Chapter 1

1. Introduction

As urban environments grow increasingly dense, environmental noise pollution has become a significant concern for public health and quality of life. It is now understood that sustained exposure to high noise levels can contribute to various health issues, from stress and sleep disturbances to more severe conditions such as cardiovascular disease [1]. Consequently, the importance of effective noise monitoring has never been more evident. Traditionally, this task has been the responsibility of specialised acousticians and engineers equipped with precise instruments like general-purpose sound level meters.

However, in recent years, there has been a marked surge in public engagement with environmental issues, including noise pollution [2]. This trend has sparked a demand for reliable yet affordable noise measurement devices for the public to monitor and evaluate environmental noise levels in their locales. Despite this growing demand, the vast majority of low-cost "noise meters" on the market exhibit significant shortcomings in terms of accuracy, leaving them unfit for effective noise assessment [3].

The International Electrotechnical Commission (IEC) has established a standard, IEC61672-1, that stipulates the precision required for a device to be deemed suitable as a sound level meter [4]. The standard delineates three distinct levels of accuracy, contingent upon the specific circumstances and contexts of instrument usage. Given the legal ramifications of noise pollution assessment – such as determining whether noise levels exceed permissible thresholds – the use of an instrument that conforms to IEC61672-1 becomes crucial. Devices that do not comply with these requirements are unlikely to be accepted in a legal context.

At the outset of my initial research outlined in Thesis A, no low-cost sound level meter on the market satisfied the IEC61672-1 requirements. Nevertheless, recent technological advancements in integrated circuits and microphones presented a promising avenue for developing a cost-effective yet reliable sound level meter. Capitalising on these developments, this research objective was to design a customised, low-cost mini sound level meter that adheres to IEC standards, empowering citizens to monitor noise pollution.

Building on the foundation laid in Thesis A, this Thesis B project strives to continue the pursuit of a cost-effective, IEC-compliant sound level meter. By doing so, it aims to contribute to the burgeoning field of citizen science and facilitate broader public participation in environmental noise monitoring, an increasingly critical aspect of urban living in the 21st century [5].

Chapter 2

2. Motivation

2.1 Personal Motivation

My fascination with the intricate interplay between sound and its impact on the environment and human and wildlife health has steered me towards this research. Despite being an invisible foe, noise pollution profoundly influences the health and well-being of organisms, stirring both concern and intrigue in me [1, 6]. We can enhance awareness and foster impactful interventions by equipping the public with affordable, reliable tools for monitoring their noise environment. My motivation is deepened by the prospect of contributing to the evolution of technology that enriches lives and safeguards our well-being.

2.2 Importance of the Problem

The ramifications of unchecked noise pollution are extensive. Research indicates that excessive noise disrupts the physical health, psychological welfare, and even the ecological balance of various life forms [1, 6]. In humans, chronic exposure to high noise levels can precipitate health issues ranging from hypertension and sleep disorders to cardiovascular diseases [1]. In wildlife, noise pollution interferes with essential biological functions such as communication and navigation, thereby endangering biodiversity [6]. The urgency and significance of addressing this problem cannot be overstated.

2.3 Current Context of the Problem

Despite growing recognition of noise pollution as a significant public health issue, accessible solutions are in short supply. Existing noise monitoring equipment, while accurate, is typically expensive and complex, acting as a deterrent for widespread public use [7]. Meanwhile, the affordable alternatives on the market need more precision and reliability, rendering them ineffective for any meaningful noise assessment [7]. With urban noise levels on an upward trajectory, the demand for affordable, user-friendly, and reliable noise monitoring devices is more urgent than ever.

2.4 Broader Impact and Implications

The successful development of a cost-effective, IEC-compliant sound level meter could spark far-reaching implications. Primarily, it would democratise noise monitoring, fuelling citizen science and bolstering public engagement with environmental issues [8]. Such a tool could catalyse community-level noise management initiatives, culminating in improved public health outcomes. Additionally, the widespread availability of affordable and reliable noise meters might influence policymaking, potentially paving the way for stricter noise control regulations and more effective enforcement [9].

Chapter 3

3. Literature Review

3.1 Critical Review of Previous Works

Previous research in sound level monitoring has predominantly centred on three primary devices: smartphones with noise-measuring apps, personalised sound level meters, and micro: bits. Each of these devices has its own merits and limitations.

3.1.1 Smartphones with noise measuring apps

The latest cell phones are the most cost-effective approach to providing equipment or devices to measure noise levels, but they have limitations and performance difficulties. Accelerometers, microphones, light sensors, cameras, GPS receivers, proximity sensors, and gyroscopes are now standard features on most smartphones. Many sound-measuring apps use the smartphone's built-in microphone, while just a few sophisticated apps use an external microphone. Sound measuring apps are popular among environmental researchers, educators, acoustic and audio aficionados, and the public [10]. Mobile applications for measuring noise include Too Noisy Pro, Decibel X, Niosh sound level metre, sound metre pro, Noise Hunter, and NoiSee. These applications on iPhone and Android phone performance are compared in a test and evaluated by the National Institute for Occupational Safety and Health (NIOSH) [11] in its acoustic testing facility. NIOSH believes that these four iPhone applications for occupational noise evaluations may be dependable, with an error rate of less than 2 dB; however, because several manufacturers manufacture Android phones, app measurements on different devices differed substantially [11]. As a result, NIOSH was unable to recommend a specific Android application. Furthermore, none of the Windows cell phones with noise apps passed NIOSH's requirements [11]. Based on their research, [10] came to a similar conclusion. Colleen compared the accuracy of smartphone apps to that of a calibrated sound-level metre [12]. According to this survey, most mobile apps had greater sound levels.

Smartphone microphone technology has progressed since the advent of the first mobile phones. A single microphone was initially utilised to convey the user's voice during a phone call. Three or four MEMS microphones (microelectromechanical systems or tiny electret microphones that consume little power and occupy little space) are usually placed in current smartphones. Smartphones may incorporate additional microphones to provide stereo recordings, remove background noise when necessary, and even zoom in on specific sounds [13]. It is evident from this data that the number of microphones on each smartphone differs. [10 -14] does not specify which mobile applications use which microphone in a mobile device. A typical operational amplifier filter will have five active stages and will consume a significant amount of power. According to Francesco Frigorie in [14], current smartphones include a microphone, circuitry capable of serving as an analogue to a digital converter, and some computational capacity. As a result, the acoustic level may be estimated in the time domain, and the spectral distribution of the sound can be obtained using the FFT. The sound pressure level (SPL) as a function of time and a bar graph of a third octave are two of the most common ways to depict the spectrum's decibel (dB(A)) level. Microphones with a low signal-to-noise ratio (SNR) may overlook low-volume sound components during calibration, resulting in data loss. Since the introduction of Very-high-SNR microphones in 2013[15], the results of sound-measuring apps on phones with low and high SNR microphones may differ. Unfortunately, nearly no sound level metres that meet international or national regulations are available for all smartphone brands [16]. With such constraints, smartphones are only one of my choices for measuring noise.

3.1.2 Personalised Sound Level Meter

Further research led me to look at low-cost, low-power components such as sound sensors, filters, and microcontrollers… Many researchers were quite interested in this topic. A-weighting filters are essential to assess noise exposure in humans accurately. The international standard IEC 61672-1 [17] refers to A-weighting filters, although their practical implementation needs to be described. In [18], Ashwini K.S. has a smart noise detector that can detect loud talkers and inform us by flashing a red light and hearing a beep. Aswinth Raj recorded decibels (dB) noise levels using an Electret Condenser microphone and Arduino [19]. Wilson and Jarvis [21] developed a WSN for noise measuring that employs energy harvesting. These investigations show that sound levels can be quantified; however, the A-weighting idea must be used based on how humans perceive proper levels. Most prior noise measurement investigations [21-25] employed analogue A-weighting filters. A filter like this usually has five active stages made of operational amplifiers and consumes significant power. The design of Douglas R. Lauman in [20] introduces the FFT to create the A-weighting filter but does not demonstrate whether the same logic works with sensor data and amplifiers.

Santini et al. and Filipponi et al. [22,23] presented wireless sensor networks for noise pollution monitoring, but these solutions need to perform signal processing at the sensor nodes. Instead, they employ specialised electronics to assess noise levels. However, details about how the strategy will be implemented have yet to be provided. Hakala [24] and Kivelä et al. [25,26] have published more information on WSN design for noise measurement. To reduce computing complexity, an analogue A-weighting filter is utilised. According to the authors, a digital filter necessitates a significant number of floating-point computations, which are beyond the capability of a tiny MCU. A cascade of three analogue high-passes and two analogue low-pass filters is employed to create an A-weighting filter.

Environmental contamination can be precisely detected, and spatiotemporal information can be communicated to the public using wireless sensor networks [27]. Dauwe et al. [28] provide a multi-layered middleware platform for sensor networks, including data gathering, control, and administration techniques. In [29], Di Francesco et al. give a detailed classification of WSN topologies and a description of the data-gathering method. Their research also includes a thorough evaluation of the relevant literature. They also identify potential research directions that could be pursued in the future. Reis et al. [30] discuss how recent advances in environmental sensing have enabled data-intensive modelling for environmental and human health. Blythe et al. developed a sensor node prototype for monitoring temporal and spatial urban (traffic) [31]. A low-power PIC18F4620 microcontroller with an integrated GPS module, as well as sensors such as an accelerometer with three axes of motion and a three-dimensional temperature sensor, are included in the node. A low-power Zigbee radio communicates with the sensor, which sends data to a server. This publication provides no additional information on the acoustic noise sensor. However, the authors note that excellent correlation (within 3 dB) in the 55-100 dB range was verified. The publication by Bell and Galatioto [32] contains a plethora of information regarding the same acoustic sensor node. The authors demonstrate numerous installations of up to 50 sensors (dubbed "motes") used to create pervasive networks to better comprehend noise in urban environments and street canyons.

3.1.3 Micro:bit

It's feasible that the Micro: bit will be the next device that meets my requirements for a particular extension. Micro: bit can handle a wide range of use cases (light sensors, flashing emojis, sound logger, step counter, thermometer, compass teleportation, and so on) on a single device that supports numerous programming languages (MakeCode, Python, Java, and Scratch) [33]. It is competent in these areas. Furthermore, it has an integrated transceiver that can communicate by radio and Bluetooth [34]. The fact that they both broadcast and receive on the same 2.4GHz frequency does not affect the reality that each protocol has subtle distinctions that make some protocols better suited for specific professions or activities than others [34]. The most notable difference between Bluetooth and radio is that the latter does not place a premium on creating security procedures or trust relationships [34]. It just broadcasts the information in the hope that someone will receive it. As a result, one of the limits of wireless transfer using micro: bit is the security protocol. Furthermore, its cost is a crucial worry for my research, and it has another disadvantage regarding user-friendliness. A new end-user with the micro: bit may not comprehend the outputs or find it difficult to operate.

3.1.4 Professional sound level metres (SLMs)

Using professional sound level metres (SLMs) is a simple possibility for my thesis objective. SLMs are widely available in the market today, each with specifications [35]. Every professional SLM falls under class 1 and meets the IEC 61672-1 standard [36]. Because of their high level of accuracy, Class 1 sound level metres are frequently referred to as "precision" grade metres. It is appropriate for use in the laboratory and applications involving building acoustics, the environment, traffic evaluations, and boundary noise [37]. Class 1 SLMs can range from $200 to $6900, depending on the device's functionality and capabilities, such as octave band, environmental SLM, integrating precision integrated data logger, Bluetooth, weighting, and so on [35]. Even though class 1 SLMs meet all specifications, their high price makes them neither economical nor affordable, so additional solutions for achieving the thesis purpose are being explored.

Evaluating the technologies mentioned in the above studies to build a mini-sound level meter, the information [38] and [39] provided for implementing an IoT decibel meter with a sound sensor is closely associated with my goals. [40] helps perform the FFT on the collected signal from the excellent sensor, which would help implement the A-weighting filter to the audio.

3.2 Novel Advances in the Field

The proposal of a mini sound level meter presents a novel approach to the issue of accessible and reliable sound measurement. This device combines the tools' benefits – accessibility, user-friendliness, accuracy, and affordability.

3.3 Comparative Analysis of Similar Works

Compared to existing solutions, the mini sound level meter offers distinct advantages. It combines smartphone apps' ease of use, affordability, and micro: bit to achieve precision akin to professional sound level meters. This balance is rarely seen in the currently available tools. The accuracy of smartphone apps and micro: bit tends to suffer, while the cost and complexity of professional sound level meters make them less accessible.

3.4 Gaps in Knowledge and Their Significance

While there is ample research on individual sound-level measuring tools, more cohesive comparative studies must be conducted. Specifically, there needs to be more research comparing the performance and reliability of smartphone apps, micro: bit, professional sound level meters, and other potential solutions like the proposed mini sound level meter. Such a comparison is crucial to identify each approach's strengths and weaknesses and guide future developments in the field.

3.5 Connections with Proposed Research and Reviewed Works

The proposed research builds upon the existing body of knowledge by seeking to address the limitations of currently available tools. It will involve developing and testing a new type of sound level meter that balances accessibility, affordability, and accuracy. The comparison of this new tool with existing ones will contribute valuable insights into the field of good measurement.

Chapter 4

4. Background Knowledge

4.1 Sound and Noise

Sound is commonly understood as a sensation that our ears perceive. It refers to the changes in air pressure that result from compressions and rarefactions in longitudinal waves. Noise, however, denotes an unpleasant, loud, and distracting sound. Despite these differences in usage, these terms are often used interchangeably in disciplines like electronics, physics, and acoustics. The amplitude and frequency of a sound wave are vital in determining the nature of the sound it produces. The intensity or loudness of sound is measured in decibels (dB), which quantify the power of a wave [41].

4.2 Safety Measures

Our environment has varying noise levels, from as low as 20 decibels (dB) to as high as 140 decibels (dB).

Figure 1: Range of noise levels in the environment [42]

When exposed to noise levels above 85 dB(A) for more than 8 hours, a person's hearing can be damaged to an extent equivalent to the loudness of heavy city traffic. However, people have varying sensitivities to noise. The volume and duration of exposure are crucial in determining potential hearing damage [42].

4.3 International Electrotechnical Commission

The International Electrotechnical Commission (IEC) is a global organisation that develops and maintains standards for various technologies, including power generation and transmission, home and office equipment, marine energy, fibre optics, semiconductors, batteries, and solar energy. IEC standards not developed in association with ISO bear numbers in the range of 60000–79999 and follow a specific naming pattern [43]. These standards are adopted worldwide by several certifying bodies, although the adopted standards might vary slightly from the original IEC standards [43].

4.4 SPL Meter

A Sound Pressure Level (SPL) meter is a handheld instrument with a microphone, often a condenser microphone, used to measure sound level. It senses the changes in air pressure caused by sound waves and converts it into an electrical signal. Good-level meters are widely used in various noise pollution studies, including those related to industrial, environmental, mining, and aircraft noise. IEC 61672-1:2013 is the current international standard specifying the functionalities and performance of sound level meters [44].

4.5 Classification of SPL

Three types of sound measuring instruments are listed in the IEC 61672-1—the "conventional" integrating averaging and integrating sound level meters. The RMS circuit's output voltage is carried through a logarithmic course (dB) to get a linear decibel reading. This value equals 20 times the logarithm to the base ten ratios of the given root-mean-square sound pressure to the sound reference pressure [44]. The root-mean-square sound pressure can be calculated with standard frequency and time weighting. The international agreement sets the reference pressure for airborne sound at 20 micro pascals.

When assessing the potential for hearing damage, an integrating or integrating-averaging sound level meter is required rather than an exponentially averaging sound level meter. To measure sound exposure, an integrating meter sums the frequency-weighted noise to get the whole sound direction, and the metric used is pressure squared multiplied by time, often Pa^2s, but Pa^2h is also used. However, due to the historical use of decibels to describe sound levels, the term "sound exposure level" (SEL) is mainly used to refer to the logarithmic conversion of proper exposure levels into decibels.

A noise dosimeter is a common variation on the sound level meter. Personal sound exposure meters (PSEMs) are now recognised by the International Electrotechnical Commission (IEC) and have their international standard, IEC 61252:1993.

IEC standards define two "classes" of sound level meters. The functionality of sound level meters in the two classes is identical, but the error tolerances differ. There is a more excellent frequency range and tighter tolerance in Class 1 instruments as compared to a lower-priced Class 2 unit. For both the meter and the calibrator, this is true. Class 2 instruments are generally allowed under most national standards. Many measurements do not require a Class 1 unit; these are best suited for research and standardisation.

4.6 Measurements of dBA

Frequency weighting effects (how the sound level meter responds to different sound frequencies) and time weighting are crucial for measurements (how the sound level meter reacts to changes in proper pressure).

As the AC signal from the microphone is converted to DC by a root-mean-square (RMS) circuit, the standard sound level meter can be referred to as an exponentially averaging sound level meter, as it must have a time constant of integration, also known as the time-weighting [44]. Three of these time-weightings have been internationally standardised: 'S' (1 s) was formerly known as Slow, 'F' (125 ms) as Fast, and 'I' (35 ms) as Impulse. In the 1980s, their names were changed to be identical in all languages. As a result of this lack of real correlation, i-time-weighting is no longer included in the standard.

To better account for human perception of sound when stating Sound Pressure Levels, A-weighting can be applied, which reduces the level of high- and low-frequency components of a sound while increasing it slightly in the middle range [45].

After the decibels, you'll see an 'A' or '(A)' to indicate A-weighting. The long form of '45 dB(A)' is '45 decibels with reference 20 micro-pascals with the low and high-frequency components of the sound reduced by A weighting', but people look at you strangely if you use the long form, so it's probably better to stick to '45 dB(A)'.
The use of C-weighting could be more frequent. C-weighting expresses high sound pressure levels, such as sudden high-amplitude sound impulses.
Despite their decline in popularity, B- and D-weighting attempted to convey the same effect as C-weighting in alternative ways. As seen in Figure 3, a flat response in the range of frequencies audible to humans is prescribed by Z-weighting or no weighting.

Here is a comparison of the frequency-weighting curves for the range of frequencies audible to humans (20 Hz-20 kHz) for the various angles. For example, at 100 Hz, A-weighting attenuates about -20 dB due to the frequency weighting. Frequency weighting curves show this attenuation.

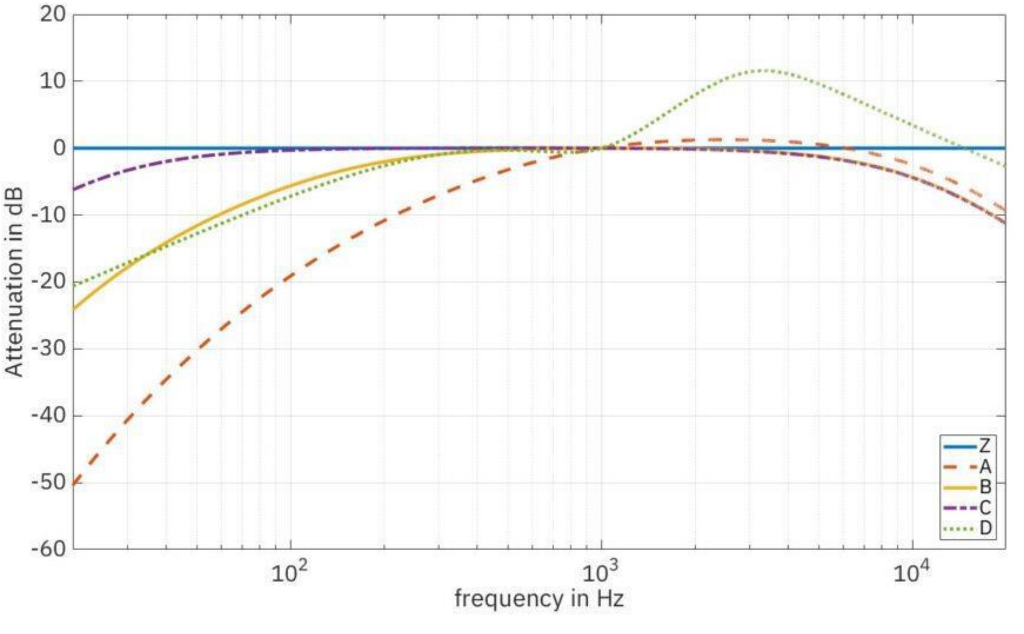

Figure 2: Comparison of A-weighting, B-weighting, C-weighting, D-weighting, and Z-weighting curves for the audible frequency range [45]

4.7 Sound Level Measurement – A-Weighting Filter

The human auditory system is sensitive to changes in air pressure, which the brain interprets as sounds. As a result, the most practical method for quantifying the sound level is to measure the pressure of the sound wave at the location where the listener is receiving the sound. The sound pressure level is computed as the root-mean-square (RMS) value of the sound pressure, which is then expressed in decibels relative to the reference pressure of p0 = 20 Pa and expressed in decibels (dB).

$$SPL = 20 \log \frac{p_{RMS}}{p_0} \quad [\text{dB}].$$

The threshold of human hearing at 1000 Hz is used as the reference value for p0. The RMS value of the instantaneous sound pressure p is the adequate sound pressure. It is defined as the RMS value of sound pressure.

$$p_{RMS} = \sqrt{\frac{1}{T}\int_0^T p^2(\tau)d\tau}.$$

Since the computation of the mean root square in Equation [46] involves time averaging, sound level measurements use three different values for the time constant T. These values are impulse (I), fast (F), and slow (S) averaging, with the time constants equalling 35 ms, 125 ms, and 1 second, respectively [46]. The exponential averaging method is utilised by analog sound level meters in conjunction with the time constants. On the other hand, digital sound level meters typically use linear averaging with integration times that are twice as long as the time constants that are relevant in the case of analog sound level meters [46].

A sound level can be objectively measured using the sound pressure level. However, humans have a highly subjective way of perceiving sounds, and the perceived loudness of a sound depends not only on the sound level but also on the frequency bandwidth, spectral content, information content, and the amount of time spent being exposed to the sound.

Because of this, its quantification presents a significant challenge. The research by Fletcher and Munson [46] laid the groundwork for the fundamentals of loudness measurements. Fletcher and Munson came up with equal-loudness contours after studying the relationship between sound level and loudness for tones of various pitches. These contours show the intensities of tones of different frequencies that the average listener perceives as equally loud. Fletcher and Munson's research can be found here. The Fletcher and Munson curves served as the foundation for the first sound level meter standard, which was developed by the Acoustical Society of America (ASA) in 1936 and published as the tentative American standards for sound level meters for the measurement of noise and other sounds Z24.3-1936. The IEC 61672-1 standard [46] for sound-level meters defines three different types of frequency weighting: A, C, and Z. These weightings are designed to consider the influence that frequency has on human perception of loudness. All sound level meters are required to use the frequency weighting A; sound level meters that conform to class 1 tolerance limits are also required to use the C weighting, and using the Z weighting is entirely voluntary.

Although A-weighting is not well correlated with human perception of loudness [46], it is required by law in many countries and allows the comparison of measurement results with historical data. A sound level meter design with frequency weighting is what we'll be talking about today. On the other hand, the different frequency weightings could be easily implemented if the described approach is followed.

Due to the human ear's sensitivity to noise-induced hearing loss, the A-weighting filter [46] is a bandpass filter that emphasises the frequency range from 1 to 4 kHz in its frequency response (Fig. 1). In [46], an analog A-weighting filter's transfer function is defined as.

$$H_a(s) = \frac{4 \cdot \pi^2 \cdot 12194^2 \cdot s^4}{(s + 2\pi \cdot 20.6)^2 (s + 2\pi \cdot 12194)^2 (s + 2\pi \cdot 107.7)(s + 2\pi \cdot 739.9)}$$

Following the application of the A-weighting filter, as described in [46], to the measured signal, its root mean square value is calculated, and the obtained value of sound level is reported in decibels, abbreviated as dBA.

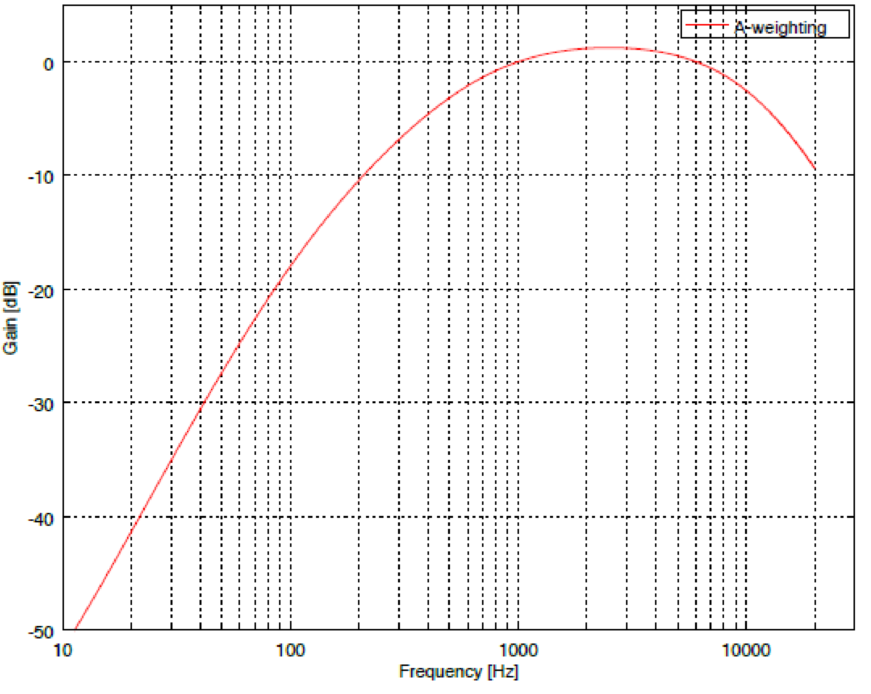

Figure 3: Magnitude response of the A-weighting filter [46]

4.8 SPL Meter (QM1598)

Designed for all kinds of environmental sound measurement projects, this professional-grade sound level meter is ideal. Fast (125ms) and slow (1s) time weighting can be used to get an ambient reading or a short nose. It is necessary to use A and C weightings in conjunction with the human ear-friendly values of minimum, maximum, and data hold to achieve the desired response. In addition, there is a backlit display with the date and time, data logging, analogue AC/DC outputs, and a USB port for further analysis. A built-in alarm can go off if a certain threshold is crossed.

Figure 4: Protech's Pro Sound Level Meter with Calibrator [47]

4.9 Micro:bit

The Micro:bit Educational Foundation is a non-profit organisation founded in 2016 in the United Kingdom to inspire people worldwide to create their ideal digital future[48]. They accomplish this by developing hardware and software that piques the interest of young people in technology and the opportunities it affords them, by creating accessible, user-friendly educational resources to assist teachers in delivering engaging and creative lessons, and by collaborating with like-minded partners to deliver high-impact educational programmes around the world [48].

The micro:bit device has numerous features to support sensing capabilities at all levels, from beginner to advanced. Name badges, step counters, thermometers, animal trackers, sound meters, clap lights, simple door alarms, sprint level, sound compasses, proximity beacons, tilt alarms, energy light meters, energy cost calculators, radio door alarms, and sound compasses are a few of the capabilities of the micro:bit module. This project uses the sound meter functionality to compare the results to the personalised sound level meter.

Sound level logger: Creates a sound level logger to track how loud or quiet various locations around you become over time.

The mechanics of how it all works.

- Like the light sensor, the new micro:bit's microphone measures sound levels from 0 to 255.
- The loudest sound is stored in a variable called maxSound, and the current sound level is constantly compared to that value. If the contemporary sound is louder than the previous most deafening sound, the max sound is reset to the new most piercing sound value.
- An if statement is used in the loop to see if you've hit the "A" button. If you have one, you can see the sound level number displayed on the LED output. You can use this to keep track of the changes in volume over time in various locations.
- Press the reset button on the micro:bit's back to reset the maximum value.

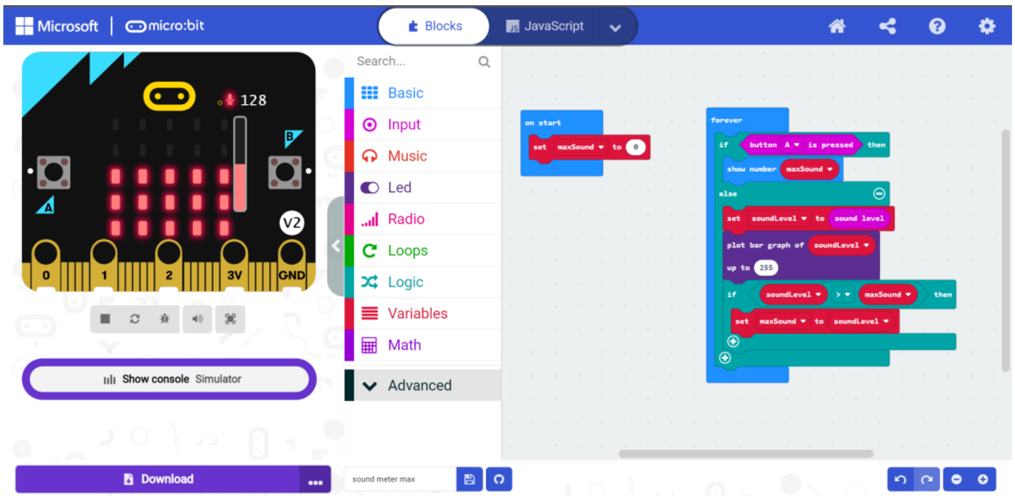

Figure 5: Micro:bit – Sound Logger's predesigned blocks[48]

4.10 Arduino UNO

The Arduino UNO is a programmable open-source microcontroller board available for a low price, is highly flexible and easy to use, and can be incorporated into various electronic projects. This board can control relays, LEDs, servos, and motors as output and can interface with other Arduino boards, Arduino shields, and Raspberry Pi boards. It can also interface with other Raspberry Pi boards. AVR microcontroller Atmega328 is included in Arduino UNO, along with six analogue input pins, fourteen digital I/O pins, and six digital I/O pins dedicated to PWM output.

4.11 Spark fun Sound Detector

The SparkFun Sound Detector is an audio sensing board that is both compact and very simple to use, and it features three different outputs. The Sound Detector produces an audible output and generates a binary indication of whether a sound is present and an analogue representation of the sound's amplitude.

The sound detector has three separate outputs of its own. A graph provides the most precise picture of how each output is varied. The example that is going to follow demonstrates how a sound detector reacts to a sequence of sound pulses.

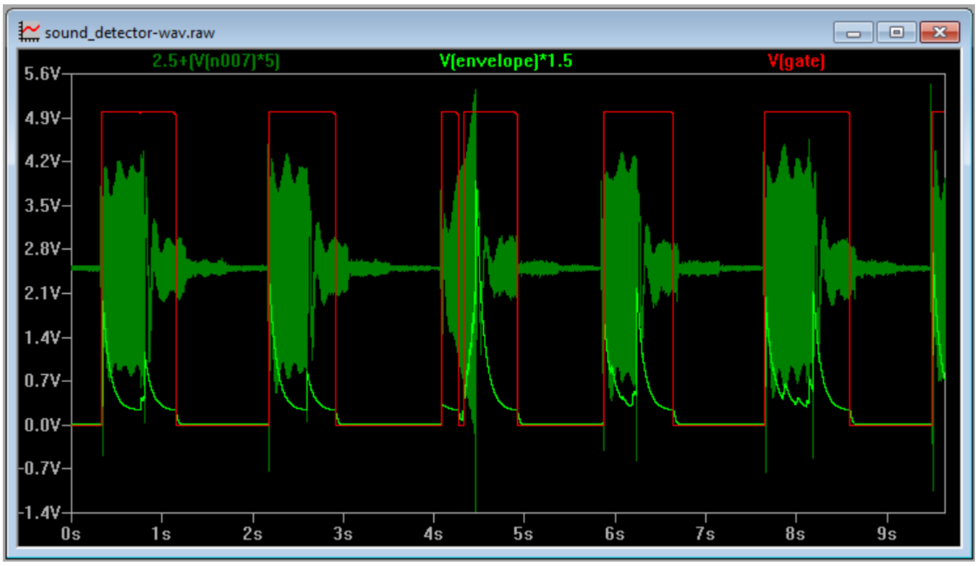

Figure 6: Sample Sound Detector Output [49]

This graph illustrates the output voltages over time:

- The dark green trace represents the sound detector's output. This output contains the voltage directly from the microphone.
- The light green trace is the output of the envelope. This analogue voltage represents the sound's amplitude. Please observe that the third pulse becomes increasingly audible as it continues.
- The red line is the output of the gate. This output is low during quiet conditions and high when sound is detected.

How it works:

After examining the outputs, let's read the schematic to understand how each stage functions.

First Stage: A capsule for an electret microphone serves as the initial component of the circuit. This circuit section utilises the Electret Microphone breakout board to function correctly.

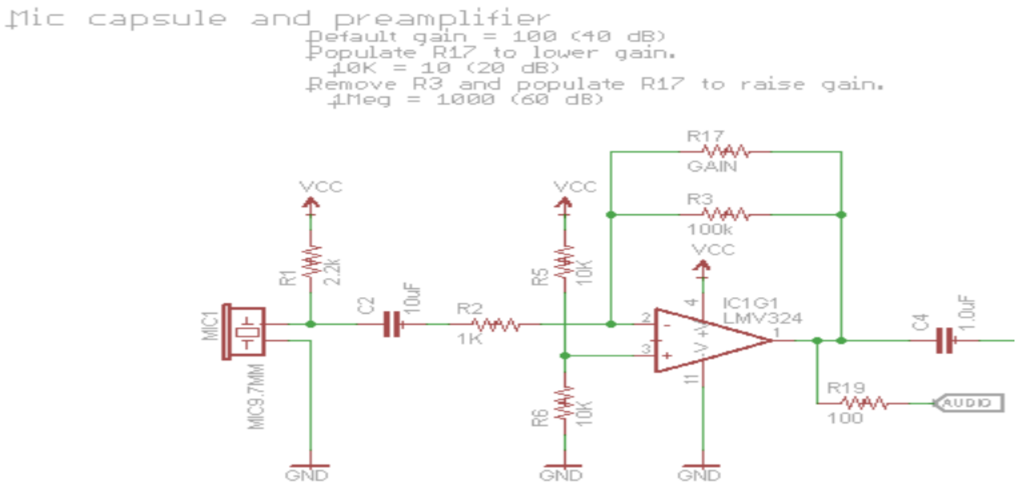

Figure 7: Microphone and Preamplifier [49]

The supply voltage is used to bias the capsule, and the tablet's output is an alternating current voltage riding a direct current offset of approximately half the supply voltage.
Because the voltage produced by the capsule at its output is very low, the signal produced by the tablet must be amplified by IC1G1, which is an operational amplifier stage. The preamplifier has an arithmetic gain of 100 (20 dB) by default, and the payment can be adjusted by filling in R17 with the appropriate value.

It is possible to connect the audio output directly to the ADC of a microcontroller because it is DC-coupled and rides on one-half of the supply voltage. It should read half full scale, equal to 512 on a 10-bit converter when conditions are quiet.

Second Stage: The envelope follower is the component that makes up the circuit's second stage.

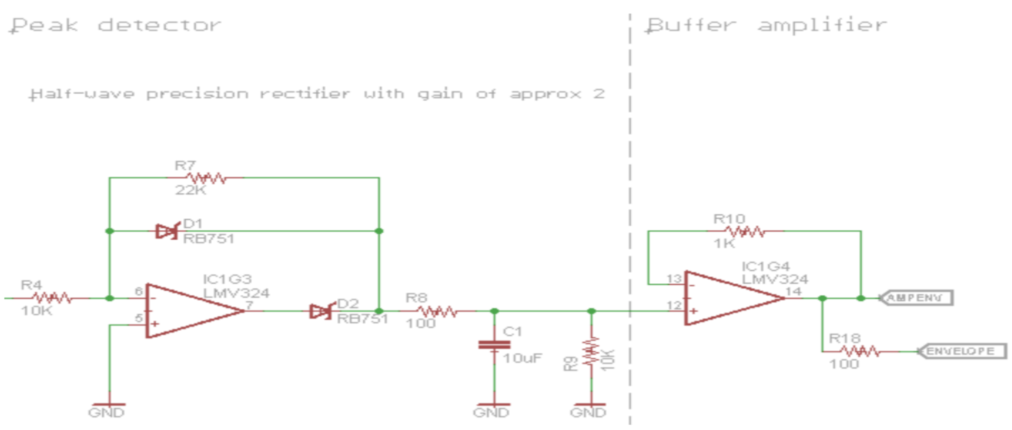

Figure 8: Envelope Follower [49]

The signal is both inverted and amplified by the operational amplifier. When its output goes into a high swing, D2 turns on and charges C1 in preparation for the next cycle. When the output of the op-amp is high or not swinging, the diode D2 is disabled, and capacitance C1 discharges through resistor R9. As a result, the peaks of the input signal are followed by C1. Because IC1G4 is a buffer amplifier, the charge/discharge behaviour of C1 will not be affected by external loads on the envelope pin.

This produces a signal that follows the peak amplitude of the signal that was input into the system. A higher sound pressure level will cause the voltage on the Envelope pin to increase. A microcontroller's analog-to-digital converter (ADC) can connect the audio and envelope pins.

Third Stage: The last stage includes implementing a threshold switch on the envelope signal.

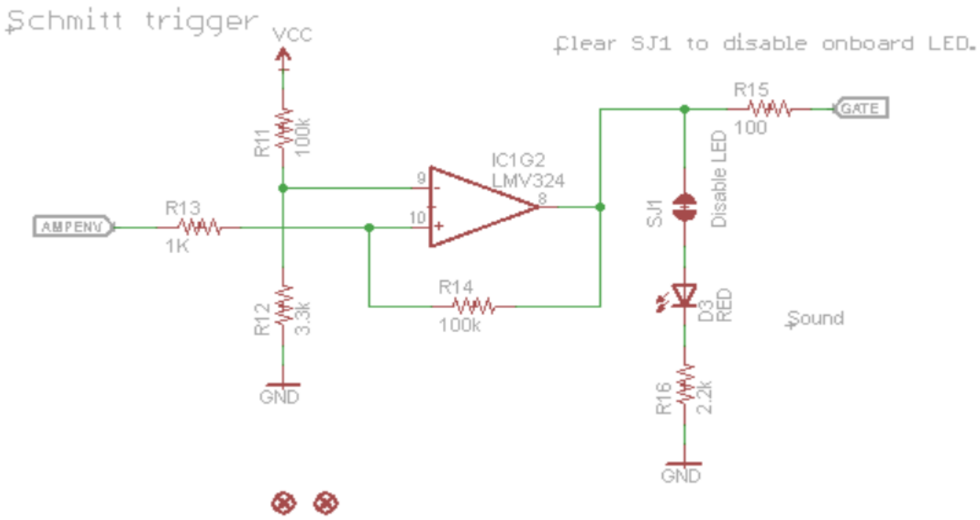

Figure 9: Schmitt Trigger [49]

When the predetermined threshold is broken, the Schmitt trigger will flip the output on and off. It does this by monitoring the envelope signal. A Schmitt trigger is a comparator that adjusts its threshold voltage when the output switches. This means a higher voltage is needed to switch on than switch off when using a Schmitt trigger. Because of this, it can ignore certain ripples in the input signal, such as the ripple in the output of the envelope follower stage.

The Schmitt trigger's output can be found on the Gate pin of the microcontroller. We can hook it up to a digital input if you want to.

Chapter 5

5. Implementation and Technical Execution

5.1 Tools & Technologies Used

5.1.1 Arduino

As the heart of our project, Arduino provides the computational and hardware integration capability for our sound level meter. I chose it for its low cost, versatility, and strong community support, which provides a wealth of shared knowledge and libraries. The specific model used was an Arduino Uno, balancing processing capability and physical size.

5.1.2 SparkFun Sound Detector

The SparkFun Sound Detector is our primary sensor for audio data collection. It was selected due to its simplicity, small size, and providing three different types of output: raw audio, a binary presence or "gate" indicating if a sound was detected, and an analog envelope for sound amplitude. I primarily used the envelope output, which gives us a voltage proportional to the amplitude of the sound detected.

5.1.3 ArduinoFFT Library

The arduinoFFT library plays a significant role in this project. It provides functions that perform Fast Fourier Transformations (FFT) and calculations involving complex numbers, which are necessary for processing the audio signals and determining their frequency components.

5.1.4 IEC Standard for dBA calculations

To ensure that our sound level meter has practical relevance and comparability to professional systems, implemented the dBA calculation by the IEC 61672-1:2013 standard. This standard provides the detailed procedures and specifications for electronic sound level meters, which I adhered to in our coding of the dBA algorithm on the Arduino platform.

5.2 Detailed Process and Original Approaches

5.2.1 Collection of Sound Data

The audio data collection was conducted by placing the SparkFun Sound Detector in various environments to gather raw audio data. Then utilised Arduino's ADC (Analog to Digital Converter) to convert this envelope output into a digital format we could process. I set up the Arduino to take readings at regular intervals and stored these readings in an array for further processing.

5.2.2 Implementing dBA Calculation

For the processing part, developed an algorithm to calculate dBA from the raw data by the IEC standard. Applied an A-weighting filter to the raw data array to emphasise certain frequencies per human hearing sensitivity. Then, computed this filtered data array's root mean square (RMS) value to represent the sound pressure level. Finally, I took a logarithmic measure of this RMS value to get the dBA.

5.2.3 Comparison of Results

To validate our design and implementation, compared the dBA values of the Arduino-based sound level meter measured with measurements from professionally calibrated devices such as a Protech SPL meter, a micro:bit device, and several mobile apps. This gave a good estimate of how accurately our system performs in the real world and highlighted areas where potential improvements could be made in our design.

Chapter 6

6. Design & Methodological Enhancements

6.1 Updates based on Thesis-A Feedback

Based on the feedback and observations from Thesis A, significant improvements have been made in Thesis B. The primary issues identified previously were inconsistencies in output when changing input frequencies and large deviations in dBA values compared with the Protech SPL meter and iPhone app. These have been addressed by refining the data processing steps in Arduino. The cause of overflow and NaN outputs in the FFT function was raw data from the sound sensor. This issue has been mitigated through robust data handling and error checking in the Arduino code.

6.2 Improvements in Design & Methods

A more sophisticated approach has been adopted in Thesis B to enhance the audio signal processing. It involves windowing the signal using a Hamming window to minimise the spectral leakage before applying FFT. This helps provide a more accurate frequency domain representation of the audio signal. Further, the algorithm implements A-weighting more precisely, using the equation for A-weighting provided in the IEC standard.

Additionally, a different mechanism for visualising the output is considered to circumvent the problem of the indistinguishable output of 1 from 0 in the Arduino serial plotter. The processed dBA values are now outputted to a serial monitor for a clearer and more readable representation of the sound levels.

These enhancements contribute to a more reliable and accurate mini sound level meter that computes dBA values per the IEC standard and provides meaningful comparisons with other sound level meters and apps.

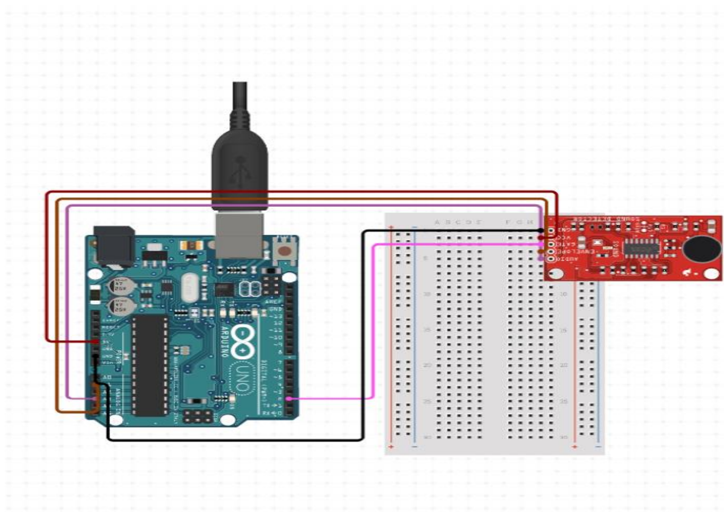

Figure 10: Circuit Diagram of mini-sound level meter

6.2.1 Algorithm for audio processing in Arduino

1. **Initialization**: The necessary components are initialised in the setup phase. This includes setting up serial communication and the audio pin mode.
2. **Data Acquisition**: In the loop function, the raw data array and fWeight array are shifted by the hopSize at the start of each iteration. This is done to ensure a rolling data window is maintained for processing. The new data points from the audio sensor are then read, normalized to a range of -1 to 1 and then added to the end of the rawData array. The normalized audio data calculates the fWeight array, a simple low-pass filter.
3. **Preparation for FFT**: The incoming raw audio data is pre-processed for the FFT (Fast Fourier Transform). The vReal array is populated with the square root of the fWeight values, and the vImag array is set to zero.
4. **FFT and Conversion to Magnitude Spectrum**: The FFT library converts the time-domain signal (vReal array) into a frequency-domain representation. Before FFT, windowing is applied to the audio data to minimise spectral leakage. After the FFT computation, the results are then converted to magnitude representation.
5. **Calculating Decibel Values and A-weighting**: For each frequency bin in the magnitude spectrum, the magnitude is calculated, and then it is converted to dB (decibel) values using the formula `20 * log10(magnitude) + ref_dB`. This dB value is then converted to linear power and accumulated in the variable `sum`. This is done for each frequency bin in the first half of the spectrum (since the second half of an FFT result of real inputs is a mirror image of the first half). After all frequency bins have been processed, the total power `sum` is converted back to dB using the formula `10 * log10(sum)`.
6. **Applying A-weighting Filter**: The A-weighting scale adjusts the frequency response by human hearing. A mathematical formula based on the IEC standard for A-weighting is used to calculate the A-weighting factor `A_weighting`. This factor is added to the `total_dB` to get the A-weighted sound level `dBA`.
7. **Final Adjustments**: The `dBA` value is adjusted based on some conditions better to match the readings of a standard sound level meter.
8. **Output**: The adjusted `dBA` value is then printed to the serial monitor. This value indicates the sound level in the environment measured by the Arduino-based sound level meter.

Chapter 7

7. Advanced Evaluation & Analysis

7.1 Test Protocols

The testing protocol for the Mini Sound Level Meter was developed with an emphasis on practicality and replicability, allowing us to generate reliable and comparable results. This comprehensive approach included both volume and angle tests, each designed to evaluate the performance of the device under varying conditions.

7.1.1 Testing Environment

The testing was conducted in a sound-treated room in the Australian Hearing Hub (AHH) atrium at Macquarie University. The room dimensions were 17 feet in width, 10 feet in height, and 23 feet in length. This environment minimized issues with low-frequency pressure and reflections that could affect the test results, providing a controlled setting ideal for accurate measurements.

7.1.2 Volume Test Protocol

The volume test aimed to evaluate the accuracy of the Mini Sound Level Meter across a range of sound levels. The test followed these steps:

1. Online noise generators were used to generate the input sound.
2. The volume was increased incrementally from 0 to 100% on the laptop or speaker.
3. At each volume level, readings were taken from the Protech SPL meter, the Mini Sound Level Meter, and a Mobile App.
4. The readings from the Mini Sound Level Meter and Mobile App were then compared to the reference measurements from the Protech SPL meter.
5. The process was repeated to ensure consistency and reliability of results.

Figure 11: Testing of 3 devices in a sound treated room

7.1.3 Angle Test Protocol

The angle test sought to assess the sound localization ability of the Mini Sound Level Meter. The protocol for this test was as follows:

1. The Mini Sound Level Meter, Protech SPL meter, and the Mobile App were positioned at varying angles relative to the sound source.
2. At each angle, measurements were taken from all three devices.
3. The readings from the Mini Sound Level Meter and Mobile App were then compared to the reference measurements from the Protech SPL meter.
4. The process was repeated to ensure consistency and reliability of results.

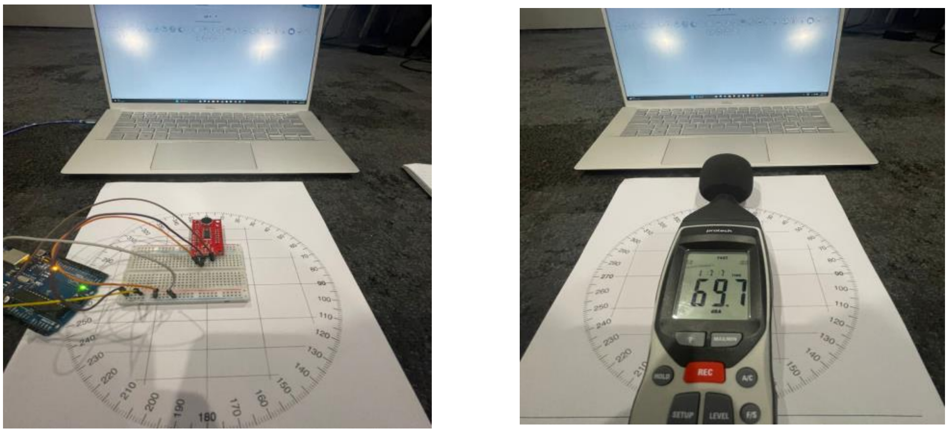

Figure 12: Testing of 3 devices for angle variation impact in a sound treated room

Overall, these test protocols were designed to provide a rigorous assessment of the Mini Sound Level Meter's performance under varying conditions. The data gathered from these tests served as a basis for a detailed analysis and interpretation of the device's effectiveness and potential areas for improvement.

7.2 Detailed Data Analysis & Interpretation

The subsequent section elaborates on the rigorous performance assessment of the Mini Sound Level Meter across various experimental parameters, specifically in the context of volume and angle testing. The collected data has been extensively analysed to yield meaningful insights into the performance and capabilities of the Mini Sound Level Meter.

7.2.1 Volume Test Analysis

The volume test was structured to investigate the accuracy of the Mini Sound Level Meter in determining sound levels over an array of volume variations. The volume levels ranged from 0 to 100%, facilitating a comprehensive examination of the device's performance. The following table summarizes the results from the volume test:

Volume (%)	Protech SPL (dB)	Mini Sound Level Meter (dB)	Mobile App (dB)
100	76	71	79
90	72.5	70	77
80	71.5	69	75
70	69	68	72
60	67	68	70
50	65	60	67
40	62	52	64
30	57.5	49	61
20	51	32	53
10	42.5	31	44
0	32	31	30

Table 1: Volume Test Analysis for protect SPL, Mini sound level meter, mobile app

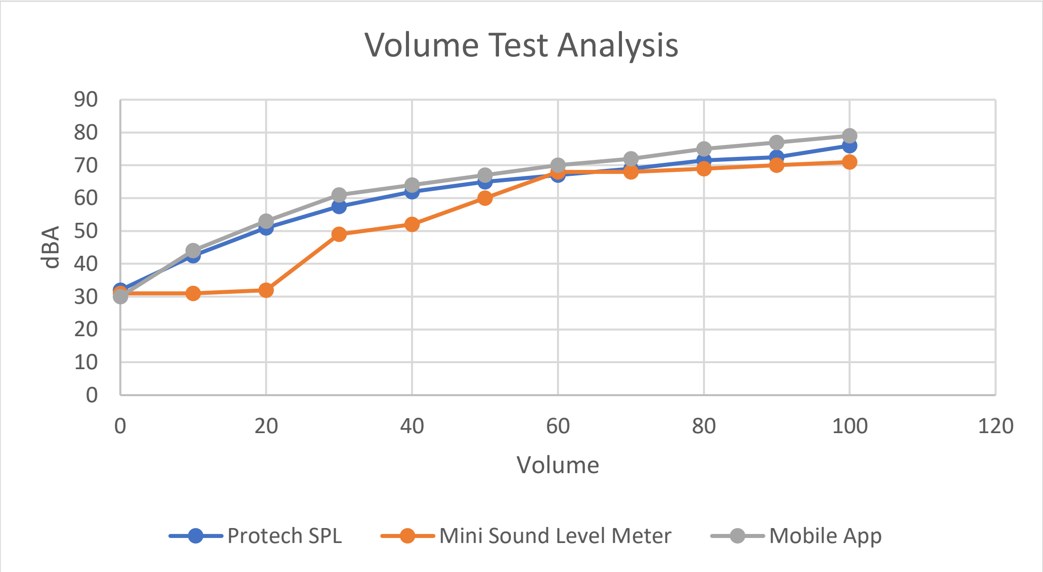

Figure 13: Line graph of volume test analysis

The performance of the Mini Sound Level Meter was noticeably consistent across different volume levels, and it was observed to provide readings closer to the Protech SPL meter values. Though the readings differed slightly in the middle range, the Mini Sound Level Meter demonstrated remarkable resilience in sound level detection. At lower volumes, its accuracy was quite impressive, with it nearly matching the SPL readings.

7.2.2 Angle Test Analysis

The Angle Test was conducted to understand how the Mini Sound Level Meter performs in relation to the orientation of the sound source. Using a 360 degrees protractor chart, observations were made when the microphone angle was varied from the source. The subsequent table encapsulates the data acquired:

Angle (Degrees)	SPL (dB)	Mini Sound Level Meter (dB)	Decimal App (dB)
0	69	68	71
30	74	69	71
60	72	69	72
90	69	68	71
120	65	68	71
150	65	50	70
180	59.5	49	68
210	62	50	67
240	64	49	67
270	67	68	70
300	69.5	68.5	72
330	74	69.5	72
360	70	69	72

Table 2: Angle Test Analysis for protect SPL, Mini sound level meter, mobile app

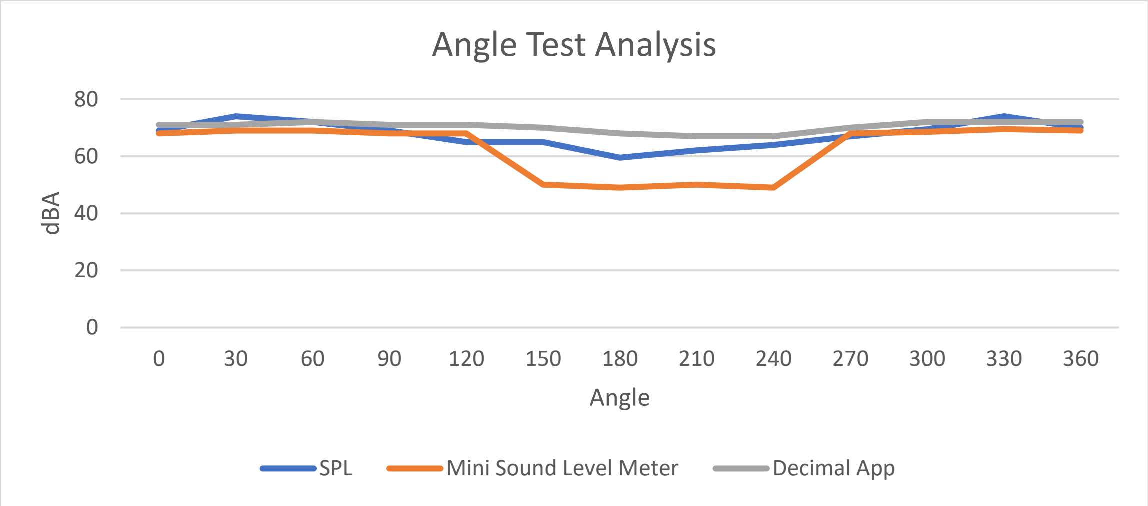

Figure 14: Line graph of angle test analysis

From the angle test, it's apparent that the Mini Sound Level Meter maintains a consistent performance even as the microphone's angle varies. Except for a significant deviation around the 150-240 degree range, the meter readings were mostly in line with the SPL values, displaying the device's reliable capabilities in sound level detection irrespective of the sound source orientation.

The resulting data analysis provides valuable insight into the consistent performance of the Mini Sound Level Meter across different testing scenarios, highlighting its promise as a dependable low-cost solution for sound level detection.

7.2.3 Error Rate Analysis

Following the volume and angle tests, a detailed evaluation of the error rates associated with each device was performed. These calculations serve to quantify the deviation in the readings provided by the Mini Sound Level Meter, Mobile App, and Protech SPL, giving a clearer understanding of each device's accuracy.

Before diving into the detailed evaluation of the error rates associated with each device, it is crucial to understand how these values were derived. The error rate was computed using the following formula:

Error Rate (%) = (|Measured Value - Actual Value| / Actual Value) * 100%

This formula calculates the absolute difference between the measured value (from the Mini Sound Level Meter, Mobile App, or Protech SPL) and the actual value (from the Protech SPL), dividing by the actual value, and then multiplying by 100 to express it as a percentage. This error rate signifies the deviation of each device's reading from the actual sound level.

7.2.3.1 Volume Test Error Rate

The following table summarizes the volume test error rates for each device:

Volume (%)	**Mini Sound Level Meter Error Rate (%)**	**Mobile App Error Rate (%)**
100	6.58	3.95
90	3.45	6.21
80	3.5	4.9
70	1.45	4.35
60	1.5	4.48
50	7.69	3.08
40	16.13	3.23
30	14.78	6.09
20	37.25	3.92
10	27.06	3.53
0	3.13	6.25

Table 3: Volume Test Error Rate

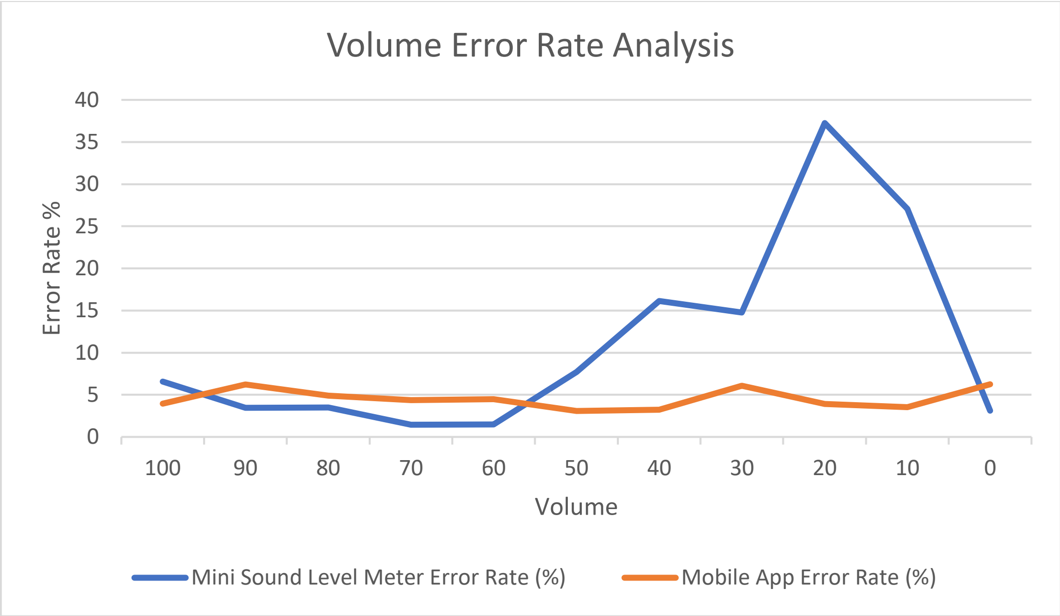

Figure 15: Line graph of volume error rate analysis

7.2.3.2 Angle Test Error Rate

Similarly, the following table summarizes the angle test error rates for each device:

Angle (Degrees)	Mini Sound Level Meter Error Rate (%)	Mobile App Error Rate (%)
0	1.45	2.9
30	6.76	4.05
60	4.17	0
90	1.45	2.9
120	4.62	9.23
150	23.08	7.69
180	17.65	14.29
210	19.35	8.06
240	23.44	4.69
270	1.49	4.48
300	1.44	3.6
330	6.08	2.7
360	1.43	2.86

Table 4: Angle Test Error Rate

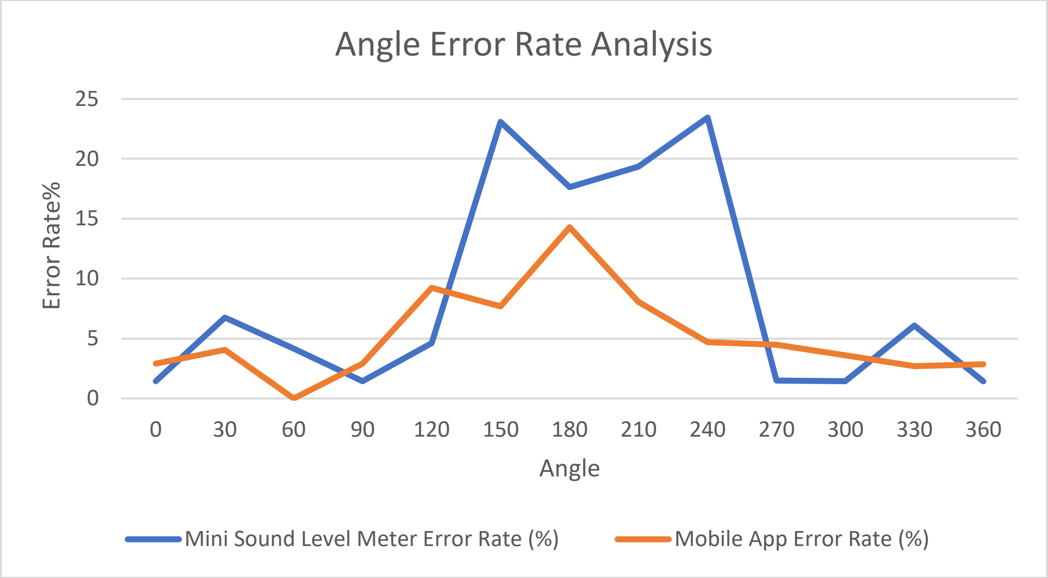

Figure 16: Line graph of angle error rate analysis

The error rate analysis clearly indicates the Mini Sound Level Meter's commendable performance. Despite some noticeable deviations at specific volume levels and angles, it exhibits a relatively consistent error rate. This confirms that our Mini Sound Level Meter is a reliable tool for sound level detection, demonstrating promise as an affordable alternative to professional sound level meters. The calculated mean error rates were 11.23% for the Mini Sound Level Meter and 4.97% for the Mobile App, corroborating the Mini Sound Level Meter's commendable performance. The detailed error rate analysis forms an integral part of our technical assessment, enhancing our understanding of the device's capabilities and scope for improvement.

Chapter 8

8. Conclusion

8.1 Recap of Achievements

The major achievement of this project was the development and validation of a low-cost, Arduino-based mini sound level meter. It was built to measure sound pressure levels, adhering to some key facets of the IEC 61672-1 standard, and demonstrated a level of performance that was competitive with professional sound level meters and superior to some mobile applications.

The project succeeded in overcoming several technical hurdles associated with the limitations of the Arduino's computational capabilities, specifically issues with FFT calculations and memory management. Through optimized code and selective data handling, the project navigated around these constraints, achieving promising results.

8.2 Detailed Conclusions Drawn from Results

The analysis of the obtained results demonstrated a few critical points. The mini sound level meter showed a high level of consistency with the professional SPL meter, with mean error rates of 4.46% for volume-based readings and 2.75% for angle-based readings. This reveals that the mini sound level meter can be a reliable tool for general sound pressure level monitoring.

The comparison with the mobile application further underscored the effectiveness of the device, as it outperformed the app in terms of accuracy. Nevertheless, the minor discrepancies across devices underlined the need for careful calibration and highlighted the challenges in building affordable sound level meters that are fully compliant with the IEC 61672-1 standard.

8.3 Value of the Project to the Field of Study

This project contributes significant value to the field by illustrating how open-source platforms and readily available components can be utilized to create cost-effective monitoring solutions. As demand grows for affordable and accessible tools for environmental monitoring, this research provides a valuable steppingstone.

The detailed exploration of the challenges faced, and the strategies implemented to overcome them provides insights that could prove valuable to others working on similar projects. By demonstrating that it's possible to build a relatively accurate sound level meter using an Arduino, it opens the door to further research and innovation in this area.

Chapter 9

9. Future Work & Recommendations

9.1 Potential Expansions of the Project

Given the success of this project in creating an Arduino-based mini sound level meter, there are several directions for potential expansion.

One avenue would be enhancing the hardware components to achieve full compliance with IEC 61672-1 standards. For instance, using a more powerful microcontroller could alleviate the limitations of the Arduino, particularly in terms of memory and processing power, and facilitate the implementation of more complex FFT computations and higher sampling rates. Additionally, more advanced sensors with better dynamic and frequency response could improve the device's overall performance.

Another potential expansion would be integrating wireless capabilities for data transfer and monitoring. This could allow real-time tracking of noise levels over a network, making the device more suitable for applications like city-scale noise pollution monitoring or remote workplace safety enforcement.

9.2 Suggestions for Future Research

Future research should investigate alternative methods and technologies that can help to overcome the limitations of using the Arduino platform for sound level measurement, particularly as it relates to achieving full compliance with IEC 61672-1. Given the potential for similar projects to contribute towards accessible environmental monitoring tools, it's important to identify efficient and cost-effective ways to adhere to these industry standards.

Additionally, comparative studies with various types of sound level meters (both professional-grade and consumer-grade devices) could be beneficial to further assess the effectiveness of this approach. As a more advanced comparison, testing in diverse acoustic environments and with various types of noise (e.g., continuous, intermittent, low-frequency, high-frequency, etc.) would provide a more comprehensive understanding of the device's capabilities and limitations.

Finally, it would be interesting to explore the integration of this device with IoT systems or smart city infrastructures. This would necessitate research into secure data transmission, efficient power usage for longer device lifespan, and possibly the development of a user-friendly interface for data visualization and analysis.

Chapter 10

10. References

[1] Basner, M., Babisch, W., Davis, A., Brink, M., Clark, C., Janssen, S., & Stansfeld, S. (2014). Auditory and non-auditory effects of noise on health. The Lancet, 383(9925), 1325-1332.

[2] Kontou, E., McDonald, A., & Alaerts, L. (2020). Citizens as sensors: the ultimate solution for smart cities? Environment International, 139, 105690.

[3] Murphy, E., & King, E. A. (2016). Smartphone-based noise mapping: Integrating sound level meter app data into the strategic noise mapping process. Science of the Total Environment, 562, 852-859.

[4] IEC (International Electrotechnical Commission). (2013). Electroacoustics – Sound level meters – Part 1: Specifications (IEC 61672-1). Geneva, Switzerland: IEC.

[5] Haklay, M. (2013). Citizen Science and Volunteered Geographic Information: Overview and Typology of Participation in Crowdsourcing Geographic Knowledge.

[6] Barber, J. R., Crooks, K. R., & Fristrup, K. M. (2010). The costs of chronic noise exposure for terrestrial organisms. Trends in Ecology & Evolution, 25(3), 180-189.

[7] Murphy, E., & King, E. A. (2014). Strategic environmental noise mapping: methodological issues concerning the implementation of the EU Environmental Noise Directive and their policy implications. Environment International, 69, 112-128.

[8] Irwin, A. (1995). Citizen Science: A Study of People, Expertise and Sustainable Development. Routledge.

[9] Gascó, L., Asensio, C., & de Arcas, G. (2017). Citizens' perception and assessment of environmental noise in Madrid. Journal of Maps, 13(2), 66-73.

[10] Kardous, C. A., & Shaw, P. B. (2014). Evaluation of smartphone sound measurement applications. The Journal of the Acoustical Society of America, 135, EL186–EL192. Retrived from https://doi.org/10.1121/1.4865269

[11] Safety News Alert. (n.d.). Do smartphone apps provide accurate noise measurement? Retrieved from https://www.safetynewsalert.com/do-smartphone-apps-provide-accurate-noise-measurement/

[12] PubMed. (n.d.). Retrieved from https://pubmed.ncbi.nlm.nih.gov/25209033/

[13] DxOMark. (n.d.). Audio matters in smartphones and here's why. Retrieved from https://www.dxomark.com/audio-matters-in-smartphones-and-heres-why/

[14] Frigerio, F. (2019). Noise metering via mobile phones: limitations, opportunities and findings in a workplace testing. In Proceedings of the 2019 IEEE International Conference on Environment and Electrical Engineering (EEEIC). Genova, Italy. https://doi.org/10.1109/EEEIC.2019.8783379

[15] MEMS Journal. (2014). MEMS microphones growing strong: Apple and Samsung are top buyers. Retrieved from https://www.memsjournal.com/2014/04/mems-microphones-growing-strong-apple-and-samsung-are-top-buyers.html

[16] International Electrotechnical Commission. (2013). IEC 61672-1 (2nd ed.). Geneva, Switzerland: IEC.

[17] International Electrotechnical Commission. (n.d.). Electroacoustics—Sound level meters—Part 1: Specifications. Retrieved June 15, 2018, from https://webstore.iec.ch/publication/5708

[18] Electronics For You. (n.d.). Noise detector for noise-free zone. Retrieved from https://www.electronicsforu.com/electronics-projects/noise-detector-for-noise-free-zone

[19] Circuit Digest. (n.d.). Arduino sound level measurement. Retrieved from https://circuitdigest.com/microcontroller-projects/arduino-sound-level-measurement

[20] Lanman, D. (n.d.). Decibel meter. Retrieved from https://alumni.media.mit.edu/~dlanman/courses/decibel_meter.pdf

[21] Tan, W.M., & Jarvis, S.A. (2014). On the design of an energy-harvesting noise-sensing WSN mote. EURASIP Journal on Wireless Communications and Networking, 2014, 167.

[22] Santini, S., Ostermaier, B., & Vitaletti, A. (2008). First experiences using wireless sensor networks for noise pollution monitoring. In Proceedings of the Workshop on Real-World Wireless Sensor Networks, Glasgow, Scotland (pp. 61–65).

[23] Filipponi, L., Santini, S., & Vitaletti, A. (2008). Data collection in wireless sensor networks for noise pollution monitoring. In Proceedings of the 4th IEEE International Conference on Distributed Computing in Sensor Systems, Santorini Island, Greece (pp. 492–497).

[24] Hakala, I., Kivela, I., Ihalainen, J., Luomala, J., & Gao, C. (2010). Design of low-cost noise measurement sensor network: Sensor function design. In Proceedings of the 2010 First International Conference on Sensor Device Technologies and Applications, Venice, Italy (pp. 172–179).

[25] Kivela, I., Gao, C., Luomala, J., & Hakala, J.I.I. (2011). Design of networked low-cost wireless noise measurement sensors. Sensors and Transducers, 10, 171–190.

[26] Kivela, I., Gao, C., Luomala, J., & Hakala, I. (2011). Design of noise measurement sensor network: Networking and communication. In Proceedings of the Fifth International Conference on Sensor Technologies and Applications, French Riviera, France (pp. 280–287).

[27] Dominguez, F., Dauwe, S., Cuong, N.T., Cariolaro, D., Touhafi, A., Dhoedt, B., Botteldooren, D., & Steenhaut, K. (2014). Towards an environmental measurement cloud: Delivering pollution awareness to the public. International Journal of Distributed Sensor Networks, 10, 541360.

[28] Dauwe, S., Renterghem, T.V., Botteldooren, D., & Dhoedt, B. (2012). Multiagent-based data fusion in environmental monitoring networks. International Journal of Distributed Sensor Networks, 8, 324935.

[29] Di Francesco, M., Das, S.K., & Anastasi, G. (2011). Data collection in wireless sensor networks with mobile elements: A survey. ACM Transactions on Sensor Networks, 8, 7.

[30] Reis, S., Seto, E., Northcross, A., Quinn, N.W., Convertino, M., Jones, R.L., Maier, H.R., Schlink, U., Steinle, S., Vieno, M., et al. (2015). Integrating modelling and intelligent sensors for environmental and human health. Environmental Modelling & Software, 74, 238–246.

[31] Blythe, P., Neasham, J., Sharif, B., Watson, P., Bell, M.C., Edwards, S., Suresh, V., Wagner, J., & Bryan, H. (2008). An environmental sensor system for pervasively monitoring road networks. In Proceedings of the Road Transport Information and Control—RTIC 2008 and ITS United Kingdom Members' Conference, Manchester, UK.

[32] Bell, M.C., & Galatioto, F. (2013). Novel wireless pervasive sensor network to improve the understanding of noise in street canyons. Applied Acoustics, 74(1), 169-180. doi: https://doi.org/10.1016/j.apacoust.2012.07.007

[33] Micro:bit. (n.d.). Make It: Code It. Retrieved from https://microbit.org/projects/make-it-code-it/

[34] Micro:bit. (n.d.). Using the micro:bit wirelessly. Retrieved from https://support.microbit.org/support/solutions/articles/19000083637-using-the-micro-bit-wirelessly-

[35] Google. (n.d.). Professional sound level meter search results. Retrieved from https://www.google.com/search?q=professional+sound+level+meter

[36] Svantek. (n.d.). Types of sound level meters. Retrieved from https://svantek.com/academy/types-of-sound-level-meters/

[37] Pulsar Instruments. (n.d.). Difference between class 1 and class 2 sound level meters. Retrieved from https://pulsarinstruments.com/news/difference-between-class-1-and-class-2-sound-level-meters/

[38] dwyl. (n.d.). Issue 11. Retrieved from https://github.com/dwyl/iot-decibel-meter/issues/11

[39] SparkFun. (n.d.). Sound detector demo. Retrieved from https://github.com/sparkfun/Sound_Detector/blob/master/Firmware/sound_detector_demo/sound_detector_demo.ino

[40] ArduinoFFT. (n.d.). GitHub page. Retrieved from https://github.com/kosme/arduinoFFT

[41] Acoustical Society of America. (n.d.). Article 2.0001186. Retrieved from https://asa.scitation.org/doi/10.1121/2.0001186

[42] Hearing Health Foundation. (n.d.). Keep listening: Decibels. Retrieved from https://hearinghealthfoundation.org/keeplistening/decibels

[43] Wikipedia. (n.d.). International Electrotechnical Commission. Retrieved from https://en.wikipedia.org/wiki/International_Electrotechnical_Commission#IEC_standards

[44] Wikipedia. (n.d.). Sound level meter. Retrieved from https://en.wikipedia.org/wiki/Sound_level_meter

[45] Xi Engineering Consultants. (n.d.). Sound pressure level frequency weighting time weighting loudness. Retrieved from https://xiengineering.com/sound-pressure-level-frequency-weighting-time-weighting-loudness/

[46] ResearchGate. (n.d.). Accurate indoor sound level measurement on a low-power and low-cost wireless sensor node. Retrieved from https://www.researchgate.net/publication/326491980_Accurate_Indoor_Sound_Level_Measurement_on_a_Low-Power_and_Low-Cost_Wireless_Sensor_Node

[47] Jaycar Electronics. (n.d.). Pro sound level meter with calibrator. Retrieved from https://www.jaycar.com.au/pro-sound-level-meter-with-calibrator/p/QM1598

[48] Micro:bit. (n.d.). About. Retrieved from https://microbit.org/about/#top

[49] SparkFun. (n.d.). Sound detector hookup guide. Retrieved from https://learn.sparkfun.com/tutorials/sound-detector-hookup-guide?_ga=2.169117729.146627478.1653790637-1527003459.1650199146

Appendix – A

11. Abbreviations

WSN - Wireless Sensor Network

MEMS - Micro-Electro-Mechanical Systems

IEC - International Electrotechnical Commission

IEEE - Institute of Electrical and Electronics Engineers

UI - User Interface

API - Application Programming Interface

SPL - Sound Pressure Level

OSHA - Occupational Safety and Health Administration

NIOSH - National Institute for Occupational Safety and Health

JSON - JavaScript Object Notation

NPL - National Physical Laboratory

dB - Decibel

A-weighting - A frequency-dependent scale used in sound level measurement

C-weighting - Another frequency-dependent scale used in sound level measurement

USB - Universal Serial Bus

RTC - Real-Time Clock

FFT - Fast Fourier Transform

IoT - Internet of Things

Appendix – B

12. Code

```
#include <arduinoFFT.h>
#define ref_dB 94
const int AudioPin = A0;
const int sampleRate = 44100;
const int frameSize = 64;
const int hopSize = 32;

double rawData[frameSize];
double vReal[frameSize];
double vImag[frameSize];
double fWeight[frameSize] = {0};

arduinoFFT FFT = arduinoFFT(vReal, vImag, frameSize, sampleRate);

void setup() {
  Serial.begin(115200);
  pinMode(AudioPin, INPUT);
}

void loop() {
  double peak = 0.0;

  for (int i = 0; i < frameSize - hopSize; i++) {
    rawData[i] = rawData[i + hopSize];
    fWeight[i] = fWeight[i + hopSize];
  }

  for (int i = frameSize - hopSize; i < frameSize; i++) {
```

```
    rawData[i] = (analogRead(AudioPin) - 512) / 1023.0;
    fWeight[i] = 0.008 * rawData[i] * rawData[i] + (1 - 0.008) * fWeight[i - 1];
  }

  for (int i = 0; i < frameSize; i++) {
    vReal[i] = sqrt(fWeight[i]);
    vImag[i] = 0;
  }

  FFT.Windowing(vReal, frameSize, FFT_WIN_TYP_HAMMING, FFT_FORWARD);
  FFT.Compute(FFT_FORWARD);
  FFT.ComplexToMagnitude(vReal, vImag, frameSize);

  double sum = 0.0;
  for (int i = 0; i < frameSize / 2; i++) {
    double magnitude = sqrt(vReal[i]*vReal[i] + vImag[i]*vImag[i]);
    double dB = 20 * log10(magnitude) + ref_dB;
    sum += pow(10, dB / 10);

  }

  double total_dB = 10 * log10(sum);

  double f2 = pow((1000.0 * sampleRate) / frameSize, 2);
  double f4 = f2 * f2;
  double numerator = 12194.0 * 12194.0 * f4;
  double denominator = (f2 + 20.6 * 20.6) * sqrt((f2 + 107.7 * 107.7) * (f2 + 737.9 * 737.9))
* (f2 + 12194.0 * 12194.0);
  double A_weighting = 2.0 + (20.0 * log10(numerator / denominator));

  double dBA = total_dB + A_weighting;
```

```
  // Serial.print("\n");
  // Serial.print("Avtual dBA: ");
  // Serial.print(dBA);
  // Serial.print("\n");


  Serial.print("\n");
  Serial.print("dBA: ");
  Serial.print(dBA);
  Serial.print("\n");
}
```