Hearing Damage Detector and Alarm System

Alex Yuan (ayuan20)
Jinzhi Shen (jinzhis2)
Jake Fava (jfava2)

Design Document
TA: Hojoon Ryu

Fall 2022

Table of Contents

1 Introduction

1.1 Problem

Middle and high school musicians can be subjected to harmful levels of noise on a daily basis between rehearsals, practice sessions, and performances. Cheap and effective hearing protection is available, but many students neglect using it until they start noticing the effects of their hearing damage years later. Even without considering hearing loss, long-term hearing damage can disturb the normal “balance between excitation and inhibition in the central auditory system” that can last several times longer than the time required to cause the damage [1].

1.2 Solution

Our solution is a device that provides live feedback to musicians about their noise exposure in an attempt to encourage more regular use of existing hearing protection equipment.

Our solution provides feedback in three ways:

1. Instantaneous sound pressure level (SPL) readings, appropriately weighted to match the human hearing curve (dBA), fed to the user with a simple set of LEDs corresponding to safe (green), potentially dangerous (yellow), and dangerous (red).
2. Sound exposure levels over the course of a day or practice session, fed to the user through a flashing indicator LED when dangerous threshold levels are breached.
3. An optional but highly recommended detailed sound report from a given section indicating the overall integrated and average SPL and the peak and average exposure level for a given frequency in the human hearing curve.

Our sound processing subsystem will consist of a digital microphone and a microcontroller. The digital microphone will convert external sound waves into usable digital signals for the microcontroller to process. Our microcontroller will be able to distinguish frequencies, especially those pertinent to humans, as well as the sound pressure level (SPL) at a given moment, or instantaneous SPL, and over a certain period, integrated SPL. The microcontroller will take this information and use it to drive the LEDs as well as send a detailed sound data file to a local computer to process into an accessible sound report. Last, all of the on-board components will be powered by a power subsystem composed of a battery charger taking power from a USB port, a 3.7 V lithium-ion battery, and a 3 V fixed linear voltage regulator. Thus, our system will be powered by a lithium-ion battery, able to read relevant sound from the environment, convey both instantaneous and daily exposure danger through LEDs without the need for a local computer, and be able to give a detailed report through USB connection and a software program from a local computer to aid musicians in combating dangerous sound.

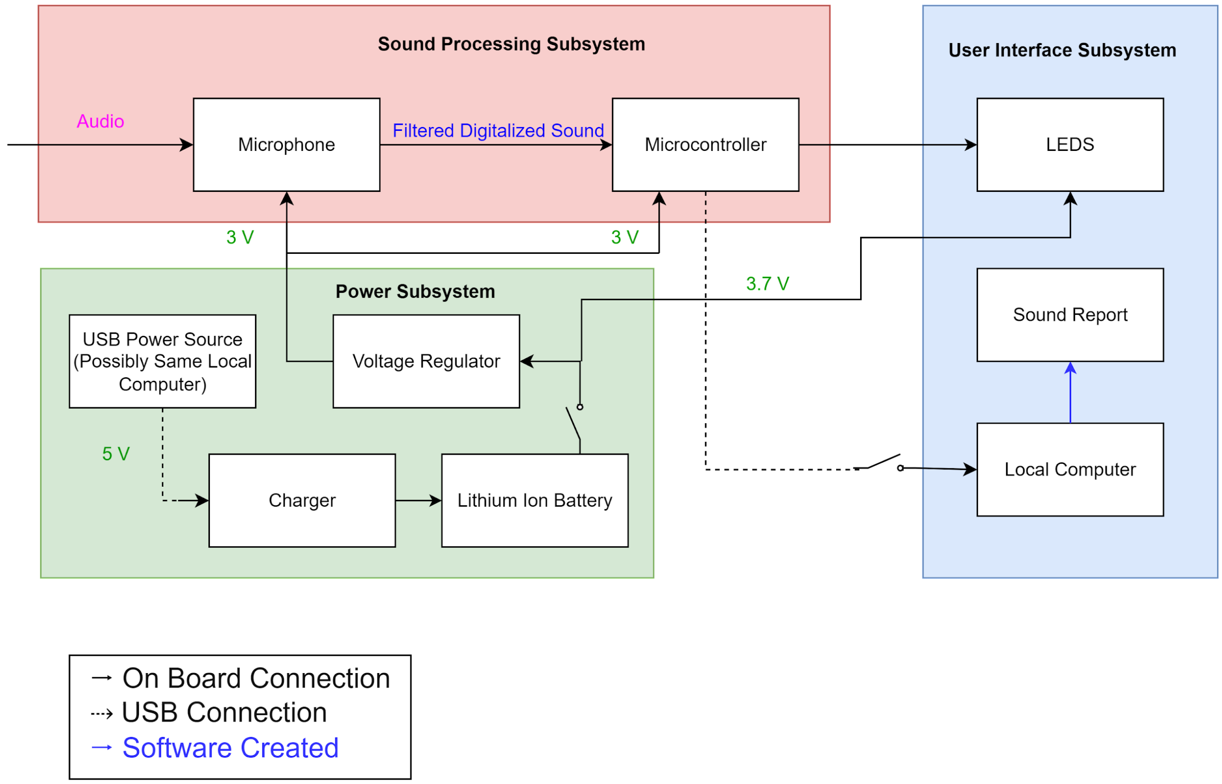
1.3 Visual Aid

1.4 High-Level Requirements

1. **Battery Life:** The device needs to last at least 8 hours from a full charge.
2. **Sound Processing:** The sound processing subsystem needs to convert external sound into usable output signals comparable, i.e., within a 1 dB tolerance for all frequencies within 20 Hz-20kHz, to an industry grade sound dosimeter or equivalent.
3. **LED User Feedback:** The device needs to be able to display instantaneous and integrated SPL data in the form of lit-up LEDs. These LEDs should accurately convey instantaneous SPL within reasonable tolerance range and integrated SPL exposure as given by the requirements and verification section below.
4. **Sound Report User Feedback:** The device needs to be able to upload recorded SPL data to a computer to perform the integration and generate a report. This report should include frequencies primarily that the human ear can listen to as well as the instantaneous and integrated SPL over the period the device was active. The report should be consistent with raw oscilloscope readings within a ± 1 dB tolerance.

2 Design

2.1 Block Diagram



2.2 Subsystem Overview

2.2.1 Subsystem 1 — Power Subsystem

This subsystem should above all else safely draw power from a standard 5 V USB source and safely provide power to the rest of the circuit. Additionally, it should provide portability to allow the user to use the alarm system, albeit without the sound report, if a nearby computer is unavailable. We will describe how we choose our design to accomplish not only this goal but additional goals as well.

Our power subsystem is responsible for safely powering the entire on-board integrated circuit. The power unit consists of a charger, the MAX1736EUT42+T; a 3.7 V output battery, the ASR00007; a simple slide switch to control whether the circuit is on or off; and a 3 V fixed linear voltage regulator.

First, we choose the MAX1736EUT42+T as it is fairly inexpensive at $4.16; simple to use; is a single-cell lithium ion battery charger that matches with our battery; has safety features including safely charging ner-dead cells and input-supply detection that ; small dimensions at a width of

1.63 mm; and allows us to charge the battery with a standard 5 V from a USB connection as it accepts 2.7 V to 22 V.

Next, we choose the ASR00007 lithium-ion battery because its output, 3.7 V, is fairly close to the voltage regulators fixed output voltage and thus is more efficient than batteries with a higher output voltage; is relatively cheap at $5.95; small dimensions at 25.0 mm x 23.0 mm x 5.5 mm; has a good capacity at 290 mAh; and has basic protection against overcharging or discharging.

The presence of this battery and the battery charger in lieu of directly connecting a USB power supply to the circuit means that the LED alarm system, i.e., the system with only the LEDs as user feedback, can be used without a nearby computer. We believe that this option will allow users who may not, in a given circumstance, have immediate access to a local computer due to either lack of access or some other cause to still have access to LED warnings so long as the alarm system is charged. We note that we still recommend using the alarm system with a local computer to provide both power and the sound report, but we stress that due to the portability afforded by the charger and battery, the alarm system will still work in a limited fashion to provide basic sound exposure warnings via the LEDs.

We then put a switch in between the battery and voltage divider to ensure that the circuit does not run without user input. We use the MS12ASG13 slide switch as it has a fairly simple design, it either connects pins 2 and 1 or 2 and 3; high voltage rating compared to our battery, 28 V vs. our battery's typical output of 3.7 V; fairly low price at $5.48; and small dimensions at 3.8 mm wide.

Finally, we chose a linear voltage regulator, the TLV70230DBVR, due to certain key characteristics. It has a fixed voltage of 3V, perfect for our digital microphone, microcontroller, and LEDs; should be easy to work with as a linear voltage regulator as opposed to a switching regulator; has relatively small dimensions at 2.90 mm × 1.60 mm in the worst case; good capabilities for battery powered handheld applications; is fairly low cost at $0.61, and is not especially inefficient as its input voltage from the battery should typically be around 3.7 V, which is fairly close to the fixed voltage.

Thus, we designed our power subsystem mainly with considerations to cost, safety, portability, and efficiency. As such, the power subsystem should provide a safe, easy to use power unit that the rest of the circuit can rely on. Additionally, the relatively low cost and low size of the system should make the entire project more accessible.

Requirements	**Verification**
1. Power subsystem must last for at least 8 hours on a full charge.	1. Charge the battery until we see with an oscilloscope an output

	of around 4.2 V ± 5%, i.e., until we see close to the voltage that should be seen at full charge. 2. Take the output of the regulator and attach it to one end of a 100 ohm resistor whose other end is ground. 3. Measure the output of the battery with an oscilloscope and verify that it does not output 3 V ± 5%, i.e., we should not see close to the voltage that should be seen at fully discharged. 4. Repeat procedure at least twice for 50 and 200 ohm resistors.

2.2.2 Subsystem 2 — Sound Processing Subsystem

This subsystem will take in outside sound data, process it through the digital microphone IC and the microcontroller, and convert it into useful outputs for the LEDs and local computer to use. This subsystem is essential for our goal of accurately and quickly providing information about the environmental noise and any possible health concerns.

To capture the sound pressure levels, we need a microphone that is omni-directional; responsive to the frequencies that the human ear is responsive to, about 20 Hz-20 kHz; and has a suitably high signal-to-noise ratio, around 60 dB or above. We would also prefer a digital microphone that includes a preamp and other features to eliminate possible points of failure that would come with using a distinct microphone, preamp, ADC, etc. system. One digital microphone that fits these criteria is the DMM-4026-B-I2S-R.

The DMM-4026-B-I2S-R has a high signal to noise ratio, 64 dB; small dimensions at 4 mm x 3 mm x 1 mm; the prerequisite 20 Hz to 20 kHz frequency range, an omnidirectional pickup pattern; compatibility with our microcontroller using one of the microcontroller’s SPI/I2S ports; and is fairly inexpensive at $2.42. Additionally, the DMM-4026-B-I2S-R has I2S, which is purely digital and thus does not need encoding or decoding [2]. This convenience means that we do not have to worry about our microcontroller having to preprocess the signal before it is able to use it.

One the other hand, our microcontroller will need to be able to take input data from the digital microphone and turn it into useful information for the user in order to indicate possible sound hazards. To do this, our microcontroller must be suitably fast and powerful memory and frequency wise. Additionally, there are two types of feedback we would like to be able to provide: instantaneous SPL readings in dBA and integrated SPL over time, also called "sound exposure", to gauge potential hearing damage accumulated over a session. A potential MCU to use for our device is the STM32F103C8T6TR.

The STM32F103C8T6TR features Arm® 32-bit Cortex®-M3 CPU core, 72 MHz maximum frequency, worst case 64 kB of flash memory, 20 kB of SRAM, and up to 2 SPI ports, which all will allow it to read the digital microphone output [3]. While the source illustrates a PDM microphone to the STM32, we have found other sources showing that I2S microphones work just as well, with one source even giving a tutorial on I2S connection to a STM32 [4]. The microcontroller also comes with a USB 2.0 full speed interface that will allow it to effectively communicate with a local computer. Additionally, the STM32F103C8T6TR is fairly cheap at only $7.64 and fairly small at worst case 14 mm x 14 mm.

Thus, we designed our sound processing subsystem as such to be fairly simple, with errors only coming from the digital microphone or microcontroller, but also sophisticated enough to capture and transmit all the data we require namely: the 20 Hz-20 kHz frequency range, an above 60 dB signal-to-noise ratio, communication between the digital sensor and MCU, an MCU able to process the signals, and the ability to send the processed data to LEDs and a local computer. Additionally, we chose relatively inexpensive and portable components that serve to make our project more accessible.

Requirements	**Verification**
1. Instantaneous and SPL readings are correct within a ±1 dB margin of error.	1. Feed curated sound data to the sound processing subsystem and to an commercially available SPL meter. 2. Either directly measure the output of the sound processing subsystem using an oscilloscope or through data files on a local computer if the USB connection is functional. 3. Compare outputs, verify that the sound processing subsystem does not vary more than ±1 dB from the SPL meter.

2.2.3 Subsystem 3 — User Interface

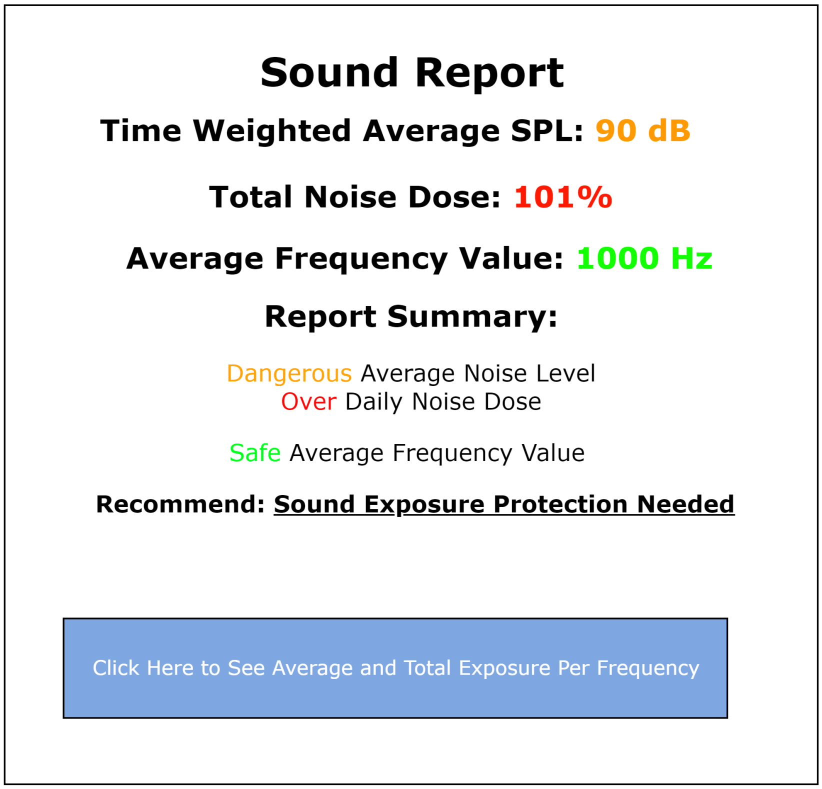
To present live feedback on instantaneous SPL, the device will feature a series of LEDs that light up or flash in response to recorded dBA, i.e., A-weighted decibels. For sake of convenience, when we refer to dB in the context of indication or measurement below including in the R/V table, we are talking about A weighted dB. Three LEDs should range from green for safe sound levels up to red for potentially dangerous sound levels. Statically, for instantaneous exposure, green should indicate from 0 dB to 90 dB, the limit of safe exposure according to OSHA [5]. Yellow should indicate from 90 dB to 105 dB, or the maximum noise level OSHA limits for a 1 hour period. Last, red should denote anything above 105 dB, when 1 hour of exposure is too much. A fourth LED distinct from the other three will flash after the daily exposure limit is reached.

We use and define our LEDs to be easily discernible at glance and relevant to our problem, i.e., to musicians. Since 1 hour intervals are fairly common for structures like middle and high school bands, we would like to warn if the exposure for a 1 hour period is too high.

One the topic of LEDs, we choose to use HV-5RGB60 LEDs as they are relatively cheap at $1.02; support the full color range to aid in differentiating the LEDs; have appropriate voltages, 2V Red, 3.4V Green, 3.4V Blue, for our battery; and are fairly small at a maximum height, its largest dimension, of 8.90mm.

Last, we use the same slide switch, the MS12ASG13 for the same reasons as above, except instead of controlling whether the device is on or off in the power subsystem, here, the switch will control whether the output of the sound process goes to the USB port or to a high impedance load in the case that the user does not want or is not able to connect to a local computer via USB.

To present a report of sound exposure over the course of a session, we will plan to allow the user to pull the data from the device onto a computer, e.g., the computers that would be located in a practice room or classroom, and perform a simple sound report. This report will be optional, as explained in the power subsystem section, but highly recommended due to the detail that the report has in comparison to the simple LED feedback. This report will pull the current frequency and SPL values from a given period from the device and detail the average SPL and overall sound exposure per frequency and what hearing damage the overall session's sound exposure/the cumulative sound exposure has the potential to cause.



Example Sound Report Main Page

The report could take arbitrarily many frequency values, but if need be, we could keep track of only around 100 frequency values from 20 Hz-20 kHz on a log scale and round other values on a log scale to the nearest frequency value without significant loss of information. For example, if we used 101 frequency values using the frequencies values $[20(1000)^{i/100}] \text{ for i in range}(0{,}101)$, we would only have a maximum frequency difference of $1000^{0.5/100}$, or around 3.5% for a value exactly geometrically between two frequencies. If we used 201 frequency values instead, we would only have a maximum frequency difference of $1000^{0.5/200}$ or around 1.7%. We reason that such a small tolerance of at most 3.5%, but realistically around 1.7% or below as long as the software keeps track of 201 or more frequencies logarithmically spaced from 20 Hz-20 kHz, will be good enough for the average user considering our other tolerances of around 10%.

If need be, this report can reduce the load on the microcontroller by allowing it to only store daily sound exposure/integrated SPL, the necessary values to capture daily sound exposure, and current peak SPL instead of storing such values for each frequency. The specific frequency and SPL values would then come from the output of the microcontroller from a given sampling period, somewhere around 1s, that would capture the peak SPL value and its associated frequency. The microcontroller would then output the frequency and SPL value to the USB port on a regular interval, likely between 0.1s to 1s, then overwrite the frequency and SPL value with a new sampled value from the digital microphone. This process would repeat until the battery of the device was depleted or the user turned off the device. While we would prefer to store the frequency and SPL values in the microcontroller itself, we stress that we have the capability to lower the processing load on the microcontroller if we must.

Requirements	**Verification**
1. Sound processing unit can accurately process sound values into frequency and A-weighted dB within a ± 1 dB and 5% frequency range.	1. Play a 60 dB recording of an audio file predominantly made of 20 Hz signals. 2. Record the values using a commercially available sound level meter and the sound processing unit. 3. Verify that the sound processing unit is able to convey the correct dBA and frequency value within a ± 1 dB and 5% frequency range. 4. Repeat for files made of predominantly 200 Hz, 2 kHz, and 20 kHz.
1. LED readings accurately convey instantaneous SPL within a ± 1 dB range.	1. Input 5 dB values via a simulated input for the green range: 88 dB, 89 dB, 90 dB, 91 dB, and 92 dB. 2. Verify that at most, 89-91 dB readings show up as green. 3. Repeat procedure for yellow, i.e., 103-107 dB with 1 dB intervals for yellow and verify 106 dB as the largest dB value accepted for yellow. 4. For red, repeat the procedure starting from 105-107 dB with 1 dB intervals. Verify that the lowest accepted red value is 106 dB.
1. LED readings accurately convey integrated SPL within 10% tolerance.	1. Using OSHA's standard for noise regulation, we know that an increase of 5 dB corresponds to half the time of exposure. 2. Send mock dB input values of 105 dB, 110 dB, and 115 dB to the interface. 3. Verify that LED flashing occurs between 54-60 minutes for 105 dB, 27-30 minutes for 110 dB, and 13.5-15 minutes for 115 dB.
1. LED readings switch between a	1. Use mock sound intensity data to send

certain value to another within 2s.	a valid green dB reading and yellow dB reading using the prior defined ranges. 2. Verify that once the dB intensity changes ranges, the LEDs switch within 2s. 3. Repeat the procedure with yellow to red, green to red, red to yellow, red to green, and yellow to green.
1. The sound report software needs to be correct within 1 dB tolerance intensity wise (dB) for average sound level and within 10% for sound exposure.	1. Send mock sound data simulating background noise in the human frequency range, a majority of sound between 50-60 dB, to the software. 2. Verify that the software conveys the correct overall average sound level and sound exposure within 1 dB and 10% tolerance respectively by checking against the raw data. 3. Repeat for 80-90 dB and 100-110 dB respectively.

All dB values are A-weighted.

2.3 Tolerance Analysis

The most critical part of our project is the sound processing subsystem, as it conveys the entire purpose behind our project. If the subsystem causes the LEDs to erroneously go off too much when there's no danger, then our project will rarely be used. However, if the sound processing unit causes the LEDs to go off too little when there is danger, then our project will not have fulfilled its purpose: it has not warned musicians when they are unsafe.

However, we believe that our sound processing unit will still be feasible. First, our digital microphone is more than capable of capturing the correct frequencies and sound levels, as it has a 20 Hz-20 kHz range, a high acoustic overload point at 120 dB, and 69 dB signal-to-noise ratio. Thus, we believe that unless there is a manufacturer error, problems with calculating SPL will not come from the microphone. Also, calculating the dB values from the I2S signal should be trivial given the microcontroller's built in log function [6]. Given the abundance of commercially available instantaneous exposure devices as well as the simplicity of calculating instantaneous SPL, i.e., storing and sending the maximum SPL in a certain interval, likely 5s for us to prevent too much flashing, the static warning should be more than effective. Sending the sound report data to the local computer should also be as simple as simply passing on the digital microphone's input from the SPI port to the USB port. Thus, the only difficulty is then showing the industry grade integrated SPL, i.e., LEP'd/Time Weighted Average (TWA) via the flashing. In this case, we are considering the scenario that all frequency processing is done by the local computer.

The LEP'd equation is given by [7]:

$$LEP'd\ or\ L_{EX}, 8h = L_{eq} + 10 \times log_{10}\left[\frac{T_2 - T_1}{T_n}\right]\ \mathrm{dB}$$

- *Leq* = frequency weighted (A or C), equivalent-continuous sound pressure level in dB
- T_n = normalization period on criterion duration (8 hours by standard)
- $T_2 - T_1$= measurement period or Run Time

Here, we would be measuring A weighted frequency. We note that we can measure dBA by converting from dB using an A-weighting table that we could easily store on the MCU [8]. We note that the above equation measures weighted SPL, and to find daily exposure, we would use the argument, Percent of Daily Dose = D = 100(C1 / T1 + C2 / T2 + Cn / Tn), where Cn is defined as total exposure time spent in a given noise level and Tn is the 8 hour limit reference time in a given noise level [9]. Also, we know that as our microcontroller comes with 3 16-bit internal timers, keeping track of time will be as trivial as incrementing a register or address in time with a timer. We note that since we allow ourselves a 1 dB tolerance, we can round any dB values not at an integer value to the closest integer. That being said, we can calculate the maximum possible error at a given sound level can be computed by $T = \frac{8}{2^{(L-90)/5}}$, or the 8 hour reference time for a given A-weighted sound level. Calculating $T_{err}/T = \frac{2^{(L-90/5)}}{2^{(L-90+0.5)/5}} = 2^{(-90+89.5)/5} = 2^{-0.1} = 0.933032991537$, where T_{err} has an arbitrary SPL 0.5 above T, as we rounded the SPL of T. Here, we have a maximum rounding error of 1-0.933032991537=0.066967008463 or around 6.7%. We notice, however, that this considers only one reading, and that since we capture a large magnitude of SPLs, our rounded values should cancel out. In the worst case that all of our rounded SPL values do not cancel out and we have the worst case rounding, i.e., 90.49999 to 90, we will underestimate exposure time by 6.7%. We reason that this error is acceptable because exposure times where 6.7% would be relevant, e.g., more than ten minutes would be 10 minutes/6.7% = 149.25 minutes, or around 2 and a half hours. However, since during this period we would be sampling many more SPL values than for a shorter but more intense period, we reason that the likelihood of this maximum underestimation is astronomically low.

Given that we can round to the nearest dB for any given SPL without noticeable error, we see that calculating daily exposure comes down to simply taking the weighted average of n values, where n is the amount of dB values we keep track of. We reason we only have to keep track of values between 85 dB to 125 dB since anything below 85 dB requires more than 16 hours to reach the 8 hour daily exposure threshold and anything above 125 dB will take less than 0.063*1 hour or around 4 minutes to reach the 8 hour daily exposure threshold. As such, we will only need to keep track of 41 different exposure dB values i.e., 41 different Cn. Each Cn value will be incremented based on a given digital microphone input over a certain period. Then, to calculate the daily dose, all we need to do is divide each Cn by its respective Tn and add it up. We reason that 41 division, 40 addition operations, and 1 multiplication operation will not take more than at worst 5 seconds given our microcontroller clock speed of 72 MHz. We also reason that this daily dose subroutine will thus certainly be above 10% tolerance defined by our R/V table for decibel levels below 125 dB.

Thus, given our above reasoning, we believe that our sound processing unit should be more than qualified to calculate integrated SPL/daily exposure dose.

3 Cost and Schedule

3.1 Cost Analysis

First, we would like to consider the labor costs of our project. To begin with, we consider the average salary of an ECE graduate. We use the most recent data from the official ECE and average the Electrical and Computer Engineering starting salaries to get an average yearly salary of $92,824 [10]. Next, dividing that yearly salary by 52 weeks and 40 hours/week, we get to an average hourly salary of around $44.63/hour. Next, we assume that each group member works an average of around 10 hours a week on the project, which should be about reasonable given in person meetings with TAs, independent research and design, etc. Finally, we see that we have 11 weeks of work from 9/19 to 12/5 not including the break. Summing everything together, we calculate our labor cost as $44.63/hour x 2.5 x 10 hours/week x 11 weeks x 3 group members = $36819.75 total. Thus, we calculate a total labor cost over the entire project as $36819.75.

Next, we would like to calculate the costs of the parts of the project. While we may use more or less parts as we further design our project, we reason that these parts will be most crucial.

Component	**Manufacturer**	**Part Number**	**Quantity**	**Unit Cost**	**Total Cost**
Battery Charger	Maxim Integrated	MAX1736EUT42+T	1	$4.16	$4.16
Battery	TinyCircuits	ASR00007	1	$5.95	$5.95
USB 2.0 Type B Connector	Allied Components International	AUSB1-4600	2	$1.10	$2.20

Linear Voltage Regulator	Texas Instruments	TLV70230DBVR	1	$0.61	$0.61
Digital Microphone	PUI Audio, Inc.	DMM-4026-B-I2S-R	1	$2.42	$2.42
Microcontroller	Microchip Technology / Atmel	STM32F103C8T6TR	1	$7.64	$7.64
LEDs	Inolux	HV-5RGB60	10	$0.663	$6.63
Slide Switches	NKK Switches	MS12ASG13	2	$5.48	$10.96
				TOTAL:	$40.57

In total, our parts will cost $40.57.

We will not be using the machine shop.

In total, our project will cost $36819.75+$40.57 = $36860.32.

3.2 Schedule

Week	Deliverables	Alex	Jake	Jinzhi
9/19	Design Document	Finish initial draft of design doc	Finish initial draft of Design Doc	Finish initial draft of Design Doc
9/26	Design Document Check PCB Design Prototype	Finalize design doc Work on PCB, focus on user interface.	Finalize design doc Work on PCB, focus on sensor interface.	Finalize design doc Work on PCB, focus on microcontroller interface.
10/3	Design Review PCB Design	Revise PCB design based on feedback and focus.	Revise PCB design based on feedback and focus.	Revise PCB design based on feedback and focus.

10/10	PCB Order 1 Teamwork Evaluation	Complete teamwork evaluation Start sound report code Edit PCB design if needed	Complete teamwork evaluation Start microcontroller code Edit PCB design if needed	Complete teamwork evaluation Start microcontroller code Edit PCB design if needed
10/17	Finish Initial Design Assembly	Assemble Design Finish initial sound report code	Assemble Design Start verification process on sensors	Assemble Design Finish microcontroller code
10/24	Debug/Verify Modify Design (If Initial Design Inadequate)	Start verification process on user interface Revise sound report code if necessary	Continue verification process on sensors If needed, modify design or continue assisting in verification process	Start verification process on microcontroller
10/31	PCB Order 2 (If Needed) Individual Progress Reports	Complete individual progress report Continue verification	Complete individual progress report Continue verification	Complete individual progress report Continue verification
11/7	Continue Debug/Verification	Revise sound report software if necessary Prepare for mock demo	Prepare for mock demo	Revise microcontroller software if necessary Prepare for mock demo
11/14	Mock Demo	Final verification of full design	Final verification of full design	Final verification of full design
11/21	**Thanksgiving Break**			
11/28	Final Demo	Complete final presentation Work on final report	Complete final presentation Work on final report	Complete final presentation Work on final report
12/5	Final Presentation	Complete final report and	Complete final report and	Complete final report and

	Turn in Lab Notebook Teamwork Evaluation	teamwork evaluation	teamwork evaluation	teamwork evaluation

Reference for schedule as well as other minor design document features from Spring 2022 UV Sensor and Alert System design document [11].

4 Ethics and Safety

Given that our project aims to give industry level safety to groups that otherwise would not have it, ethics and safety are a primary focus.

On ethics, one issue we have to keep in mind is that some people may be suspicious that since the project uses a microphone, it may be used to secretly record people. This misuse in turn would violate the IEEE code of ethics I.1 [12]. To dissuade these fears, we believe we would have to make sure the microcontroller only keeps track of what it needs to: instantaneous and integrated SPL and frequency.

On safety, as our project is primarily devoted to promoting safe SPL according to the previously mentioned OSHA standards, we would have to guarantee that our final product accurately, within our defined tolerance levels, conveys SPL safety through both the LEDs and report. During the project, we would have to be careful to not expose any of ourselves and/or others to unsafe sound levels during testing. We would alleviate these concerns through simulating when possible and soundproofing and testing remotely when necessary.

Last, we would have to make sure that users of our project understand that the project can only accurately convey noise exposure warnings in the area that it is in comparable to an industry grade SPL dosimeter, and that the users themselves are responsible for noises outside of the range of the microphone and for protecting themselves against sound exposure they are warned against. Additionally, we would have to confirm that users know that if they use our noise alarm without an accompanying local computer, they will not receive the full benefits from the LEDs and sound report but merely the sound report.

5 Works Cited

[1] J. J. Eggermont, “Effects of long-term non-traumatic noise exposure on the adult central auditory system. hearing problems without hearing loss,” *Hearing Research*, vol. 352, pp. 12–22, 2017.

[2] V. P. of A. E. and M. C. at C. U. I. D. Jeff Smoot, “A comparison of digital PDM and I²S interfaces in MEMS microphones,” *Digi*, 11-Jul-2022. [Online]. Available: https://www.digikey.com/en/articles/a-comparison-of-digital-pdm-and-i2s-interfaces-in-mems-microphones. [Accessed: 29-Sep-2022].

[3] “AN5027 application note - stmicroelectronics.” [Online]. Available: https://www.st.com/resource/en/application_note/an5027-interfacing-pdm-digital-microphones-using-stm32-mcus-and-mpus-stmicroelectronics.pdf. [Accessed: 29-Sep-2022].

[4] D. Ramsay, “How to get i2s working on an STM32 MCU,” Medium, 17-Feb-2018. [Online]. Available: https://medium.com/@davidramsay/how-to-get-i2s-working-on-an-stm32-mcu-33de0e9c9ff8. [Accessed: 29-Sep-2022].

[5] “Department of Labor Logo United Statesdepartment of Labor,” Occupational Noise Exposure - Overview | Occupational Safety and Health Administration. [Online]. Available: https://www.osha.gov/noise. [Accessed: 24-Sep-2022].

[6] *St Community*. [Online]. Available: https://community.st.com/s/question/0D50X00009XkYmFSAV/logarithm-and-exponential-function. [Accessed: 29-Sep-2022].

[7] “Noise dosimetry terminology,” *Larson Davis*. [Online]. Available: http://www.larsondavis.com/learn/industrial-hygiene/noise-dosimetry-terminology. [Accessed: 13-Sep-2022].

[8] “A-weighting table,” *Acoustical Engineer*. [Online]. Available: https://acousticalengineer.com/a-weighting-table/. [Accessed: 29-Sep-2022].

[9] “Department of Labor Logo United Statesdepartment of Labor,” *1910.95 App A - Noise Exposure Computation | Occupational Safety and Health Administration*. [Online]. Available: https://www.osha.gov/laws-regs/regulations/standardnumber/1910/1910.95AppA#:~:text=If%20the%20dose%20as%20read,accumulated%20dose%20in%20percent%20exposure. [Accessed: 28-Sep-2022].

[10] Grainger Engineering Office of Marketing and Communications, “Salary averages,” Electrical & Computer Engineering | UIUC. [Online]. Available: https://ece.illinois.edu/admissions/why-ece/salary-averages. [Accessed: 22-Sep-2022].

[11] E. Boehning, G. Chan, and J. Huh, “UV Sensor and Alert System for Skin Protection (Design Document),” 05-Mar-2022. [Online]. Available: https://courses.engr.illinois.edu/ece445/getfile.asp?id=20032. [Accessed: 22-Sep-2022].

[12] “IEEE code of Ethics,” *IEEE*. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed: 13-Sep-2022].