LONG-TERM, LOW-COST ACOUSTIC MONITOR

Student Name:	Mr Jonathan Paul Hendricks
Student Number:	20332947
Study Leader:	Professor Jaco Versfeld
Date:	September 2020

UNIVERSITEIT
iYUNIVESITHI
STELLENBOSCH
UNIVERSITY

Report submitted in partial fulfilment of the requirements of the module Project (E) 448 for the degree Baccalaureus in Engineering in the Department of Electrical and Electronic Engineering at the University of Stellenbosch

Acknowledgements

Thank you to my study leader, Professor Jaco Versfeld. Your guidance and support over the three years of receiving knowledge through your lectures and personal discussions aided to the completion of undertaking of this project. Thank you for letting me solve this problem individually and help me gain confidence before stepping into the real world. Thank you to Hilton Johnson for providing the necessary lab equipment and components needed. Thank you Jenny Martin for ordering the hardware required for this project. Thank you Johan Booysen and Anneke Bester in assisting me with getting access to the anechoic chamber. Thak you Dr Danie Ludick for providing information on common-impedance problems. I would also like to thank the personal friends and family in my life who supported me throughout my engineering degree and my final year. Thank you to my parents who provided me with the necessary funds to receive an education whilst living comfortably. Thank you Mike Brewis, you have inspired me to believe I am worth more than what I sometimes believe. Thank you to my girlfriend, My Nguyen, for pushing me to excel since the day I met you. Lastly, to my fallen brother, Ashwill Boer. You did not make it, but your spirit carries on in all of us who have struggled this year especially with Covid-19. May you rest in peace.

UNIVERSITEIT·STELLENBOSCH·UNIVERSITY
jou kennisvennoot · your knowledge partner

Plagiaatverklaring / Plagiarism *Declaration*

1 Plagiaat is die oorneem en gebruik van die idees, materiaal en ander intellektuele eiendom van ander persone asof dit jou eie werk is.
Plagiarism is the use of ideas, material and other intellectual property of another's work and to present is as my own.

2 Ek erken dat die pleeg van plagiaat 'n strafbare oortreding is aangesien dit 'n vorm van diefstal is.
I agree that plagiarism is a punishable offence because it constitutes theft.

3 Ek verstaan ook dat direkte vertalings plagiaat is.
I also understand that direct translations are plagiarism.

4 Dienooreenkomstig is alle aanhalings en bydraes vanuit enige bron (ingesluit die internet) volledig verwys (erken). Ek erken dat die woordelikse aanhaal van teks sonder aanhalingstekens (selfs al word die bron volledig erken) plagiaat is.
Accordingly all quotations and contributions from any source whatsoever (including the internet) have been cited fully. I understand that the reproduction of text without quotation marks (even when the source is cited) is plagiarism.

5 Ek verklaar dat die werk in hierdie skryfstuk vervat, behalwe waar anders aangedui, my eie oorspronklike werk is en dat ek dit nie vantevore in die geheel of gedeeltelik ingehandig het vir bepunting in hierdie module/werkstuk of 'n ander module/werkstuk nie.
I declare that the work contained in this assignment, except where otherwise stated, is my original work and that I have not previously (in its entirety or in part) submitted it for grading in this module/assignment or another module/assignment.

20332947	JHendricks
J.P Hendricks	**14 September 2020**

Summary

Passive acoustic monitoring (PAM) in oceans can be used to monitor marine animal vocalisations through the use of passive acoustic monitors. This project aims to investigate the development of a long-term, low-cost acoustic monitor to record dolphin sounds in False Bay. Knowledge of acoustical waves and its properties are required to interpret results to make an effort to improve the system designed. The acoustic monitor developed requires the gain to be varied during real-time application, provide low noise levels on board, and record during specific time frames set in software. Ultimately, the device could not be deployed into the ocean. Experiments were performed through simulating the acoustic monitor with an electret microphone in an anechoic chamber, and in an aquatic environment such as a swimming pool. The results conclude that the device may be suitable for long-term PAM, and improvements on the system is recommended.

Opsomming

Passiewe akoestiese monitering in oseane kan gebruik word om diere wat in die oseaan lewe se klankvorms te monitor deur gebruik te maak van passiewe akoestiese monitors. Hierdie projek se doelwit is om ondersoek in te stel na die ontwikkeling van ‘n langtermyn, lae koste akoestiese monitor wat dolfyn klanke in Valsbaai kan opneem. Kennis van akoestiese golwe en hul eienskappe is benodig om uitslae te interpreteer en ontleed ten einde die sisteem wat ontwerp is te verbeter. Die akoestiese monitor wat ontwikkel is benodig dat die wins deur (real-time) aanwending gevarieer moet word,dit moet lae vlakke van geraas aanboord kan aanbied en moet kan opneem gedurende spesifieke tydrame wat in sagteware vasgelê is. Uiteindelik kon die toestel nie in die oseean ontplooi word nie. Eksperimente is uitgevoer deur die akoestiese monitor te simuleer met ‘n elektret mikrofoon in ‘n anekiese kamer asook in ‘n akwatiese omgewing soos ‘n swembad. Die uitslae bevestig dat die toestel wel geskik is vir langtermyn passiewe akoestiese monitering en verbeterings op die sisteem is aanbeveel.

Contents

List of Figures VII

List of Tables VIII

List of Abbreviations IX

1 Introduction 1
- 1.1 Problem statement . . . 1
- 1.2 Objectives . . . 1
  - 1.2.1 Data logger . . . 1
  - 1.2.2 Varying the gain . . . 1
- 1.3 Scope . . . 2
- 1.4 Summary of work . . . 2

2 Literature Study 3
- 2.1 Underwater Soundscape Monitoring . . . 3
  - 2.1.1 Fish Acoustic Monitoring . . . 3
  - 2.1.2 Long-term PAM methods . . . 3
- 2.2 Software tools for system identification . . . 4
  - 2.2.1 Audacity . . . 4
  - 2.2.2 Matlab . . . 4
  - 2.2.3 Arduino Software (IDE) . . . 4
- 2.3 Acoustics fundamentals . . . 4
  - 2.3.1 Condenser Microphone . . . 5
  - 2.3.2 Audio Waveform . . . 5
  - 2.3.3 The Decibel . . . 7
  - 2.3.4 Gain of the system . . . 8
  - 2.3.5 Common-impedance coupling . . . 9
  - 2.3.6 Acoustic Test Facilities . . . 10
  - 2.3.7 Acoustics Test and Measurement Practices . . . 11
- 2.4 Passive Acoustic Monitoring of Underwater Sound . . . 11
  - 2.4.1 Hydrophone and Sound Pressure Levels . . . 11
  - 2.4.2 Sound propagation and transmission loss . . . 12
- 2.5 Signal Processing . . . 13
  - 2.5.1 Signal-to-noise ratio and Dynamic Range . . . 13
  - 2.5.2 Total Harmonic Distortion Plus Noise (THD+N) . . . 14
  - 2.5.3 Notch Filter Design . . . 14

3 Methodology 16
- 3.1 Design Requirements . . . 16
  - 3.1.1 ARM Cortex-M Processor Family and MCU . . . 16
  - 3.1.2 Audio Codec . . . 17
  - 3.1.3 Transducer input . . . 17

3.1.4 Data Storage . . . 17
3.1.5 Power . . . 17
3.1.6 Scheduling Operation . . . 18
3.1.7 Testing practice . . . 18

4 System Design 19
4.1 System Overview . . . 19
4.2 Hardware Design . . . 20
4.2.1 Microcontroller Unit . . . 20
4.2.2 Audio Adapter Board . . . 21
4.2.3 Electret Microphone . . . 21
4.2.4 Hydrophone . . . 22
4.2.5 Micro SD Card . . . 22
4.2.6 Latching Switch . . . 22
4.2.7 Power source . . . 22
4.2.8 Jumper connections for LININ/LINOUT . . . 23
4.2.9 Diode . . . 23
4.2.10 Coin cell battery . . . 23
4.2.11 Ferrite bead . . . 23
4.2.12 Waterproof casing . . . 23
4.3 Software Design . . . 24
4.3.1 Audio System Design Tool . . . 25
4.3.2 Varying Gain . . . 26
4.4 System Testing . . . 27
4.4.1 Oscilloscope . . . 27
4.4.2 Anechoic Chamber Testing . . . 28
4.4.3 Swimming pool . . . 29
4.4.4 Signal Generator . . . 29
4.4.5 Varying the gain . . . 30

5 Results 31
5.1 Ground voltages from oscilloscope . . . 31
5.2 Anechoic Chamber Testing for Microphone Self-Noise . . . 32
5.2.1 Software related issues and low frequency components . . . 32
5.3 Signal Generator Test . . . 33
5.3.1 THD+N . . . 34
5.3.2 SNR . . . 34
5.4 Varying Gain . . . 35
5.5 Hydrophone Results . . . 36

I Conclusion 37
5.5.1 Recommendations for the future . . . 38

References 39

Appendix A: Project planning schedule 41

Appendix B: Outcomes compliance 42

Appendix D: Final Design Project Cost **43**

Appendix E: Code **44**

List of Figures

2.1 Visualising the crest factor 6
2.2 Visualising the crest factor in Audacity 7
2.3 Common-impedance coupling circuit 9
2.4 Anechoic chamber Stellenbosch University 10
2.5 Dynamic Range 13
2.6 Poles and Zeros of a Notch Filter 15
3.1 ARM Processor Family 16
4.1 System Overview 19
4.2 Acoustic monitor final design and its peripherals 20
4.3 Waterproof case 24
4.4 Scheduling operation overview 24
4.5 Audio System Design Tool for Teensy Audio Library 25
4.6 Peak detection set in software 26
4.7 Ground Loop Isolator 27
5.1 Ground voltages of the acoustic monitor 31
5.4 THD+N for initial and final design 34
5.6 Transition in peak-to-peak voltage resulting in clipping 35
5.7 Serial monitor display when a peak is detected 36

List of Tables

2.1 Frequency range of sound sources affecting marine animals and recordings . 4
4.2 Microcontroller comparison 20
4.3 Microphone comparison 21
4.4 Hydrophone characteristics 22
5.5 Project Planning Schedule 41
5.6 Exit level outcome (ELO) assessments 42
5.7 Component list for final design 43

List of Abbreviations

3D . . . Three-dimensional
ADC . . . Analog to Digital Converter
Ah . . . Amp hour
ARM . . . Advanced RISC Machines
CPU . . . Central Processing Unit
dB . . . Decibel
Fs . . . Sampling frequency
Hz . . . Hertz
I . . . Current
I/O . . . Input/output
I2C . . . Inter Integrated-circuit Communication
I2S . . . Inter Integrated-circuit Sound
k . . . kilo
MCU . . . Microcontroller unit
OS . . . Operating System
P . . . Power
PAM . . . Passive Acoustic Monitoring
R . . . Resistance
RC . . . resistor-capacitor
RF . . . Radio Frequency
RL . . . Received Level
RMS . . . Root Mean Squared
RTC . . . Real-time clock
RX . . . Receive
SD . . . Secure Digital
SL . . . Source Level
SPI . . . Serial Peripheral Interface
SPL . . . Sound Pressure Level
SNR . . . Signal-to-Noise Ratio
SoC . . . System on Chip
SPI . . . Serial Peripheral Interface
THD+N . . . Total Harmonic Distortion Plus Noise
TX . . . Transmit
V . . . Voltage
WAV . . . Waveform Audio File Format
Z . . . Impedance

1 Introduction

In order to grasp the content of this report, please refer to the list of contents especially with regards to symbols and abbreviations used. Since a digital version of this report is submitted only, this report is set out for easy navigation by making use of the internal links to sections, figures, equations, references, tables or the Appendices.

1.1 Problem statement

Passive acoustic monitoring (PAM) of underwater sources such as oceans, lakes and streams is a practice where marine fauna is observed. This project aims to develop a low-cost, long-term acoustic monitor in order to measure dolphin sounds as implied by the title of this report. Furthermore, it aims to monitor the effects of noise on marine fauna present in the ecosystem, and allow for variation of the gain during real-time application.

The complexity surrounding this task is that knowledge in underwater acoustics and general small room acoustics is required. For a device to measure and record data successfully, it also has to have minimal noise present in the system. A proper gain structure is needed, since audio waveforms are generally small in magnitude, and produce low voltages. Therefore, it can already be established that a system is required which can maximise the gain in order to produce an optimal voltage level signal to receive the recording at the highest volume whilst not amplifying accompanying noise.

1.2 Objectives

The objectives may be narrowed down from the problem statement, as certain conclusions were already made. Therefore, two main criteria are specified to to measure the adequate performance and success of the acoustic monitor development.

1.2.1 Data logger

The first objective is for the passive acoustic monitor to be able to record and log audio data in a desired manner. This will include minimal noise recordings and software being developed in order to save the data with the correct data and time. Audio data needs to be logged during intervals and record for fixed lengths of time.

1.2.2 Varying the gain

The second objective is to be able to detect voltage peaks during real-time in the recordings due to noise or loud sound sources, and vary the gain accordingly. Therefore, the gain should be set at its optimal gain value to amplify low input signals during normal operation, but decrease whenever a disturbance is detected.

1.3 Scope

The nature of the project considers a low-cost device, which could be applied in order to monitor dolphin sounds and other marine fauna. Therefore, the project is limited by obtaining equipment for extensive tests. These results could lead to further verification and data in order to better the system during the design phase. Basic field-tests are done to characterize the system. Equipment which could aid to the project to do more analysis on the device before deployment is mentioned throughout, and a study is done on relevant topics for the development and testing of an acoustic monitor.

1.4 Summary of work

The procedure of investigating the development of an acoustic monitor is demonstrated throughout the following chapters. A literature review is completed to investigate current PAM practices, acoustical signals and its various properties. Furthermore, the literature review contains information regarding various topics of acquired knowledge and knowledge required to understand and interpret results. The methodology is then provided to showcase the design requirements and testing practice which should be used when developing an acoustic monitor. The methodology precedes a detailed system design, where hardware and software is developed. Metrics of the acoustic monitor are considered and then measured. Finally, a conclusion is made based on the results obtained and recommendations for the future are made. Further information regarding the project and acoustic monitor developed is provided in the appendices.

2 Literature Study

2.1 Underwater Soundscape Monitoring

Soundscape ecology is used to study the acoustic relationships between living organisms, human and other, and their environment [1]. This allows for information to be gathered on marine fauna and the habitats in which they reside by the use of PAM on underwater soundscapes. Furthermore, it creates the opportunity for PAM to describe acoustical waves present underwater to monitor health, biodiversity, spawning patterns, and other biological patterns [2].

2.1.1 Fish Acoustic Monitoring

Fisheries science often need to determine where fish spawn. The discovery that fish produce certain sounds when spawning has led to the development of a ”spawn-o-meter”, which detects and counts these specific sounds over time [3]. This provides important data on habitats utilised by different species, and can aid to the management of of these habitats, especially where the habitat and species are vulnerable to anthropogenic noise. Anthropogenic noise continues to rise and has become a global concern since it masks the communication of fish and marine animals. The ambient noise of a system may influence when and how fish communicate. Increased background noise may interfere with communication between marine life and how they behave.

The fish vocalisations consist of short pulses or grunts lasting a few seconds to a few milliseconds. The best studies made are on coral reefs, where the water is clear, and sounds made by different biotic species are often heard by scuba divers. Other quieter vocalisations may only be heard through the use of a hydrophone to monitor these sounds.

2.1.2 Long-term PAM methods

Omnidirectional hydrophones are implemented with the appropriate hardware and software developed to produce automatic acoustic recorders with high sensitivities. This sensitivity needs to be calibrated with regards to the intended source to be measured. Studies which do not calibrate their hydrophone systems often show why this is necessary. The acoustic data gathered is mainly limited through battery life and storage. The challenges faced are the vast amount of biotic sound present underwater. Waves breaking, wind, rainfall, shipping noise and snapping shrimp can sometimes create challenges through drowning out other biotic sounds with regards to the audio data.

Research shows that dolphins often produce tonal whistles of mid to high frequencies while socialising and appear unique to dolphins [4]. Whistles are narrow-band, frequency-modulated sounds that usually last from a tenth of a second to three seconds. Table 2.1 shows the frequency ranges of the various biotic sounds and compares them to the dolphin whistles which is the intended study of this project.

The goal of the study at hand for monitoring underwater soundscapes determines how long acoustic recorders should record for. A spawning periodicity of a species may be determined, and therefore calibrate the system to record during certain time frames around the specific species' spawning times. Variables such as the required length of the study, storage available, hydrophone used and bandwidth determine the recording rates required.

Source	Frequency
Breaking Waves	200 Hz -2000 kHz (peak)
Shipping Noise	30 Hz -100 Hz
Snapping shrimp	2.5 kHz -15 kHz
Fish and whales	<500 Hz
Dolphin whistles	2 kHz -20 kHz
Rainfall	15 kHz-20 kHz (peak)

Table 2.1: Frequency range of sound sources affecting marine animals and recordings

2.2 Software tools for system identification

The following software tools are used to develop an acoustic monitor.

2.2.1 Audacity

Audacity is free, open source, cross-platform audio software [5] used throughout the undertaking of developing an acoustic monitor.

2.2.2 Matlab

Matlab combines a desktop environment tuned for iterative analysis and design processes with a programming language that expresses matrix and array mathematics directly [6]. The live editor script is used to combine code and output in an executable notebook. This is used for short-term post processing of audio data acquired by the acoustic monitor.

2.2.3 Arduino Software (IDE)

The open-source Arduino Software (IDE)makes it easy to write code and upload it to a MCU.

2.3 Acoustics fundamentals

To allow for interpretation of the data produced by the acoustic monitor, an investigation has to be done on the fundamentals of acoustics, so as to proceed with empirical analysis of the data. Furthermore, an investigation is also required on testing conditions and other considerations.

2.3.1 Condenser Microphone

All condenser microphones work by converting a changing capacitance to a charging electrical signal [7]. The condenser microphone used for simulation and testing is an electret condenser microphone consisting of two parallel plates, a diaphragm, and the backplate. An incoming sound wave causes the diaphragm to vibrate, the capacitance changes due to the movement and change in distance between the backplate and diaphragm. With a fixed polarizing voltage applied to the plates, the movement as a result of acoustical waves will produce a change in the capacitance, and therefore produce a time-varying voltage, which will represent the audio signal. Thus, acoustical energy is converted to electrical energy by means of changes in pressure resulting in movement of the diaphragm. The microphone can be classified according to several features which it may contain.

2.3.1.1 Directivity

Microphones can be categorised according to their pickup patterns. This refers to how the microphone reacts to sound coming from different directions and how it discriminates the sounds. It follows that it can be categorized according to the following:

- **Omnidirectional:** can pickup sounds from all directions.
- **Bidirectional:** typically referred to as noise-cancelling microphones, and picks up audio from the front and rejects noise from its sides.
- **Unidirectional:** can pickup audio from one direction only.

2.3.1.2 Bias voltage

The bias voltage of the microphone is achieved through applying a voltage to the diaphragm and backplate. The voltage received from the power supply will be biased around a certain value to center the output voltage of the microphone around the bias voltage. Therefore, negative values are scaled up and this allows for the acoustical waves to pass through the MCU ADC.

2.3.1.3 Sensitivity

Microphone sensitivity is the measure of the electrical output in volts of a microphone with respect to the acoustic sound pressure level input [8]. The standard reference input is is a 1 kHz sine wave at 94 dB sound pressure level (SPL).

2.3.1.4 SNR

The signal-to-noise ratio(SNR) is a ratio of a reference voltage to the noise level of a microphone [9]

2.3.2 Audio Waveform

Audio waveforms originate in the form of an acoustical wave. For development of a sound system such as the acoustic monitor, the waveform must be in the electromagnetic domain prior to being presented to the sound system [10]. This is achieved through using a transducer in the form of both an electret condenser microphone and a hydrophone for the purpose of developing and testing an acoustic monitor.

Audio waveforms are complex of nature and typically vary over time. Several characteristics are of interest when used to describe an audio waveform in order to characterise the system and further develop insights on what is occurring within the system. One may easily identify these characteristics from a sinusoidal waveform with amplitude symmetry, but the task is more complex when working with waveforms of complex nature which vary over time. Thus, first consider the ideal case of evaluating these characteristics from a pure sinusoidal input:

- **Amplitude:** The amplitude relates to the amount of pressure generated by a sound wave and is measured through calculating the difference between the peak and the equilibrium point of the wave. For humans, this value is the perceived loudness of the sound.
- **Peak-to-Peak voltage:** The voltage measured between the largest positive peak and largest negative peak of the waveform.
- **Peak Voltage:** The largest peak present in the waveform regardless of whether it is positive or negative.
- **Average Voltage:** The average of all positive and negative amplitude values of the waveform
- **Root-Mean-Square (rms) Voltage:** Also known as the effective value of the waveform, where the ac voltage would describe the equivalent DC voltage that would produce the same amount of heat into a resistive load[10].

Figure 2.1 describes a waveform with amplitude symmetry and can be used to demonstrate the characteristics described above. However, figure 2.2 shows a typical audio waveform recorded on the acoustic monitor as measured on Audacity.

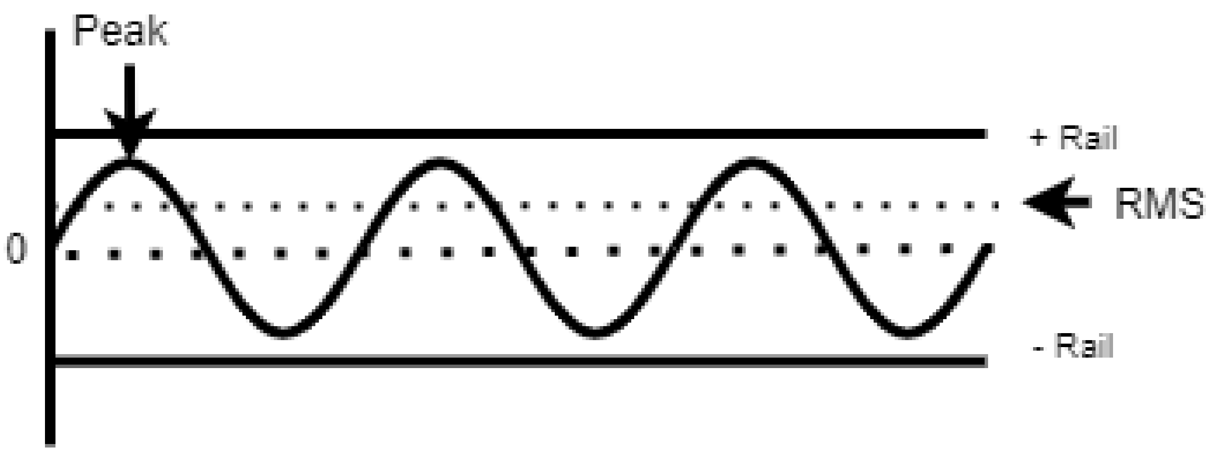


Figure 2.1: Visualising the crest factor

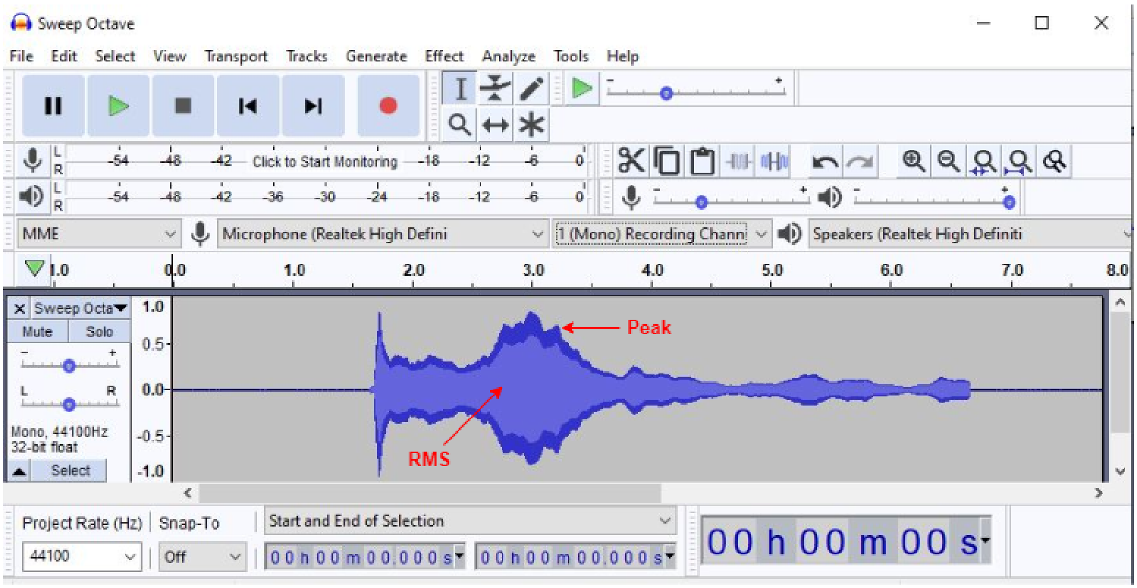


Figure 2.2: Visualising the crest factor in Audacity

The light blue shaded area describes the waveform's RMS voltage value, and the dark-blue envelope is the peak amplitude the voltage reaches. Thus, the values of concern are the peak and RMS values in the pure sine wave and the spectrum measured on Audacity. The peak-to-rms ratio can be observed which is also known as the crest factor[10]. The crest factor of a signal determines the energy content, and therefore the power produced by the amplifier.

Some of these variables are measured extensively , and conclusions are made based on the results obtained through analysing the waveforms in post-processing. Therefore, to understand and interpret audio, to produce meaningful results and calibrate the system for optimum performance, one has to be able to identify the key features described above.

2.3.3 The Decibel

The decibel was defined as the logarithmic form of a power ratio as having a value of $10^{0.1}$, where a measured power and reference power are compared. Two amounts of power differ by 1 dB where they are in the ratio $10^{0.1}$ and two amounts of power differ by (N)dB where they are in the ratio $10^{N(0.1)}$ [11]. The result of deriving an expression for decibels can be show in Equation 2.5. Most measurements taken throughout the undertaking of developing an acoustic monitor, however, are voltage amplitude measurements generated by the transducer. Thus, one may use reference voltage measurements versus output voltage measurements and manipulate Equation 2.5 when one realises that Equation 2.6 and 2.5 are equivalent due to Ohm's law, since the decibel works with ratios. The term "dB" is reserved for power ratios in this report, whereas "dBV" is used when working with voltages. The derivation when working with voltage or power is produced as follows:

$$\frac{P_1}{P_2} = 10^{N(0.1)} \tag{2.1}$$

$$log_{10}\left(\frac{P_1}{P_2}\right) = log_{10}(10 * N * 0.1) \tag{2.2}$$

$$log_{10}\left(\frac{P_1}{P_2}\right) = 0.1 * N \tag{2.3}$$

$$\frac{log_{10}\left(\frac{P_1}{P_2}\right)}{0.1} = N \tag{2.4}$$

$$10log_{10}\left(\frac{P_1}{P_2}\right) = NdB \tag{2.5}$$

Where P_1 represents the output power, P_2 the reference power, and N the number in decibels. Now, the following relationship holds true if only both values are measured at identical points in the circuit [11]

$$10log_{10}\left(\frac{V_1^2}{V_2^2}\right) = 10log_{10}\left(\frac{P_1}{P_2}\right) \tag{2.6}$$

which can be further simplified in order remove the exponent and be written as:

$$10log_{10}\left(\frac{V_1^2}{V_2^2}\right) = 20log_{10}\left(\frac{V_1}{V_2}\right) \tag{2.7}$$

where V_1 and V_2 now represent the voltage levels instead of the power levels. Thus, the decibel is based upon a power ratio and when you are working with any other kind of ratio (voltage, current or sound pressure), it should be converted to a power ratio by squaring the variables.

2.3.4 Gain of the system

The power supply provided by the system is a representation of the maximum amplitude that a waveform may take when it passes through a component within the system. Exceeding this peak value will result in clipping of the waveform. When no input is applied, there will still be a measurable output on components due to thermal noise. The level of thermal noise determines the noise floor of the device [11]. Careful design considerations are of utmost importance when designing an acoustic monitor in order to reduce the impact of thermal noise. However, one can never truly get rid of it.

When no input source is applied to the input of the electret microphone used to simulate the result of the acoustic monitor meant to record marine fauna, one may accurately measure the noise present in the system by means of recording in an anechoic chamber. In the anechoic chamber, one sets the gain of the system to unity such that the voltage at the device's input is the same as its voltage at its output. Since no acoustical signal is present at the input, the only measurement available is that produced by the microphone itself just through being an impedance. This describes the microphone's self-noise, or as mentioned above, the thermal noise present in the system. For this reason, a proper gain structure is desirable in order to reduce the impact of thermal and environmental noise, and still be able to extract a meaningful signal.

Since the voltage output of the microphone and hydrophone is generally low, one also needs to establish a proper gain structure for the system in order to hear the recordings and be able to extract meaningful information. As a result, when increasing the output voltage by increasing the gain of the system, thermal noise will most likely also increase by the same gain factor. The signal-to-noise ratio (SNR) will therefore not be improved. Through increasing the source voltage and assuming that the sending device has a noise floor lower than the drive device, results in an improvement in SNR.

2.3.5 Common-impedance coupling

In building an acoustic monitor, the system essentially uses two wires, one to provide a signal and one for ground. From Ohm's law it is known that current flows through a resistor and a voltage drop occurs. Similarly, if two different circuits share the same conductor or wire, a current flowing in either circuit will produce a voltage drop across the wire [12]. Thus, common-impedance coupling occurs when two circuits must share the same current path. This may lead to unwanted behaviours in systems such as hum which appears when the device is either switched on or off.

Figure 2.3 provides a basic demonstration of a common-impedance circuit. Two circuits share the same return path through Z_g and as a result, the interconnecting cable becomes the common impedance. The voltage across the load of R_{L1} and R_{L2} is affected by the return currents I_{s1} and I_{s1}. The implication of this phenomena is that the two circuits will not be able to distinguish between the signal and hum. Therefore, if the signal is amplified as required of the acoustic monitor, the accompanying hum will be amplified as well.

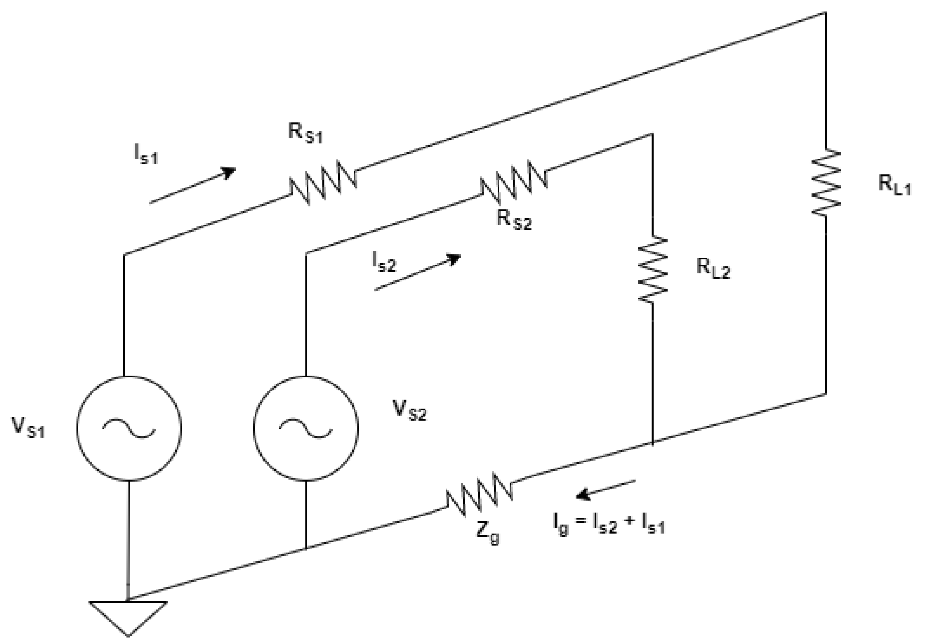


Figure 2.3: Common-impedance coupling circuit

2.3.6 Acoustic Test Facilities

An acoustic chamber is used for various tests in order to calibrate the system and measure its performance in near ideal conditions without external noise from the environment. An anechoic chamber can be considered similar to a precision acoustical measurement instrument, providing a free-field environment without noise interference or sound reflection [13]. An anechoic chamber has the following noise and vibration characteristics [13]:

- Provide good sound isolation against external noise so that resulting internal noise will not invalidate acoustic measurements.
- Requires the use of single or double wall construction with appropriately designed vibration isolation to adequately reduce air - and/or structure borne noise transmission.
- For best results, should be individual structures, separate from any host building walls.

Figure 2.4: Anechoic chamber Stellenbosch University

The anechoic chamber at Stellenbosch University's Engineering building is one of the instruments used in order to test the acoustic monitor. It is situated on the fifth floor and is used mainly by the RF department, but for this project it is an ideal place to measure system noise and microphone self-noise. It has sound absorbing material shaped into wedge configurations and mounted on the interior surface, which is a common practice in order to achieve a free-field environment. Figure 2.4 shows the anechoic chamber at Stellenbosch University.

2.3.7 Acoustics Test and Measurement Practices

Sound systems must be tested to assure that all components are functioning properly. The test and measurement process can be subdivided into two major categories: electrical tests and acoustic tests. The main difference between electrical and acoustical testing is that the interpretation of the latter must deal with the complexities of 3D space, not just amplitude versus time at one point in a circuit [8].

2.3.7.1 Electrical testing

The purpose for electrical testing include, but are not limited, to the following [8]:

- to determine if all system components are functioning properly,
- to diagnose electrical problems in the system, which are usually manifested by some form of distortion,
- and to establish a proper gain structure.

The electrical tests involve measuring voltage levels in the system and can aid in establish the desired gain structure. Most tools needed for basic field tests can be found at the electrical and electronic laboratory at Stellenbosch University.

2.3.7.2 Acoustical Testing

Audio waveforms can be evaluated through producing recordings from a transducer in the form of an electret microphone. The audio waveform is in the form of having an amplitude which varies over time. Therefore, features can be identified through performing numerical and scientific calculations, and experiments may be done in order to characterise the system through the data recorded.

2.4 Passive Acoustic Monitoring of Underwater Sound

In a fluid, sound propagates as a longitudinal pressure wave, in which particles move parallel to the direction the wave is travelling [14]. Changes in pressure levels may be measured by the use of a hydrophone.

2.4.1 Hydrophone and Sound Pressure Levels

A hydrophone detects sound and pressure waves underwater. Most hydrophone sensors are made of piezoelectric materials which convert mechanical energy to electrical energy [15]. Changes in acoustic pressure is converted to an electrical signal which may normally produce the same electrical equivalent output as an electret-condenser microphone. The magnitude of sound is measured using the sound pressure level (SPL), expressed in dB, and the following equation may be applied:

$$SPL = 20log_{10}(p/p_0) \tag{2.8}$$

where p is the received sound pressure measured, and p_0 is a fixed reference pressure. For underwater sound, a reference of 1μPa is used.

2.4.2 Sound propagation and transmission loss

The source level (SL) is the acoustic intensity of a sound emitted from a source. As the sound pressure wave propagates outward from the source, the decrease in acoustic energy of the source is known as the transmission loss (TL), expressed in dB. TL increases the further the sound travels away from the source due to spreading loss and attenuation [14]. The spherical spreading law describes the decrease in level when a sound wave propagates away from a source uniformly in all directions [16] and results in spreading loss. Attenuation of the acoustic energy occurs by means of absorption losses or scattering losses. The absorption losses are frequency dependent and occur due acoustic energy being absorbed by a medium and the acoustic energy converted to heat. Scattering losses form as a result of the acoustic wave deviating from its original path due to contact with the seabed or other external factors.

The attenuation in an underwater acoustic channel over a distance l for a signal frequency f is given by [17]

$$A(l, f) = l^k a(f)^l \tag{2.9}$$

where a is the absorption coefficient and k is the spreading factor. This result may also be expressed in dB,

$$10log\ A(l, f) = k\ .\ log\ l\ + l\ .10\ log\ a(f) \tag{2.10}$$

where the first term describes the spreading loss and the second term the absorption loss.

Four sources of ambient noise (turbulence, shipping, waves and thermal noise) can be described with Gaussian statistics:

$$10log\ N_t(f) = 17\ -\ 30\ logf \tag{2.11}$$

$$10log\ N_s(f) = 40\ +\ 20(s - 0.5) + 26logf - 60log\ (f + 0.03) \tag{2.12}$$

$$10log\ N_w(f) = 50\ +\ 7.5w^{\frac{1}{2}} + 20logf - 40log\ (f + 0.4) \tag{2.13}$$

$$10log\ N_w(f) = -15\ + 20logf \tag{2.14}$$

and the overall ambient noise is given by

$$N(f) = N_t(f) + N_s(f) + N_w(f) + N_{th}(f) \tag{2.15}$$

The attenuation and noise can be combined and may be used to evaluate the SNR observed at the receiver over a distance l when the transmitted signal is a tone of frequency f and power P [17], resulting in the following equation:

$$SNR(l, f) = \frac{\frac{P}{A(l,f)}}{N(f)\Delta f} \tag{2.16}$$

with Δf the receiver noise bandwidth.

2.5 Signal Processing

Audio files obtained from the acoustic monitor may be used for short-term post processing. Two methods are used to characterise system noise, namely THD+N and SNR.

2.5.1 Signal-to-noise ratio and Dynamic Range

The SNR is a measure of the maximum output voltage compared to the integrated noise floor over the audio bandwidth, expressed in dB [18]. Figure 2.5 shows the spectrum of a recording captured by the acoustic monitor where the input is driven just below clipping. A 1 kHz sine wave is placed at the microphone input of the device by means of using a signal generator. The lower level represents the lowest level the signal can take without being buried in noise, whereas the top level is the highest possible undistorted peak. The difference is known as the dynamic range, where the spurs at 2 kHz and 3 kHz appears due to the electrical components of the signal generator.

An appropriate means of determining SNR is established in the sense that one may measure the noise floor and compare its value to the output signal.

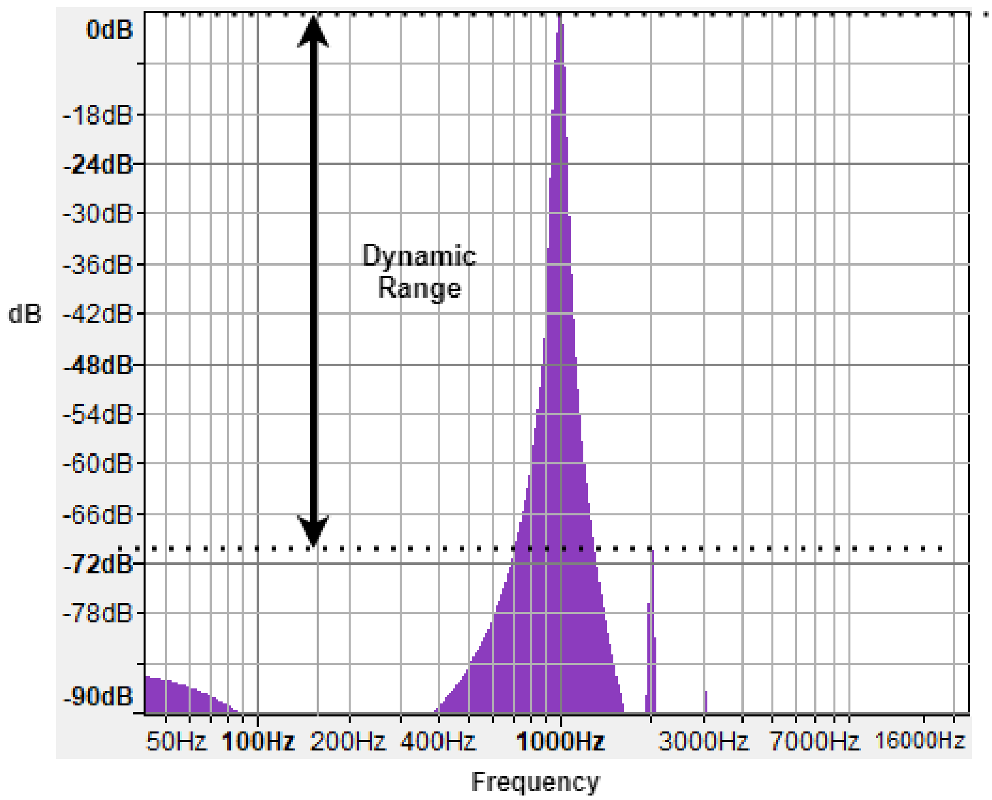


Figure 2.5: Dynamic Range

The following equations are used to compare differences in noise levels produced when calibrating the system.

$$SNR = 20log\left(\frac{V_{OUT}^{RMS}}{V_{NOISE}^{RMS}}\right) \tag{2.17}$$

$$SNR = dBV_{OUT} - dBV_{NOISE} \tag{2.18}$$

where V_{OUT}^{RMS} represents the output rms voltage and V_{NOISE}^{RMS} represents the noise floor rms voltage.

2.5.2 Total Harmonic Distortion Plus Noise (THD+N)

THD+N measurements combine the effects of noise, distortion and other undesired signals into one measurement and relates it to the fundamental frequency. THD+N is usually defined as the square root of the ratio of the sum of powers of all harmonic frequencies plus noise to the total power [19]. The following equation may be applied:

$$THD + N = \frac{\sqrt{V_{total}^2 - V_1^2}}{V_{total}} \tag{2.19}$$

with V_{total} representing the RMS amplitude of the total signal measured, and V_1 the amplitude of the fundamental frequency component. This value is often expressed as a percentage or in dB. The equation may be written in the form of the distortion harmonics (V_i), which is an integer multiple frequency of the input signal, and the RMS noise voltage (V_n) and be compared to the level of input signal (V_f) through using the equation:

$$THD + N = \sqrt{\frac{\sum_{i=1}^{\infty}(V_i^2) + V_n^2}{V_f}} \tag{2.20}$$

In order to measure the distortion harmonics and noise voltage, the fundamental frequency is normally filtered out using a notch filter.

2.5.3 Notch Filter Design

A notch filter is a filter that contains one or more deep notches or, ideally, perfect nulls in its frequency response characteristic [20]. When the objective is to eliminate frequency components of a signal at specific frequencies, a notch filter may be applied.

This method is achieved through introducing a pair of complex-conjugate zeros on the unit circle at an angle of ω_0. The bandwidth of the notches can be reduced by placing a pole pair at the same frequency close to the unit circle as can be seen on figure 2.6. The effect of the poles is to introduce a resonance in the vicinity of the null and thus reduce the bandwidth of the notch. The resulting transfer function can be seen from equation

$$H(z) = b_o.\frac{(1 - e^{j\omega}.z^{-1})(1 - e^{-j\omega}.z^{-1})}{(1 - re^{j\omega}.z^{-1})(1 - re^{-j\omega}.z^{-1})} \tag{2.21}$$

$$H(z) = b_o.\frac{(1 - (2cos\omega) + z^{-2})}{(1 - (2rcos\omega) + rz^{-2})} \tag{2.22}$$

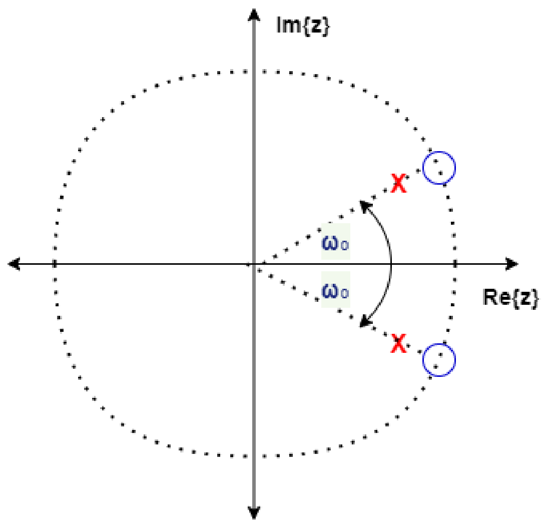


Figure 2.6: Poles and Zeros of a Notch Filter

3 Methodology

3.1 Design Requirements

The acoustic monitor developed will be based on the design requirements which follow. This will allow for an initial design to be developed and finally result in a final design which will be deployed. Therefore, to complete the objectives, the appropriate peripherals are necessary for hardware and software development.

3.1.1 ARM Cortex-M Processor Family and MCU

The acoustic monitor will require hardware to implement long-term coverage of acoustics underwater. Therefore, a suitable processor is necessary which does not require a large surface area and is energy efficient.

ARM has a range of different processor products. Figure 3.1 [21] shows the ARM processor family. The processors can be divided into three categories based on its intended use cases. Thus, the Cortex-A family, Cortex-R family and Cortex-M family may be evaluated for the following use cases [22]:

- application processors support OS and high performance use cases,
- real-time processors support real-time processing and mission-critical control,
- and microcontrollers are used when the desired outcome is cost-sensitive. It also supports SoC.

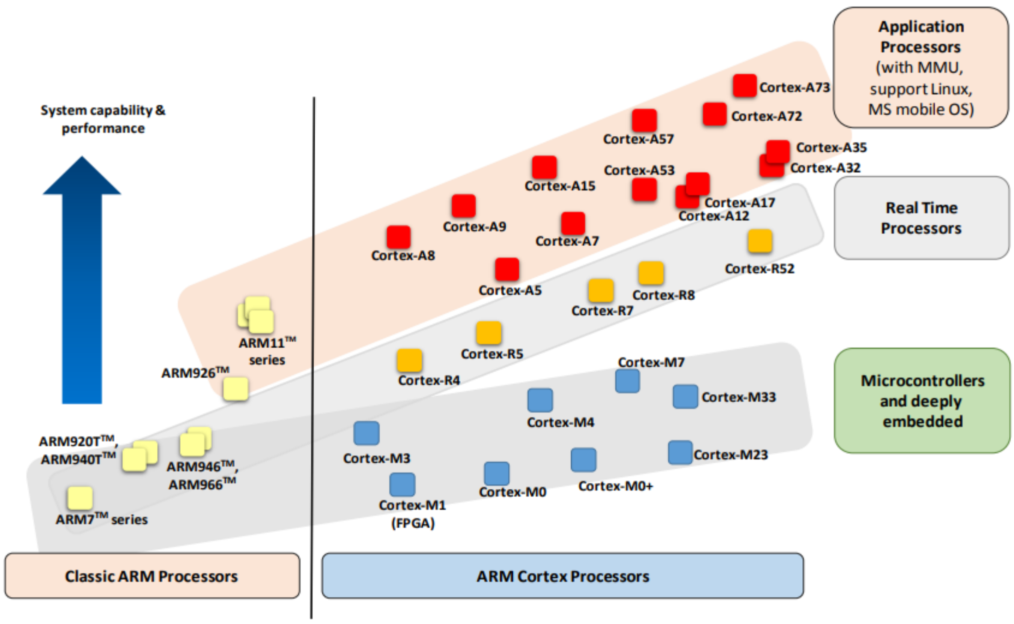


Figure 3.1: ARM Processor Family

Cortex-M best suits the project, since it provides a great balance in cost, efficiency and performance. Specifically, the Cortex-M7 processor is the most desirable since it maximizes performance and is popular in high-end MCU's and processing intensive applications [21]. This also narrows down the decision when it comes to selecting an appropriate MCU.

3.1.2 Audio Codec

An audio codec is a peripheral that enables analog signals to be converted and coded to a digital data steam or conversely the data steam to be decoded and converted back to an analog signal [23]. An audio codec is needed to provide for a portable device which can support a microphone or hydrophone input and digital I/O. Further features such as digital audio processing, low power consumption and product longevity is required as well since the acoustic monitor is intended for long-term use.

3.1.3 Transducer input

An appropriate transducer is needed in the form of both an electret microphone and a hydrophone which will serve to connect to the audio codec microphone input and have its analog signal converted by the ADC. The electret microphone is inexpensive and easy to use, since it only has an output pin and a pin which has to be connected to ground. With most audio codecs, one connects the output pin of the microphone to the microphone input pin of the audio codec, and similarly connect the ground pin of the microphone to ground. The hydrophone can be coupled in the same manner.

The electret microphone needs to have similar features as the hydrophone. Features such as sensitivity, current consumption, directivity and SNR are a few important features when selecting the microphone to provide the best test results. Improved test results provides for better system characterisation and decision making regarding system design and optimisation.

3.1.4 Data Storage

In order to save the data of the acoustic monitor, an appropriate storage system is required. The system should be able to save the data and create a time stamp when the recording begins and when the recording finishes. Furthermore, the system should be able to save a large amount of data for the intention of being a long-term device.

The approach considered will be to save the data onto a micro SD card. This approach leads to further simplification, as Arduino IDE has user friendly libraries in order to implement the required SD card functionality.

3.1.5 Power

The acoustic monitor will be portable and will therefore require external power. Testing and simulation will provide power by means of using a USB connection with the MCU, but when the device is ready for deployment, external power is required.

The device should be able to record for a few hours at a time and should be easily turned on or off. Therefore, batteries can be used in a battery pack in order to achieve this. A battery pack with an on/off switch is required in order to be able to turn the acoustic monitor on right before deployment, and switch it off during retrieval.

3.1.6 Scheduling Operation

The acoustic monitor is required to record during intervals for a fixed period of time set in software. This determines the duty cycle of the acoustic monitor, and an idle mode is required for when the device is not recording, to prolong battery life through lowering the power consumption.

Scheduling operation requires the use of the MCU's real-time-clock (RTC). This allows for timing operations based on the MCU's RTC. The scheduling operation should be synchronised with the data storage unit to provide for saving the data with the correct date and time. The date and time should also be stored whilst the device is off. This is due to the device needing to be switched over to external power and remain off right until deployment.

3.1.7 Testing practice

As the title of this report suggests, a low-cost device is needed in order to measure dolphin sounds and marine fauna. Therefore, results obtained in order to calculate metrics such as SNR and THD+N are limited to the following tools:

- Oscilloscope
- Software such as Matlab and Audacity
- Signal generator
- Anechoic chamber
- Swimming pool

As a result of budget constraints, the system may not be fully characterised as one would hope, but is still suitable enough to produce a meaningful result which will be demonstrated.

4 System Design

4.1 System Overview

The system is designed to develop a long-term recording device in order to measure frequencies at a sampling rate of 44.1 kHz. This will be appropriate for measuring dolphin whistles at frequencies of 2 kHz to 20 kHz. Figure 4.1 shows the desired result. The hydrophone is connected to the microphone input of the audio adaptor board, where signal conditioning takes place via a built-in RC circuit and pre-amplifier, before the analog signal is passed through the audio board's ADC. The audio board's ADC then streams audio data via audio packets to the MCU through I2S communication, where a buffer is used to store the data and is saved to Audio Memory. This data is then processed by the MCU in order to store the audio data on to an SD-card in a WAV format through communicating in SPI mode. Furthermore, I2C communication is used between the MCU, and a low power stereo codec found on the audio adapter board in order to control the codec and adjust its parameters, which is software programmable. This low power stereo codec contains the circuitry for the ADC and other signal conditioning circuitry present.

The device is powered through combining 3x 1.5V alkaline batteries to meet the specification of a 3.6V to 5.5V input required by the MCU. The batteries are connected to the input pin of the MCU for power. The input voltage is regulated internally down to 3.3V, which is then fed through to the audio adaptor board by means of a 3.3V pin also found on the MCU. To begin the sequence of recording, a latch button is used. This sets the device to record only during intervals for a fixed period of time set in software. The MCU's ground pin serves to provide a common ground path for the functional blocks present.

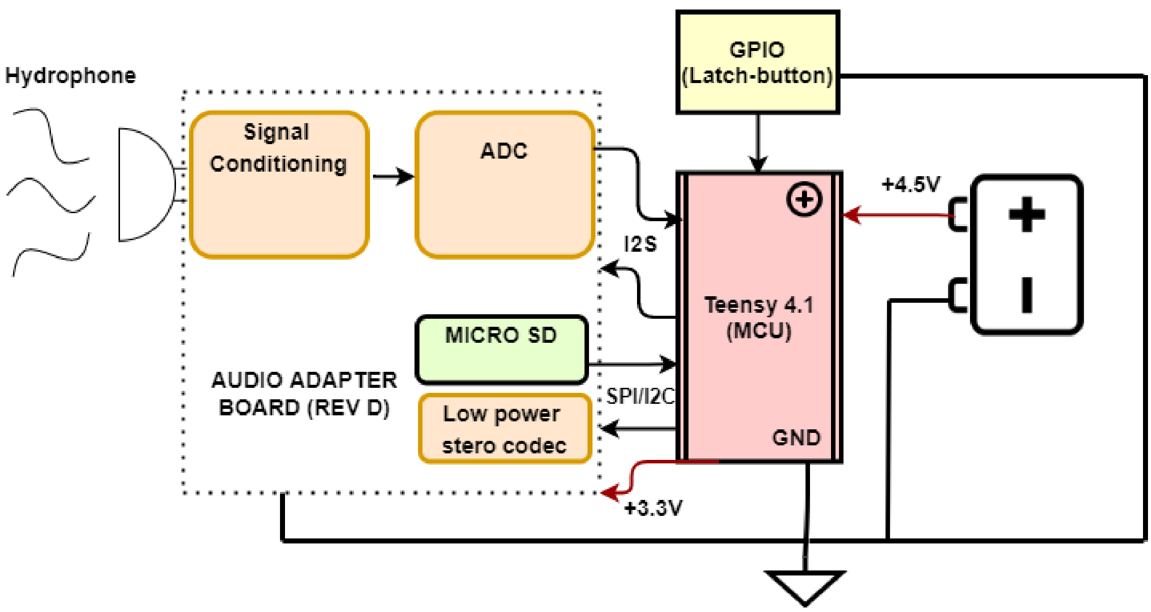


Figure 4.1: System Overview

4.2 Hardware Design

Figure 4.2 shows the final design of the acoustic monitor which was used for simulating an underwater acoustic monitor.

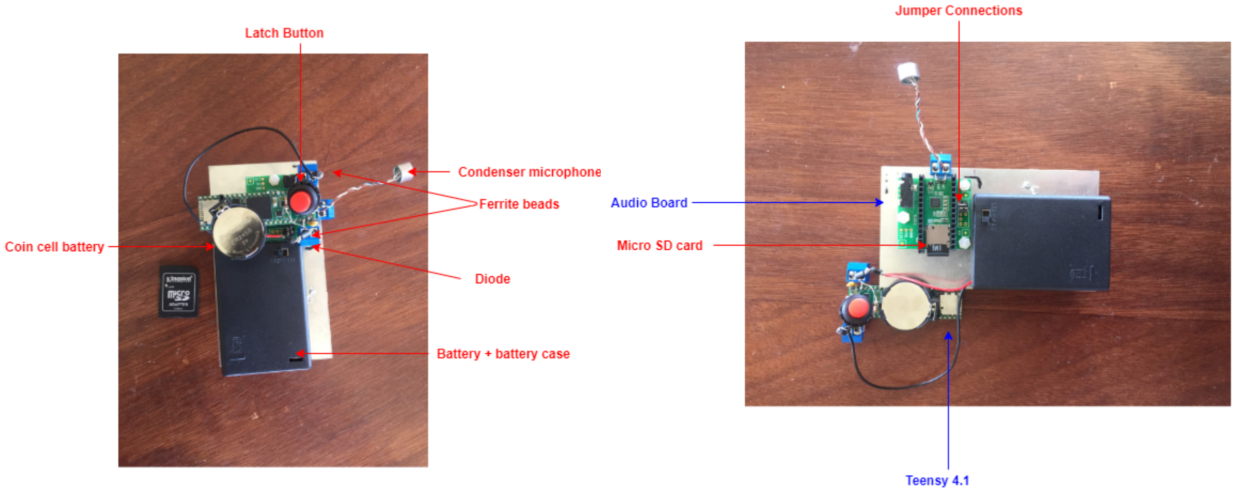


Figure 4.2: Acoustic monitor final design and its peripherals

4.2.1 Microcontroller Unit

Feature	Teensy 4.1	Teensy 4.0
Price	R917.70	R539.35
CPU	ARM Cortex-M7(NXP iMXRT1062 chip)	ARM Cortex-M7 (NXP iMXRT1062 chip)
SDIO	Micro SD socket	8 SMT Pads
SD Card Signals	6	0
Flash Memory	8 Mbyte	2 Mbyte
QSPI Memory	2 Chips Plus Program Memory	Program Memory Only
Communication	UART, I2C, SPI	UART, I2C, SPI
Date/time	RTC	RTC
Dimensions	2.4 by 0.7 inch	1.4 by 0.7 inch

Table 4.2: Microcontroller comparison

Two considerations were considered for the selection of which microcontroller to be used based on CPU performance and memory, namely the Teensy 4.1 and Teensy 4.0. Both microcontrollers feature an ARM Cortex-M7 processor at 600 MHz, with an NXP iMXRT1062 chip to provide for best real-time response in order to cater for the metrics considered under the scope of this project. Both of these microcontrollers functionality are described in Table 4.2. The Teensy 4.1 was selected based on the added peripherals regarding memory and added SD signals for ease of use. Specifically, the flash memory was enhanced from 2 Mbyte to 8 Mbyte from the Teensy 4.0 to the Teensy 4.1 for improved program storage which serves improved functioning of a data-logging device.

4.2.2 Audio Adapter Board

Audio adapter boards are available for the Teensy 3.x and Teensy 4.x. This allows for added 16 bit, 44.1 kHz sample rate audio which will be used for the project. Two versions are available, namely the Rev D and Rev C audio adapter boards. Rev D is compatible with the Teensy 4.0 and Teensy 4.1 and is used for the recording device, and Rev C is used for older versions. The audio board supports stereo headphone and stereo line-level output, and also stereo line-level input or mono microphone input. For the application of this project, the main interest is that of the mono microphone input which can be used to easily attach an electret condenser microphone or hydrophone.

The audio adapter board has a low power stereo codec, SGTL5000, which connects to the Teensy through 7 pins. I2C, SDA and SCL pins control the chip. Audio data uses I2S signals, with TX and RX pins used to transmit and receive audio data.

4.2.3 Electret Microphone

To simulate results prior to the undertaking of recording with the hydrophone, an electret microphone is used which can be connected to the audio board's microphone input. Two options were considered to provide accurate simulation results, namely the MDO9765APN and MDN6027CSC Condenser microphones. Parameters such as the microphones sensitivity, SNR, current consumption and directivity were all considered when making this selection.

Electrical Characteristic	MDO9765APN	MDN6027CSC
Directivity	Omnidirectional	Noise Cancelling
Sensitivity	Max $-34dB$	$-40dB$
Current Consumption	$0.5mA$	$0.5mA$
SNR	more than $60dB$	more than$60dB$

Table 4.3: Microphone comparison

Ultimately, the MDO9765APN was selected. Similar to the H2c hydrophone, its directivity is omnidirectional. These types of microphones are designed in order to receive vibrations from virtually any direction. Therefore, unlike the MDN6027CSC, this microphone cannot discriminate between unwanted sounds and therefore noise can be picked up from the environment and amplified. This plays a vital role in being able to simulate how the system will respond to noise under water, and allows for more accurate software simulations in order to vary the gain when the system detects unwanted noise.

4.2.4 Hydrophone

The H2c hydrophone from Aquarian Hydrophones is used. It incorporates a matched sensor and FET buffer amplifier assembly that produces an output electrically equivalent to electret-condenser microphones [24]. A 3.5mm TRS output jack is also provided which is configured for dual mono, thus both left and right stereo channels are from the same source, and either may be connected to the mono input of the audio adaptor board. This connection is the same as with the electret microphone. Therefore, when the acoustic monitor is deployed into an underwater environment, the electret-microphone may easily be replaced with the hydrophone. Table 4.4 described the characteristics of the H2c hydrophone.

Electrical Characteristic	MDO9765APN
Directivity	Omnidirectional
Sensitivity	-180 dB re: 1V/μPa
Power	$0.3mA$
Useful range	10 Hz to 100 kHz

Table 4.4: Hydrophone characteristics

4.2.5 Micro SD Card

The Arduino library supports up to 32 GB size SD cards. The recommended SD card for the Teensy 4.1 is the 32 GB SanDisk Ultra. This comes with an adapter and has read speeds of up to 80 MB/s and write speeds of up to 10 MB/s. The selection was clear from the recommendations made on the Teensy support page. However, when placing the order the Kingston SDCS2/16GB arrived instead.

4.2.6 Latching Switch

The system is designed in order to respond to a switching mechanism only while physically actuated. The MCU supports this via means of GPIO pins, where the state can be set from low to high depending on the initial state using a push button switch. For the intended functionality of the recording device, a latching switch is used instead of a normal push button. This is used in order to maintain the state after being activated, i.e. the state is maintained since the switch remains actuated until physically released again. Therefore, once the button is pressed a sequence starts within software, and can only be stopped by means of releasing the button or the MCU losing power.

4.2.7 Power source

In order to provide the MCU with power, two options were considered, using alkaline or lithium batteries. For the scope of the project and its intended purpose, the battery with the lower cost was used, which is the alkaline type. Thus, the MCU is powered by means of 3x 1.5V AA Duracell Alkaline batteries with a capacity of 2.85Ah. The capacity indicates that the battery may provide 2.85 A of current for 1 hour. The batteries are applied to the VCC pin and ground of the Teensy to supply the required input voltage.

The Teensy 4.1 draws about 100 mA current, and the duration can be calculated as follows:

$$Duration(h) = \frac{Capacity(Ah)}{Average\ current\ draw\ over\ an\ hour\ (A)} \tag{4.23}$$

$$Duration(h) = \frac{2.85}{100m} \tag{4.24}$$

$$Duration(h) = 28.5\ hours \tag{4.25}$$

4.2.8 Jumper connections for LININ/LINOUT

A jumper connection is applied to the line-in of the audio board in order to reduce noise of the system. This is an experimental decision, and is demonstrated further in Section 5.2.1.

4.2.9 Diode

A 1414G diode is connected with the orientation being that the cathode connects between the MCU VCC pin and the positive terminal of the battery. This diode protects the MCU if the battery is connected backwards.

4.2.10 Coin cell battery

The Teensy has a pin, VBAT, located near the SD-card socket which allows for the RTC to keep track of date and time while the power is switched off. Therefore, a 3V coin cell battery is connected to VBAT and ground.

4.2.11 Ferrite bead

In order to address the common-impedance problem, a ferrite bead is used. The ferrite bead is placed in parallel with a 100 nF capacitor.

4.2.12 Waterproof casing

A waterproof casing is provided by the machine labs at Stellenbosch University. This allows for the MCU to be placed underwater without getting damaged. The hydrophone is also attached to the lid of the case, therefore, the MCU can detach the electet condenser microphone from the audio adaptor board's microphone input pins and easily replace it with that of the hydrophone pins and place the MCU safely inside. The casing can be seen from Figure 4.3.

Figure 4.3: Waterproof case

4.3 Software Design

Figure 4.4 demonstrates the recording procedure of the acoustic monitor. Once the program starts, the *millis*() function is called in order to keep track of the milliseconds since the device was powered. With this function, scheduling operation becomes possible in the sense that one may set user-specific parameters to implement the desired functionality.

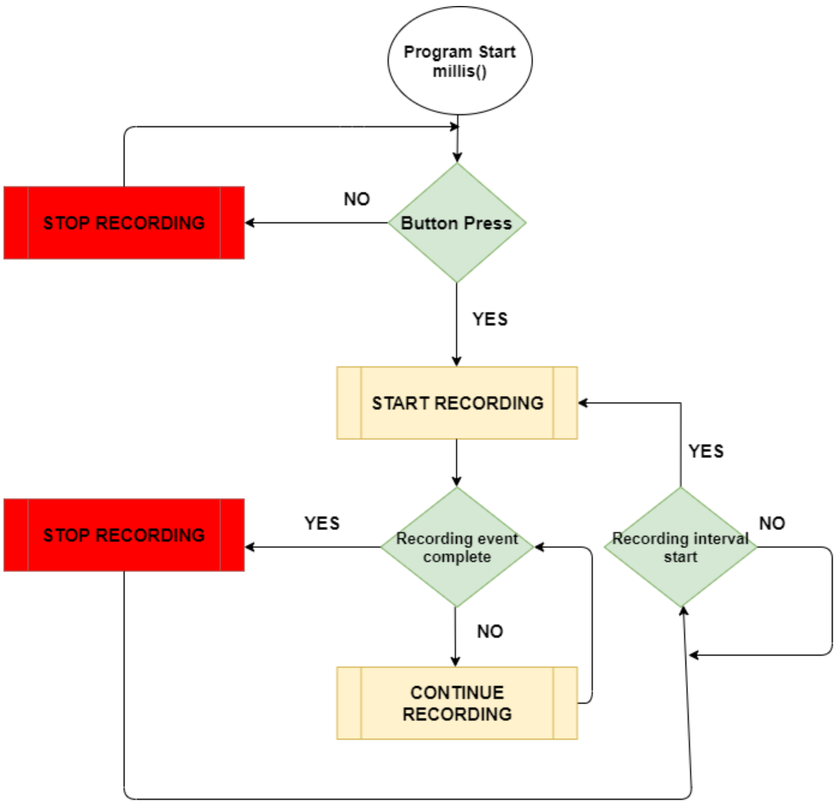


Figure 4.4: Scheduling operation overview

The recording event starts once the latch-button is pressed and continues to record for a duration set by the user until completion. Once the recording duration is complete, the device stops recording. Finally, the device is set to record for fixed duration during intervals. The acoustic monitor's desired functionality is to record 20 minutes every hour.

4.3.1 Audio System Design Tool

The Audio System Design Tool lets you draw a system to process 16 bit, 44.1 kHz streaming audio while your Arduino sketch runs and is meant to be used strictly for the Teensy. It provides objects which can be called from setup() or loop() to control the desired functionality required. Figure 4.5 represents the objects used for the software developed to record and save the audio files to an SD card and vary the gain.

The design described above receives 16 bit stereo audio from the audio adaptor using I2S master mode. The I2S object has no functions to call from the Arduino Sketch and simply streams data from the I2S hardware to its two outputs. The Queue object records the data by sending it to the Arduino sketch, allowing it to receive audio packets. The AudioPlaySdWav is simply used for playback to play a WAV file stored on the SD card. The SGTL5000 block provides the functions of the low power stereo codec. Finally, the peak object allows for peak detection of a signal.

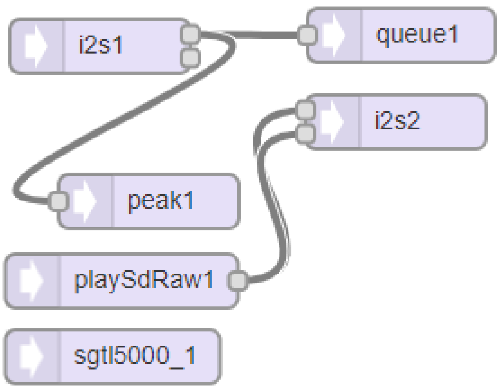


Figure 4.5: Audio System Design Tool for Teensy Audio Library

Once the design is complete, code is generated in order to implement the system and can be exported and copied into the Arduino editor. The design tool is available on the the Audio Design Tool website where the design provided above can be displayed.

4.3.1.1 SD Card Operation

As mentioned in section 4.3.1, audio data is recorded with the queue object and is streamed via audio packets. Each packet contains a 128 sample array with 16 bit integers. 2 blocks are fetched at a time and stored into a 512 byte buffer. Recalling that 16 bits is equal to 2 bytes, therefore each packet contains 128*2 = 256 bytes.

Equation 4.26 is applied to calculate the duration of the audio file, and in turn be able to calculate different sizes of audio files when specifying the desired duration:

$$duration = \frac{size}{fs} \tag{4.26}$$

where the duration is given in terms of the size in MB and fs is the sampling frequency. Since the size contains the number of samples, it has to be multiplied by 2 because each 2 blocks are writing at a time to the SD card. Therefore, a 1 minute recording will result in the following:

$$duration = \frac{size}{fs} \tag{4.27}$$

$$60 = \frac{size}{44100} \tag{4.28}$$

$$size = 60 * 44100 \tag{4.29}$$

$$size = 264600\ Samples \tag{4.30}$$

and thus, the size on the SD card will be 264600*2, resulting in 529 200 bytes.

4.3.2 Varying Gain

The gain of the acoustic monitor is controlled through the SGTL5000. The microphone programmable gain accepts values of 0 dB, 20 dB, 30 dB and 40 dB.

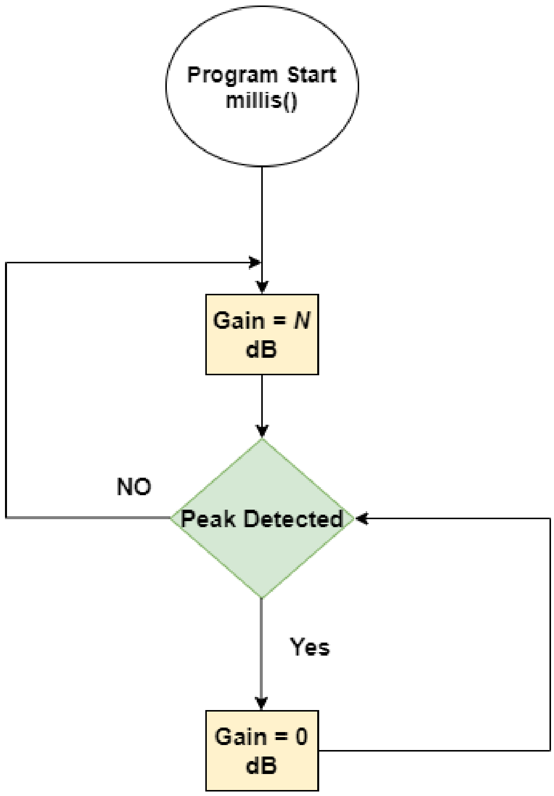


Figure 4.6: Peak detection set in software

Furthermore, the ADC of the SGTL5000 has an additional gain stage which provides a further 0 to 22.5 dB of gain. The acoustic monitor is set to vary the gain through the software functions provided. This is done through peak detection on the audio signal applied during real-time. When a peak is detected, the gain should be regulated down to reduce the sound intensity of the audio signal. The gain is set according to user specification and amplifies low-voltage signals produced by the transducer. The desired gain level is evaluated through testing the acoustic monitor and calibrating it accordingly. Figure 4.6 shows the process set in software, where N is the desired gain expressed in dB and may range from 0-63 dB.

To guarantee against clipping, it is important that the gain in a signal path in addition to the signal level does not exceed 0 dB at any point [25]. The gain value to be set depends on the results obtained through system testing. Thus, during normal operation, the gain value is fixed and only changes when a peak is detected. A peak represents a higher amplitude, resulting in an increased sound intensity, and therefore a louder volume. Peaks may occur due to noise, resulting in clipping of the signal, and therefore should not be amplified further. For this reason, when a peak is detected, the gain is set to 0 dB. As soon as the noise source is no longer present, the gain is set back to the desired specification.

4.4 System Testing

4.4.1 Oscilloscope

Since audio travels along the Audio Adaptor Board, the signal carries a current derived from the power line as well as the audio signal. The Teensy is mounted on top of the Audio Adaptor Board with header pins and therefore share the common ground path leading to common-impedance coupling.

From the pjrc site, it is mentioned that when the Audio Adaptor Board is connected in a system with other devices sharing the same power source, or even separate power supplies with common grounds, a "ground loop" can be formed causing noise to be heard related to the power supply currents. This phenomenon was well established early on as recordings were often overshadowed by a constant hum in the background of recordings.

Figure 4.7: Ground Loop Isolator

Therefore, a ground loop isolator is available and recommended for purchase to prevent this problem with the Audio Adaptor Board and can be seen in figure 4.7. However, the specific product mentioned is not available in South Africa and other approaches are followed in order to reduce the impact of the common-impedance phenomena occurring. Ultimately, it cannot be confirmed whether or not this product would have resulted in a system which performed better than the current system.

To test for common-impedance coupling, one simply has to measure the ground voltage present in the system by using an oscilloscope. To identify differences in ground voltages, a measurement is done first without a power supply present. This indicates the device's thermal noise properties since a voltage is still present regardless of being powered or not. Once power is added to the system via means of a USB or battery, a measurement is taken again, and an increase is expected since the common-impedance path is set from low to high through powering the device and establishing a connection. These two tests are also done with the Teensy connected and disconnected from the Audio Board in order to verify the existence of a ground loop occurring due to the two components communicating with each other.

4.4.2 Anechoic Chamber Testing

The anechoic chamber is used for measuring the electret microphone's self-noise. This test is useful in order to measure the microphone's frequency response with no input supplied to view unwanted frequencies present in the system and can later be compared to the results obtained with that of the hydrophone under similar conditions. Furthermore, a speaker will be used in order to generate a sine sweep within the room and verify that the electret microphone's frequency response is within the desired range of 20 Hz - 20 kHz and the device is functioning as required. This result is only for demonstration purposes, but is not ideal to measure metrics beyond that of microphone self-noise and frequency response, since the speaker used is not ideal for testing purposes. The speaker is placed close to the microphone at full volume, but the signal recorded still has to be normalised to a maximum amplitude in Audacity. This is done in order to get the maximum volume from the audio signal, whilst retaining its relative dynamics. The frequency response of the speaker in an ideal case would have to be a flat frequency response, which no equipment has, but can come close to. Special equipment like this is beyond what was available and therefore measurements may show inaccuracies due to the speaker.

A test is also done to measure the speaker's effect on the room and recordings. A sine wave generated from a signal generator is recorded by the acoustic monitor through applying the signal generator's inputs directly to the microphone input, and is saved to the SD card in a WAV format. The same signal is then replayed through the speaker, normalised in Audacity, and the resulting difference between the recorded signal and playback signal is compared.

Normalising audio signals produce the same amount of gain for the entire signal and thus only the speakers dynamics are added and can therefore be compared to the original output of the signal generator.

4.4.3 Swimming pool

Similar tests are done in a swimming pool as are done in an anechoic chamber. The swimming pool is used in order to test the acoustic monitor's ultimate role, which is to be able record in an underwater environment.

4.4.4 Signal Generator

A signal generator is applied directly to the microphone input of the audio adapter board. This is done in order to produce a test wave and vary the waves' parameters such as its amplitude and frequency. Therefore, the system can be characterized in order to measure metrics such as SNR and THD+N. The signal generator also serves to produce an input to the acoustic monitor to demonstrate the a gain which varies due to peak detection as mentioned in Section 4.3.2.

4.4.4.1 SNR

The SNR is calculated using the Matlab's SNR function:

$$r = snr(x, fs, n) \tag{4.31}$$

returns the SNR of a real sinusoidal input signal x, sampled at a rate of fs. The computation excludes the power contained in the lowest n harmonics, including the fundamental. The microphone input receives a 1 kHz sinusoidal wave with its voltage ranging from 4 mV_{pp} to 1.7 V_{pp} at unity gain and the SNR is calculated for each input voltage increment.

Another calculation is done with the same samples taken above. However, this time the output is measured and compared to that of the input and the difference is calculated. Equation 2.18 is used and the difference between the input and output is used for the V_{NOISE}^{RMS} parameter, and the output measured is used for V_{OUT}^{RMS}.

4.4.4.2 THD+N

THD+N versus frequency is calculated through varying the frequency from 20 Hz to 20 kHz [18]. At each frequency step linearly along a semilog plot, a notch filter is applied to filter out the fundamental frequency. The RMS voltage of what remains is then calculated and equation 2.20 is applied.

4.4.4.3 Notch Filer Design

As mentioned in section 4.4.4.2, a notch filter is required to filter out the fundamental frequency of the various frequency steps by sweeping the frequency from 20 Hz to 20 kHz. The notch filter is designed in software, using Matlab's *iirnotch* function:

$$[num, den] = iirnotch(\omega_0, b_0)$$

returns the numerator coefficients, *num*, and the denominator coefficients, *den*, of the digital notching filter with the notch located at ω_0 and the bandwidth at the -3 dB point set to b_w. The quality factor, q, for the filter is determined by

$$q = \omega_0 / b_w \tag{4.32}$$

where ω_0 is the notch frequency and b_w the bandwidth.

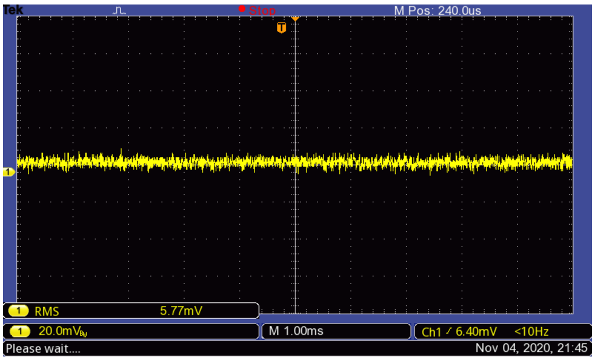
4.4.5 Varying the gain

Due to the limitations of the SGTL5000 chip, producing a peak at the microphone input has risks involved. The ADC may be damaged if tests done involve knowingly producing a signal which would result in clipping of the waveform. The signal generator is therefore used to change the voltage whilst recording. Whenever a large enough change in voltage output of the signal generator is produced while connected to the microphone input, a peak occurs almost instantaneously. This peak sets the $adjustMicLevel()$ function which varies the gain as described in Section 4.3.2.

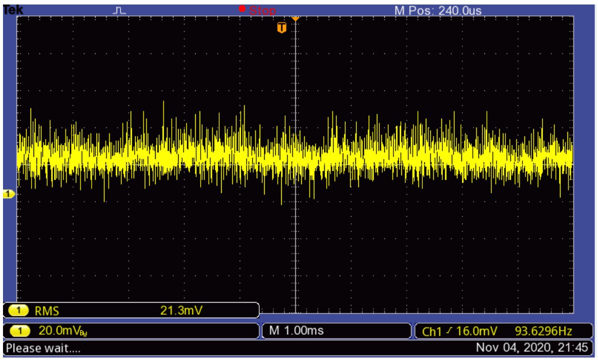
5 Results

5.1 Ground voltages from oscilloscope

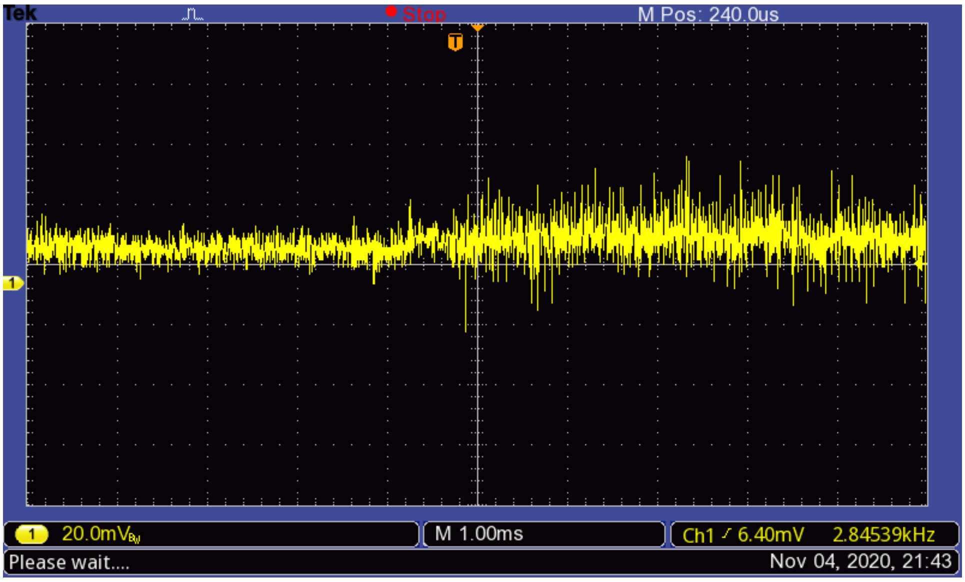
Using an oscilloscope probe, measurements are made when the common-impedance path is set high and low. Figure 5.1a shows the ground voltage when the acoustic monitor is off and the impedance path is low. Figure 5.1b shows the voltage when the acoustic monitor is on and the impedance path is high. The transition from low to high impedance can be demonstrated in Figure 5.1c.



(a) Acoustic monitor turned off



(b) Acoustic monitor turned on



(c) Transition from off to on

Figure 5.1: Ground voltages of the acoustic monitor

The change in ground voltage verifies the presence of a common-impedance path in the system. There is a noticeable transition which causes a degradation of the signal. As a result, hum occurs in the audio recorded.

5.2 Anechoic Chamber Testing for Microphone Self-Noise

System design led to an initial design being made to gather data on and measure its performance. It was built and tested before any additions were made to reduce noise or disturbances present in the system. The results obtained demonstrate how the initial design was calibrated, and improved the microphone's self-noise through viewing the frequency plots with Audacity.

The initial design was tested using the anechoic chamber to view the frequency plots. Various problems were faced and addressed to try and analyse what is causing a disturbance or noise in the system.

5.2.1 Software related issues and low frequency components

The system design tool as described in Figure 4.5 provides information on each block present in the design. Furthermore, it shows the most popular functions which may be used with each particular block. The $sgtl5000$ has various signal conditioning functions. The $adcHighPassFilterDisable()$ function completely disables the analog input filter. The consequence of this is that sub-audible frequencies are allowed to enter, but the claim is still made in the description as well that it *may reduce noise* in some cases. Thus, it was included in the initial design.

Figure 5.2a shows the first test for the microphone's self-noise. The noise floor reaches a maximum of -52 dB and the noise is mainly centred around low frequency components. To reduce the impact of the low frequency components, jumper connections are added to the LINEIN/LINEOUT pins of the audio adaptor board.The frequency components between 100 Hz and 200 Hz are no longer present after the jumper connection has been added. Thus, for a peak of -52 dB of the noise floor, the dynamic range will allow for the lowest level a signal can take of:

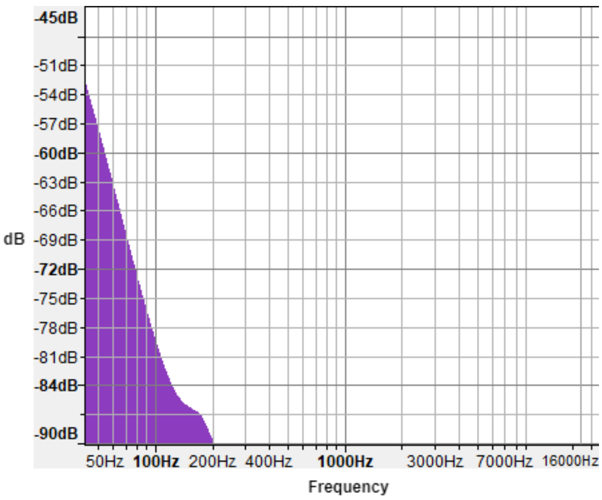
$$20\ log\ (V_{min}) = -52\ (dBV) \tag{5.33}$$

$$log\ (V_{min}) = \frac{-52}{20} \tag{5.34}$$

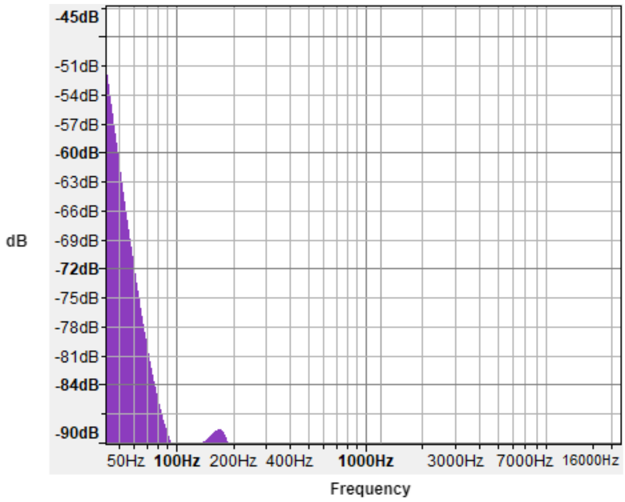
$$V_{min} = 10^{\frac{-52}{20}} \tag{5.35}$$

$$V_{min} = 2.5\ mV \tag{5.36}$$

The noise floor present is not ideal as the microphone input voltage is generally low, and an appropriate gain stage is needed to amplify the signal. Therefore, if the gain is altered too much, this signal will be amplified with the desired output signal of the microphone, and there will be excessive resultant interference.

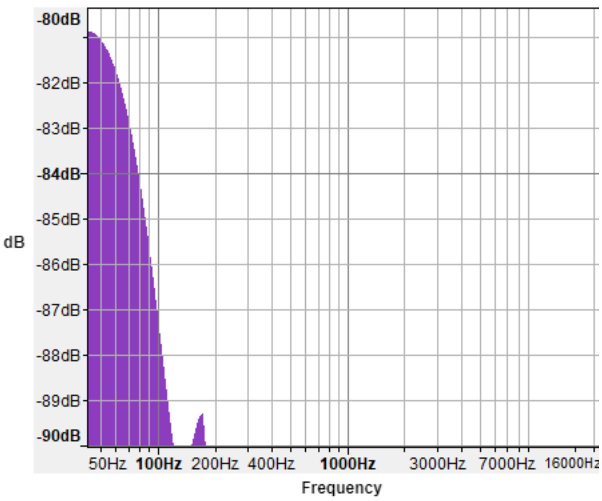


(a) Acoustic monitor turned off

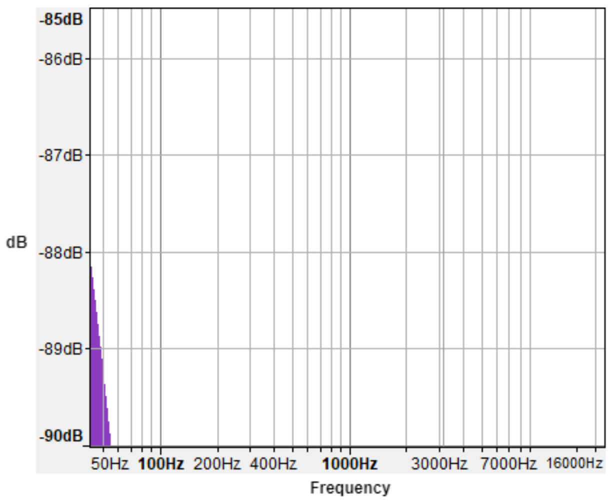


(b) Acoustic monitor turned on

After performing the test with the *adcHighPassFilterDisable*() function included in software, the high pass filter is then enabled to view the difference and can be seen from figure 5.3a.



(a) High pass filter enabled with jumpers



(b) High pass filter enabled with jumpers and ferrite bead

The lowest peak noise level decreases from 52 to 81 dB. The ferrite beads are also then added and figure 5.3b shows the final result. Thus, the initial system provided a noise floor with a peak of -52 dB and improved to -88.1 dB for the final design. This corresponds to a voltage level of 39.4 μV.

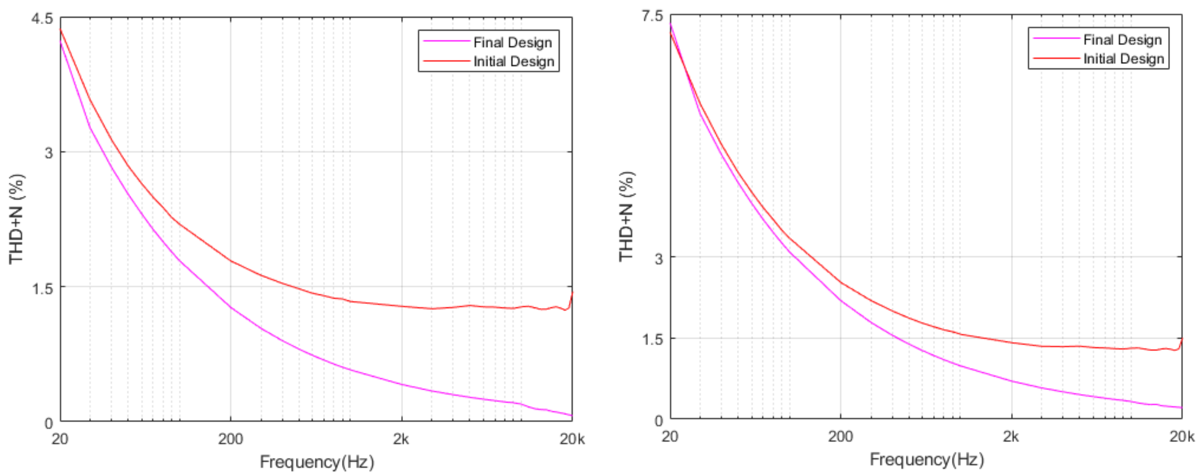
5.3 Signal Generator Test

The results obtained from using the signal generator is described in this section. The SNR and THD+N is determined and plotted.

5.3.1 THD+N

In order to measure THD+N, a sine wave is generated by the signal generator and the amplitude kept fixed while the frequency is varied from 20-20kHz. This is done physically through incrementing the signal generator frequency linearly along the linear x-axis of the semi-log plot. Thus, 37 samples are produced which includes the samples with frequencies ranging from 20-20kHz.

A Matlab script can test each frequency step independently through iterating through the samples. During each iteration, a Notch-filter is applied to the current fundamental frequency in order to measure the noise and harmonics left in the system after the fundamental input has been removed. The rms value of the remaining harmonics and noise are calculated and equation 2.20 is used to calculate the THD+N measurement. This measurement can be plotted as a percentage versus frequency, as shown in figure **??**.



(a) THD+N for Q = 1 (b) THD+N for Q = 3

Figure 5.4: THD+N for initial and final design

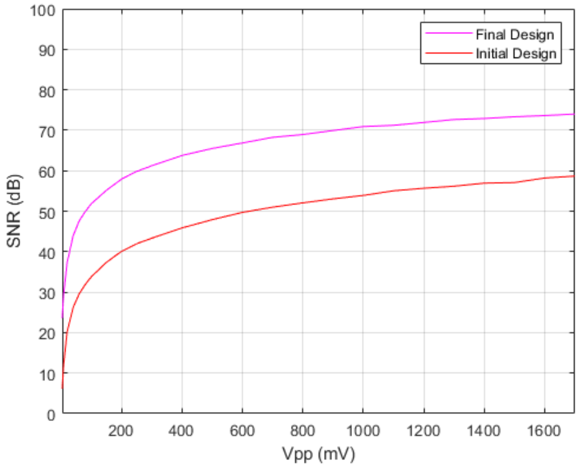
The result shows the difference in THD+N between the initial and final design described in section 5.2.1. The graph reveals that most of the noise occurs due to electrical signals at low frequencies where the plot is the highest. The change in frequency of the input signal results in an exponential decaying slope of THD+N. The THD+N ratio improves by approximately 1.3% as the higher frequencies are reached.

5.3.2 SNR

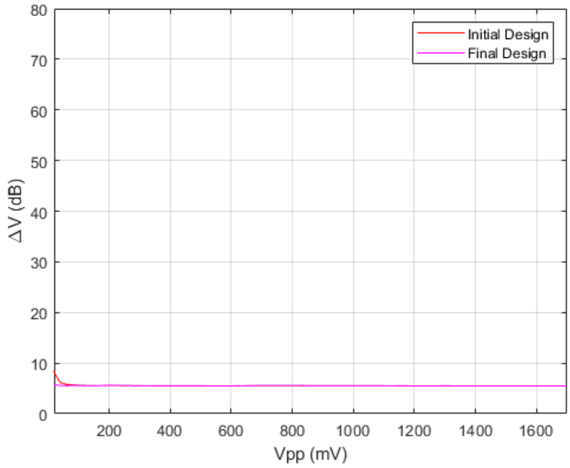
The SNR is measured and calculated through varying the amplitude of a sine wave generated by the signal generator and keeping the frequency fixed at 1kHz with unity gain. The amplitude is varied from the lowest possible setting on the signal generator $4mV_{pp}$ up to $1.7V_{pp}$, which is just below clipping of the signal. This value was not exceeded in order to prevent damage on the audio adaptor board.

The result is shown in figure 5.5a, where the difference can be seen from the initial to the final design. The SNR improves as the peak-to-peak voltage is increased. The final design improves by 25 dB as the peak-to-peak value increases. This value is determined from Matlab's $snr()$ function.

Further tests were done in order to characterise system noise. Through using the sine wave input as described above, the output is measured and the difference calculated between the output and input rms voltages. Figure 5.5b shows the result.



(a) SNR of initial and final designs



(b) Difference between input and output voltage

A constant energy loss of 5.8 dB is observed in the system for both the initial and final designs. This loss is present throughout testing the acoustic monitor.

5.4 Varying Gain

The teensy is used to stream audio data through a USB connection to Audacity. Therefore, the record button can be used in Audacity to record a sinusoidal input during real-time and the method described in Section 4.4.5. The peak-to-peak voltage is adjusted from 50 to 100 mVpp, and then to 200 mVpp. During the first transition, the change in voltage is small and does not result in a peak being generated. During the second transition, a peak is detected and the gain is set to 0 dB.

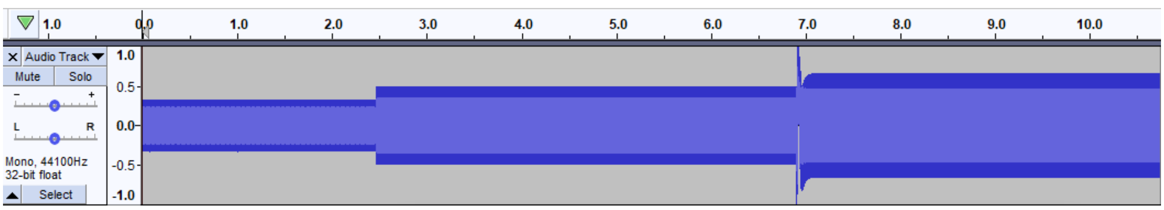


Figure 5.6: Transition in peak-to-peak voltage resulting in clipping

At 2.5 seconds the voltage is altered, but it does not result in a peak occurring. Around the 7 second mark another change in voltage is produced, but this time a voltage spike is visible and results in clipping of the waveform. The serial monitor is set to print a message when a peak is detected and display the change in gain as seen in Figure 5.7. The message is displayed for the duration of the spike.

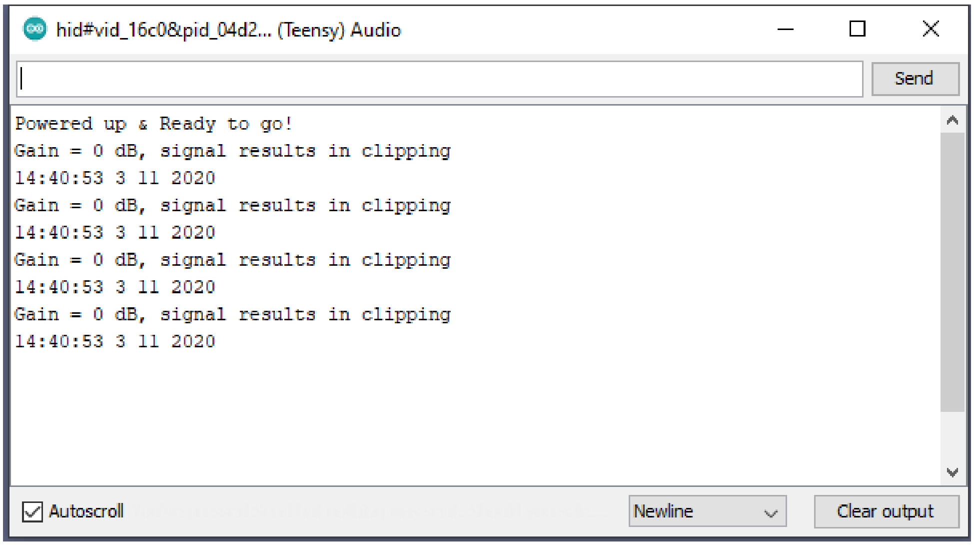
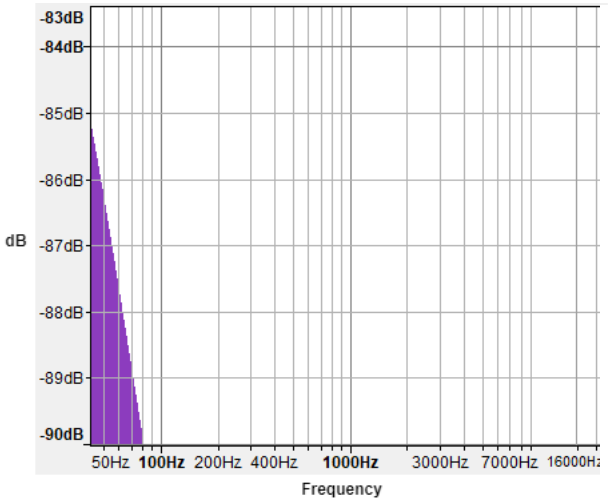


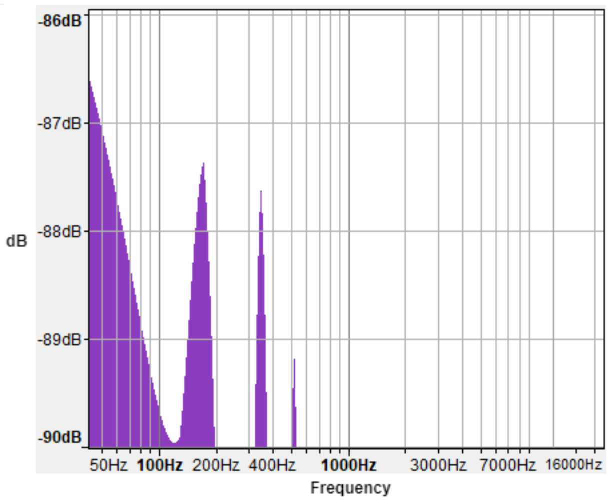
Figure 5.7: Serial monitor display when a peak is detected

5.5 Hydrophone Results

The hydrophone could only be tested by means of a swimming pool. A waterproof speaker was the intended means to produce frequency plots, but did not work as accurately enough to provide a visual display. However, the hydrophone's self-noise could still be evaluated. The gain is tested for 0 dB and 20 dB. 20 dB introduces small noise spikes visible closer to the peak of the noise floor, but the dynamic range is still maintained with the maximum noise floor peak around 86 dB.



(a) Self-noise of hydrophone at 0 dB



(b) Self-noise of hydrophone at 20 dB

This result reveals that the acoustic monitor may be set to record with a gain of 20 dB to allow for amplification of low-level voltages whilst maintaining a good SNR.

Part I

Conclusion

The objectives required to determine the success of a long-term, low-cost acoustic monitor is satisfied from the results obtained. The scheduling operation of the acoustic monitor developed allows for monitoring acoustic sounds underwater during fixed time frames. Therefore, the device may be programmed for certain species spawning periodicity during certain times of the day. Furthermore, the WAV files which are stored are saved using the RTC which allows for the correct date and time formats of the WAV files. Extra functionality such as varying gain is also accomplished. The system designed improved from the initial design up until the final design and is visible from the results obtained.

The SD card introduced noise on to recordings and was mainly visible through listening to the recordings. This is speculated to be noise introduced by SPI communication between the Teensy 4.1 and the SD card module on the audio adaptor board. It cannot be confirmed whether or not this was due to the wrong SD card arriving.

The fact that the acoustic monitor could not be deployed into the ocean at False Bay does affect the validity of whether the device may perform during real-time conditions for its intended use. Simulation of the device in an anechoic chamber and swimming pool only allows for representation of the system at ideal conditions. This result did show that the system has a low noise floor after the modifications were done.

Finally, the cost involved for producing the acoustic monitor was found to be expensive. The performance of the acoustic monitor versus the expenses of hardware required is disproportional, as the SNR and THD+N values may be achieved through the use of different hardware components.

5.5.1 Recommendations for the future

Practical recommendations with regards to project planning is of importance. Initially, a proper project planning schedule was not established. The goal was to develop the system as quick as possible and then try and improve on it. The scope was ill defined from the start, and led to work being done on irrelevant topics with regards to the intended purpose of this project. This may be complex to establish at the start of the project, but a framework should be established in accordance with the study leader.

The MCU and audio adaptor board used to develop the acoustic monitor faced many problems. Apart from the hardware problems faced such as common-impedance coupling, and a large difference in input and output voltage at unity gain, software implementation also had faced issues. The software provided basic implementation of libraries to allow for certain tasks to be done, but when further operations were tried to be implemented on the data, problems were faced. A digital filter could be used to filter the audio signals before being saved onto the SD card for real-time signal processing. This has many advantages, as noise occurring in the system may then already be reduced prior to being stored. Thus, an Arduino MCU or Teensy 4.0 with a audio adaptor board developed through PCB design should be considered, and long term investigation on the Teensy libraries and hardware involved.

A battery management system should also be implemented. The costs involving the project limited the use of acquiring the necessary hardware to implement this, but should be used for the future. The waterproof casing provided also needs to be worked on, as it floats on its side when released into an underwater environment. A weight may be attached on the lid where the hydrophone sits, which will allow for the hydrophone to be directed towards the seabed whilst still floating.

References

[1] B. Traux, *Handbook for Acoustic Ecology.* ARC Publications, 1978.

[2] A. Lindseth and P. Lobel, "Underwater Soundscape Monitoring and Fish Bioacoustics: A Review," 2018.

[3] P. Lobel and D. Mann, *Spawning sounds of the damselfish, Dascyllus albisella (Pomacentridae), and relationship to male size. Bioacoustics.*

[4] K. Dudzinski, *Dolphin Mysteries: Unlocking the Secrets of Communication.* 2008.

[5] "Audacity® —Free, open source, cross-platform audio software for multi-track recording and editing." https://www.audacityteam.org/.

[6] Mathworks, "Matlab." https://www.mathworks.com/products/matlab.html.

[7] Electro-Voice, "THE PA BIBLE: ADDITION NUMBER NINETEEN CONDENSOR MICROPHONES,"

[8] G. Ballou, *A Sound Engineers Guide to Audio Test and Measurement.* Routledge, 2012.

[9] InvernSense, *Microphone Specifications Explained*, 2016.

[10] G. Ballou, *Handbook for Sound Engineers.* Focal Press, 2008.

[11] D. Davis, E. Patronis, and P. Brown, *Sound System Engineering, Fourth Edition.* Focal Press, 2013.

[12] B. Whitlock, "HUM AND BUZZ IN UNBALANCED INTERCONNECT SYSTEMS,"

[13] iac acoustics, "Acoustic Test Facilities: A complete solution to all types of acoustic test facilities,"

[14] V. Todd, I. Todd, J. Gardiner, and E. Morrin, *Marine Mammal Observer and Passivbe Acoustic Monitoring Handbook.* Pelagic Publishing, 2015.

[15] O. Corporation, *Hydrophone Handbook.* 2015.

[16] "Cylindrical vs Spherical Spreading," 2020.

[17] M. Stojanovic, "On the Relationship Between Capacity and Distance in an Underwater Acoustic Communication Channel," 2006.

[18] R. Palmer, "Guidelines for Measuring Audio Power Amplifier Performance," 2001.

[19] H. Wang, "Measurement of Total Harmonic Distortion (THD) and its related parameters using Multi-Instrument," 2020.

[20] J. Proakis and D. Manolakis, *Digital Signal Processing: Principles, Algorithms and Applications, 4th Edition.* Prentice Hall, 2007.

[21] J. Yui, "An overview of the ARM Cortex-M processor family and comparison ," 2017.

[22] "ARM Processor Architecture: Embedded Systems with ARM Cortext-M," 2018.

[23] M. Fisher, *ARM Cortex M4 Cookbook.* Packt Publishing, 2016.

[24] Aquarian Audio, "Aquarian audio products: H2c hydrophone user's guide." https://www.aquarianaudio.com/AqAudDocs/H2c%20manual.pdf.

[25] NXP Semiconductors, *Low Power Stereo Codec with Headphone Amp*, 2013.

Appendix A: Project planning schedule

Week	Date	Project progression
1	27 Jul	Project definition and analysis of similar systems.
2	03 Aug	Research into Acoustic theory.
3	10 Aug	Research into Underwater Acoustic theory.
4	17 Aug	Hardware Development.
5	24 Aug	Hardware Development.
6	31 Aug	Familiarization with microcontroller and Teensy libraries.
Tests	7 Sep	- -
Recess	14 Sep	Software Development.
7	21 Sep	Software development.
8	28 Sep	Software Development.
9	5 Oct	Test and measurement, system calibration.
10	12 Oct	Test and measurement, system calibration.
11	19 Oct	Test and measurement, system calibration and software development.
12	26 Nov	Report Writing.
13	2 Nov	Report Writing.

Table 5.5: Project Planning Schedule

Appendix B: Outcomes compliance

Exit level outcome (ELO)	Motivation	Chapter Addressed
ELO 1: Problem Solving	The acoustic monitor developed required a detailed design in both hardware and software. This resulted in knowledge being required to solve the problem and to acquire new skills. With these new skills, the acoustic monitor was developed and can be viewed in chapter 2,3,4 and 5 details how the problem was solved.	3,4
ELO 2: Application of scientific and engineering knowledge	Scientific relations from digital signal processing and specialist knowledge in software was shown in system design and in the mathematics used to understand acoustics. This can be seen from chapter 2 and 4	2,4
ELO 3: Engineering design	An acoustic monitor was designed and tested through both hardware and software design. This is visible from the methodology, then the detailed design and the results obtained in chapters 3,4 and 5.	3,4 and 5
ELO 4: Investigations, experiments and data analysis	Chapter 2 was necessary to provide information on the testing process with regards to audio systems and what causes noise. Chapter 5 highlights the results obtained from the experiments performed.	2,5
ELO 5: Engineering methods, skills and tools, including information technology	Software tools as described in chapter 2 was used. Digital signal processing tools were used and applied in chapter 4.	2,4,5
ELO 6: Professional and technical communication	Meetings were held whenever possible with my supervisor. Furthermore, communication with other lecturers and staff and the engineering building demonstrate the ability to work with others professionally.	All
ELO 8: Individual work	My supervisor and I held meetings for progress updates. Therefore, all the work done for this project required individual work and learning whatever was necessary to complete the task	All
ELO 9: Independent learning ability	This project required Audacity to be used and learned. It also required the ability to learn about audio waveforms and acoustics in general. My knowledge prior to the undertaking of this project was insufficient and required independent learning.	2,3,4

Table 5.6: Exit level outcome (ELO) assessments

Appendix D: Final Design Project Cost

Hardware	Cost
Teensy 4.1	R917.70
Audio Adapter Board	R506.00
H2c Hydrophone	2 636,20
Latching switch (4's pack)	R51.75
2x Ferrite Beads	R7.48
3x 1.5V batteries	R79
Coin cell battery	R19.55
10kΩ Resistor	Lab Availability
100nF Capacitor	Lab Availability
Schottky diode	Lab Availability
2x Jumper Connectors	Lab Availability
Total	R4217.68

Table 5.7: Component list for final design

Resistors, capacitors, diodes and jumper connections were made available from the Electrical and Electronic Lab at Stellenbosch University. Since only 1 of each were used, the cost is neglected in Table 5.7

Appendix E: Code

Listing 5.1: Arduino and matlab code used to implement software for acoustic monitor functionality

```
/*
  - PROJECT (E) 448 (SKIPSIE == FINAL YEAR PROJECT)
  - Jonathan Paul Hendricks, University of Stellenbosch, ...
     2020
  - TOPIC: Long term-low cost acoustic monitor
  - Monitor dolphin sounds and other marine fauna.
*/

//Includes & Declarations & Constants

#include "WavFileWriter.hpp"
#include <SerialFlash.h>
#include <Audio.h>
#include <Wire.h>
#include <TimeLib.h>
#define FILE_BASE_NAME "REC"
#include <OpenAudio_ArduinoLibrary.h>


const int Fs = 44100;        // Frequency to sample
const int myMic = AUDIO_INPUT_MIC;         // Mic input
unsigned long prevMil = 0;
unsigned long prevMil2 = 0;
unsigned long prevMil3 = 0;
const long interval = 10000;
const long a = 5000;
int flag;
int flag2;
int flag3;
const uint8_t BASE_NAME_SIZE = sizeof(FILE_BASE_NAME) - 1;
char fileName[] = FILE_BASE_NAME "00.wav";
int buttonState = 0;
const int buttonPin = 1;

AudioPlaySdWav         kingSD;
AudioInputI2S          audioIn;
AudioOutputI2S         audioOut;
AudioAnalyzePeak       peak1;
AudioRecordQueue       queue1;
AudioConnection        patchCord1(audioIn, 0, queue1, 0);
```

```
AudioConnection       patchCord2(kingSD, 0, audioOut, 0);
AudioConnection       patchCord3(kingSD, 0, audioOut, 1);
AudioConnection       patchCord4(audioIn, 0, peak1, 0);
AudioControlSGTL5000_Extended  AudioShield;

WavFileWriter wavWriter(queue1);



// ---------------

// Initial setup run at start of program

void setup() {

  SdFile::dateTimeCallback(dateTime);
  Serial.begin(9600);
  AudioMemory(60);
  AudioShield.enable();
  AudioShield.inputSelect(myMic);
  AudioShield.micGain(0);          //0-63
  AudioShield.volume(0.75);         //0-1
  //AudioShield.adcHighPassFilterDisable();
  AudioShield.micBiasEnable(3);
  setSyncProvider(getTeensy3Time);
  Serial.println("Powered up & Ready to go!");
  pinMode(buttonPin, INPUT);


}

// ---------------

//loop function

void loop() {

  if (digitalRead(buttonPin) == LOW) {
    unsigned long currMil = millis();
    if (flag3 == 0) {
      flag3 = 1;
      Serial.println("Start Recording");
      wavWriter.open(fileName, Fs, 1);
    }
    if (currMil - prevMil ≥ interval) {
      while (SD.exists(fileName)) {
        if (fileName[BASE_NAME_SIZE + 1] != '9') {
          fileName[BASE_NAME_SIZE + 1]++;
```

```
          } else if (fileName[BASE_NAME_SIZE] != '9') {
            fileName[BASE_NAME_SIZE + 1] = '0';
            fileName[BASE_NAME_SIZE]++;
          } else {
            Serial.println(F("Can't create file name"));
            return;
          }
        }
        prevMil = currMil;
        Serial.println("Start Recording!");
        wavWriter.open(fileName, Fs, 1);
      }
      if (currMil - prevMil2 ≥ a + interval) {
        prevMil2 = currMil - a;
        wavWriter.close();
        Serial.println("Stop Recording");
        digitalClockDisplay();
      }
      if ((currMil - prevMil3) ≥ a && flag2 == 1) {

        Serial.println("Stop recording");
        wavWriter.close();
        flag2 = 0;
        digitalClockDisplay();
      }
      if (wavWriter.isWriting()) {
        wavWriter.update();
      }
    } else {
      unsigned long currMil = millis();
      prevMil = currMil;
      prevMil2 = currMil;
      prevMil3 = currMil;
      flag2 = 1;
      flag3 = 0;
      wavWriter.close();
      delay(500);
    }
  if (myMic == AUDIO_INPUT_MIC) adjustMicLevel();
}


// ---------------

/*Additional functions*/

//Timing functions

```

```
time_t getTeensy3Time() {
  return Teensy3Clock.get();
}

void digitalClockDisplay() {
  // digital clock display of the time
  Serial.print(hour());
  printDigits(minute());
  printDigits(second());
  Serial.print(" ");
  Serial.print(day());
  Serial.print(" ");
  Serial.print(month());
  Serial.print(" ");
  Serial.print(year());
  Serial.println();
}

/*  code to process time sync messages from the serial port   */
#define TIME_HEADER  "T"

unsigned long processSyncMessage() {
  unsigned long pctime = 0L;
  const unsigned long DEFAULT_TIME = 1357041600;

  if (Serial.find(TIME_HEADER)) {
    pctime = Serial.parseInt();
    return pctime;
    if ( pctime < DEFAULT_TIME) {
      pctime = 0L;
    }
  }
  return pctime;
}

void printDigits(int digits) {

  Serial.print(":");
  if (digits < 10)
    Serial.print('0');
  Serial.print(digits);
}

void dateTime(uint16_t* date, uint16_t* time) {
  *date = FAT_DATE(year(), month(), day());
  *time = FAT_TIME(hour(), minute(), second());
}
```

```

void adjustMicLevel() {
  if (peak1.available()) {
    if (peak1.read() == 1) {
      AudioShield.micGain(0);
      Serial.println("Gain = 0 dB, signal results in ...
         clipping");
      digitalClockDisplay();
    } else {
      AudioShield.micGain(20);
      //Serial.println(peak1.read());
    }
  }
}
```

```
%MATLAB CODE FOR THD+N CALCULATIONS

files2 = ...
    dir(fullfile('C:\Users\jhend_000\Documents\MATLAB\Final_Design', ...
    '*.WAV'));
files3 = ...
    dir(fullfile('C:\Users\jhend_000\Documents\MATLAB\ADCDISABLE_PC', ...
    '*.WAV'));

my_Files2 = natsortfiles({files2.name});
my_Files3 = natsortfiles({files3.name});
y = [20 30 40 50 60 70 80 90 100 200 300 400 500 600 700 800 900 1000 ...
    2000 3000 4000 5000 6000 7000 8000 9000 10000 11000 12000 13000 ...
    14000 15000 16000 17000 18000 19000 20000];

thdPlusN2 = zeros(1,37);
thdPlusN3 = zeros(1,37);
thdPlusNDiff = zeros(1,37);

rmsAns2 = zeros(1,37);
rmsAns3 = zeros(1,37);

rms2 = zeros(1,37);
rms3 = zeros(1,37);

for i = 1:37

    w0 = y(i)/(44100/2);
    bw = w0/1;

    [d, c] = iirnotch(w0,bw);
    [f, e] = iirnotch(w0,bw);

    [myRead2, F2] = audioread(my_Files2{i});
    [myRead3, F3] = audioread(my_Files3{i});

    rms2(i) = rms(myRead2);
    rms3(i) = rms(myRead3);

```

```
    rmsAns2(i) = rms(filter(d,c,myRead2));
    rmsAns3(i) = rms(filter(f,e,myRead3));

    thdPlusN2(i) = sqrt((power(rmsAns2(i),2))/(power(rms2(i),2)))*100;
    thdPlusN3(i) = sqrt((power(rmsAns3(i),2))/(power(rms3(i),2)))*100;
    thdPlusNDiff(i) = thdPlusN3(i) - thdPlusN2(i);

end

    semilogx(y,thdPlusN2, 'm');
    xlabel('Frequency(Hz)')
    ylabel('THD+N (%)')
    xticks([20 200 2000 20000])
    yticks([0 1.5 3 4.5])
    xticklabels({'20','200','2k','20k'})
    yticklabels({'0','1.5','3','4.5'})
    xlim([0 20000])
    ylim([0 4.5])
    grid on
    hold on

    semilogx(y,thdPlusN3, 'r');
    xlabel('Frequency(Hz)')
    ylabel('THD+N (%)')
    xticks([20 200 2000 20000])
    yticks([0 1.5 3 4.5])
    xticklabels({'20','200','2k','20k'})
    yticklabels({'0','1.5','3','4.5'})
    xlim([0 20000])
    ylim([0 4.5])
   % title('THD+N versus Frequency Comparisson Between Initial and ...
      Final Designs')
    grid on
    legend('Final Design','Initial Design')
```

```
%MATLAB CODE FOR SNR CALCULATIONS
snrFiles1 = ...
    dir(fullfile('C:\Users\jhend_000\Documents\MATLAB\SNR_varyin_FINALDESIGN', ...
    '*.WAV'));
snrFiles2 = ...
    dir(fullfile('C:\Users\jhend_000\Documents\MATLAB\Test_A', '*.WAV'));

my_snrFiles1 = natsortfiles({snrFiles1.name});
my_snrFiles2 = natsortfiles({snrFiles2.name});

rmsOut1 = zeros(1,25);
rmsOut2 = zeros(1,25);
rmsRatio1 = zeros(1,25);
rmsRatio2 = zeros(1,25);
SNR1 = zeros(1,25);
SNR2 = zeros(1,25);

rmsT1 = zeros(1,25);
rmsTrueNoise1 = zeros(1,25);
rmsTrueNoise2 = zeros(1,25);
x1 = [4 8 20 40 60 80 100 150 200 250 300 400 500 600 700 800 900 ...
    1000 1100 1200 1300 1400 1500 1600 1700];
```

```
mySnr1 = zeros(1,25);
mySnr2 = zeros(1,25);
SNRDiff = zeros(1,25);

for i = 1:25

    [myOutRead1, FS1] = audioread(my_snrFiles1{i});
    [myOutRead2, FS2] = audioread(my_snrFiles2{i});

    rmsT1(i) = ((x1(i)/2)*0.7071)/1000;
    rmsOut1(i) = rms(myOutRead1);
    rmsOut2(i) = rms(myOutRead2);

    rmsTrueNoise1(i) = abs(rmsT1(i) - rmsOut1(i));
    rmsTrueNoise2(i) = abs(rmsT1(i) - rmsOut2(i));

    SNR1(i) = 20*log(rmsOut1(i)/rmsTrueNoise1(i));
    SNR2(i) = 20*log(rmsOut2(i)/rmsTrueNoise2(i));

    mySnr1(i) = snr(myOutRead1, FS1);
    mySnr2(i) = snr(myOutRead2, FS2);
    rmsRatio1(i) = rmsT1(i)/rmsOut1(i);
    rmsRatio2(i) = rmsT1(i)/rmsOut2(i);

    SNRDiff(i) = mySnr1(i) - mySnr2(i);
end

figure
plot(x1,SNR2, 'r');
xlabel('Vpp')
ylabel('dB')
grid on
hold on

plot(x1,SNR1 , 'm');
xlabel('Vpp (mV)')
ylabel('{\Delta}V (dB)')
xlim([20 1700])
ylim([0 80])
%title('Difference between Input and Output Voltage Versus ...
    Peak-To-Peak Voltage Comparisson Between Initial and Final Designs')
grid on

legend('Initial Design','Final Design')
hold on

figure
plot(x1,mySnr1, 'm');
xlabel('Vpp (mV)')
ylabel('SNR (dB)')
xlim([4 1700])
ylim([0 100])
grid on

hold on

plot(x1,mySnr2, 'r');
xlabel('Vpp (mV)')
```

```
ylabel('SNR (dB)')
%title('SNR Versus Peak-To-Peak Voltage Comparisson Between Initial ...
    And Final Designs')
grid on
legend('Final Design', 'Initial Design')

hold on
```