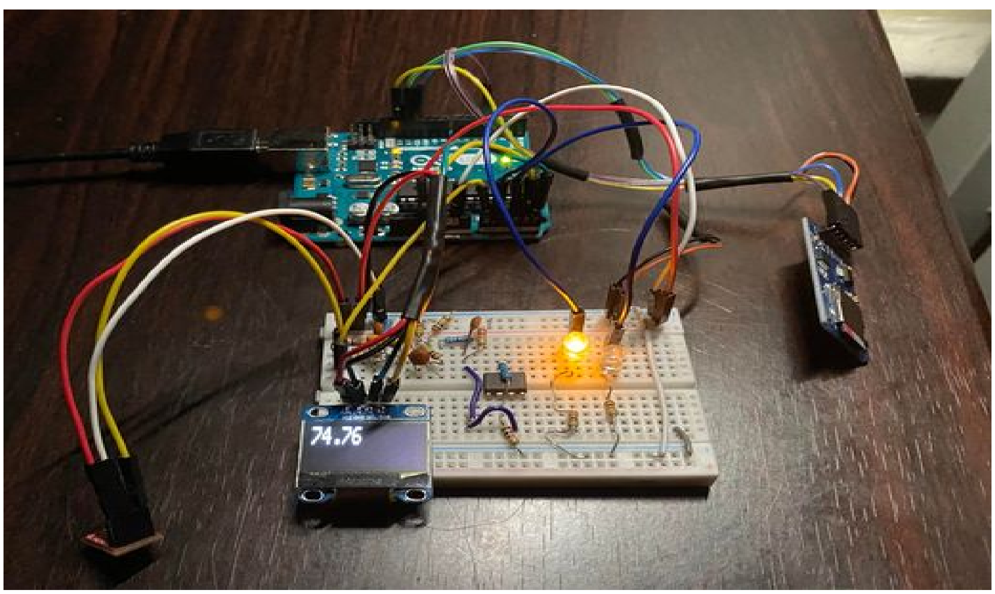
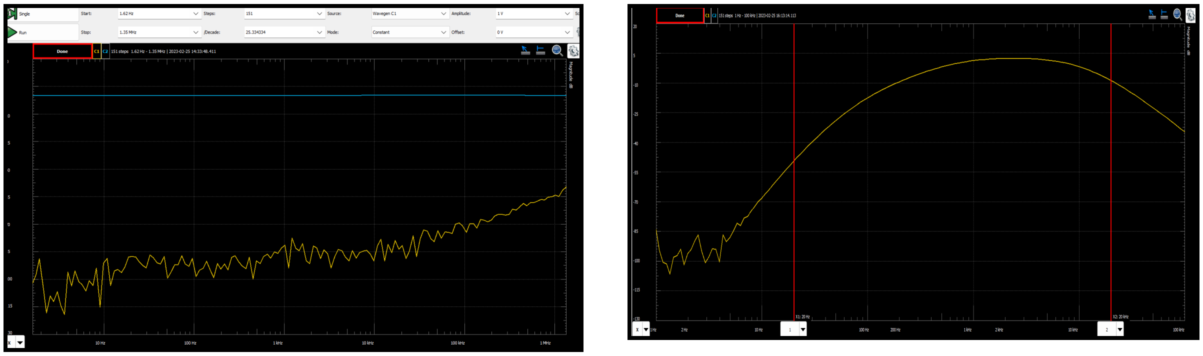


Figure 61. Addition of LED indicator, ANSI A-weighted filter, and storage card module to the previous prototype. The sensor detected a reading of 74.76 dBA, causing the LED to light up upon reaching a threshold of 70 dBA.

Hardware Validation

In order to validate whether the A-Weighting filter suppressed the designated frequencies or not, we performed magnitude frequency analysis using the Analog Discovery Studio feature and then tested the chosen filter with the MEMS microphone where we showed the signal's frequency vs. magnitude (Fig. 63). Fig. 63a shows that the magnitude response was not as expected (Fig. 59a), thus another A-weighting design was implemented and is shown in Fig. 16b the result was satisfactory. Then in order to verify that the Mems microphone signal would get attenuated by the A-Weighting filter, we passed the signal through the filter and did magnitude analysis, and compared it with the input signal from the microphone (Fig. 64). As it can be shown in Fig. 64 the output signal from the filter is suppressed at lower frequencies and amplified at higher frequencies, which matches with the magnitude response analysis (Fig 63b). Furthermore, the smaller prototype of the initial ANSI A-weighting was validated using the same process. From the magnitude response graph shown in Fig. 65, it is observed that the curve is more fitting an A-weighting profile which means it is a better representation of the A-Weighting. This was verified by comparing the magnitude responses of the final A-weighted filter, simulation, and the OSHA guideline values for an A-weighted filter (Fig. 66). As it can be observed from fig 66 the final A-weighted filter magnitude response (shown in orange) is overlaying the OSHA standard (shown in green). To validate the battery life of the device with the design specs, the device was connected to a power source to measure the current draw which was found to be 111 mA (Fig 67). This means that using a 2500 mAh battery will power the device for 22.7 hours on one full charge. Furthermore, in order to test whether the audio signal will be suppressed by the right filter weighting, 1 kHz audio was subjected to the mic and the signal was observed prior and after the A-weighting were compared (fig 68). According to the OSHA standard, the audio signal should not be impacted at 1 kHz (0 gain), and as it can be observed the signal was unchanged (fig. 38).



a) b)

Figure 63. Magnitude Response of the a) A-Weighting Filter (Fig. 9) b) Adjusted ANSI A-Weighting Filter (Fig. 8)

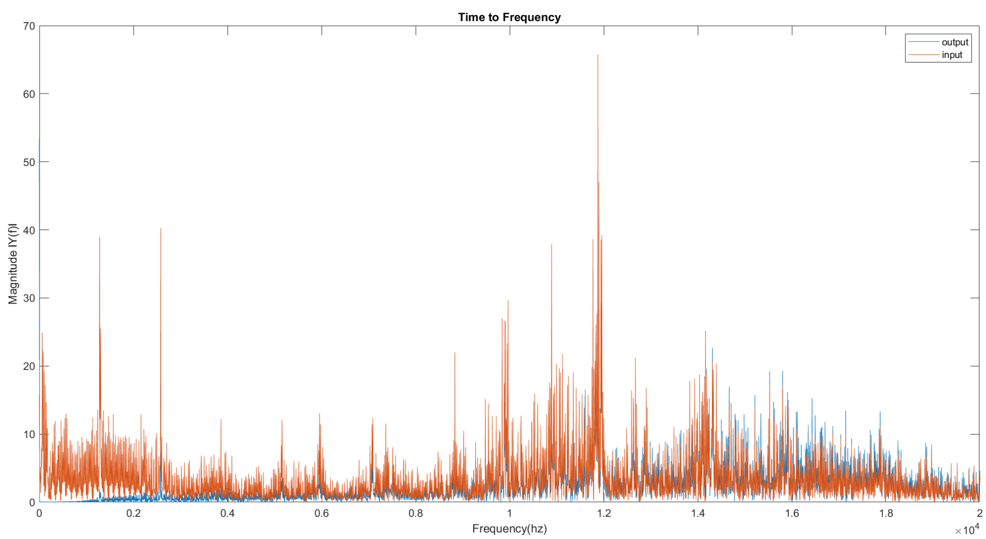


Figure 64. Microphone Frequency Signal input (Orange) vs. Microphone Frequency Signal passed through the filter output (Blue)

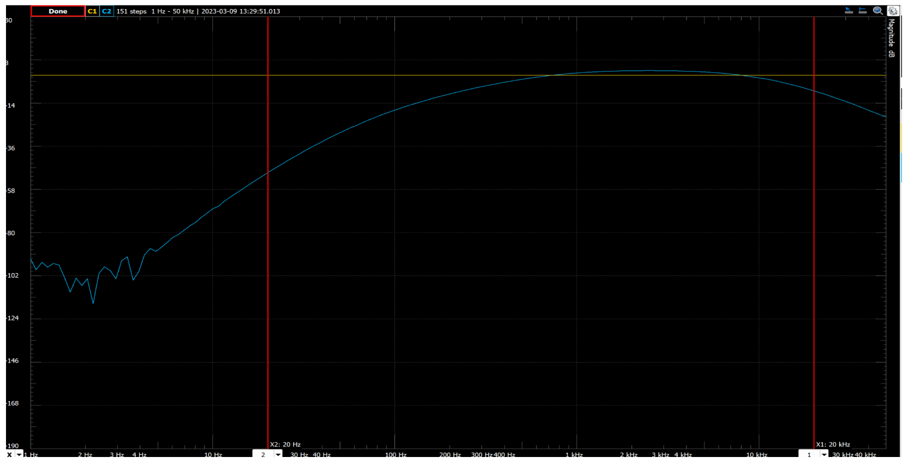


Figure 65. ANSI A-weighting filter smaller prototype (Fig xx) magnitude response. The blue line shows the magnitude response to a signal (which range is suppressed and which range is amplified). The yellow line represents 0V and the red lines represent the cutoffs of 20 Hz and 20 kHz

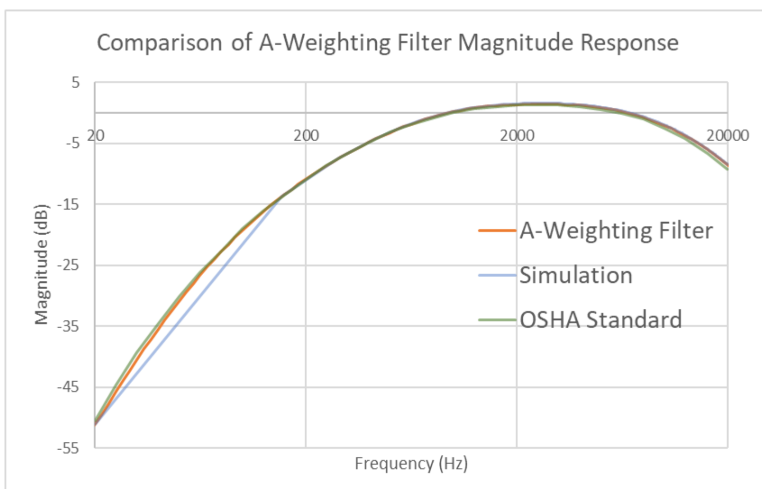


Figure 66. The magnitude response of the A-weighted filter (orange), the Multisim simulation (blue), and the OSHA standard (green)

Figure 67. The current draw of the full device (highlighted in green) from a power source

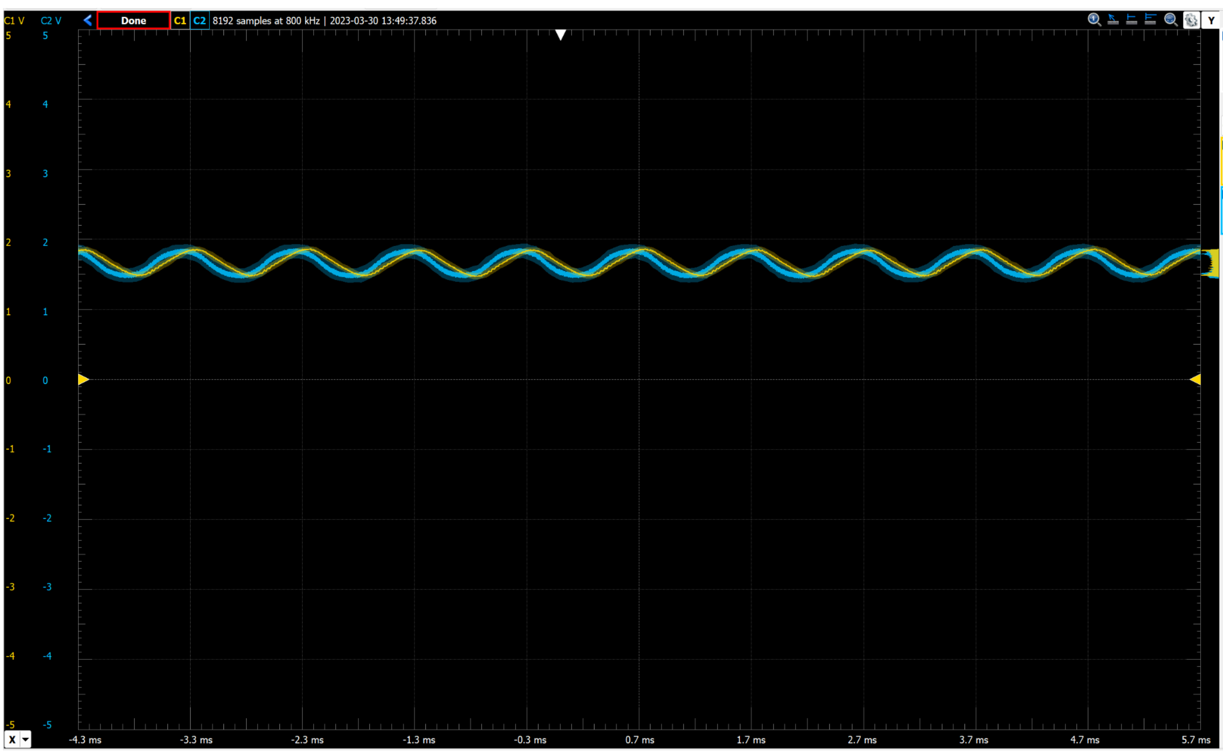


Figure 68. Audio signal processed by the filter: Input Signal (yellow), Output Signal (blue)

.

Quality Control: Case

To ensure that the device was capable of withstanding stress from impact, the drop test was performed on the housing after it was covered in foam to increase its resistance. To perform this, the housing was dropped from shoulder height, three times, each in a different orientation that the device is most likely to be in while falling. Upon completing this, the housing was inspected for damage. No discernible damage was found upon inspection (Fig. 69).

Figure 69: a) The acrylic screen remains intact b) No damage was sustained

Software Manufacturing, Validation, and Testing

Sound Collection

In the MATLAB-based code, we generated a collection of raw audio data that utilized the 'Data Acquisition' package to generate real-time audio data for intervals of 5 seconds. This was able to be created into a 'for-loop' function that gave an output of continuous 5-second interval audio data to get processed into a built algorithm (Fig. 70B, 70C). Before it can be processed in dBA, we have to convert the voltage values getting inputted into the microphone into decibels. We use the equation $dB = 20Log_{10}(\frac{V}{V_{ref}})$ to convert the voltage values into working decibel values. Once this is done, the dBA values are averaged over the 5-second interval and can be implemented into the algorithm to check for its associated weight.

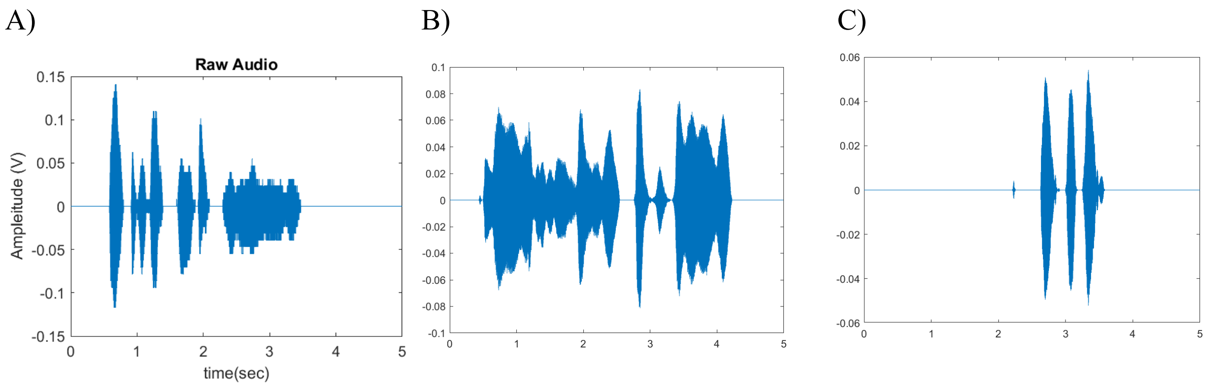


Figure 70: Transition to a collection of continuous raw data output 20 A) Saved audio single file 20 B), 20 C) Continuous audio output to be saved

Sound waves are constantly fluctuating which causes the voltage readings to vary. Taking a random sample in one-second intervals will not give over the proper noise readings as it could be taken during any point of the wave. Peak-to-peak voltage values were implemented to receive the wave's amplitude or loudness. The peaks were found using simple if statements, if the voltage is greater than the maximum, it is the new maximum (peak), and if the voltage is less than the minimum, it is the new minimum (negative peak). The amplitude of the signal was then the peak minus the negative peak divided by two (formula for the amplitude of a wave). Peak-to-peak voltage gave a good estimate of the amplitude values but unfortunately, sound signals are often extremely noisy. When using peak-to-peak, there is a large chance that the peak values will signal noise rather than the desired signal amplitude.

To correct the noisy sound voltage signals, a root mean square of the signal was taken. The basic approach is to take voltage readings over a known period (1 second) with a known sample rate (approx. 14,000). The RMS equation is as follows: $\sqrt{\frac{1}{N}(x_1^{\ 2} + ... \ + x_n^{\ 2})}$, where x represents the voltage values incoming to the microphone. This value will take all voltages of the period into account which will reduce the effect of the noise.

This new RMS signal will be a better indicator of the actual noise level because it is stable over time. The RMS voltage values will be plugged into the calibrated dBA equation to give accurate noise level readings. The dBA values will then be averaged every 5 seconds to help avoid noisy artifacts such as tapping or clothing ruffling the microphone. These measurements will be plugged into the allowance algorithm taking the 5-second interval length into account to properly weigh the sound level.

Microphone Calibration

Microphones must be properly calibrated to output accurate dBA readings. We used a SPL meter from Dr. Cardoso's lab and the anechoic (sound resistant) room to test for calibration. The microphone and SPL meter were placed in alignment to gather the same sound signal (Fig. 71).

In order to calibrate the microphone:

1. The generation of a 1kHz pure tone noise with varying distances and volumes to display desired decibel level on the SPL meter starting at 65 dBA up to 115 dBA.
2. The RMS voltage values were recorded on an Excel spreadsheet by running the code in conjunction with the SPL meter after the stabilization period (5 sec).
3. The voltage vs decibels was plotted and fitted to obtain the calibration equation.
4. Optional
    a. Using the basic voltage to dBA formula, plug in the voltage values to get potential decibel values
    b. Plot the potential dBA values to the output from the SPL meter, and fit equation
5. If a full equation doesn't fit the entirety of the curve, split the curve into its linear and curve components.

Based on the data we obtained from our microphone, there wasn't an equation that could fit the entirety of the curve so we partitioned the data into two components:

a. From 80 - 90 dBA: x = RMS; $y = -3.1672x^2 + 545.23x - 23372$
b. From 90 - 115 dBA: x=20log(RMS/0.00002); y = 2.1571x - 93.162

Through both of these equations, we were able to acquire near-accurate decibel levels similar to the SPL meter. These values, however, were calibrated with data using a 1kHz pure-tone sound. According to the frequency response of our microphone (Fig 72a), we should be receiving a near-flat line from 40 - 6000 Hz. This was proven to be false when we tested against multiple frequencies to get a logarithmic curve (Fig 72b). Hence, this was a limitation of the microphone, however, our equations don't waver past a 10-decibel difference across extreme frequencies.

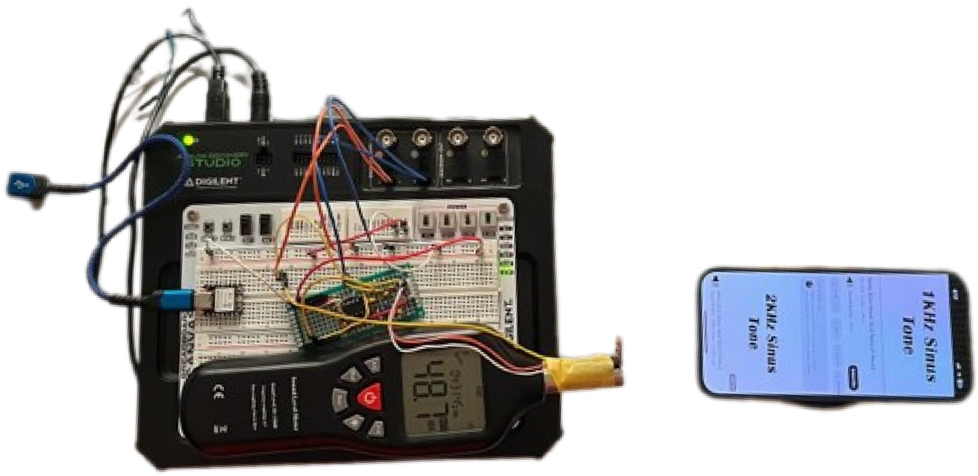


Figure 71: Calibration set-up with an SPL meter

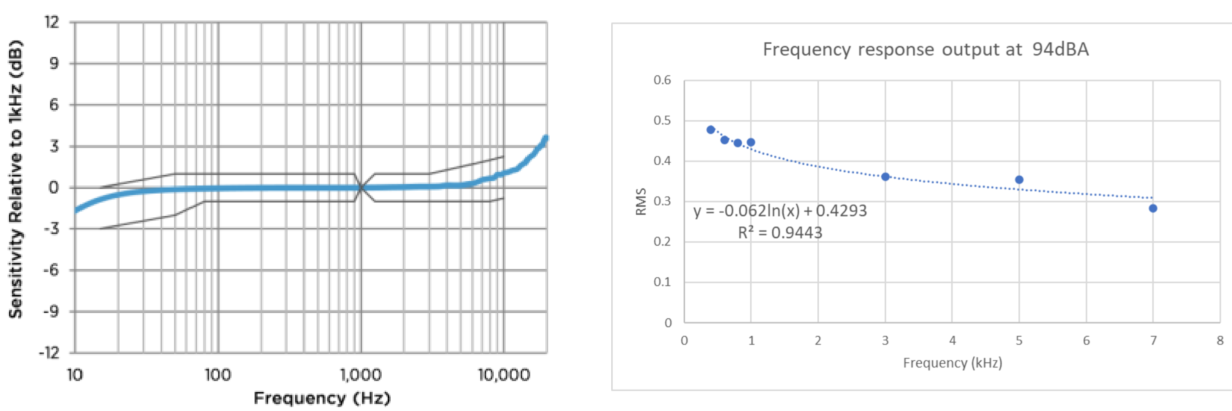


Figure 72: The validation of frequency response of the microphone (a) The frequency response of the microphone (b) Tested response

Algorithm Manufacturing

The goal of the algorithm is to create an allowance system that will take in dBA values from the sound level meter (SLM) portion of the device and give feedback to the user. The algorithm will be implemented subsequently to the SLM by averaging 12 dBA values per minute, one measurement every 5 seconds, and then plugging it into the algorithm's equation to properly weight the sound levels. The dBA level weights will be deducted from the total allowance and continuous user feedback will be provided.

The algorithm was made in Python using OSHA's 5 dB exchange rate as a base and the Polyfit function to give continuous weighting for all dBA levels including indiscrete values (ex: 92.4 dBA, 144 dBA, 71.9 dBA). The plot and equation for the weightings are logarithmic (Fig. 73). This equation will allow for the seamless integration of the SLM dBA value into the weighted sound level. Each minute's value will be weighted and then taken off of the day's total allowance.

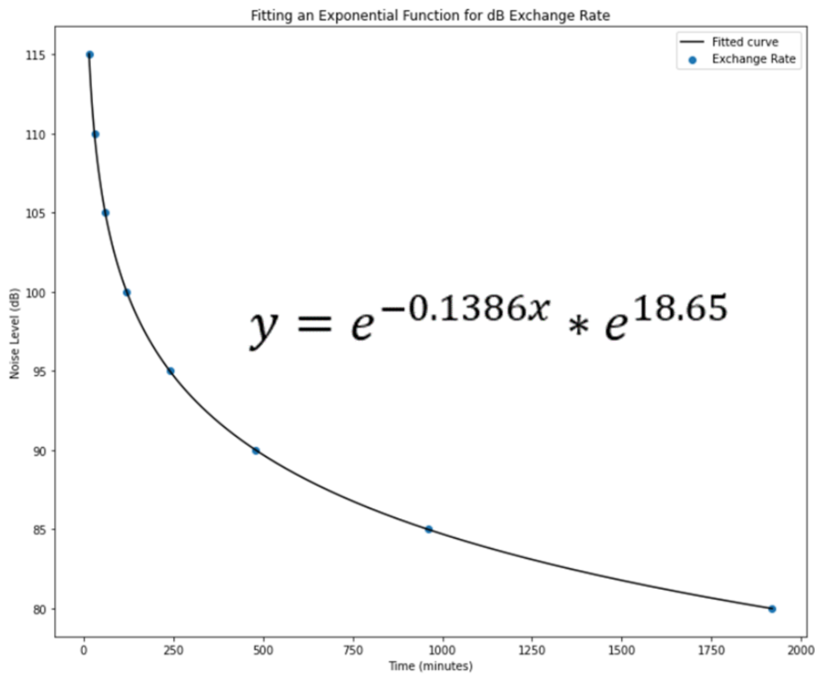


Figure 73: Exponential Curve and Equation that Describes Continuous dBA Values Weightings

To test the feasibility of the algorithm, a randomized 100 x 12 matrix of values 80-125 (dBA) was generated to mimic the output of the SLM. Each 12-column row was averaged to model the samples per minute and the 100 rows will be 100 minutes. The high 80-125 dBA range was chosen to inflate the measurements to show an exaggerated change in allowance. The averaged dBA values were plotted to show that most measurements were above 100 dBA (Fig. 74). The averaged dBA array was inputted into the equation which deducted percentages from the daily allowance (Fig. 75). A high dBA level for a long duration was used to demonstrate the ability of the algorithm on an exaggerated scale. In a normal situation, the dBA values would be lower, and the allowance will not deduct as rapidly. The large percentage drops will be indicated by the output of the algorithm. A percentage and information about the allowance level will be provided for each new measurement (Fig. 76).

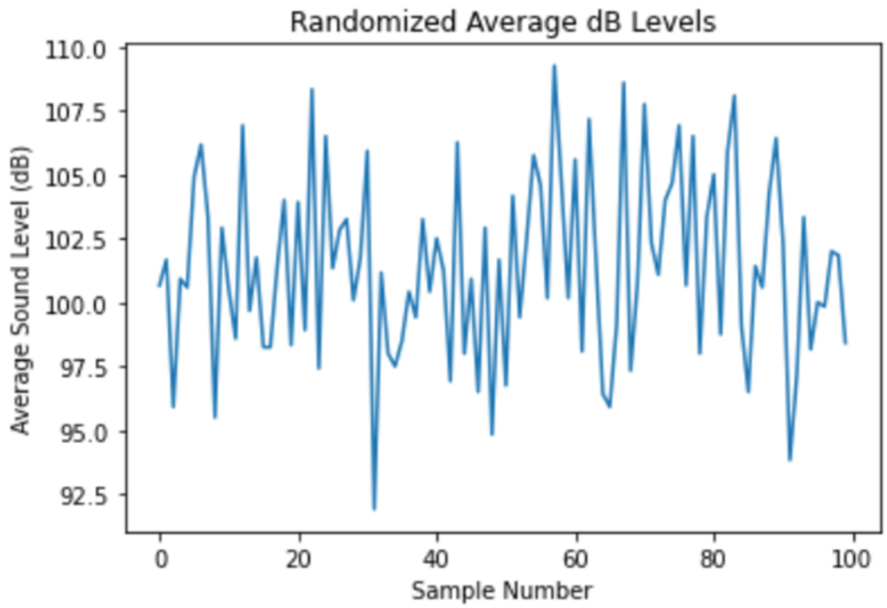


Figure 74: Randomized dBA Averages are Plotted

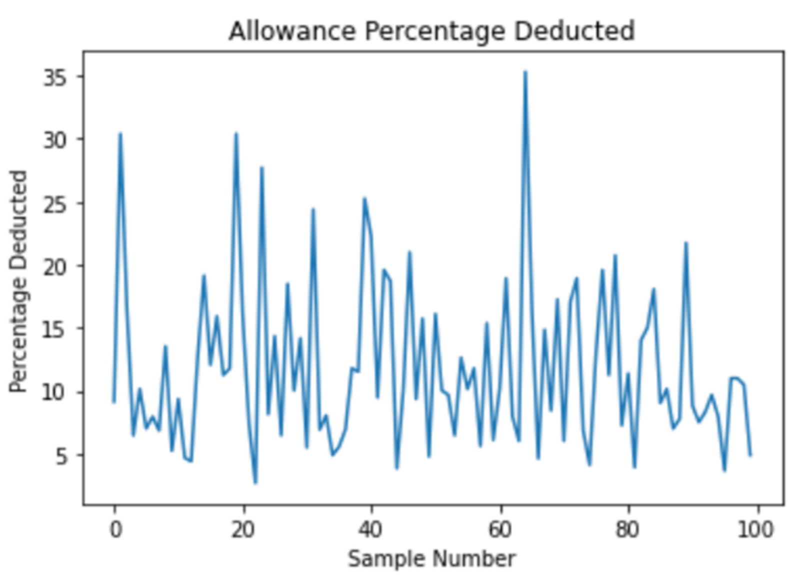


Figure 75: Exaggerated Allowance Percentage Deducted per Sample

```
100 % Start Allowance for Day
78 % Remaining, Healthy Enviornment
68 % Remaining, Healthy Enviornment
58 % Remaining, Be Careful go to Quieter Enviornment
39 % Remaining, Be Careful go to Quieter Enviornment
28 % Remaining, Be Careful go to Quieter Enviornment
19 % Remaining, Be Careful go to Quieter Enviornment
12 % Remaining, Be Careful go to Quieter Enviornment
0 % Remaining, Limit Reached, Risk for Hearing Loss!
22 % Over Limit, Danger for Hearing Loss!
28 % Over Limit, Danger for Hearing Loss!
```

Figure 76: Allowance Percentages and User Feedback

Algorithm (Debugging) Validation

To validate the algorithm, a more realistic data source was generated instead of making a matrix of dBA values, averaging the values, and then applying it to the algorithm. The task was to perform all the previous steps together in a continuous fashion. This was achieved by applying a while and a for loop (Fig. 77). The randomized 80-125 (dBA) data was produced continuously, which meant having many samples per second which are similar to the output of the SLM. Obtaining just one value every 5 seconds was made possible using the sleep function. Each minute 12 measurements were averaged and plugged into the algorithm for weighting. Each weighting was then deducted from the allowance and feedback was provided. This was all done in a loop, allowing each minute to be a separate calculation rather than using a matrix where all values were obtained simultaneously (Fig. 78).

```python
seconds = 0
AVG_dB=[]
Time=10 #This will be 1 for 1 minute
Allowance_Start=100#Allowance Will start at 100%
print(round(Allowance_Start),'% Start Allowance for Day')
while (seconds<24):
        dB=[]
        for x in range(12):
            dB.append(np.random.randint(80,125)) #going to be dB values from SLM
        #print(dB)
        seconds = seconds + 1
        time.sleep(0)
        AVG=np.mean(dB)
        Allowance_Start=Allowance_Start-(100*Time*(1/(np.exp(p[0]*AVG)*np.exp(p[1]))))
        if Allowance_Start > 60:
            print(round(Allowance_Start),'% Remaining, Healthy Enviornment')
        elif Allowance_Start>10:
            print(round(Allowance_Start),'% Remaining, Be Careful go to Quieter Enviornment')
        elif Allowance_Start>0:
            print(round(Allowance_Start),'% Remaining, Limit Reached, Risk for Hearing Loss!')
        else:
            print(round(abs(Allowance_Start)),'% Over Limit, Danger for Hearing Loss!')
        #AVG_dB= AVG_dB+[AVG]
#print(AVG_dB, 'dBA')
```

Figure 77: The Python Code Containing While and For Loop Used to Generate Realistic SLM Data for Algorithm Validation

```
100 % Start Allowance for Day
[117, 111, 104, 81, 82, 97, 110, 93, 81, 120, 122, 105]
101.91666666666667
99 % Remaining, Healthy Enviornment
[111, 120, 113, 123, 97, 100, 122, 81, 113, 85, 123, 87]
106.25
97 % Remaining, Healthy Enviornment
[90, 120, 105, 82, 112, 123, 92, 98, 83, 101, 105, 82]
99.41666666666667
96 % Remaining, Healthy Enviornment
[106, 106, 109, 103, 113, 80, 83, 99, 96, 95, 118, 92]
100.0
95 % Remaining, Healthy Enviornment
[123, 82, 85, 101, 112, 107, 100, 98, 112, 101, 110, 98]
102.41666666666667
94 % Remaining, Healthy Enviornment
```

Figure 78: The 12 dBA values for Each Minute, The Average dBA Value, and The Percentage Remaining with User Feedback for Multiple Measurements

There was an issue averaging the dBA on a one-minute scale as averaging over a long period of time will cause the information to be lost. A high dBA noise, even for a short period of time, will have a large impact on the daily allowance. If the high noise level is combined with lower noise levels in the minute duration, it will minimize the effects. Linear averaging cannot be used due to the logarithmic properties of sound, (5 dBA increase has double the impact). For example: having 120 dBA for 30 seconds will be heavily weighted. However, combining 120 dBA with 60 dBA over the minute will average 90 dBA which will not have a large impact. To combat this, 1-second measurements (slow acquisition)were taken which will be able to keep the integrity of the data.

There is a possible issue with continuously running the algorithm's exponential equation after each dBA measurement, because it may deplete the available RAM on the microcontroller. A possible solution is to create a lookup table which will have all of the dBA and weighting values stored, using indexing the proper weighting values for each dBA can be received and subtracted from the allowance. This will allow us to make proper calculations without constantly rerunning the exponential allowance equation (algorithm). The lookup table was created in Python and then exported to Excel to be useful for Arduino devices (Fig. 79 & 80). Testing will be done to verify if processing power is actually saved.

```
dBA=np.round(np.arange(80, 140.1, .1),2)
Voltage=np.round(((10**(dBA/20))*0.00002),5)
Weighting=(1/(np.exp(-0.13862944*dBA)*np.exp(18.65043535)))
df=pd.DataFrame({'dBA': dBA, 'Voltage': Voltage,'Weighting':Weighting},dBA)
print(df)
```

```
         dBA    Voltage  Weighting
80.0    80.0    0.20000   0.000521
80.1    80.1    0.20232   0.000528
80.2    80.2    0.20466   0.000535
80.3    80.3    0.20703   0.000543
80.4    80.4    0.20943   0.000551
...      ...        ...        ...
139.6  139.6  190.99852   2.018257
139.7  139.7  193.21018   2.046431
139.8  139.8  195.44744   2.074998
139.9  139.9  197.71062   2.103964
140.0  140.0  200.00000   2.133335

[601 rows x 3 columns]
```

Figure 79: Partial Lookup Table Created in Python Using Equations for dBA and Weighting

	A	B	C	D
1		**dBA**	**Voltage**	**Weighting**
2	**80**	80	0.2	0.000521
3	**80.1**	80.1	0.20232	0.000528
4	**80.2**	80.2	0.20466	0.000535
5	**80.3**	80.3	0.20703	0.000543
6	**80.4**	80.4	0.20943	0.000551
7	**80.5**	80.5	0.21185	0.000558
8	**80.6**	80.6	0.2143	0.000566
9	**80.7**	80.7	0.21679	0.000574
10	**80.8**	80.8	0.2193	0.000582
11	**80.9**	80.9	0.22183	0.00059
12	**81**	81	0.2244	0.000598
13	**81.1**	81.1	0.227	0.000607
14	**81.2**	81.2	0.22963	0.000615
15	**81.3**	81.3	0.23229	0.000624
16	**81.4**	81.4	0.23498	0.000632
17	**81.5**	81.5	0.2377	0.000641
18	**81.6**	81.6	0.24045	0.00065
19	**81.7**	81.7	0.24324	0.000659
20	**81.8**	81.8	0.24605	0.000668
21	**81.9**	81.9	0.2489	0.000678

Figure 80: Partial Lookup Table in Excel Using Equations for dBA and Weighting

Arduino IDE Integration Manufacturing

The full working code was produced in MATLAB language that outputted the percentage over time. That code was translated into the Arduino language (C++) so that it can integrate with the microcontroller (Appendix).

Two methods were brought up to output the 'Weighting' of the dBA conversion. Processing power and output time had to be taken into consideration to choose the best methods. We are utilizing both the look-up table shown previously and an equation that was originally computed. For the code that uses the look-up table, the difficult part of the conversion is the indexing because the table is read differently on this platform. However, utilizing the equation would possibly use higher computing power. Both methods will be produced and tested to make sure we take the best route.

The two methods were tested, and they performed very similarly. We decided to implement the algorithm approach rather than the table method, as the code becomes much simpler to manipulate and debug.

Arduino IDE Integration Testing Methods

To make the Python/MATLAB code compatible with the Arduino IDE, it was converted to C++. Debugging the code posed a difficulty as Arduino IDE can only run if a microcontroller is present and the hardware group was testing them simultaneously. A simulation website was employed to accommodate the situation. WOKWI.com allowed us to run and debug the code without the hardware available. The Arduino IDE code worked well in the simulation (Fig. 81). To debug the code, the interval, voltage, decibel, weighting, allowance, and feedback were all printed to make sure each phase of the algorithm was working.

The allowance, OLED display, and SD card module was tested using the simulation (Fig. 82). Each part worked independently but there were certain issues when combined. The OLED and SD card modules were not able to run together as the Arduino UNO only had 2 kB of RAM. A much smaller font was used to allocate more RAM for the OLED and both modules now worked simultaneously.

This was not an issue when using the SEEED Xiao as the microcontroller had over 400 kB of onboard RAM. The SEEED Xiao needed the ESP32 library to function. This library was very large and slowed down compiling tremendously. The SEEED Xiao also needed to be rebooted each time new code was compiled, if not an error would appear. The SEEED Xiao also needed to be reset each time code was compiled. The device runs much more smoothly and has much larger capabilities after compiling code onto the SEEED Xiao rather than the Arduino UNO.

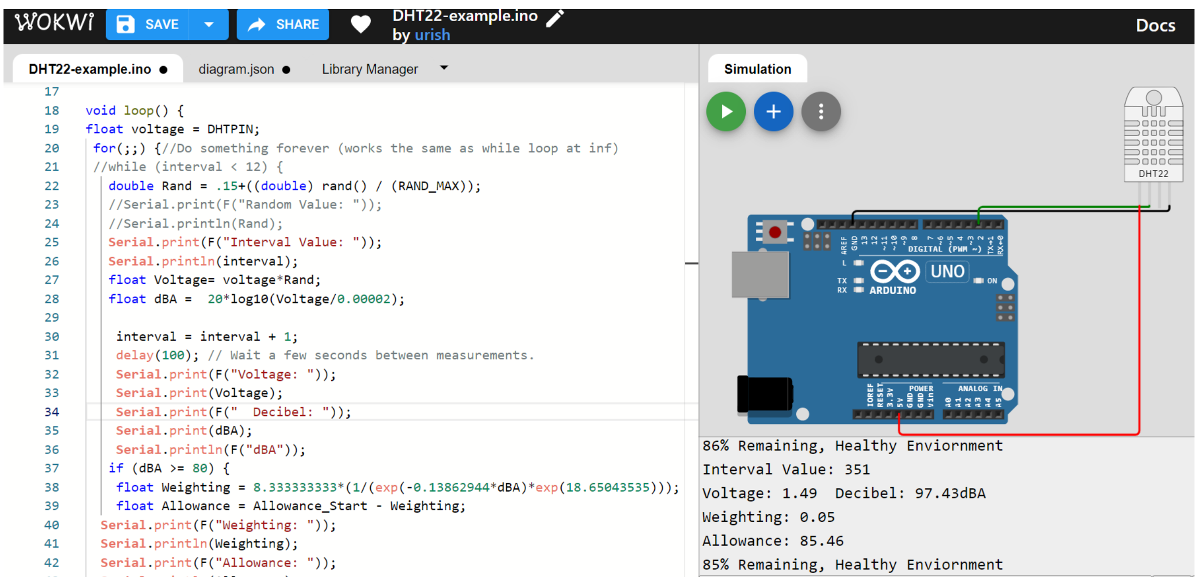


Figure 81: Arduino IDE Code Simulation on WOKWI

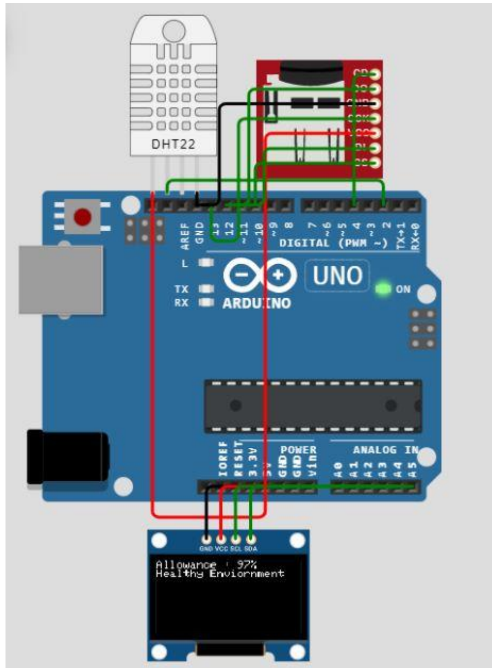


Figure 82: OLED and SD Module Simulation on WOKWI

Display Notifications

The OLED display was coded to present noise dose information clearly and continuously to the user. The OLED has 3 types of feedback; written, visual, and numerical representation. The written representation will be a notification message located at the top of the display and will print: “Healthy” above 60%, “Careful” below 60% and above 10%, “Limit” below 10% until 0%, and “Over Limit” after the total allowance was reached (Fig. 83a, 83b, 83c, 83d). The visual representation will be an allowance bar located under the notification message on the display. The bar will be full at 100%, the start of use, and will diminish according to noise exposure (Fig. 83b). The numerical representation will be the allowance percentage which is located at the bottom of the display. It will indicate 100% at the start of use and will decrease accordingly, continuously showing the available allowance percentage (Fig. 83c).

a) b) c) d)

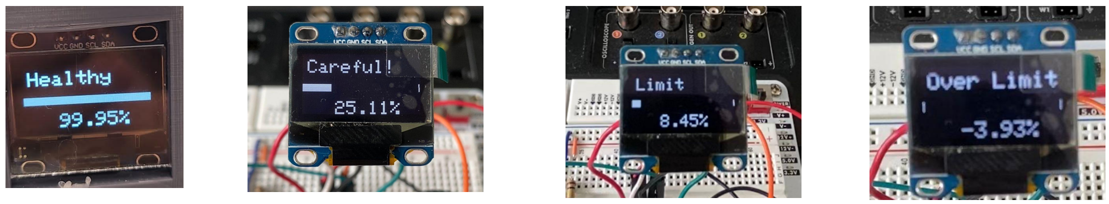


Figure 83. OLED Feedback types a) “Healthy” message with full allowance bar and high percentage b) “Careful” message with emptying allowance bar and low percentage c) “Limit” message with almost empty allowance bar and critically low percentage d) “Over Limit” message with empty allowance bar and negative percentage

Another form of feedback will be an LED that serves as an indication of dangerous instantaneous decibel levels. The LED will begin to indicate at a threshold of 80 dBA, and will initiate a blinking sequence at 100 dBA. While the OLED displays information that reflects the daily noise dose to the ear, the LED’s purpose is to inform the user of possible dangerous noises.

Data Saving

Daily data will be saved onto a SD card in the form of a .txt file (Fig. 84a). The time, dBA, and allowance will all be saved for analysis. In order to use the data it must be saved with a comma in between each value, this ensures that the file can be used as a .csv file which allows for easier conversion to a Pandas dataframe or an Excel spreadsheet (Fig. 84b & 84c). The ‘time’ will be the time from when the user turned on the device. The ‘dBA’ is the allowance data of the same day. The file created for data will be constantly appended to add data. Each reset of the device will have a message “Allowance was Reset” to separate trials and or days. It will be obvious as the allowance will jump back to 100% at each reset.

a)

Dr.Richter.txt — Edited

```
Allowance Was Reset
Time,dBA,Allowance
0:0:1,117.90,99.83
0:0:2,117.93,99.67
0:0:5,117.93,99.50
0:0:10,117.93,99.33
0:0:11,117.93,99.17
0:0:12,117.93,99.00
0:0:13,117.93,98.83
```

b)

	Time	Voltage	Decibel	Weighting	Allowance
0	0:0:0:52	1.11	86.97	0.01	99.99
1	0:0:1:111	1.04	86.45	0.01	99.98
2	0:0:2:172	1.09	86.85	0.01	99.97
3	0:0:3:234	1.05	86.56	0.01	99.96
4	0:0:4:296	1.06	86.58	0.01	99.95

c)

Time	Voltage	Decibel	Weighting	Allowance
0:0:0:52	1.11	86.97	0.01	99.99
0:0:1:111	1.04	86.45	0.01	99.98
0:0:2:172	1.09	86.85	0.01	99.97
0:0:3:234	1.05	86.56	0.01	99.96
0:0:4:296	1.06	86.58	0.01	99.95
0:0:5:358	1.11	87.02	0.01	99.93
0:0:6:420	1.11	87.02	0.01	99.92
0:0:7:483	1.05	86.56	0.01	99.91
0:0:8:545	1.08	86.79	0.01	99.90
0:0:9:607	1.08	86.77	0.01	99.89
0:0:10:669	1.07	86.69	0.01	99.88

Figure 84: Example Files generated (a) text file (b) Pandas Dataframe and (c) Excel Spreadsheet

Equipment needed

An Analog Discovery studio was needed for prototyping the circuitry (provided by the BME department). In order to manufacture and assemble the device, a Soldering kit was needed, in addition to training on soldering skills, both of which were provided by the Grove School of Engineering makerspace and its staff. In order to manufacture the housing, a 3D Printer (provided by the makerspace) is needed and the material (filament) as well. Furthermore, an SPL meter (provided by the Cardoso Lab) is needed to validate the final readings of the mic.

Future Directions

Hardware

- Further scale the dimensions and weight of the device.
- Add different color LEDs to differentiate the warning indicator (Red, Yellow, Green).
- Add a wifi module to transmit data without the need to remove the memory card.
- Increase battery capacity

Software

- Incoming sound will be dissociated into octave frequencies and weighted separately to calculate the overall decibel level.
- Microphone will be calibrated with an SPL meter with a broader frequency range.
- Timestamps will be implemented to give precise times of high exposure to provide better feedback to the user.
- Connect devices with headphones/earbuds to account for total daily noise exposure.
- Create a Graphic User Interface (GUI) for better user experience.

Detailed Budget

This is a detailed list of each component and cost (Tab. 11). The “Cost” column includes the price if the product was going to be made from scratch without access to materials. The “Actual Cost” only includes the price of components that we needed to pay for. If the price was zero, it was provided for us by the Makerspace or another Laboratory. The cost of some of the components is lower because we purchased in bulk, so we only included the price of the needed part. Also, some of the parts were provided to us by different labs so it resulted in an actual cost of $0.00.

Table 11. Detailed Budget		
Component	Cost	Actual Cost
Mems Microphone on a Breakout Board	$6.95	$6.96
Microcontroller (Seeed Xiao)	$5.40	$5.40
Capacitors with various values	$12.99	$0
Resistors with various values	$13.99	$0
Breadboard for prototyping	$13.99	$0
PCB for prototyping/assembly	$11.99	$1.00
Battery	$15.99	$15.99
Display	$10.99	$8.99
SD card module	$7.99	$7.99
SD card	$8.88	$8.88
Solder material	$8.99	$0
3D Printing Filament	$2.98	$0
Housing Mount	$6.99	$1.99
Total	$128.13	**$59.19**

Conclusion

We have created an affordable, wearable, easy-to-use, device to provide continuous feedback on a daily noise dose based on OSHA standards to adults (Table 12) (Fig. 85). This will combat peoples' unawareness of the potential harm they can suffer from extended exposure to loud noise. By knowing more about the noise they are exposed to on a daily basis, it will help them take the necessary steps to protect their hearing.

Thank you, have a good summer!

Table 12: Product Specifications Accomplished By DoseClip Device		
Frequency	20 - 20kHz	✓
Measuring (dB SPL)	60 - 115 dB	✓
Filter	A - Weighted	✓
Accuracy	1-2 dB SPL	✓
Amount of Data measurement	10 - 18 hours per day	✓
Memory Capacity	2 GB	✓
User Experience	Easily wearable and comfortable	✓
Processing	• Calculate equivalent noise level • Use of a threshold based on OSHA 29 CFR 1910.95 standard • Calculate remaining time exposure • Warning system	✓
Interface	Display the percentage available to the user	✓
Size and Weight	Around or less than 8 cm x 8 cm x 4 cm while weighing less than 200 grams	**5.6x8x3.75cm** **423 g**
Final Price	Under $250	**$59.19**

Figure 85: Final Prototype

APPENDIX

Final_DoseClip_Noise_Dosimeter.ino

```
using namespace std;
// Require Library for code to run. Download first before uploading code into microcontroller
#include <Wire.h>
#include <Adafruit_SSD1306.h>
#include <SD.h>
#include <iostream>
#include <algorithm>
#include <elapsedMillis.h>
#include <TimeLib.h>
// Objects //
elapsedMillis timeElapsed;
File myFile;
#define SCREEN_WIDTH 128  // OLED display width, in pixels
#define SCREEN_HEIGHT 64  // OLED display height, in pixels
#define BAUD_RATE 115200
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);
const int micPin    = A3;    // Collects the sound data and process sound
const int interval = 1000; // 1-sec Slow interval
float Allowance_Start = 100;
// used in RMS average.
int   Count=0;
float dBA=0;
float sum_square=0;
/* Definition */
// initialization of device
void initialization(int ADRR, int CS_pin){
  // Configuration of the display bar
 int width = 128;
 const int Color = 0xFFFF; // White in RGB565 format
```

```
const int x = 0;
const int y = 32;
const int height = 10; // Height of Bar
Serial.print("Initializing SD card...");
 if (!SD.begin(CS_pin))
  {
    Serial.println("initialization failed!");
    while (1); }

Serial.println("initialization Complete."); Serial.println(""); Serial.println("");
Serial.print("Initializing OLED Display...");
  if(!display.begin(SSD1306_SWITCHCAPVCC, ADRR))
    {
      Serial.println(F("SSD1306 allocation failed"));
      for(;;);
    }
Serial.println ("Initializing Complete."); Serial.println("");Serial.println("");
 /* Data logging Section */
SD.mkdir("/Noise_Dosimeter"); // Creates a folder to place the user information
if(SD.exists("/Noise_Dosimeter/Sound_User.txt"))
  {
    myFile = SD.open("/Noise_Dosimeter/Sound_User.txt", FILE_APPEND); // Appends data into file
      {
        myFile.printf("\n");
        myFile.printf("Allowance has been reset\n");
        myFile.printf("%-12s%-10s%-19s\n", "Time", "dBA", "Allowance");
        myFile.printf("%-12s%-10s%-19s\n", "----", "---", "---------");
        myFile.close();

        Serial.println("File is ready to append.");
        Serial.println("");
      }
  }
else
```

```
  {
    myFile = SD.open("/Noise_Dosimeter/Sound_User.txt", FILE_WRITE); // Creates an txt file for
collecting data.
    if (myFile)
     {

       myFile.printf("%-12s%-10s%-15s\n", "Time", "dBA", "Allowance");
       myFile.printf("%-12s%-10s%-15s\n", "----", "---", "---------");
       myFile.close();

       Serial.println("File has been created.");
       Serial.println("");
     }
  }
```

```
 display.clearDisplay();
 display.drawRect(x,y, (width - 1), height, Color);
 display.fillRect(x,y, (width - 1) * 1, height, Color); // width * fraction of the allowance
 display.setTextSize(2);
 display.setTextColor(SSD1306_WHITE);
 display.setCursor(5, 6);
 display.println("Healthy");
 display.setTextSize(2);
 display.setTextColor(SSD1306_WHITE);
 display.setCursor(30, 50); // x,y
 display.print(100.00);
 display.println("%");
 display.display();
}
// For the conversion of A/D to volts to dBA
float bits_2_Vsquare(float Bits){
 // float Bits is an input from analogRead()
 // bias_offset is from half the battery capacity.
 // Vcc = 3.3V so bias_offset is 1.65;
```

```
 float dev_resolution = 3.3/4096;
 float bias_offset    = 1.65;
 float Bitsv  = (Bits * dev_resolution) - bias_offset;
 float square = pow(Bitsv,2);
 return square;
}
float RMS_v(int &Count, float &sum_square){
 // Conversation AD reading into Root Mean Square (RMS)
 float RMS_v = sqrt(sum_square/Count);
 Serial.print("Summed volts: "); Serial.println(sum_square);
 Serial.print("Samples: ");      Serial.println(Count);
 Serial.print("RMS: ");          Serial.println(RMS_v);
 return RMS_v;
}
float RMSv_to_dBA(float RMSv){
    float x = 20*log10(RMSv/0.00002);
    float x_square = pow(x,2);
  if (RMSv > 0.316274 && RMSv < 0.354264) // 80 dBA - 90 dBA
      {
       dBA = (-3.1672*x_square)+(545.23*x)-23372 ; Serial.print("dBA: "); Serial.println(dBA);
      }
  else if (RMSv >= 0.354264) // 90 dBA and above
      {
       dBA =  (2.1571*x)-93.162; Serial.print("dBA: "); Serial.println(dBA);
      }
  else
      {
       dBA = 0; Serial.println("dBA was less than 80. Allowance not taken off"); // Anything below 80 is
not necessary and is assigned a dummy variable.
       Serial.println("");
      }
  return dBA;
}
}
```

```
// For configuration of the screen and allowance
float Allowance(float dBA, float Allowance_Start) {
 float factor = 1.6667; // 1-sec measurements ;
 float Weighting = factor*(1/(exp(-0.13862944*dBA)*exp(18.65043535)));  // 1-sec measurements
 float Allowance = Allowance_Start - Weighting;

 Serial.print("Weighting: ");  Serial.print(Weighting,5); Serial.print("; ");
 Serial.print("Allowance: ");  Serial.println(Allowance);
 Serial.println("");
 return Allowance;
}
void Allowance_Disp_Setting(float &allowance){
 int width = 128;
 const int Color = 0xFFFF; // White in RGB565 format
 const int x = 0;
 const int y = 32;
 const int height = 10; // Height of Bar
 float allowance_percent = (allowance/100);
 display.clearDisplay();
 display.drawRect(x,y, (width - 1), height, Color);
 display.fillRect(x,y, (width - 1) * allowance_percent, height, Color); // width * fraction of the allowance
 display.setTextSize(2);
 display.setTextColor(SSD1306_WHITE);
 display.setCursor(5, 6);
 //  Display text when a certain percentage has been depleted
 if (allowance > 60)
   {
    display.println("Healthy");
   }
 else if (allowance > 2)
   {
    display.println("Careful!");
   }
 else if (allowance > 0)
```

```
  {
    display.println("Limit");
  }
 else
  {
    display.println("Over Limit");
  }
  display.setTextSize(2);
 display.setTextColor(SSD1306_WHITE);
 display.setCursor(35, 50); // x,y
 display.print(allowance,2);
 display.println("%");
 display.display();
}
// For Data logging
void Data_Log(long int time,String Filename,float dBA, float Allow){
/* Time      = elapse time after device is one
   Filename  = Naming the file to append dBA and Allowance percentage.
 */
 Serial.print("Time:");
 // To convert elapse time into hour, minutes and seconds
 unsigned long hours =   (time /3600000UL) % 24;
 unsigned long minutes = (time /60000UL) % 60;
 unsigned long seconds = (time /1000UL) % 60;
 Serial.print(hours);   Serial.print(":");
 Serial.print(minutes);  Serial.print(":");
 Serial.println(seconds);
 myFile = SD.open(Filename, FILE_APPEND);
  if (myFile)
   {
    myFile.printf("%02lu:%02lu:%02lu    %-10.2f%-10.2f\n", hours, minutes, seconds, dBA, Allow);
//Format of the time in H:M:S
   }
}
```

```
// Main Portion of the Code
void setup()
{
 Serial.begin(BAUD_RATE);
 initialization(0x3C,4);
  pinMode(D7, OUTPUT);    // Turning on the LED
 digitalWrite(D7,LOW);  // WHen the device is on, the LED is initially off
}
void loop(){
 long int time = millis();  // elapsed time
  float square =  bits_2_Vsquare(analogRead(micPin));
 sum_square  += square; Count++;   // used in calculating RMS
 if (timeElapsed > interval)
   {

    float RMSv =  RMS_v(Count,sum_square); // Returns RMS voltage
    float dBA  =  RMSv_to_dBA(RMSv);       // Converts from RMS volts to dBA based on calibration
    if (dBA >= 80) { // The 80 dBA threshold is from OSHA standard requirement

       digitalWrite(D7,HIGH); // LED is turn on
       float Allow = Allowance(dBA, Allowance_Start);  // This is the portion of the code where the
allowance gets depleted
       Allowance_Disp_Setting(Allow);  // Display is updated

       Allowance_Start = Allow;  // Updates the allowance percentage

      if ( dBA >= 100){
       // Code to blink the LED once the dBA reading is at 100 and above.
       digitalWrite(D7,HIGH);
       digitalWrite(D7,LOW);
       digitalWrite(D7,HIGH);
       digitalWrite(D7,LOW);
```

```
    }
    Data_Log(time,"/Noise_Dosimeter/Sound_User.txt",dBA, Allow); // where the data gets append
   }
   else {
     // When the reading is less than 80 then the LED is turned off.
     // LED should only be on when the reading is at 80 and above
     digitalWrite(D7,LOW);       }
```

```
   // Reset parameter for next iteration used in calculating data.
   square=0;
   sum_square=0;
   timeElapsed = 0;
   Count=0;
  }
}
```

References

1. Oghalai J.S., & Brownell W.E. (2012). Chapter 44. anatomy & physiology of the ear. Lalwani A.K.(Ed.), CURRENT Diagnosis & Treatment in Otolaryngology—Head & Neck Surgery, 3e. McGraw Hill. https://accessmedicine.mhmedical.com/content.aspx?bookid=386&sectionid=39944085

2. Alberti, P. W. (2001). The anatomy and physiology of the ear and hearing. Occupational exposure to noise: Evaluation, prevention, and control, 53-62.

3. The Four Types of Hearing Loss | CCHAT Sacramento. (n.d.). Retrieved October 17, 2022, from https://www.cchatsacramento.org/blog-and-events/the-four-types-of-hearing-loss

4. Degree of Hearing Loss. (n.d.). American Speech-Language-Hearing Association; American Speech-Language-Hearing Association. Retrieved October 17, 2022, from https://www.asha.org/public/hearing/degree-of-hearing-loss

5. Purves D, Augustine GJ, Fitzpatrick D, et al., editors. Neuroscience. 2nd edition. Sunderland (MA): Sinauer Associates; 2001. The Audible Spectrum. Available from: https://www.ncbi.nlm.nih.gov/books/NBK10924/

6. *What is A-weighting?* Siemens DISW. (2019, August 29). Retrieved December 9, 2022, from https://community.sw.siemens.com/s/article/what-is-a-weighting

7. John Hopkins. (2021). Hearing loss and the Dementia Connection. Retrieved October 17, 2022 from https://publichealth.jhu.edu/2021/hearing-loss-and-the-dementia-connection

8. WHO.(2021). Deafness and hearing loss. Retrieved October 17, 2022 from https://www.who.int/news-room/fact-sheets/detail/deafness-and-hearing-loss

9. Kardous, C. (n.d.). *Understanding noise exposure limits: Occupational vs. General Environmental Noise.* Centers for Disease Control and Prevention. Retrieved December 9, 2022, from https://blogs.cdc.gov/niosh-science-blog/2016/02/08/noise/#:~:text=The%20EPA%20document%20also%20specified,dBA%20for%20indoor%20activities

10. *State prohibited occupations for minors.* Department of Labor. (n.d.). Retrieved December 9, 2022, from https://dol.ny.gov/state-prohibited-occupations-minors#:~:text=Employment%20After%20Childbirth-,Under%2018%20Years%20of%20Age,structure%20from%20an%20elevated%20surface

11. Roberts, B., & Neitzel, R. L. (1970, January 1). *Noise exposure limit for children in recreational settings: Review of available evidence*. Scitation. Retrieved December 9, 2022, from https://asa.scitation.org/doi/10.1121/1.5132540

12. OSHA. (n.d.). Occupational Noise Exposure. Occupational Noise Exposure - Standards | Occupational Safety and Health Administration. Retrieved October 17, 2022, from https://www.osha.gov/noise/standards

13. Statista.(2022). Awake time per day in the United Kingdom (UK) in 2014, by age and gender. Retrieved October 17 from https://www.statista.com/statistics/326907/awake-time-per-day-uk-by-age-and-gender/

14. Power Electronics Corp.(2019). Design Consideration for Maximum Allowable Temperature. Retrieved October 17 from https://slpower.com/data/collateral/AN_Maximum_Allowable_Temperature.pd

15. Mouyal, Natalie.(2021). New standards for wearable technologies. Retrieved October 17 from https://etech.iec.ch/issue/2021-04/new-standards-for-wearable-technologies

16. Byrne, D., & Reeves, E. (2008, January 24). *Analysis of Nonstandard Noise Dosimeter Microphone Positions*. Taylor & Francis Online. Retrieved December 9, 2022, from https://www.tandfonline.com/doi/abs/10.1080/07488008808408783

17. dBadge2 Personal Noise Exposure Meter. DBADGE2. (n.d.). Retrieved October 17, 2022, from https://www.casellasolutions.com/us/en/support/product-support/dBadge2.html

18. Dosebadge noise dosimeter datasheet - ose.directory. (n.d.). Retrieved October 18, 2022, from https://ose.directory/wp-content/uploads/2018/08/dosebadge-noise-dosimeter.pdf

19. *Minut: Protect your home, rental property, airbnb and short-term rental.* (n.d.). Retrieved October 17, 2022, from https://www.minut.com/

20. The ultra low power dosimeter's, US 7836771 B2, https://patents.google.com/patent/US7836771B2/en?oq=US+7836771+B2

21. “A Method and system for noise dosimeter with quick-check mode and earphone adapter”,US 8322222B2, https://patents.google.com/patent/US8322222B2/en?oq=US+8322222B2+

22. “Integrated wearable noise dosimeter”, US10408671B2, https://patents.google.com/patent/US10408671B2/en?oq=US10408671B2

23. In Ear Noise Dosimeter, US 20170289670 A1, https://patents.google.com/patent/US20170289670A1/en?oq=US+20170289670+A1

24. Use the noise app on your Apple Watch. Apple Support. (2021, December 3). Retrieved October 17, 2022, https://support.apple.com/en-us/HT209593#:~:text=The%20weekly%20limit%20at%20this,can%20cause%20temporary%20hearing%20loss

25. “User interfaces for managing audio exposure”, US 11223899 B2, https://patents.google.com/patent/US11223899B2/en?oq=US+11223899+B2

26. “User interfaces for managing audio exposure”, US 11228835 B2, https://patents.google.com/patent/US11228835B2/en?oq=US+11228835+B2

27. “User Interfaces For Monitoring Noise Exposure Levels”, US 10764700 B1, https://patents.google.com/patent/US10764700B1/en?oq=US+10764700+B1

28. “Noise Index Detection System And Corresponding Methods And Systems”, US 9877128 B2, https://patents.google.com/patent/US9877128B2/en?oq=US+9877128+B2

29. “Noise exposure computer and method”, US 3747703 A, https://patents.google.com/patent/US3747703

30. Image of SPM1423HM4H-B microphone’s frequency response, (n.d.). Retrieved December 11, 2022, from https://m5stack.oss-cn-shenzhen.aliyuncs.com/resource/docs/datasheet/core/SPM1423HM4H-B_datasheet_en.pdf

31. Image of CMB-6544PF microphone’s frequency response, (n.d.). Retrieved December 11, 2022, from https://www.google.com/url?q=https://www.cuidevices.com/product/resource/cmb-6544pf.pdf&sa=D&source=docs&ust=1670793504844596&usg=AOvVaw0D2ob9QPH-ep_E-2StxYqD

32. Image of analog A-weighting filter circuit, (n.d.). Retrieved December 11, 2022, from https://sound-au.com/project17.htm
33. Image of SPH1878LR5H-C microphone's frequency response, (n.d.). Retrieved December 11, 2022, from https://www.google.com/url?q=https://media.digikey.com/pdf/Data%2520Sheets/Knowles%2520Acoustics%2520PDFs/SPH1878LR5H-C_DS.pdf&sa=D&source=docs&ust=1670793504840687&usg=AOvVaw1WIzqdMX424P2UC7zxU_jk
34. Image of 2nd generation AirPods Pro, (n.d.). Retrieved December 11, 2022, from https://www.googleadservices.com/pagead/aclk?sa=L&ai=DChcSEwj956nD5eP7AhUY-MgKHVoXCnoYABAkGgJxdQ&ohost=www.google.com&cid=CAESa-D2JVEsvx6XW0WI9tmGPO1AywCddIbPjJzOpz87-DpJqaKRXVdmTygI5rYCG5ook7fLXlLMndvf3X7ygAQrehXgjIMWEuUnl5T27QY5yF0O0V2bZE0m1PfEYEaOacPuTCYW6oDJtbAB3Cx4&sig=AOD64_1O5u9s7ofPPqt6GhbwNWGhOKgAiQ&ctype=5&q=&ved=2ahUKEwjh4p7D5eP7AhX3K1kFHQACDBUQ9aACKAB6BQgGEPkB&adurl=
35. Image of smartwatch (n.d.). Retrieved December 11, 2022, from https://www.amazon.com/Fitness-Swimming-Waterproof-Touchscreen-Smartwatch/dp/B0BC3Y5LDP/ref=sr_1_1_sspa?gclid=Cj0KCQiAj4ecBhD3ARIsAM4Q_jGDt77iksc_KnIqjn8_BgD0_-iILFgXH44qhFXZkYYabNo-jjMrqksaAhRSEALw_wcB&hvadid=521274106292&hvdev=c&hvlocphy=9004287&hvnetw=g&hvqmt=b&hvrand=10256053877353955699&hvtargid=kwd-391217578314&hydadcr=19138_9664171&keywords=smart%2Bwatch%2Bwith%2Bmicrophone&qid=1669519536&sr=8-1-spons&spLa=ZW5jcnlwdGVkUXVhbGlmaWVyPUEzVENaVVBJUTRDTlNXJmVuY3J5cHRlZElkPUEwNjEyNTQ5MVI3SEVUVVFYSVNZNSZlbmNyeXB0ZWRBZElkPUEwMzA2MzEwM1ZRV05UVUlTWEZFNyZ3aWRnZXROYW1lPXNwX2F0ZiZhY3Rpb249Y2xpY2tSZWRpcmVjdCZkb05vdExvZ0NsaWNrPXRydWU&th=1
36. Image of microphone schematic (n.d.). Retrieved December 11, 2022, from https://www.st.com/resource/en/application_note/an5027-interfacing-pdm-digital-microphones-using-stm32-mcus-and-mpus-stmicroelectronics.pdf
37. Image of Seeed Xiao with speech recognition (n.d.). Retrieved December 11, 2022, from https://www.google.com/url?sa=i&url=https%3A%2F%2Fwiki.seeedstudio.com%2FXIAO-BLE-Sense-TFLite-Mic%2F&psig=AOvVaw3QQKT3J-MNN2rGPopi_PzR&ust=1670381891642000&source=images&cd=vfe&ved=0CA8QjRxqFwoTCMC1isz_4_sCFQAAAAAdAAAAABAE

38. Image of pdm microphone attached to microcontroller (n.d.). Retrieved December 11, 2022, from https://learn.adafruit.com/adafruit-pdm-microphone-breakout?view=all
39. Image of M5AtomU microcontroller (n.d.). Retrieved December 11, 2022, from https://shop.m5stack.com/products/atomu-esp32-development-kit-with-usb-a
40. Image of sample test tube dye process (n.d.). Retrieved December 11, 2022, from https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.stevespanglerscience.com%2Flab%2Fexperiments%2Fbaby-soda-bottle%2F&psig=AOvVaw2fZ1PFnw1NDDzSr6JweU7m&ust=1670382959662000&source=images&cd=vfe&ved=0CA8QjRxqFwoTCIiZxMmD5PsCFQAAAAAdAAAAABAE
41. Image of modified headphones, (n.d.). Retrieved December 11, 2022, from https://www.amazon.com/Portable-Bladeless-Operated-Rechargeable-Headphone/dp/B08911JGGW