FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

MASTER IN ELECTRICAL AND COMPUTERS ENGINEERING

Adaptation of an Harp for MIDI Implementation and Sound Amplification

MASTER THESIS DISSERTATION

Author:
João Miguel A. Beleza

Supervisor:
Prof. Aníbal Ferreira

24 June, 2019

Abstract

The objective for this thesis project will be to adapt an Harp for MIDI implementation together with sound amplification. To accomplish that objective, this project will focus on studying all available products that can provide these functionalities with the intention of developing its own product while remaining competitive in the market. To define priorities the MIDI implementation will be the main focus of the project while the sound amplification will be the secondary objective.

This thesis dissertation will aim to provide a walk through process justifying all the decision making regarding both implementations. The first step will be to study and develop tools that allow data to be collected from any device chosen. Secondly, this project will focus on the study of piezoelectric transducers in order to make the bridge between the string vibration of the harp and the corresponding electrical signal. Following this analysis, the MIDI implementation will be developed using the already established transducers mechanism together with the sound amplification system. Finally, a very brief market analysis justifying and validating the product will take place for the finalized product.

Resumo

O objetivo para este projeto de Tese será adaptar uma harpa para implementação MIDI em conjunto com amplificação sonora. De modo a cumprir este objetivo, este projeto irá primeiramente focar-se no estudo de todos os produtos disponíveis no mercado capazes de realizar tais funcionalidades. Este estudo servirá como base para desenvolver um protótipo funcional competitivo no mercado actual. De modo a definir prioridades a implementação MIDI servirá como foco principal do projeto e a amplificação sonora será o objetivo secundário.

Esta dissertação irá focar-se em providenciar uma descrição passo a passo da maioria das decisões tomadas para desenvolver ambas as implementações. O primeiro passo será estudar e desenvolver ferramentas que permitam recolher e guardar dados dos dispositivos escolhidos. De seguida, este projeto irá focar-se no estudo de transdutores piezoelétricos que permitam fazer a ponte entre a vibração das cordas da harpa e do correspondente sinal elétrico. Posterior a esta análise, a implementação MIDI em conjunto com a implementação de áudio será desenvolvida utilizando o mecanismo desenvolvido. Por fim, uma muito breve análise de mercado será fornecida de forma a validar e justificar o produto final.

Acknowledgements

I would like to thank my coordinator Prof. Aníbal João de Sousa Ferreira who not only accepted and made possible the initial proposal as well as helping and guiding me throughout the whole project development. I would like to thank my family especially my mother, Maria do Rosário Beleza, and my father, Álvaro Manuel Beleza, for supporting me in this project as well as my grandfather, José Antonino Beleza, who helped me develop many of the more practical aspects of this endeavour.

I would also like to thank my friends who made all the effort possible through good and bad moments of this project, those who are in my heart, those who are not here, those who kept me from failing and those who lend their ear. A special thanks to the girl with sapphires, the LM couple, the best work partner I have ever had, the future harpmaker and the best friend this man has had, you all made all of this possible.

Contents

List of Figures

List of Tables

Abbreviatures and Symbols

Piezo	Piezoelectric
ADC	Analog to Digital Converter
DAC	Digital to Analog Converter
DIY	Do IT Yourself
FT	Fourier Transform
DFT	Discrete Fourier Transform
FFT	Fast Fourier Transform
$M\Omega$	megaOhm
$k\Omega$	kiloOhm
Ω	Ohm
μs	microSecond
ms	miliSecond
V	Voltage
V_{pp}	Voltage peak-to-peak
F	Farad
pF	picoFarad
μF	microFarad

Version Control

Version	Date	Changes / Motivation
1.0	02/02/19	Document Creation
1.1	10/02/19	Validation of Preliminary Document
2.0	18/06/19	Final Thesis First Submission
2.1	22/06/19	Final Thesis First Validation
3.0	23/06/19	Final Thesis Validated

Table 1: Version Control.

1 Introduction

This section will approach the subject of *Sound to Midi* conversion regarding its context, motivation, application and overall use cases. Consecutively, an introduction will be provided to the harp as an instrument with regards to its purpose and usage. In conclusion a relation between these two topics will be provided justifying the project's motivation and objectives.

1.1 Brief Context

1.1.1 Sound to Midi Conversion

Sound can be defined as a vibration that propagates as an audible wave of pressure through a transmission medium. This wave has several characteristics associated with it however, the main ones for this project are pitch (frequency) and velocity (amplitude). In order to replicate and record these waves there have been several approaches throughout the years, all being a variation of what is defined as a microphone. Essentially, the microphone consists of a small piezoelectric transducer which converts the sound wave into an electric one. This conversion has a higher quality the better the wave characteristics are converted to an electrical signal.

This electric signal can then be modified and processed and even converted again to a sound wave through an amplifier circuit. However, this analog signal cannot be directly interpreted by a digital system, which can only accept a binary language made of 1's and 0's. For that reason ADC's (*Analogue to Digital Converters*) and DAC's (*Digital to Analog Converters*) were created to make the bridge between the digital and analog worlds. Nowadays, regarding musical applications, both analog and digital signals have their place and specific purposes.

Even though the bridge was created there is still a need to convert what is being played to a digital signal, in other words and using an analogy: if a pianist played three keys in a piano the computer would know what sound they reproduced but not which keys were pressed by the pianist. Such information could be particularly useful if, for example, the pianist wanted to write a musical sheet by sending the notes being played to a computer in real-time. The need for this technology resulted in the creation of the MIDI protocol.

MIDI, which stands for *Musical Instrument Digital Interface*, is a standard protocol created in 1982 by Dave Smith and it is used to communicate between instruments or computers using simple messages. Simply put, each time a key is pressed a message is sent informing which note was played (pitch) and how hard it was pressed (velocity). These messages are encoded using the MIDI protocol. Since its creation, the MIDI protocol has grown to accept many other inputs and controls even though most of them are still a variation of this initial design.

The MIDI protocol rapidly became common in the music industry allowing a standard language for instruments from different companies to communicate. Many digital instruments nowadays, if not all of them, are MIDI capable and have it integrated from design. In essence MIDI allows any computer, or MIDI capable machine, to interpret and eventually record and store all the notes and characteristics from the instruments that are being played in real-time.

1.1.2 Introduction to the Harp

The Harp is a stringed musical instrument with its origin dating back as early as 3500 BC. Throughout the years there have been many adaptations and variations of the harp. Nowadays, the most common designs are the Celtic Harp [Fig: 1] and the Pedal Harp [Fig: 2]. The two are used in several environments, from orchestra to street playing, and included in many music genres from classical to jazz. In latest years, there has also been an introduction of the harp in the pop culture as well.

Figure 1: Celtic Harp

Figure 2: Pedal Harp

The main difference between the Celtic and Pedal Harp are size, weight, sound richness (due to a bigger body usually referred to as the belly) and pitch control. The last topic is actually one of the main differences between the two. In the Celtic harp, the pitch of each individual note has to be changed by hand using a lever on the top of the string. This allows for each individual string to be set to natural or flat.

In the Pedal harp however the pitch is controlled using pedals which allows for a set of strings to be changed to sharp, natural or flat. In the schematic [Fig: 3] it is possible to observe that one pedal is associated with each of the individual notes on a scale (A through G) with three different positions for each one. As an example setting the A pedal in the upper position will set all the A strings in the harp to flat.

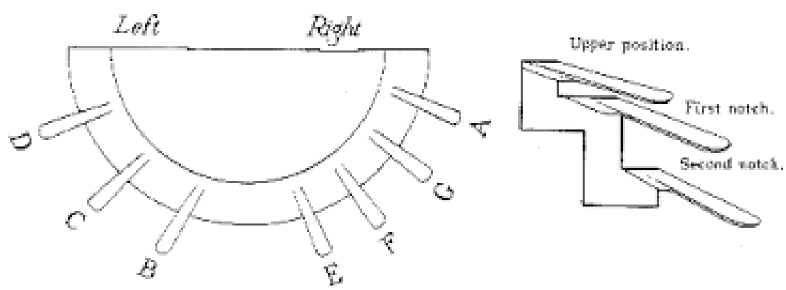


Figure 3: Schematic of the System of a Pedal Harp

1.2 Motivation for MIDI adaptation of an Harp

As previously stated, many instruments have been adapted to natively support MIDI integration and sound pre-amplification with the most common ones being keyboards and pianos. Since the *90's* decade, many other instruments such as guitars or drums, among others, have been adapted to implement the same kind of features.

Besides the reasons stated in section 1.1.1 this adaptation allows players to easily use the instrument they're most comfortable with to reproduce sounds from other instruments. Since a computer is receiving the notes being played by, for example, a piano, it can easily reproduce the same notes using a saxophone sound or any other. This scenario allows players for a larger flexibility in music production providing an alternative for learning a new instrument each time a different sound is needed. The Harp is included in this scenario with some adaptations already available but mostly leaving a lot to be desired, either because of lack of functionality or price.

Regarding sound amplification in the harp, the most common adaptation is a microphone located close to the belly of the harp. This approach can work well in an enclosed space but not only is unbalanced capturing the upper, middle and lower strings with a constant tone, due to different distances from the microphone, but also captures other instruments when playing in an orchestra. To fix this problem, the method evolved for smaller microphones which can be insert inside the harp however, these are very prone to amplify bumps and scratches and do not entirely fix the first problem presented.

Recently, some adaptations were made using single piezoelectric transducers per string which are able to capture the sound of the harp with a very complete and rich result. Harp building companies such as CAMAC [1] and Salvi [2] have launched a few models since 2009. Some can only be played when amplified and other can be both played normally and amplified. These have their respective uses and are very appreciated by players, but expensive to obtain (in the order of the 20,000$ to 30,000$).

Regarding MIDI adaptation, private developers such as Kortier [3] and Mountain Glen Harps [4] have published models using single piezoelectric transducers per string connected to an external device. These harps have received very good reviews from players and, in regards to price, the addition of a MIDI system to a pre existing harp round the 400$ to 5,000$. However these harps are only MIDI capable and cannot be amplified using the piezoelectric transducers, only by using microphones. The company Camac has also published a MIDI capable harp but it is only available for private endeavours and is not available for the general public to purchase.

As it stands today there is a need for a more competitive harp on the market which can implement both MIDI and sound amplification, by using an optimized approach such as piezoelectric transducers, while still remaining capable of being played as a regular harp. These harps should be made more available to the general market instead of alternatives that depend on custom developed harps, made for a specific kind of public.

1.3 Main Objetives

With this thesis project, the main objective will be to study and, if possible, develop a prototype which can simultaneously implement a MIDI capability and sound amplification using piezoelectric transducers. If possible the price of final implementation should be reasonable and within the typical market frame of 1,000$ to 3,000$. Furthermore, this project should have both a scientific approach to solving the problem as well as a business approach to validate the project. Several approaches will be tested and compared. The best of them will be optimized and studied in more detail. This study will conclude with a very brief market analysis of how competitive a product such as this could be in the market today.

2 Fundamentals Overview

This section will take a more detailed overview of what has been done to make a MIDI adaptation and amplification of a harp by several harp makers. Following, there will be an individual approach to each topic of the project using as a guide the diagram in Figure 4.

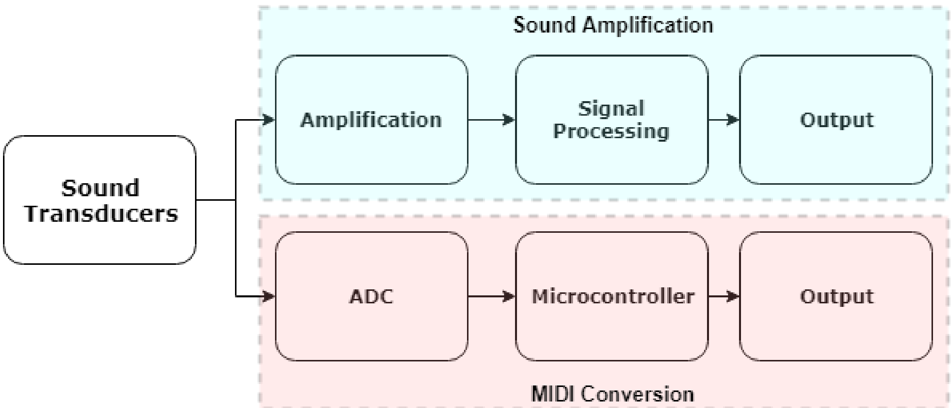


Figure 4: Overview of the Problem

2.1 Approach Comparison

2.1.1 MIDI Conversion

As of today there are three main MIDI capable harpmakers/companies: Camac [1], David Kortier Harpmaker [3] and Glen Mountain Harps [4]. All of these examples use a similar approach with individual piezoelectric transducers placed at the bottom of the string making the conversion to MIDI through an internal circuit. Considering the celtic harp, only Mountain Glen has made full adaptations to MIDI, with the harp remaining capable of being used in a regular way.

In order to illustrate Figure 5 and Figure 6 show the comparison between a harp capable and not, respectively, of being used in a regular way (without amplification), with the biggest difference being on the size of the belly.

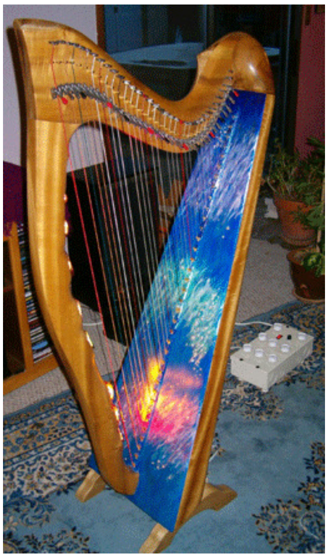


Figure 5: MIDI implementation on a regular harp

Figure 6: MIDI implementation on a non regular harp

For both Celtic and Pedal harps, these are only sold as MIDI or sound amplification harps and are individually made or adapted from pre-existing harps. In the website [4] some specifications are given stating the system provides both MIDI ON and OFF (informing when a note starts and stops vibrating) with the respective velocity. Also, for further reference while developing, it is advertised that the total sampling and acquisition time is 8 milliseconds assuming a typical harp with 45 strings, even though not directly stated.

The Kortier MIDI system is sold in separate for non regular harps or adapted to pre-existing harps. No specifications are given in detail but from a previous personal experiences they only provide a MIDI ON information even though this characteristic may have changed in more recent harps. Details regarding the meaning of these MIDI message will be explained in more detail in chapter 2.2.2.

Some adaptations have been done by other private companies using the separate Kortier system however most of them result in a poor quality due to poor tuning or lack of understanding of the implemented system. The Camac version, as it is not available to the general public, does not provide many information about its adaptation with only live performances available.

Regarding Pedal harps it is not clear for any of them if the pedal mechanism makes any influence in the MIDI adaptation, which should result in a pitch variation. For Kortier and Mountain Glen Harps the resulting price of the MIDI adaptation alone can vary from 3,000$ to 5,000$, not including the harp.

2.1.2 Sound Amplification

Sound amplification capable harps, or more generally called electroacoustic harps, are much more common nowadays. Several companies such as Glen Mountain Harps [4], Camac [1], David Kortier Harpmaker [3] and Salvi [2] have presented versions of electroacoustic harps. Likewise, all the companies mentioned before sell both Pedal and Celtic electroacoustic harps. As a note, in the beginning of the 2010 approaches varied from microphone to piezoelectric however, today, all companies use a variation of the piezo pick-up placed at the bottom of the string.

As mentioned in section 1.2 there are no harps announced which implement both MIDI and sound amplification using piezoelectric transducers. The reason for that is not mentioned in any company as the mechanism as been proven useful for both implementations. Besides, it is clearly mentioned in the Kortier and Glen Mountain websites that MIDI harps do not implement the amplification system as of today.

For all the systems mentioned the amplification is usually done by some passive filtering on the piezo which is then connected to an already made pre-amplifier from companies such as Fishman [5]. This implementation can be seen in Kortier and Salvi harps, for example, however due to the pre-amplifier not being made specifically for the intended use the results can vary in quality.

2.2 Fundamentals

2.2.1 Sound Transducers

The sound transducer is the most important part of the whole project as it makes the conversion of the sound characteristics. One of the possible approaches to convert the vibration of the string into an analogue signal is the piezoelectric transducer element, mentioned from now on as piezo for shortening. As very well explained in *The Principles of Piezoelectric accelerometers* [6] piezo materials consist of active electrical elements that produce an electrical output when excited by a varying force. They're usually made of a ceramic element, such as quartz, which possesses a crystalline structure. This structure, when submitted to a mechanical force will respond with a voltage proportional to the acceleration applied to it. In Figure 7 [7] it is observable the behaviour of a piezo element under different conditions with the respective voltage response.

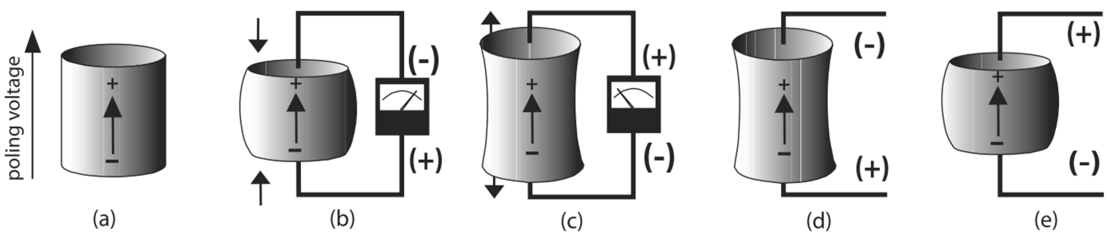


Figure 7: Piezo voltage response to an applied voltage or mechanical force

This voltage occurs as the electrons inside the crystalline structure redistribute themselves when a force is applied and, if connected to a closing circuit, even leave the crystalline structure thus creating room for new electrons to enter the crystal. This movement generates an electromotive force that will stimulate charges to move around a circuit or within the piezo. The opposite can happen as well as by submitting the piezo material to an electrical field it will cause the crystalline structure to deform and generate a proportional physical force.

The piezo material can be, and has been, studied in many details however for this project only its electrical response is required for a good appliance. Without going into much detail, for the piezo element to properly work it needs a polling treatment. Usually a non treated piezo element will have dipole moments within its crystalline structured oriented in almost random directions. The treatment usually consists of placing electrodes to the surface of the piezo and applying a high DC voltage. After the treatment the piezo material will be polarized in the direction of the applied DC voltage and contain dipole moments fixed into a permanent configuration. This allows the behaviour of the piezo to be predictable when applying a force in a specific direction.

Ideally, as shown in Figure 7 a force applied in the polarization direction will result in a positive voltage while a force applied against the polarization direction will result in a negative voltage. The polarization direction is very important as forces submitted to the piezo in different directions will result in different behaviours, even when applied outside the polarization direction. A complementing explanation of piezoelectricity especially in regards to its fabrication materials and procedure can be found in the book *Piezoelectric Sensorics* [8].

In order to use the piezoelectric within a circuit a stable model needs to be implemented. As stated above, a piezo material will convert mechanical stress into an electrical charge. This relationship is named *piezoelectric charge constant* and is defined by the constant d which can be expressed in Coulombs (charge) per Newton (force) (1). This constant is associated with each piezo and varies with the direction of the applied force, always referencing to the polarization direction.

$$d = \frac{Q}{F} \qquad I = \frac{dQ}{dt} \qquad V = \frac{1}{C}\int I\,dt = \frac{1}{C}\int \frac{dQ}{dt}\,dt = \frac{Q}{C} \tag{1}$$

Due to the lack of "charge sources" it is possible to make the conversion between charge and current using the formula in (1). As currents only varies when there is a variance in charge and the charge within a piezo is never constant when generating a voltage making the conversion to a current source gives a good approximation of its behaviour.

As an example, if a force is applied to a piezo and remains constant the piezo will produce a voltage until the equivalent opposing force of the piezo becomes the same value as the initial one. When this happens the piezo will no longer produce any voltage as there is no varying charge. Furthermore, until now voltage has been mentioned even though the circuit has been translated as current source. Usually the piezo element is connected to the circuit using two electrodes in each side as displayed in Figure 8 which forms a capacitor behaviour. This capacitor in parallel with the current source will generate a voltage behaviour presented in formula (1).

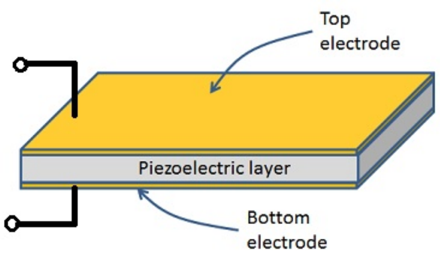


Figure 8: Capacitive behaviour of a Piezo Element

This model is almost ideal for transient and dynamic applications but a more accurate representation also needs to account for the discharging of the piezo through current leakage, which can be represented by a parallel resistor. The final model is represented in Figure 9.

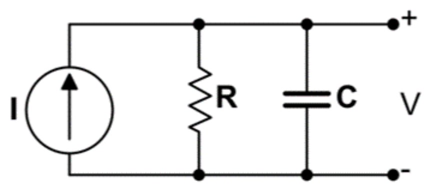


Figure 9: Equivalent Piezo circuit model

Regarding frequency response piezo elements have the typical frequency response of an high-pass filter with a resonance frequency. The behaviour is ilustrated in Figure 10 [9].

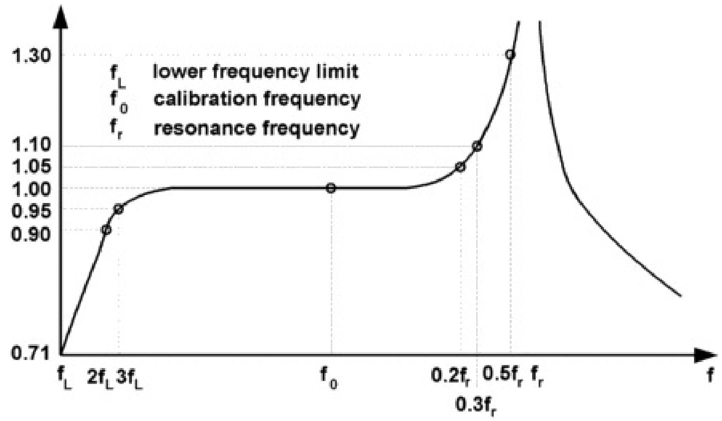


Figure 10: Frequency response of a piezo element

The values shown in Figure 10 are typical values for a piezo however these characteristics may vary a lot, especially for custom made piezos. Usually, piezos are defined by its resonance frequency. When using the piezo as a sensor this frequency should be as high as possible which allows sensors to work near the calibration frequency and provide more accurate and stable results as the gain should be 1. The resonance frequency coincides with the spot where the impedance of the piezo will be the lowest as possible. A typical piezo impedance curve is shown in Figure 11. The lowest peak represents the resonance frequency while the highest peak represents the anti-resonance frequency, where impedance takes its highest value.

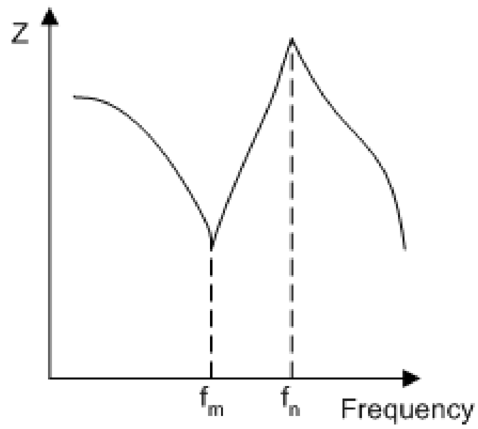


Figure 11: Impedance per frequency response of a piezo element

Some circuits, such as piezo quartz crystals oscillators, make use of this resonance frequency to oscillate at a precise frequency where the impedance is the lowest as possible. This kind of circuits behave differently from the ones in study and cannot be represented by the model presented in Figure 9. Ideally for this project, the resonance frequency will be as high as possible and the lower frequency limit as low possible guaranteeing a wide bandwidth to work. As a note, the working bandwidth can be a little extended using a low pass filter with a cut-off frequency near the resonance frequency but this can only attenuate the resonance frequency.

The frequency response described before can be suitably modeled by equation (2). This equation shows the relation between the gain output (G_o) relative to the reference gain G_r (where $\frac{f}{f_R} = 1$) and the input frequency f relative to its resonance frequency f_R. The Q value, not to mistake with charge as described above, or quality factor, represents the "sharpness of the resonance" for each given piezo. Basically it defines how sharp, or wide, its resonance peak is. The higher the value, the sharpest the resonance frequency but also the higher the equivalent output gain.

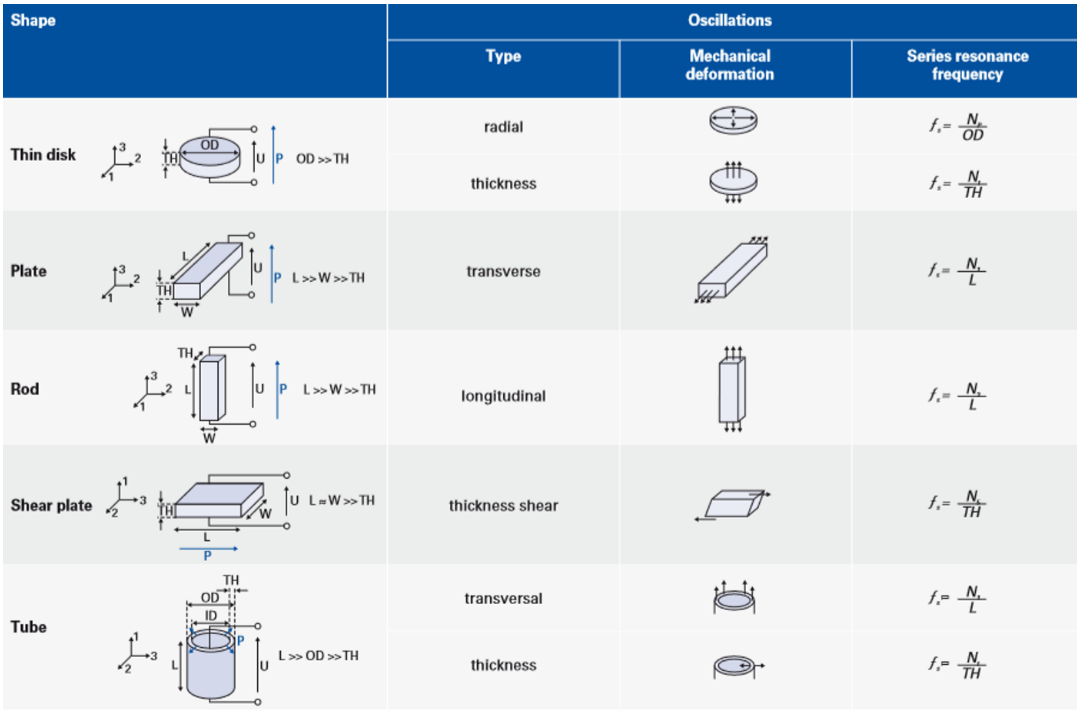
$$\frac{G_o}{G_r} = \frac{1}{\sqrt{(1-(\frac{f}{f_R})^2)^2 + \frac{1}{Q^2}(\frac{f}{f_R})^2}} \qquad phase\ lag\ (^\circ) \approx \frac{60}{Q}\frac{f}{f_R},\ for \frac{f}{f_R} \leq \frac{2}{5} \tag{2}$$

The phase lag, which determines the lag in phase since the signal is applied until it is seen at the output, is also represented in equation (2). However, the phase lag is only relevant for high frequencies and for inputs that cross the piezo. Since the piezo will originate all the signals to be process phase lag will not have much, if any, impact in the project.

As stated before, the circuit should behave outside the resonance frequency however it is possible for some piezos, especially low quality piezos, to have low resonance frequencies. This means that even if the string applied to a piezo contains a fundamental frequency below the resonance frequency its harmonics, or overtones, may be beyond that point and get further amplified. Not only these frequencies get proportionately amplified as they may generate larger voltages than the circuit can handle.

As stated, this effect could be mitigated using a low pass filter but there is a trade-off between attenuating harmonics as they are the very essence of each instrument and serve to characterize its unique sound. This serves a note for later take into account during the development phase of the project.

Also, if not indicated by the manufacturer, according to the company PiMicos guide [10] it's possible to extrapolate the piezo series resonance frequency as exposed in Figure 12 by direction of applied force and dimensions of the piezo. This series resonance frequency is a good approximation of the resonance frequency. This could be particularly useful for more undifferentiated piezos.



Shape	Oscillations		
	Type	Mechanical deformation	Series resonance frequency
Thin disk (OD >> TH)	radial		$f_s = \frac{N}{OD}$
	thickness		$f_s = \frac{N}{TH}$
Plate (L >> W >> TH)	transverse		$f_s = \frac{N}{L}$
Rod (L >> W >> TH)	longitudinal		$f_s = \frac{N}{L}$
Shear plate (L = W >> TH)	thickness shear		$f_s = \frac{N}{TH}$
Tube (L >> OD >> TH)	transversal		$f_s = \frac{N}{L}$
	thickness		$f_s = \frac{N}{TH}$

Figure 12: Resonance frequency by piezo shape

2.2.2 MIDI Conversion

To convert the output of the piezo into an electrical signal an ADC needs to be implemented. Most single board micro controllers such as the Arduino, which contains the ATMEGA328P [11] chip, or the most recently developed Teensy with a Cortex®-M4-based micro controller [12] made specially for music applications, have embedded ADCs to convert the analog signal into a digital one.

Depending on the output of the piezo the signal may need some amplification and some filtering, such as a band pass filter, in order to present a readable signal for the ADC. This signal is usually contained within a voltage range and presents a maximum output impedance associated. An external ADC could be implemented but documentation of both microcontroller's ADC have proved them effective to fulfil the need. Another advantage of using a microcontroller is the possibility of converting the ADC output to a MIDI message using the same device which reduces the overall area of the circuit and general latency. Using a single ADC for each piezo could prove effortless as not all the strings in the harp are being plucked at the same time.

A typical and simplified output signal of the piezo is represented in Figure 13. Although this response varies from instrument to instrument (and from type of string to string) and even from each piezo a common feature among all are the peaks associated with the decaying velocity of the string. By measuring the first peak, it is possible to extrapolate the force with which the string was plucked in the first time. Afterwards, by measuring each subsequent peak it is possible to make an assumption of the evolution of the velocity of the string after being plucked.

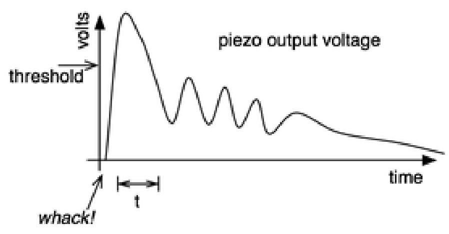


Figure 13: Typical piezo response

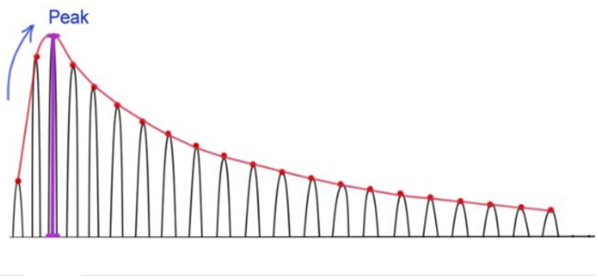


Figure 14: Simplified peak acquisition

Ideally, if the string goes above a threshold a MIDI ON message will be sent with the corresponding velocity associated to the peak. As the signal decreases in amplitude, if the signal goes below a certain threshold a MIDI OFF message is sent. If the player stops the strings while its vibrating the behaviour should also be translated into a MIDI OFF message. All these messages will follow the standard MIDI protocol [13] which allows for the communication between instruments. As a curiosity, while this project is being developed MIDI 2.0 was announced bringing some new features, even though its basic operations remain almost identical. However, this project will focus more on having a good interoperability more than state of the art technology, for now.

An example of a MIDI message is represented in Figure 15. To inform the computer that a note was played the algorithm will send three distinct message. The first, the Status Byte, will inform that a Note will be sent together with the corresponding channel, (the channel is a similar to an IP address when several computers are connected in the same network); afterwards, the Data Byte 1, which will inform exactly which note was played within a 127 possible range; lastly the Data Byte 2 will inform of the velocity associated with the corresponding note, also a value between 0 and 127. A MIDI OFF message would contain all this information with the Data Byte 2 set to 0.

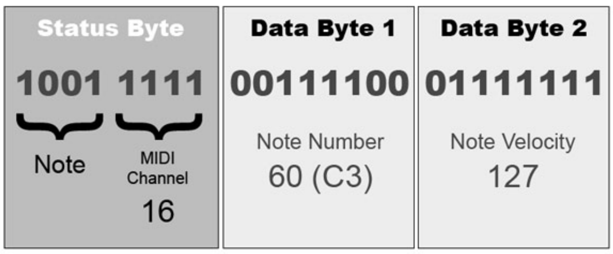


Figure 15: Example of a MIDI message

At the output a typical MIDI output socket or USB connection can serve as a bridge to connect to other instruments or MIDI capable devices..

It is important to mention that a good MIDI conversion has a general latency of ~5ms with a maximum delay of ~10ms. A reasonable conversion should not only give accurate results but also give them within a time window.

2.2.3 Sound Amplification

Regarding amplification of the piezo there are several already developed pre-amplifiers for many different applications. Regarding audio these usually vary considering the instrument or audio input that needs to be amplified. Nonetheless, the majority are usually pretty expensive and most importantly, are not developed for the specific instrument. For a guitar or piano it is possible to find a unique developed circuit however outside of this scenario it becomes rather difficult.

In this project since there will be several piezo inputs there is a need for a simple amplifier solution for each individual piezo. Further on, after this first stage amplification, the sound can be individually filtered and the sum of all the signals can be driven through a common filtering process and final tunable gain. This approach allows for a more precise tuning of each single piezo in the first stage amplification followed by a final tuning of the sum of all the signals in a second stage amplification.

Regarding the first stage many circuits have been published and proposed to amplify a piezo signal. A very simplistic approach, found in a DIY (Do it Yourself [14]) project can be found in Figure 16. This design consists of a very basic voltage divider, used to control the voltage output of the piezo, which could be replaced by a potentiometer for more accurate tuning, a high pass filter, in order to remove noise and interference, and finally a single Jfet to amplify the sound. At the output of the Jfet the the capacitor and resistor serve to remove any DC signal and bring everything to a typical line level. The values for each component should be designed according to the specific piezo in use.

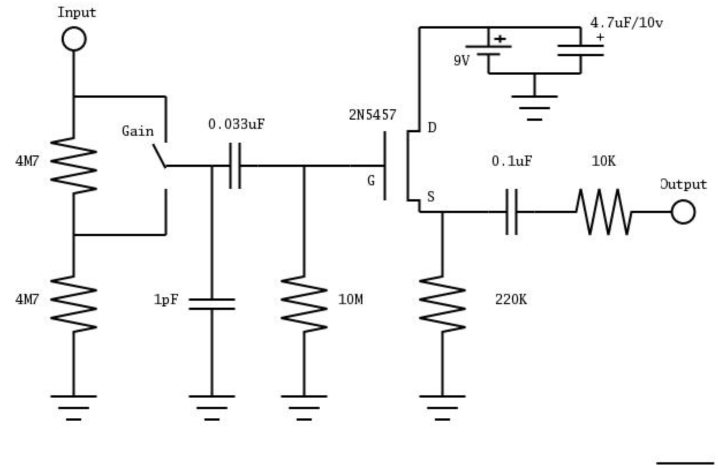


Figure 16: First Stage Amplification Example

For the second stage a simple application using an op-amp might be the best choice with a tunable filter for a more accurate approach. A design made for 5 channels is available in [15]. Despite all of these approaches it is most important to first study the response of the piezo and then build the system around it.

Also, while using piezos the most important feature is to match the output impedance of the piezo with the input impedance of the amplifier. A mismatch between these two devices can really affect the sound output, leading to inconsistent or very attenuated gains. It is also important to consider leakage current in order to minimize the circuit power consumption.

As a conclusion, ideally, it is important to keep a maximum voltage and signal quality the closest to the amplifier mechanism as it is easier to make tunings regarding audio when converting from higher to lower voltages than the opposite. Keeping minimal distortions while still conditioning the signal to an audio output is the main objective for the audio implementation. To verify these parameters, even though designed specifically for mosfets, a very well written guide regarding testing can be seen in [16].

3 Inital Approach

In this section an initial approach to the project will be described together with the discrete objectives the project will aim to fulfil. This section will not go into much detail regarding deployment of the implementation as that will heavily depend on experimentation and testing. It will serve more as a guide to develop the rest of the project.

3.1 Discrete Objectives

In order to verify and validate the project some objectives need to be defined. These are:

Main Objectives

- The transducers need to capture the vibration of a string of an harp and convert it to an electrical signal;
- The MIDI circuit, using a micro controller, must be able to convert the signal from the transducers into a MIDI message;
- The transducers must allow the musician to play without disturbing regular use of an harp;
- Player controllability of the circuit should be maximized;
- The MIDI conversion should be done within the normal latency requirements: maximum 10 ms, optimal should be around 5ms;

For this project the MIDI implementation serves as the main driver and the amplification module serves as a secondary driver. If possible both should work using the same piezo and possibly work at the same time. However if not possible there should be a switch to toggle the piezo to be amplified or converted to MIDI.

If time and cost are of no restriction the amplification module should be implemented with the following objectives:

Secondary Objectives

- The amplification module must allow for the direct connection of a regular amplifier;
- The amplification module should not condition any of the main objectives;
- The amplification must allow to be controlled and, if possible, tuned by the player.

3.2 Initial Methods

The first approach to the project will be comparing several piezo types regarding signal and frequency response. In order to achieve that a small structure will be used which can fixate strings and place piezo elements at its base, such as a guitar or ukulele. This will serve to build an initial database with each piezo characteristics. Until now the company *PI Ceramics* [17] has been contacted and supplied two samples of piezos for testing during the development phase. Besides, several typical piezo disks have been bought to test and compare.

Further on, a basic program will be implemented to convert the signal of the piezo using a Teensy or Arduino board. Together with a virtual oscilloscope this will allow to collect samples and retain them for comparison. This is important as signals collected from an actual oscilloscope and ADC can radically vary due to the internal resistance of the ADC, however this topic will be studied in more detail in section 7.2.2. After validating the piezo the MIDI module will be implemented according to the Main Objectives defined in section 3.1

After the MIDI module is finalized and validated, testing with the amplification module will start by using each developed piezo in an embedded amplification circuit. This will require isolation between the two mechanisms not to cause any interference with each other. As stated in 2.2.3 an initial stage amplification will be built for each individual piezo and its results validated. Finally, several outputs will be summed and implemented within a single audio output.

Once the validation and testing of both mechanism proves functional a small analysis on the overall benefits and place on the market will be discussed an taken into consideration. This is an important step since this project has the goal of creating an innovative project within the harp and music subject.

4 Work Methodology

This section will contain a very brief description of all the tools planned and used during the preparation and development phase of the project. It will as well contain a work schedule procedure for the development of the project.

4.1 Project Phases

The project will consist of two distinct phases, the **preparation phase** and the **development phase**. The preparation phase will be executed prior to the beginning of the project. The **development phase** will start near the end of the first phase will be continued until the final delivery of the project.

4.1.1 Preparation Phase

The preparation phase consisted in following by order of the presented topics:

- Investigation of the different approaches to convert sound to MIDI;
- Investigation of approaches to amplify sound using sound transducers;
- Matching of the two previous topics in one single implementation;
- Market study of the different components;
- Ordering of the components to be used in the Developing phase;
- Study of the tools to be used during the Developing Phase.

All of these topics have in parallel a continuous gathering of information from specialists within and outside the academic environment.

4.1.2 Developing Phase

The Developing phase will consist in following by order the presented topics:

- **Weeks 1-5** Conclusion and validation of all the information gathered during the Preparation phase
- **Weeks 2-5** Initial Testing and implementation of the pre-schematics
- **Weeks 3-5** Validation of the pre-schematics in accordance to the main objectives
- **Weeks 6-8** Implementation of the studied systems and second ordering of components
- **Weeks 9-10** Tuning and cleaning of noise within the implemented circuit
- **Weeks 11-12** Implementation of the MIDI Algorithm
- **Weeks 13** Validation of the algorithm
- **Weeks 14-16** Final Verification and Validation of the implemented schematics
- **Weeks 17-18** Preparation and Writing of the Dissertation

All of these topics have been continuously evaluated by the supervisor and submitted to approval and/or changes when requested.

4.2 Project Tools

For this project the following tools will be used in order to fulfil objectives:

- ***Oscilloscope*** To gather data from the circuit at any given point
- ***Python: matplotlib*** In order to simulate and gather data and graphics from the oscilloscope
- ***National Instruments Multisim*** In order to simulate the behaviour of the circuit and piezos
- ***Micro Controllers: Arduino, Teensy*** To read and process data from any given piezo or input
- ***Electrical Components*** To build the circuit in need
- ***Computer Station*** To process all the data and simulations

5 Research Methods and Developed Tools

This section contains a description of the used and developed tools to validate the system.

5.1 Research Tools

5.1.1 Oscilloscope

The oscilloscope in use will be DSO10112B [18]. It has 4 analog channels capable of sampling at $1GSa/s$ each, a $100MH_z$ processor clock speed and a 2 mV/div per 2 ns/div minimum resolution. Since the measured signals should not have an amplitude below 10 mV and 20 H_z this specifications should suffice. It is also capable of performing Fast Fourier Transforms which will be particularly useful during the audio amplification development and will be explained more in detail in 5.2.4. The probes in use contain a high impedance of 1MΩ and a small capacitance of 5 pF.

5.1.2 Multisim

In order to simulate the circuit behaviour the NImultisim software [19] will be used. This tool will be most usefull during the dimensioning of the audio amplification circuit specifically for filter designing. However, due to the characteristics of the piezo it is very challenging to accurately replicate its behaviour in a simulation and plan accordingly. The signal amplitude will vary in time, frequency will change in phase and the peak resulting from physically plucking the string will affect all the remaining signal.

For that reason the multisim tool will only be used as a theoretical help to implement and predict frequency response of the amplification signal more than an accurate estimation of the expected result.

5.1.3 Developed Software

In order to validate the results from the microcontroller it is important to visualize the inputs coming from the piezo signal using the internal ADC, in opposition to the oscilloscope. For that, the Arduino IDE already provides a tool to visualize the signal however, this tool, as it stands, is only capable of visualizing 300 samples with no control over its window or number of inputs on screen. To fix this issue a program was developed from the ground up allowing to visualize any samples coming from the Teensy microcontroller, or any other board, in real-time using a program in python available in [20]. As it stands the program is capable of: visualizing the signal in real-time; visualizing up to 8 concurrent signals; stop the image and zoom in/out in any location; work in continuous mode or trigger after a certain event; define any maximum voltage or bit resolution for the signal.

Furthermore, the program is also capable of displaying the x values in numerical samples or associated timing. Yet, this last option introduces a small delay while processing which delays the whole signal per sample. This is an issue that will be fixed in future versions explaining why the figures from this program will be displayed using a sample based X-axis. Also, the program shows the average delay since sending the sample until it is processed on top of the window which could give an idea of the sampling rate time and adjust the intrinsic per-sample delay.

Finally, a second program was developed on top of the previous one to visualize the incoming signal in the frequency domain also available in [20]. This program is only capable of processing one input as it stands and it is available in [**fft_git_me**]. It is possible to define several windowing methods as well as number of samples to collect until the FFT is processed. This program should serve as a tool to help validate the signal coming from the oscilloscope.

5.2 Teensy Microcontroller

5.2.1 Characteristics

For this project, as stated in section 4.2 a Teensy 3.5 development board [21] will be used. This board contains a 120MHz ARM Cortex-M4 microprocessor[12] together with 512K Flash memory and 192K Ram memory. It contains 62 I/O pins available from which 26 pins support analog signals up to 3.3 Volts. As a note it is possible to overclock the microprocessor up to 168 MH_z if needed however, this is not recommend for long periods of time as it shortens the life expectancy of the microcontroller.

Besides the ample number of I/O ports the Teensy board was chosen due to its specific development towards sound processing. It contains embedded audio codecs, capable of encoding and decoding audio files from an SD Card, 14 independent hardware timers and most important 2 dedicated ADC's capable of working independently of each other.

These are not all the specifications regarding the Teensy 3.5 but a sum of the most useful ones regarding this project. Finally, the Teensy 3.5 was chosen instead of the model 3.6, which would have a higher clock speed of 180MHz, due to its 5 Volt tolerance in the I/O despite only working up to 3.3 Volts. This characteristic can be quite important while testing and even in the final product as piezo elements can behave erratically when submitted to unpredictable pressure or heat. It allows for a more robust implementation both while testing and as a final product.

5.2.2 Teensy and Python oscilloscope - Validation of a Single ADC

One of the central aspects of this project is the ADC capability of the Teensy board. Even though the oscilloscope described in 5.1.1 is more than capable of analysing the signal from the piezo it is very important to keep in mind that the embedded ADC will have a much lower signal accuracy and sampling speed when compared to the oscilloscope. This means that signal characteristics displayed on the oscilloscope can be lost to the microcontroller's ADC if, as an example, the signal changes too fast or contain very small voltage variations.

Besides, the teensy's ADC contains an internal capacitance of 5pF and most important an internal impedance of 2 kΩ. This means that signals collected in the oscilloscope, due to its 1MΩ will have a much higher voltage than signals collected inside the Teensy. This difference increases the higher the voltage and output impedance of the whole circuit load before the ADC input.

First, in order to test accuracy, the ADC will be submitted to a $5kH_z$ signal provided by a signal generator, using a very low output impedance of 8Ω as recommended by the microcontroller datasheet. This frequency serves to guarantee that most signals from the piezo will be captured since their frequency should be mostly below $2kH_z$. The microcontroller will contain a program which will collect 2000 samples to a buffer and afterwards print them out to the Serial Interface. To control the ADC characteristics the official ADC library, available in [22], will be used. This library allows a straightforward control of the sampling speed (how fast samples are collected), bit resolution up to 16 bits (how many discrete intervals between 0 and 3.3 Volts) and conversion speed (how accurate the voltage signal is converted to a discrete value). After executing, the program will also display how much time collecting the samples and sending them took.

The Serial Interface will be connected to a computer which will run a program in Python. This program was developed from the ground up and it is available in [**github_python**]. The program will receive the samples from the microcontroller and display in real-time a graph illustrating its results. At the same time the program will calculate the average delay between samples which allows to estimate the total and single send delay from the computer side.

The results are illustrated in Annex A table 5. It is possible to observe that, as expected, the higher the conversion and sampling speed the faster the signal is obtained, with the total time to collect 2000 samples going from about 6 ms to 27 ms which is a very significant. Changing from 10 bits to 16 bits resolution affects time consumption with an increase of 10% to 20% the faster the conversion and sampling speed. It is observable that the faster the samples are collected the closer values from 12 bit and 16 bit get to each other.

Regarding signal send time, as expected, this value is independent of sampling collection and only varies from bit resolution, taking about 5 μs per sample. These times remain fairly constant as they do not depend on the ADC characteristics but rather on the information size and bit change. As an example, using 4 bits to send the number 0 (in binary: 0000) or 15 (in binary: 1111) takes less time than sending the number 10 (in binary: 1010) due to the increase in bit change. This justifies why processing times are exponentially varying the higher the bit resolution. There is also a delay between sending and receiving values of about 3 to 4 μs. This delay may be due to the python program generating a graph at the same time however, this is not a very relevant variable as only MIDI messages will be sent to the computer and not constant data.

Regarding sample quality, also in Annex A in Figure 74d tests B, E and I are illustrated for both 10 and 16 bits resolution. The results show only 500 samples for better signal analysis. By observing the results it is perceptible that as the sampling speed increases less signal periods are captured as more information is collected about each individual period. However, even when comparing a slower test to the fastest test there is no decrease in sample quality with no typical glitches present. Also, by comparing both 10 and 16 bits, side to side, there is no observable variation in quality whatsoever even for the fastest test implemented.

The same tests where implemented using a wave with the same frequency but with an amplitude of 50 milivolts peak-to-peak. A 10 bit implementation is able to detect a maximum of 3 milivolt variation while a 16 bit implement can detect up to 50 microvolt variation. However this limit is not usually imposed by the bit resolution but by the intrinsic ADC accuracy during its conversion. By looking at Figures 73a through 73f the existence of glitches becomes very noticeable increasing the faster the test is performed and the higher the bit resolution gets. For the slowest test shown at 10 bits the worst variation is 10mV with an average of 3mV error while for the fastest test at 16 bits the worst variation is 13mV with an average of 6mV error. This means that in order to detect smaller variances an averaging may be needed to decrease these kind of errors.

Finally, a final program was implemented using interrupts to generate ADC readings. After several attempts for collecting samples this method revealed the fastest and most accurate. The process consists of a timer generating an interrupt at a defined period. This timer when activated will generate a pulse in a digital output and start an ADC reading. When finished, the ADC will generate an interrupt, collect the value and generate a pulse. The difference between these two pulses reveals the timing the ADC takes to read a sample and can be measured in an oscilloscope. An example of a signal can be seen in Figure 17. The results for each test at both 12 and 16 bits are represented in Annex A in tables 7 and 8.

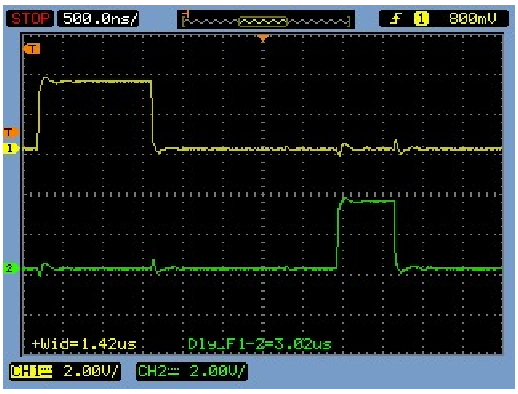


Figure 17: Example of the timing window for Medium Sampling and Conversion Speed at 12 bits

These tables serve more as reference point for later implementations of the signal sampling. As expected, the timings follow the same pattern as already discussed for table 5. Finally, some measurements were done using different averaging which are represented in Annex A in table 9. However, varying in averaging only multiplies the samples taken each time. After a certain amount averaging follows a linear pattern.

As a final conclusion, it is proved that it is possible to execute the ADC at higher speeds with the highest resolution. If changes below 10 milivolts need to be obtained, at 16 bits, some averaging may be needed to apply. Regarding the serial interface, since no signals will be sent consecutively as in this example but only single MIDI messages the teensy development board is more than capable of executing the requirements imposed in 3.1.

5.2.3 Teensy and Python oscilloscope - Validation of a Both ADCs

In section 5.2.2, each ADC was validated independently. For this project both ADCs will be used at the same while connecting to several inputs each. Due to the configuration process one ADC can only be linked to an input at each time. This means that for reading several inputs at the same time the ADC configuration needs to be changed on the fly while capturing data.

Figure 74, presented in annex A, shows four tests done in order to compare the delay between acquiring one and two inputs in each ADC. The tests were run using the fastest settings at 16 bits for 500 samples. By looking at table 6 it is noticeable that both ADCs working at the same time for a single input will take the same amount of time as a single ADC working for a single input. However by imposing a switching mechanism between two inputs for one ADC the time goes up about 2.1 times.

This means that the switching mechanism alone takes about 5% the time of collecting a sample at the fastest settings. Also by looking at Figure 74d in which one ADC is oscillating between two inputs and the other is reading a single one this delay becomes visibly noticeable by the number of periods acquired. This alone is not that problematic however when scaling for 3, 4 or more inputs at the same time in one ADC it can generate significant delays.

This is unfortunately expected and it is important to keep this detail in mind while developing and escalating the MIDI circuit for several inputs. However, for the MIDI implementation as long as the peak is collected there should be no problem in this regard. Apart from this no quality degradation is observed throughout the test.

5.2.4 Teensy and Python Fast Fourier Transform

In order to validate the piezo itself as well as the audio amplification side, it is important to verify its frequency response. The ideal way is to use a FT (Fourier Transform) which converts a signal from its time domain to its frequency domain. The result of a FT should be in a graph displayed with frequency per voltage (or dB) in amplitude. Ideally, the result of a perfect sinusoidal wave is displayed through a single peak in its frequency with the corresponding voltage amplitude. Even more complex waves can be decomposed into a sum of sinusoidal waves which can therefore be represented by single peaks, as shown in Figures 18, 19, 20.

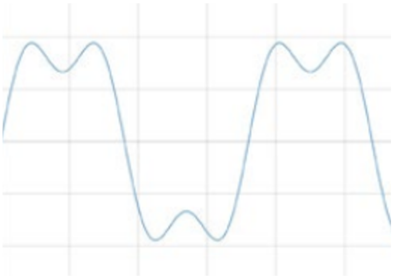


Figure 18: Complex Waveform

Figure 19: Decomposed Waveform

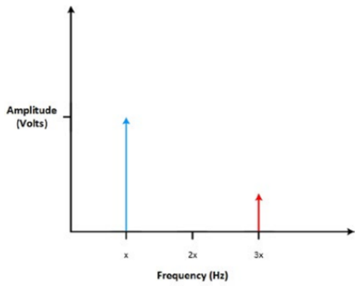


Figure 20: Fast Fourier Transform

For a digital circuit to perform a Fourier transform the solution is to implement a DFT (Discrete Fourier Transform) that applies the same principles but for discrete data sets which display the result in discrete frequency bins. These bins represent a set of frequencies in which the signal is contained and can vary in size depending on the sampling rate of the signal and windowing applied. In this case the FFT (Fast Fourier Transform) will be applied as its an optimized version of the DFT, to reduce computation time.

Also, when sampling a signal, usually the data is not contained within perfect boundaries. In other words, each individual signal, resulted from the decomposed wave, does not start and end at the beginning of a period. Because of this, the DFT result will show high frequency components not present in the original signal. Besides, since the original signal is always windowed prior to the FFT transformation, the magnitude spectrum will present leakage to near frequencies. This makes the result look more as a curve more than an impulse. This effect is called spectral leakage as is represented in Figure 21.

To mitigate this effect a windowing technique mentioned before can be applied. Windowing mainly aims to reduce the effect of the high frequency components but can substantially reduce spectral leakage. This is achieved by multiplying the finite set of data with a curve which gradually smooths the edges of the data towards zero approximating the starting and ending value of the waveform. A representation of this effect is demonstrated in Figure 22.

Applying a window is not as trivial and can change the result drastically depending on which technique implemented as well as the original signal. Windowing techniques are well beyond the scope of this project, however a good and more detailed explanation windowing and spectral analysis in general is Spectral Audio Signal Processing by Julius O. Smith [23].

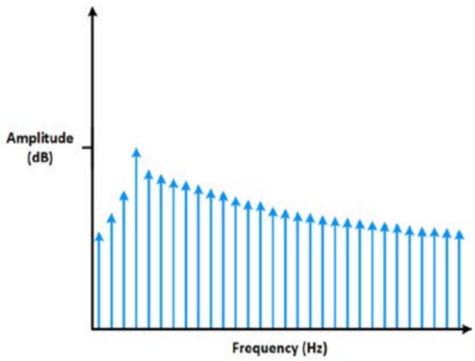


Figure 21: Spectral Leakage without windowing

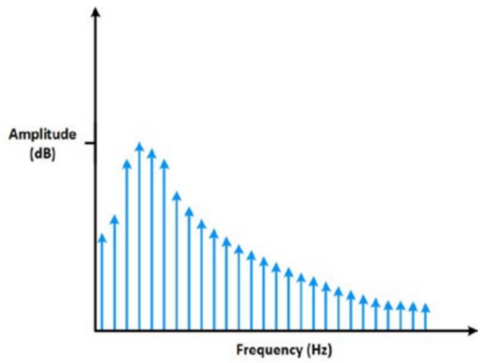


Figure 22: Spectral Leakage with windowing

There are several types of windowing techniques that can be applied with the most common ones being Rectangle (no Window), Hamming and Blackman, with many variants even within these last two. Typically the Hamming window will fit 95% of the scenarios however, for most audio applications the Blackman window technique is the best one to apply. A comparison between the Hamming and Blackman window technique can be seen in Figures 23 and 24.

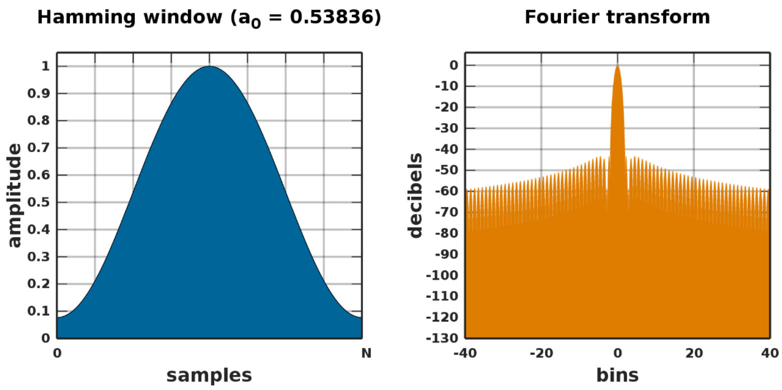


Figure 23: Hamming Windowing

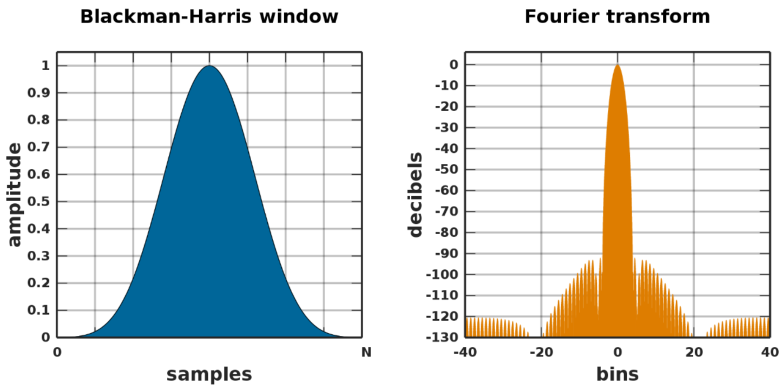


Figure 24: Blackman-Harris Windowing

This Blackman method aims to minimize frequency leakage which is rather important for this project: as the string vibrates the main frequency leakage can attenuate the harmonics produced by the string hiding its effect. However, this method widens the frequency bins to achieve this, leading to less frequency accuracy, and removes power from the signal leading to less amplitude than supposed.

However, for this application, frequency accuracy is not a big concern as the expected main and harmonics frequency are known from the string tuning. Also, even though there is a power attenuation in the signal the proportion between the frequency peaks is maintained leading to a relevant analysis at the end. By testing several variations of the Blackman windowing the Blackman-Harris variation is the most accurate representation of all.

As the oscilloscope mentioned in 5.1.1 does not possess the Blackman-Harris variation windowing one had to be implemented in the Teensy using the official audio library [24] together with the ADC library [22]. The Teensy will collect several samples and perform an FFT using the Blackman-Harris windowing. Afterwards, it will send the signal through the Serial Interface allowing the already mentioned Python program to display the graph.

It is important to mention that in order to collect samples using the Teensy the signal needs to be within the 0 to 3.3 volts margin. In some cases this can be applied, however, in many of them an oscilloscope needs to be used.

6 Piezoelectric Transducer

This section contains all the details and decision process regarding the piezo material and its design.

6.1 Concept and Initial Design

The initial approach for choosing the ideal piezo for this project was both to contact piezoelectric elements manufacturers and research piezo elements used to amplify acoustic guitars, which are the most common. The second option presented no results as guitar companies do not reveal which types of piezo elements are used. Also, many DIY (Do it Yourself) projects had information regarding piezo disks which are not ideal to be added directly right under each string. The first option resulted in the company Pi miCos [10] recommending the PIC255-0753 piezo element. This number informs that the piezo is made of the material PIC255, a modified lead zirconate-lead titanate, with the company reference number 0753 to specify dimensions of 5x2x0.5 mm (LxWxH).

This cuboid piezo seems ideal for the project due to its small dimensions and datasheet characteristics which specify a high piezoelectric charge constant (already described in section 2.2.1) of about 550 pC/N, a maximum of 100 Vpp Voltage response, and a resonance frequency of $3MH_z$. The quality factor constant Q is stated as 80. By using equation 2 presented in section 2.2.1 the curve displayed in Figure 25 is obtained.

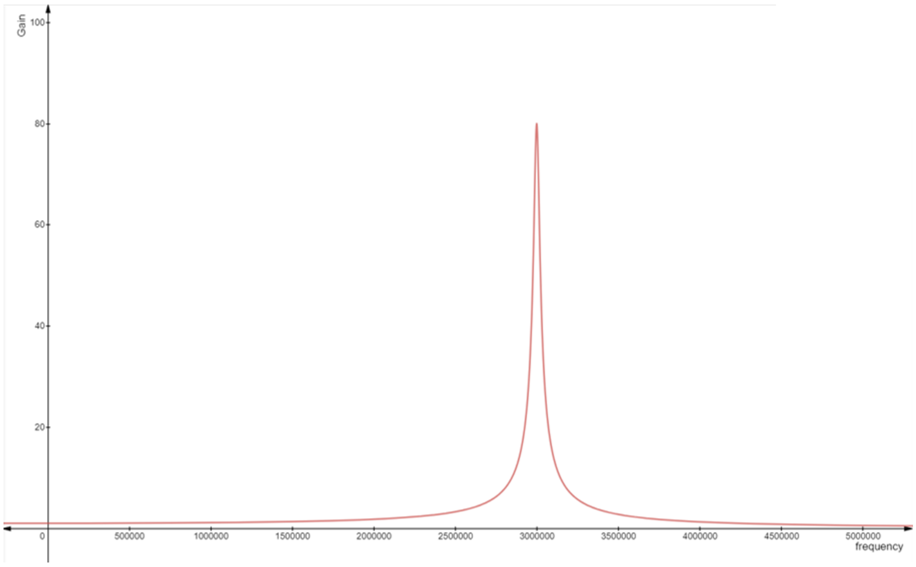


Figure 25: Frequency Response of Piezo PIC 255-0753

Reminding Figure 10 in section 2.2.1, this high resonance frequency gives a high bandwidth window to work within. Only frequencies above 2MH_z should demonstrate an average gain which is well beyond the working window established.

However, in order to get the electric signal two ways can be used: by soldering the piezo element, which requires special ovens that prevent changes in the internal characteristics to change their orientation by using low temperature and a special solder, not available; by using contact metal pieces attached to both sides of the piezo.

This last solution was applied using several types of conductive materials and pressure points on the piezo. However, the best results obtained are presented in Figure 26. This Figure displays both the output signal (in yellow) and the associated FFT (in purple). Neither display a significant result for the applied frequency of 348H_z (associated with the note F_4).

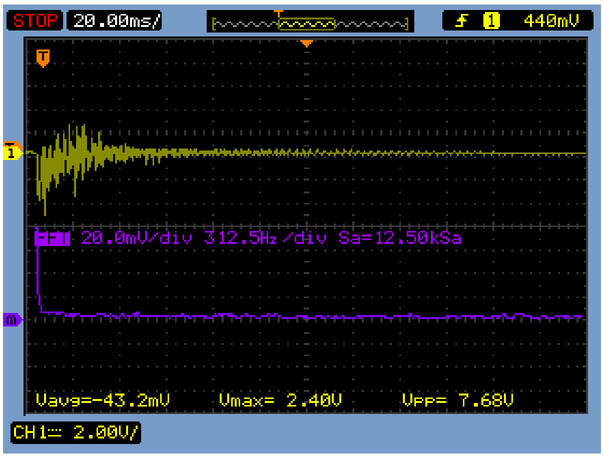


Figure 26: Results of the Cuboid Piezo element using metal contact

Since no signal can be captured from the initial piezo another possibility is to capture the signal from another piezo while attaching the first piezo to it. This would mainly maintain the characteristics of the initial piezo, denominated from now on as piezo_A, while using the second to conduct the electric signal, denominated as piezo_B. Assuming piezo_B has a weaker nature (regarding piezoelectric charge constant and voltage response) there shouldn't be much interference in the circuit. Such a piezo is easily available and consists of a piezo disk. The idea resulted in the schematic shown in Figure 27 and 28.

In the schematic piezo_A (1) is attached to piezo_B (2) which consists of a small disk. The string, in contact with the piezo_A (1) will oscillate producing a vibration which will pass to piezo_B. Piezo_B will remain on top of a hollow cylinder (3) allowing this oscillation to occur. The base support (4) will prevent any oscillation from below, such as the harp belly or near strings reverberation, to affect the piezo response.

In this case the cylinder (3) contains a gap allowing wires attached to piezo_B to pass, this could also be done from the base of the base support (4). The enclosure (5) serves only to protect the inside and provide an aesthetic improvement.

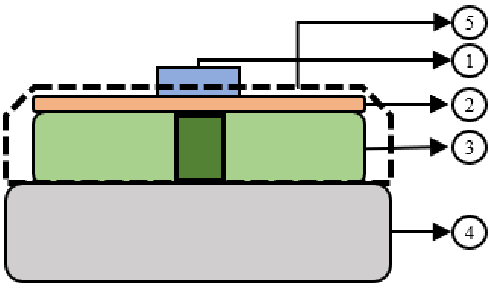


Figure 27: Side View of the Piezo Structure

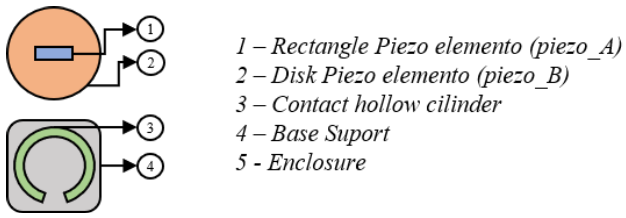


Figure 28: Top View of the Piezo Structure

As the ideal piezo disk element was well past the limit budget, the closest to ideal piezo disk element obtained was the 7BB-12-9 and the 7BB-15-9 [25]. These two piezos vary in diameter, 12mm and 15mm respectively, and resonance frequency, $9kH_z$ and $6kH_z$ respectively. The low resonance frequency can be a problem as frequencies near $1kH_z$ and above can be extra amplified. This comes only as an estimation as no quality factor is provided by the manufacturer. Regarding MIDI this will not be a problem however, for audio amplification, it can lead to more high frequency components. This effect can be more or less mitigated using a low pass filter if needed, which is a topic already discussed in 2.2.1.

6.2 Piezoelectric Disk Testing

In order to test the piezo disk individually the schematic shown in Figure 27 was assembled. The cuboid piezo element was replaced by a copper cuboid of the same size, the cylinder was made out of brass metal and the base support made out of rubber. The results of a medium force pluck can be seen in Figure 29 and 30 with the wave from the piezo represented in yellow and the FFT shown in purple. These results display a more promising result with the respective FFT showing the fundamental frequency, F4 at 348 Hz, and three of its harmonics at $670Hz$, $1047Hz$ and $1400Hz$. These harmonics coincide with the expected as they are multiples of the fundamental frequency F4. Also, as the string is pressed and then released, stretching and compressing the piezo, a negative voltage peak occurs as expected due to the initial stretch of the piezo. As the strings stops vibrating the signal tends to zero.

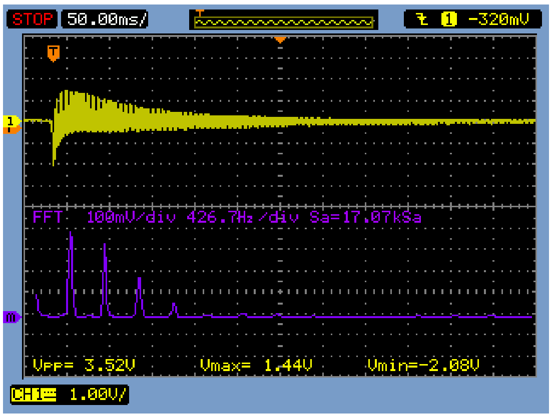


Figure 29: Piezo Result of medium pluck with FFT

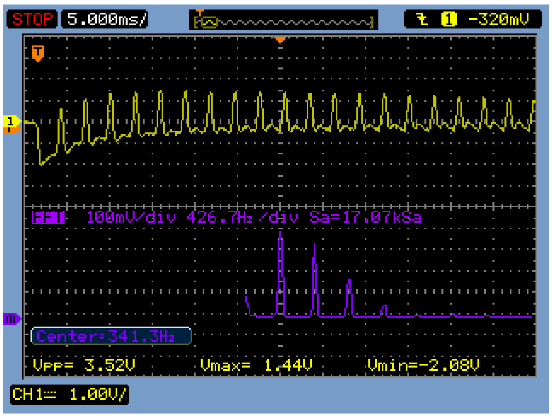


Figure 30: Amplified Piezo Result of medium pluck with FFT

By doing the same test with a stronger pluck of the string the results shown in 31 are obtained. It is important to notice the increase in amplitude, from about 3.5 to 7.5 Volts, which is to be expect as more force is applied on to the piezo. The increase in the harmonics amplitude, comparing to the fundamental frequency, happens due to the low resonance frequency and is easily noticeable. This results implies a low quality factor of piezo_B as frequencies near $1kH_z$ are further amplified than the fundamental frequency.

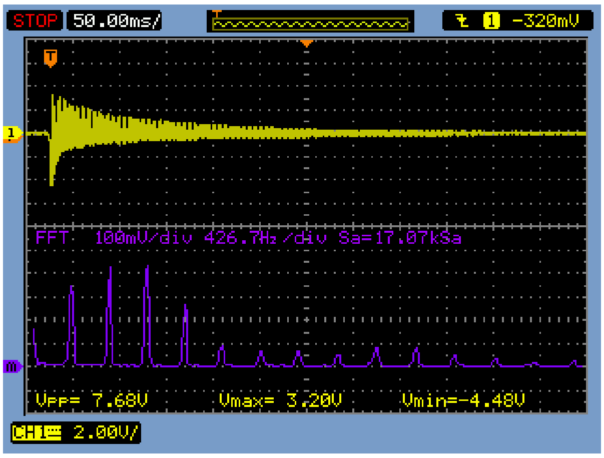


Figure 31: Piezo Result of stronger pluck with FFT

Finally by testing a larger piezo using the same setup the results shown in Figure 32 and 33 are obtained. These are of lower quality not only by the lower voltage peak-to-peak obtained as well as worse performance in the relation between the fundamental frequency and its harmonics. In this piezo the lower resonance frequency is heavily noticeable.

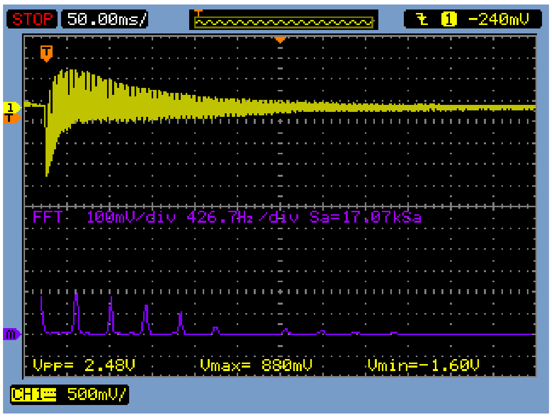


Figure 32: Piezo Result of medium pluck with FFT

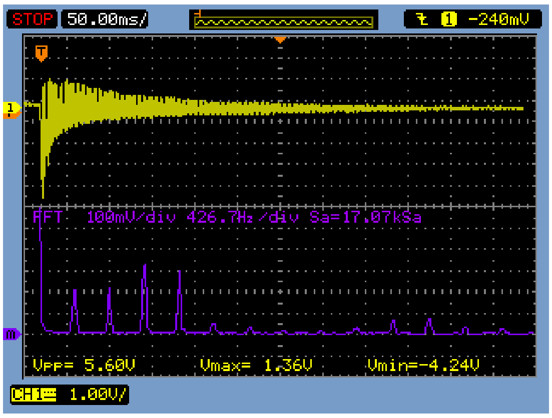


Figure 33: Amplified Piezo Result of medium pluck with FFT

These results eliminate the possibility of using a larger piezo in comparison to the smaller one. Even though the smaller piezo still has a result far from perfect it is the best result among the possibilities obtained.

6.3 Contact Hollow Cylinder Testing

To make sure metal is generically the best material for the hollow cylinder this was replaced by rubber. The rest of the components remain the same using a smaller piezo disk and a copper cuboid on top. The results are shown in Figures 34 and 35. Comparing to Figures 29 and 30 the result seems attenuated with lower harmonics and definition. For that reason a metal hollow cylinder will be used throughout the project.

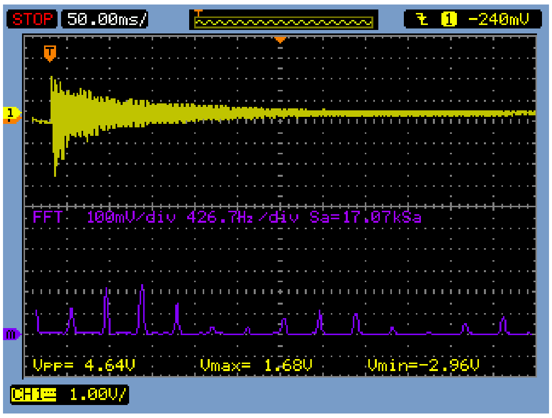


Figure 34: Piezo Result of medium pluck with FFT

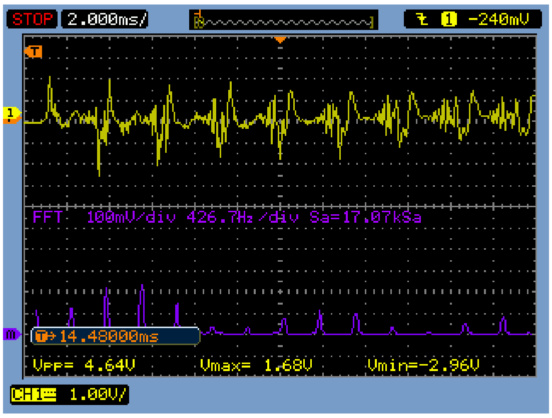


Figure 35: Amplified Piezo Result of medium pluck with FFT

6.4 Full schematic Testing

To complete the design validation, the copper cuboid was replaced with the initial planned piezo PIC255-0753 mentioned in section 6.1 together with a metal cylinder and a smaller piezo disk. The results are shown in Figures 36 and 37 which contain both the signal and the FFT. The scales were reduced in order to accommodate the results. These show a significant increase in peak-to-peak voltage when comparing to the initial results, with a strong pluck reaching about 30Vpp. This increase in voltage is expected due to the higher piezoelectric charge constant .

Not only the power of the signal has increased but also the quality. As shown in the FFT of Figure 36, for an increase in amplitude of about 5 times the initial result, shown in Figure 29, a good relation between fundamental frequency and its harmonics is still maintained. Since the disk is still amplifying frequencies near and above $1kH_z$ further than frequencies below, the results in the FFT in Figure 36 still shows higher harmonics, specially the $1047H_z$ harmonic. The cuboid piezo due to its higher resonance frequency will amplify all relevant frequencies the same, explaining why the relation between the two is 5 times higher, just like the voltage.

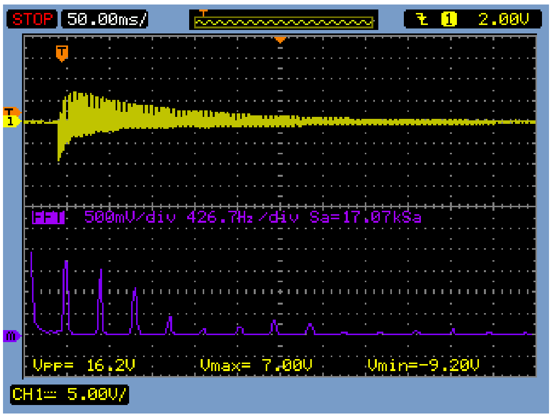


Figure 36: Piezo Result of medium pluck with FFT

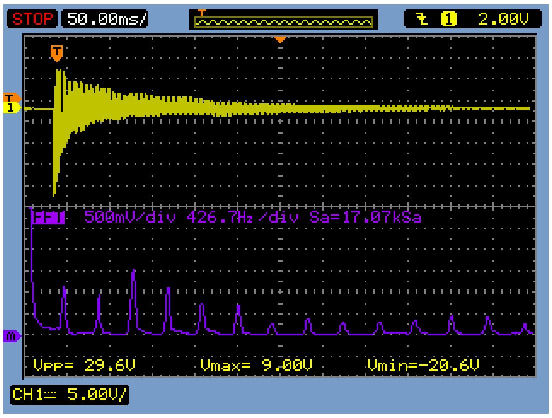


Figure 37: Piezo Result of strong pluck with FFT

As a note, Figure 38 shows an amplified version of the test done in Figure 36. This Figure is inverted for easier observation. It shows how noticeable the signal, even when oscillating at the complex string frequency, travels from a predominant distended shape to a neutral state due to its initial impact combined with the relaxation of the string near the tip.

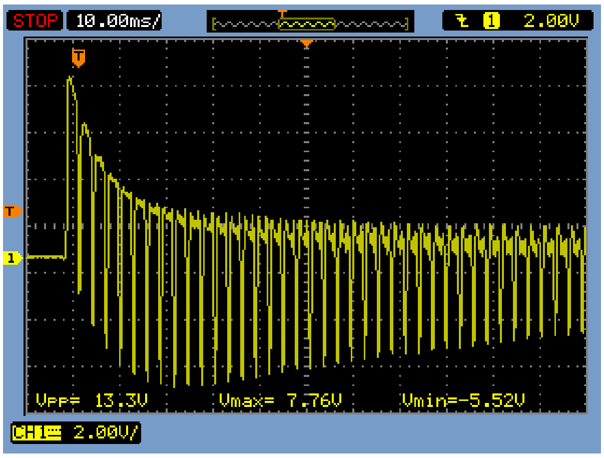


Figure 38: Amplified Piezo Result of medium pluck

The results are more than pleasant and ideal to adapt to MIDI and audio amplification. Unfortunately no more conclusions can be taken out of the piezo behaviour since the datasheet of the piezo disk is very incomplete and indicates deviations of about 10% in the resonance frequency. Surely, in regards to this project, this lack of information can be overcomed.

7 MIDI Adaptation

This section contains the description of all the process making regarding the MIDI adaptation of the circuit.

7.1 Objectives for MIDI Implementation

In order to adapt a circuit to MIDI some objectives and considerations need to be defined. The objectives can be defined as follow:

1. The MIDI conversion should be done within the normal latency requirements: maximum 10 ms, optimal should be around 5ms;
2. The signals needs to be contained within 0 and 3.3V in order to be read by the microcontroller's ADC;
3. The velocity included in the MIDI ON message needs to correspond to the force applied to the string;
4. A mechanism to define when a string stops vibrating needs to be implemented (MIDI OFF);
5. The input impedance of the microcontroller's ADC needs to be taken in account;

These objectives define all the steps that need to be considered and will serve as a guide to implement the MIDI adaptation.

7.2 Concept Design

In order to explain and better follow the design process the initial idealized circuit is presented in Figure 39.

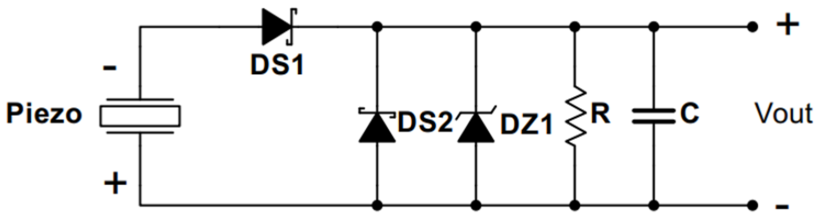


Figure 39: Idealization of a piezo Circuit Adaptation for a MIDI Implementation

7.2.1 Signal Rectification and Conditioning

First of all since pressing the string produces a positive voltage while releasing the string produces a negative voltage the first implementation is to invert the piezo polarization. As the player releases the string to produce sound this release will correspond to a positive voltage peak. The impact of releasing the string will generate a positive peak which should correspond to the force applied. Figure 40 and 41 show the output of the piezo for an inverted signal of a medium and strong pluck of the string.

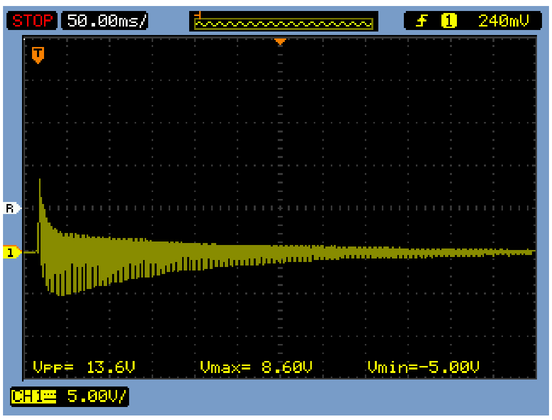


Figure 40: Natural piezo response to a a small pluck

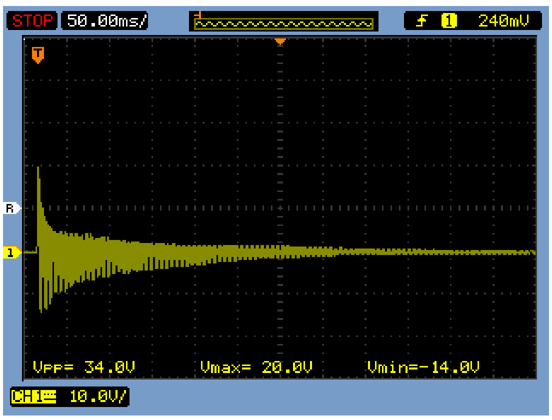


Figure 41: Natural piezo response to a a strong pluck

By looking at the values of the maximum voltage presented it's clear that stronger plucks generate stronger peak voltages. As long as the ADC is capable of capturing that corresponding peak objective (3) is fulfilled. Further on, the microcontroller is not able to interpret negative voltages of any sort . The maximum accepted negative voltage stated in the microcontroller datasheet [12] is -0.5 Volts. For that reason the signal needs to be rectified in order to be read by the ADC. For that a half wave rectifier was implemented in order to guarantee that no negative signal crosses through the ADC. This was implemented using the schottky diode DS1 IN5818 [26]. Schottky Diodes are typically used to rectify circuits due to their low forward voltage drop, in this case 0.55 V, and high reverse voltage block, in this case of 30 Volts. They are also ideal for this application due to their rapid switching behaviour meaning they possess low attenuation at high frequencies. However, by looking at Figure 42 there is still some negative voltage present at the output of about 3 Volts, depending on the force applied.

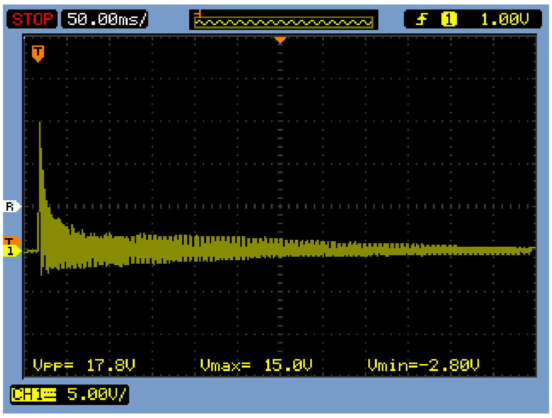


Figure 42: Rectification using One Diode

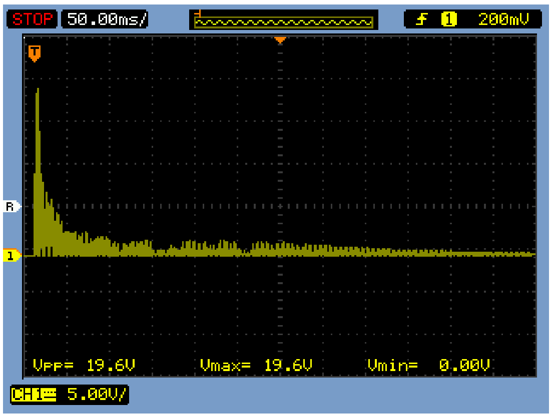


Figure 43: Rectification using Two Diodes

This effect happens due to the natural current leakage of the diode. However, it can be fixed by adding another schottky diode in parallel, diode DS2, which the results are demonstrated in Figure 43. As the negative signal goes all through the diode no negative voltage is presented at the output. Finally a Zener Diode rated for 3.3V is added in parallel as a safety mechanism to guarantee that no voltage above 3.3 V crosses the output. Zener diodes are typically used as voltage regulators due to their low reverse voltage. The Zener diode in use is the 1N4728A [27] with a reverse voltage of 3.3V. With this objective (2) is accomplished.

7.2.2 RC Pair

Using the model described in section 2.2.1 and preseted in Figure 9 the equivalent schematic for the circuit is presented in Figure 44.

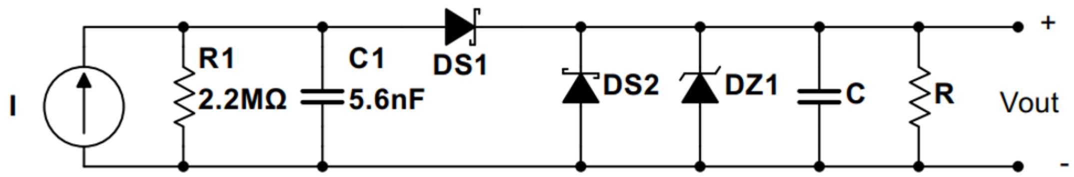


Figure 44: Idealization of Piezo circuit with piezo schematic

Using a specialized multimeter to measure the piezo impedance the final value for the piezo resistance is 2.2MΩ and its capacitance is 5.6nF at 1kH_z. By consulting the piezo datasheet the equivalent resistance has a value that only suffers relevant changes near the resonance value while the capacitance should only change with temperature variation. Even though piezo_B has a low resonance these values will be considered to simulate the piezo behaviour.

Despite the output signal being contained between 0V and 3.3V the resulting value is not yet proportional to the 0V to 3.3V window. If a medium pluck generates 10V and a stronger pluck 20V both will show around 3.3V at the output. For that an RC pair needs to be used. The R value will dictate how this conversion is performed. Since the piezo works as a current source by using Ohm's Law (3) the higher the resistance value the higher the output value. So, ideally the R value should match the impedance of the piezo in the order of the MΩ. Values above the internal resistor will not generate higher voltage values but maintain the maximum peak obtained.

$$V = R * I \qquad V_c = V_s * e^{-\frac{T}{RC}} \qquad \tau = R * C \tag{3}$$

The value of C serves mostly to define the discharging rate of the signal meaning how long the signal is kept for the ADC. The typical behaviour of a capacitor is presented in formula (3). For this purpose the C value will only be defined after defining the value of R.

As a note from this point on, the values on the oscilloscope should not be used to define the circuit as the results will differ from one another as presented in Figures 45 and 46. This occurs due to the internal impedance difference presented in both ADC's as already mentioned in section 5.2.2. This difference gets more relevant the higher the input impedance, as it is the case.

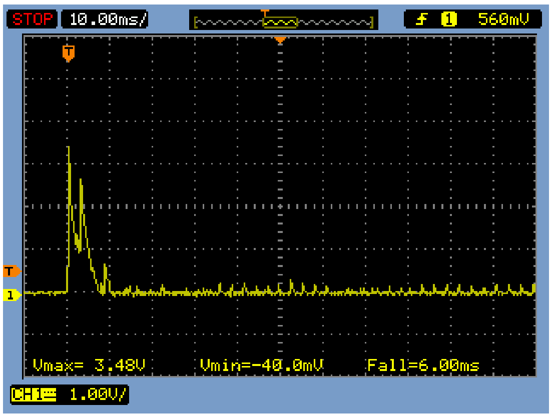


Figure 45: Medium pluck read by the oscilloscope

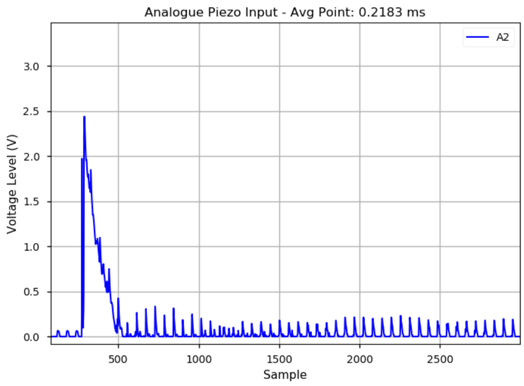


Figure 46: Medium pluck read by the microcontroller

By using a value of 2.2MΩ to match the input impedance and no capacitor the results are demonstrated in Figure 47.

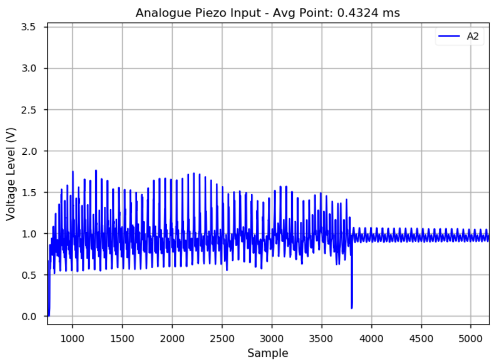


Figure 47: Results of a high output resistance

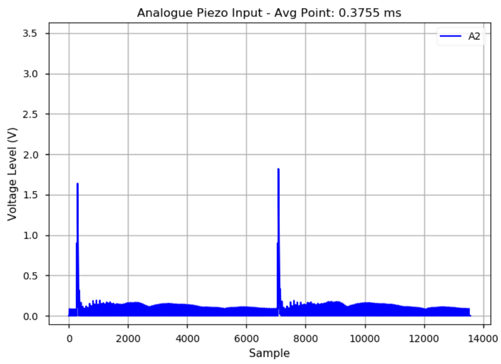


Figure 48: Results of a low output resistance for a strong pluck

For this scenario a sampling rate of 100kH_z (10μs) was used with the ADC set to medium sampling and conversion speeds at 12 bits. Consulting table 7 in annex A a sampling rate of 100kH_z is more than enough for medium speeds which only take 5.8μs to execute. What is happening is due to the high output resistance the ADC is saturating and the internal capacitor does not have time to fully discharge creating not only a bad result but also a "DC" level at about 1V. By lowering the resistance to about 100kΩ and maintaining the sampling frequency and speed the results displayed in Figure 48 are obtained. Even for a very strong pluck the output value is not able to reach 3.3V.

These examples demonstrate that there is a silver lining between output impedance and ADC speeds. A lower resistance allows for higher sampling speeds but lower output voltages while a larger output impedance allows for greater voltages but lower sampling frequency and speed. Lowering speed can also be pretty dangerous as the peak may be lost if the moment the sample is taken misses the peak time as demonstrated in Figure 49.

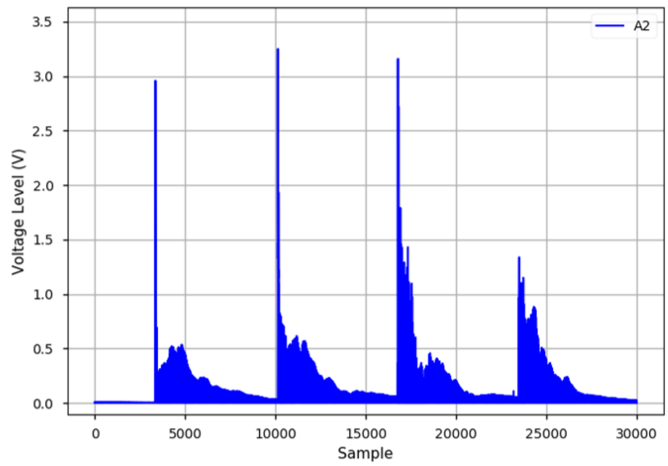


Figure 49: Example of low sampling speeds and lost peak information

In order to find an equilibrium for both parts the first priority will be to keep a stable peak sample and the second to choose a resistor value that allows the voltage to reach 3.3V. Maintaining a stable peak means that several samples need to be taken near the peak as illustrated in the comparison between Figures 50 and 51. In Figure 50 even if the highest peak is lost the next sample comes at a difference of 0.2V while in Figure

51 the next peak comes at about 1V difference.

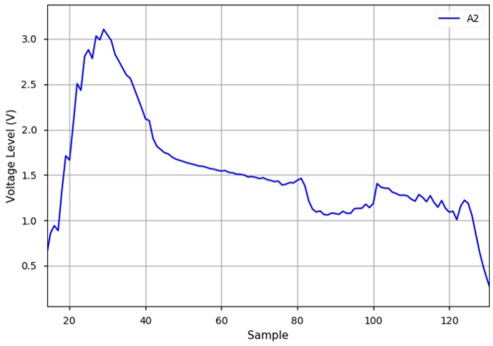


Figure 50: Results of a good quality peak capture, high sampling rate

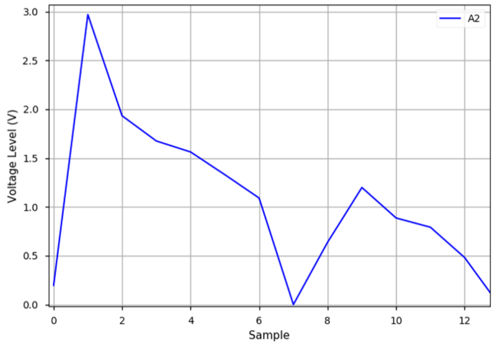


Figure 51: Results of a low quality peak capture, low sampling rate

Before advancing with the results there is a need for a background to justify the decisions done. A typical Sample&Hold ADC works by charging a capacitor for a certain amount of time (sampling time) and then connecting the capacitor to the ADC for a certain amount of time (conversion time). By using table 7 in annex A it is possible to estimate the sampling time alone at about $1\mu s$ to $5\mu s$ depending on speed. By using the formula for the capacitor in equation (3) the most important factor will be $K = e^{-\frac{T}{RC}}$ where T is the sampling time (for this example a $2.5\mu s$ will be used), R is $2.2\text{M}\Omega$ and C is the internal capacitance value at 5pF.

By calculating using the stated values $K = 0.797$. When the capacitor is fully charged the end voltage will be approximately $K * (old\,voltage) + (1 - K) * (new\,voltage)$ meaning that 0.797% of the old voltage is being retained by the capacitor. Although there are more factors in play this serves as a good approximation of the final value read by the ADC. To mitigate this effect it is possible to either increase the sampling time or lower the RC constant by decreasing R.

By consulting a very informative but simple to understand guide [28] it is recommended to keep the K value below 1/2 of the ADC bit resolution. Using 12 bit resolution and the equation presented in (4) the result amounts for a sampling time of $T >= 9\tau$ for a recommended value (7.6τ for 10 bits and 11.8τ for 16 bit).

Another possibility is to add an external capacitor and this solution is actually known as a charge reservoir as it will help charge and discharge the internal capacitor when the connection is established.

Once again following the guide [28] the optimal value for C can be calculated using the formula (4) which amounts for a minimum value of $C = 41.0nF$. It is important to keep in mind that even though this is the recommended value the capacitor also serves the purpose of defining the discharge rate of the piezo. Besides, increasing the capacitor too much may lead to the signal being "consumed" and not being able to reach 3.3V.

$$K <= \frac{1}{2^{bit_r+1}} \Leftrightarrow T >= -ln(\frac{1}{2^{bit_r+1}}) * \tau \qquad C >= (2^{bit_r+1} - 1) * Csamp \tag{4}$$

By this point an attentive reader will notice that these solutions work against each other. Increasing the C value, to $C = 41.0nF$, in order to help charge and discharge the internal capacitor implies that the sampling time should be, even for a resistor of 100kΩ, about 40ms due to the higher total capacity. If keeping the sampling time at about 2.5μs the value of the resistor goes down to 10Ω which is way to low for this application when taking in account that the output should match the piezo resistance. However, there is still another factor in play.

Until now it is assumed that the sampling rate will be as fast as possible running every time the last acquisition has ended. Nonetheless by increasing the sample rate there's more time for the internal capacitor to discharge while still maintaining some output resistance. The equation that explains this behaviour is presented in (5).

$$Q = C * (V_{new} - V_{old}) \Leftrightarrow I_{avg} = C * \Delta V * f \Leftrightarrow V_{final} = C * \Delta V * f * R \tag{5}$$

By using $R = 2.2M\Omega$, $C = 5pF$ and $f = 10kH_z$ for the worst case scenario of $\Delta V = 3.3V$ the final voltage is $V_{final} = 0.36V$ or $V_{final} \approx \Delta V * 10\%$. This result is not that bad for a worst case scenario considering only the peak should reach 3.3V with other values rounding about 1V. However notice how the value for the capacitor was the internal value and not the external. According to this point it is better not to use an external capacitor as it would only increase V_{final} (or use a very small value). With this in mind, if even by increasing the sampling rate the peak can be captured the problem should be solved.

Using all the knowledge presented a good compromise for all the variables is using an R value of 680kΩ with 5pF capacitance, resulting in a total 10pF capacitance with the internal one. Using Very Low sampling Speed and Medium Conversion speed the absolute minimum sample rate is about 50 kH_z (20μs). This itself isn't very bad however, by consulting table 7 in annex A, it shows that the signal at the mentioned speeds takes 7.6 μs to process. This means that it is taking more time to stabilize than to actually sample. Using this configuration the maximum absolute time to wait while still being able to catch the peak is 3.3kH_z (300μs). The results for both tests are presented in Figures 52 and 53.

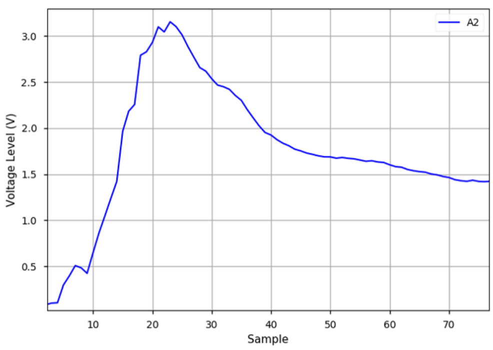


Figure 52: Results of a 50kH_z sample rate

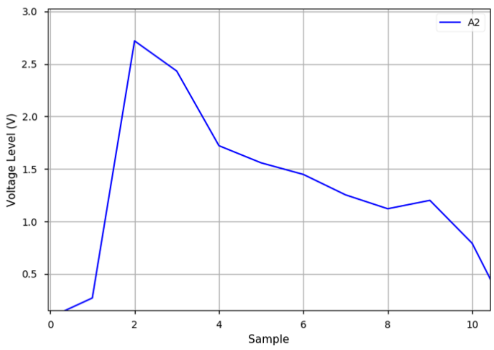


Figure 53: Results of a 3kH_z sample rate

More attention to timings will be given in section 7.2.5. For now the above timings will be considered the best and worst possible scenarios for this implementation.

7.2.3 Capturing MIDI OFF

In order to capture MIDI OFF a mechanism to capture the moment when the strings stops vibrating needs to be implemented. As explained in section 7.2.1 stopping the string, which corresponds to press the piezo, creates a negative voltage (since the piezo is inverted for this circuit). The result of a blocked string can be seen in Figure 54. However, since the signal is rectified, no negative signal can be captured by the ADC. The idea here is to introduce a DC level to define a threshold and capture voltages below the DC value.

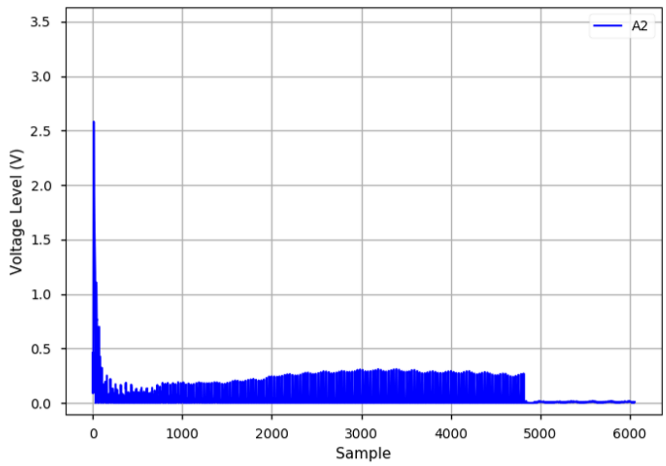


Figure 54: Results of a string vibration block

Even though it is very clear from Figure 54 the moment the string is blocked this image is drastically amplified by using a high sampling rate. Because the capacitance defined in section 7.2.2 is very low, the discharge rate RC will be about 4 μs, even though in practice this value will be much larger as the string does not stop vibrating after the initial peak.

If no force blocks the string from vibrating the signal will naturally tend to zero as it stops vibrating. In contrast, when adding a DC level, the signal will tend to this value as it stops vibrating as represented in Figure 55. Figure 56 shows the amplified recovery of the signal, after blocking, for the DC value.

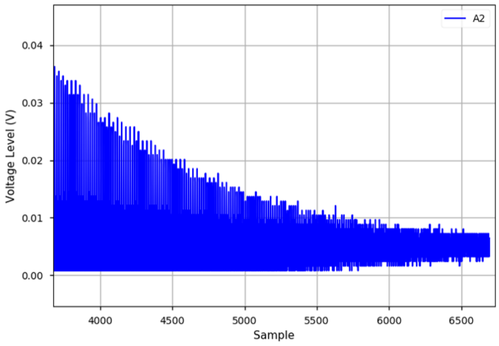


Figure 55: Resuslts of a string vibration natural tendency with DC

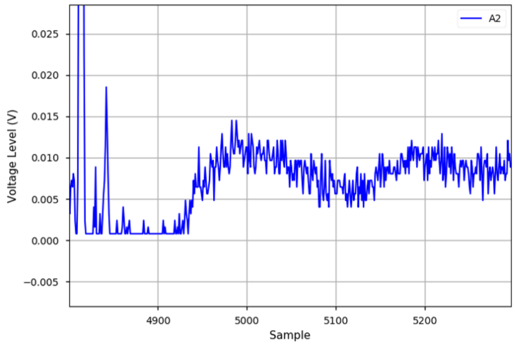


Figure 56: Amplified results of a string vibration block with DC

As for this project a 12 bit resolution will be used the minimum value the ADC can possibly capture in theory is $\frac{3.3}{2^{12}} = 800\mu V$. In practice, as already validated in section 5.2.2, this value should be about 4mV. By introducing a resistance from Vcc to ground the DC value will be created by the voltage divider between two resistances. By choosing a high value of about 10 MΩ the final DC value will be $V_{DC} = 3.3 * \frac{10M\Omega}{10M\Omega + 680k\Omega} = 15mV$. By implementing this solution objective (4) is therefore complete.

The final resulting circuit is presented in Figure 57.

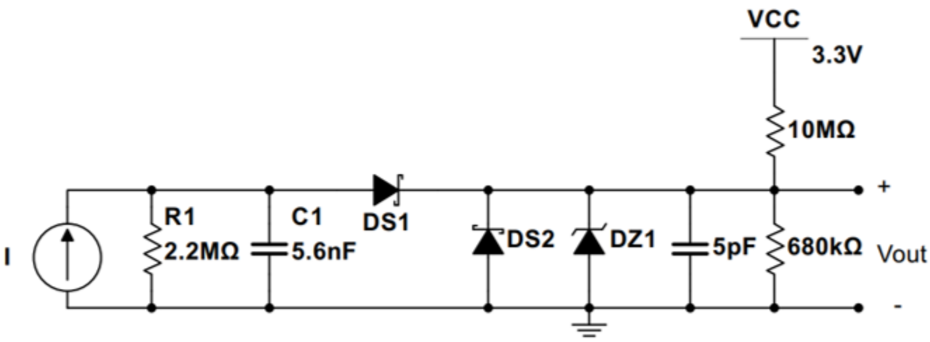


Figure 57: Final implementation of passive circuit

The final results are displayed in Figures 58 and 59. Even though this solution works it lacks scalability. For that reason a better solution was developed and is presented in section 7.2.4. This does not mean that this solution has no use however, its pros and cons will be discussed further on in section 7.2.6

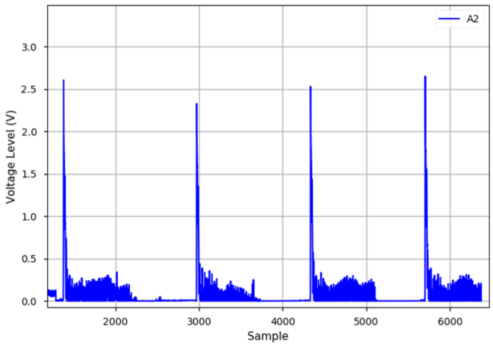


Figure 58: Resuslts of a 4 consecutive plucks using the same force

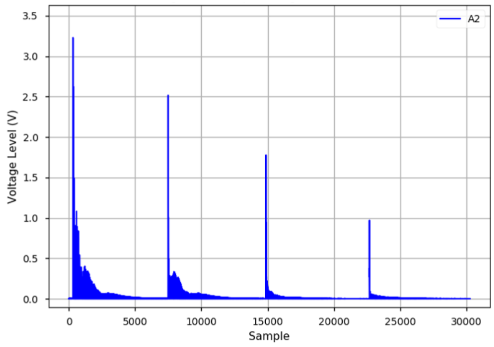


Figure 59: Results of 4 consecutive pluck with decreasing force

7.2.4 Active Alternative

As seen in section 7.2.2 the biggest problem with the previous design is output impedance which prevents using high sample rates. One way of decreasing output impedance is by adding a buffer to the circuit. Most of the designs presented during research for this paper show amplification done through a simple inverting Opamp (Operational Amplifier) configuration. However, this solution is wrong even though it works for the wrong reasons. As presented in section 2.2.1, the piezo does not behave as a voltage input but rather as a charge input. For that reason a charge amplifier needs to be implemented. The circuit idealized for this implementation is presented in Figure 60.

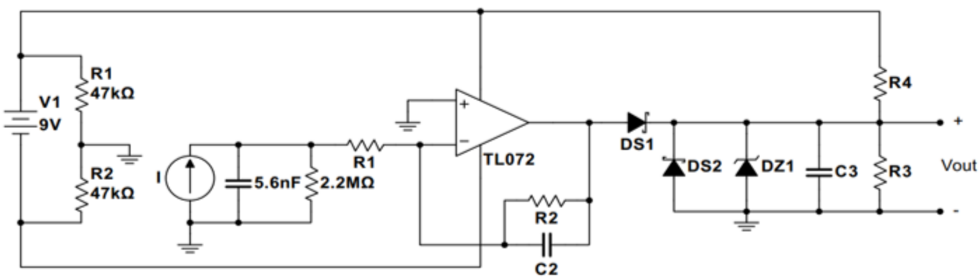


Figure 60: Initial Idealization for an Active MIDI implementation

A charge amplifier converts an input charge into an output voltage. The gain of the charge amplifier without considering the resistance R2 can be considered using the equation presented in (6). Because the gain shows a negative sign the piezo will in this case be oriented opposite to the passive implementation. Also, the amplifier will be fed with 9V with the signal operating differentially at 4.5V. Meaning the output will vary from -4.5V to 4.5V maximum. From the ADC's perspective, if the ground is set to 4.5V the behaviour results will be the same. The amplifier in use will be the TL072 [29] due to low price and availability.

$$I_1 = I_2 \Leftrightarrow \frac{v_1}{R_1} = -C_2\frac{dv_{out}}{dt} \Leftrightarrow \int \frac{v_1}{R_1}dt = \int -C_2\frac{dv_{out}}{dt}dt \Leftrightarrow$$
$$\Leftrightarrow v_o = -\frac{1}{C2}\int \frac{v_1}{R1}dt \Leftrightarrow v_o = -\frac{Q}{C2} \quad (6)$$

Equation represented in (6) shows that the gain can be controlled by the feedback capacitor and the initial charge which can be estimated by equation (1). The resistance R2 is usually added to the charge amplifier to allow current to flow and prevent the output to saturate, due to the amplifier not being an ideal model in practice. However, resistance R2 plays an important role in frequency response which can be be modelled using equation (7). This equation shows a typical behaviour of a band pass filter.

$$\frac{V_o}{V_i} = -\frac{jwC_1R_2}{(1+jwC_1R_1)(1+jwC_2R_2)} \qquad f_{hp} = \frac{1}{2\pi R_2C_2} \qquad f_{lp} = \frac{1}{2\pi R_1C_{piezo}} \quad (7)$$

Also, equation 7 shows that the feedback impedance, made from C_2 and R_2, will work as an high pass filter attenuating low frequencies and the input impedance, made from R_1 and C_{piezo}, will work as a low pass filter attenuating high frequencies. In order to define values the first step is to quantify the charge to be amplified. By connecting a 15nF capacitance the output between an amplified output and a non amplified signal is shown in Figure 61.

The relation shown is clearly a 3:1 attenuation which is equivalent to the relation between the 15nF:5nF capacitance (Added Capacitance C2 : Internal Capacitance C_{piezo}). For the MIDI implementation the R2 resistance should be high enough allowing low frequencies to pass. This means that low frequencies from pressing the piezo will pass and allow the circuit to capture and send a MIDI OFF behaviour. For that, a 50MΩ was chosen resulting in a frequency cutoff of $F_{hp} = 0.31H_z$. Finally the input resistor R1 is not very relevant for this scenario as it will only block high frequencies so a value of 100Ω was chosen resulting in a frequency cutoff of $F_{lp} = 0.28MH_z$

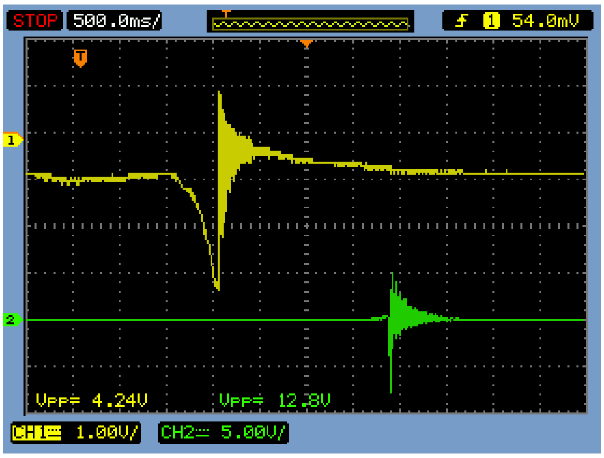


Figure 61: Amplified Output (green) compared to a non-amplified output (yellow)

By doing some tests using the circuit shown and the values mentioned, a small negative DC value shows at the output after some minutes of use. This likely happens due to some charge accumulation from the piezo input which is negatively amplified by the amplifier and prevents MIDI OFF from being captured. This issue can be fixed by adding a capacitor at the output of the amplifier.

Finally the output resistance R3, together with the gain adjusts, can be set to be as low as 50Ω and still maintain a signal between 0V and 3.3V. However having a value that low for the resistance does not bring any practical advantages. It is possible to achieve the maximum sampling rates with a resistance of about 50kΩ.

Regarding capacitor C3, because the sampling speed is at maximum speed the internal capacitance has no time to discharge and creates a very low noise at the output of about 7.5mV. This behaviour can be captured by the ADC and allows to identify the moment a string stops vibrating. Figures 62 and 63 show an example of the noise captured while pressing or not the string at very high sampling and medium conversion speeds. Having a capacitor C3 of high value would consume this noise and for that reason and very low value or no capacitor will be used. Also, because of this, no resistance is needed to implement a DC level. This is an improvement from the passive design as it will reduce current consumption, thus power consumption.

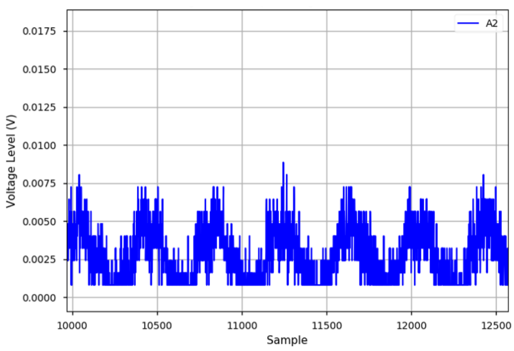


Figure 62: Natural output noise on the ADC

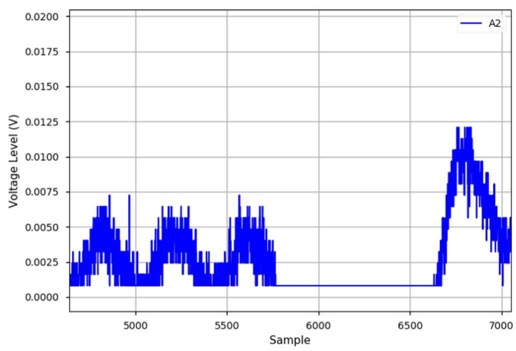


Figure 63: Result of a slightly blocked string

The final circuit design is presented in Figure 64. This implementation allows for up to very high sampling and conversion speeds with a 6μs sampling rate. This means that the signal can be processed and sampled as fast as the micro controller allows with a sample rate as low as the microcontroller can store variables. This is a much more effective implementation however it comes at the cost of power consumption. More detail to this aspect will be given in section 7.2.6.

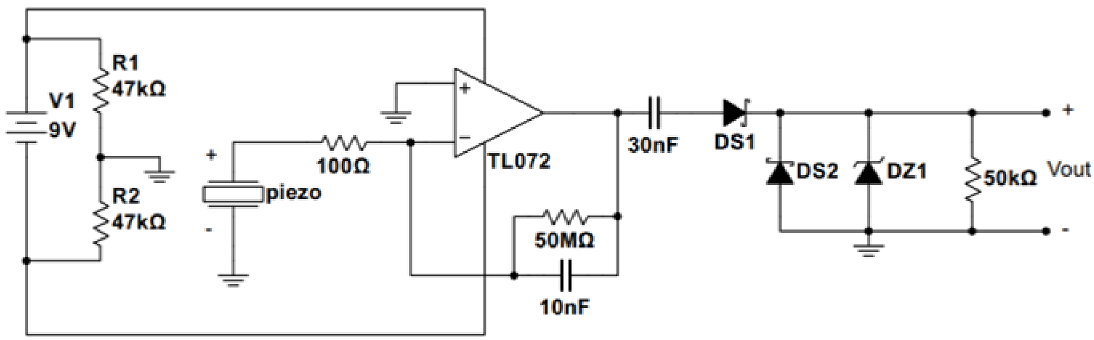


Figure 64: Final implementation of the active circuit

7.2.5 Timings and MIDI Implementation

As seen in sections 7.2.2 and 7.2.4 the minimum times for the ADC to capture the input for the passive and active implementation are 20μs and 6μs respectively, at 12 bit resolution. The absolute maximum sample rate for both implementations is 300 μs. Increasing time beneath this point would decrease signal quality translating in bad velocity values.

Since the harp has a maximum of 47 inputs all the inputs would have to be read within the 300 μs time frame. Meaning that for the passive implementation the maximum inputs that could be read would be $\frac{300\mu s}{20\mu s} = 15 * 2ADC = 40$. This means that even at maximum speed and 2 ADC working independently one microcontroller wouldn't be able to take care of the task within a reasonable amount of time. Also, this is not accounting for the time the actual MIDI program takes to run. The solution for the passive implementation is to have two units run in parallel and connected between them using a communication protocol. In contrast the active solution could account for $\frac{300\mu s}{6\mu s} = 50 * 2ADC = 100$ inputs, allowing theoretically much more inputs to be processed and no need for an external unit.

The idea for the timings implementation is presented in Annex B Figure 75. The proposal is to establish timers that will execute as interruptions for each input needed and start an ADC read for a specific pin. Once the ADC finishes processing it will generate another interrupt which will store the sample. When finished storing the sample, another interrupt will start for the next input and so on and so forth until all inputs have been read and stored. The second ADC will run in parallel and will start reading an input as soon as the timer for the first ADC stops running. Once the two alternate ADCs finish processing the remaining time will be used by the main program to cycle through all the inputs and process accordingly. When finished the timers actuate and the cycle repeats.

The main program will work as a state machine starting on state 0. The diagram for this implementation is represented in annex B, Figure 76. If for any input it detects any value above a minimum threshold it will go to state 1. Once there it will wait for 5 values to guarantee that the maximum value is captured. This maximum value should correspond to the velocity peak.

After collecting all samples it will go to state 2, normalize the input value between 0 and 127 (as MIDI velocity can only be sent as a 7 bit value) and send a MIDI ON message. After that it will generate a timer proportional to the velocity. This timer will serve as a reference to indicate when the string stops vibrating.

If the timer ends or several consecutive values below the minimum MIDI threshold are detected a MIDI OFF message will be sent and the program goes back to the initial state 0.

Considering all the program delays the results are presented in table 2.

	Total Time (μs)	**Total Time with Delay (μs)**	**Total Time to Send (ms)**	**Worst Case Delay (ms)**
Passive Implementation	20	22	1.32	1.58
Active Implementation	6	8	0.96	1.15

Table 2: Timing Response of the MIDI Implementation

The end results show that both implementations are able to accomplish objective (1). The passive implementation, even though with higher value, has its inputs distributed between two microcontrollers, which means four ADC's. Even so, the active implementation using only one microcontroller can still implement with lower time consumption and fulfil objective (1). It would still be possible to lower the bit resolution to achieve faster results even though this would come at the price of lower accuracy, not bringing any advantages. In the final result a 12 bit resolution is used because it results in a conversion of 32:1 as MIDI works at only 7 bit resolution.

Finally, the worst case scenario is considered as the time it would take if all the strings would play at the same time. Even for a worst case scenario the timings would still be in agreement with objective 1. In the end, both implementations work, their choice coming down to price and power consumption.

7.2.6 Comparison of both Models and Final Considerations

In previous sections the active model was defended as being the superior competitor against the passive model. Besides, on a personal level, the active model performs better in terms of response accuracy likely due to faster response time. However, this model comes at the price of power consumption. A very rough estimation can be seen in table 3. In its current state a fully implementation with the active model for the pedal harp would have a very low autonomy of 8 hours.

It is important to refer that the microcontroller is not considered for the active implementation as MIDI messages need a computer or MIDI interface connected to the microcontroller. This connection would generate enough power to support the device working operation. However, when using two microcontrollers for the passive solution the second microcontroller needs to be powered externally. The rest of the circuit would need an external battery as the microcontroller it not able to provide enough power for all the resistors, and amplifiers, in both passive and active implementations.

	Power Consumption (mA)	**Autonomy 9V (h)**	**Autonomy 2*9V (h)**	**Optimized 9V (h)**
Celtic Harp (Passive)	1.9	289	578	-
Celtic Harp (Active)	49.3	11	22	53
Pedal Harp (Passive)	2.6	229	458	-
Pedal Harp (Active)	68.2	8	16	39

Table 3: Power Consumption of the MIDI Implementation

An optimized version using the low power alternative for the TL071 , the TL061 [30], could be implemented raising the autonomy to about 40 hours. However, this alternative was not tested or validated and it is not guaranteed that would work. At this point it could be even considered to power the circuit from an outlet using a transformer. That would be an easy implementation and could be considered in a final implementation as it would not change anything regarding the internal circuit.

Finally, in order to account for the pedals a small button can be introduced on the top of each gap. This button indicates the position of each pedal by showing a 1 or 0 value. This only influences the pitch that will be sent in the MIDI message by -1 or +1 in terms of value. It comes as a simple implementation and will not increase power consumption as it can be powered by the microcontroller. Also, for the chosen microcontroller there are no 47 inputs readily available. This can be overcome using 4 to 1:8 input multiplexers which allow to turn one analog input into 8 inputs using only one multiplexer.

8 Audio Amplification

This section contains the description of all the process making regarding the Audio Implementation.

8.1 Objectives for Audio Amplification

For the purpose of this thesis, the MIDI Implementation was the main motivation for the project and, for that reason, most of the focus came to implement it. However, as the final ambition is to provide a competitive product, it seems natural that an audio implementation can be adapted in parallel with the MIDI implementation. Nonetheless, the subject of amplifying a piezo transducer for an audio implementation has been heavily studied and there are several already proven and implemented models for doing so with good results both scientifically and even personally validated.

Despite this, the objectives for the Audio Adaptation are as follow:

1. The audio implementation should maximize scalability and control from the part of the user;
2. The audio implementation should take into account resonance frequency behaviour from the piezos;
3. The audio amplification should be designed for output levels at nominal audio levels but not restrict the user;
4. The audio implementation, together with the midi implementation, should allow for maximum flexibility;

These objectives define all the steps that need to be considered and will serve as guide to implement the audio adaptation.

8.2 Concept Proposal

As stated, there are many adaptations and proposals for amplifying audio from piezo inputs. The initial idea, after finishing developing the MIDI implementation, was to use a charge amplifier to amplify the piezo signal. Further on, the signal could be filtered and modified as pleased. Finally, the signal should be submitted to a final gain in order to comply with typical audio levels for pre-amplifiers. This allows the user to plug a regular audio cable to the output and connect it to any regular speaker or amplifier.

However, instead of developing a new model, this section will focus on how to adapt a very well written guide, while correcting some minor mistakes, and complement it with visual and scientific justification. The guide in question was already mentioned in section 2.2.3 [15] and was written by the founder of the Logos Fundation, Godfried-Willem Raes. The author of this thesis strongly encourages any reader to backup its project if it so desires to support it.

In order to explain and better follow the design process the initial idealized circuit is presented in Figure 65. The idea is to separate the circuit in 3 different parts: Initial Gain (represented in red), filtering and effects (represented in blue) and final gain (represented in green).

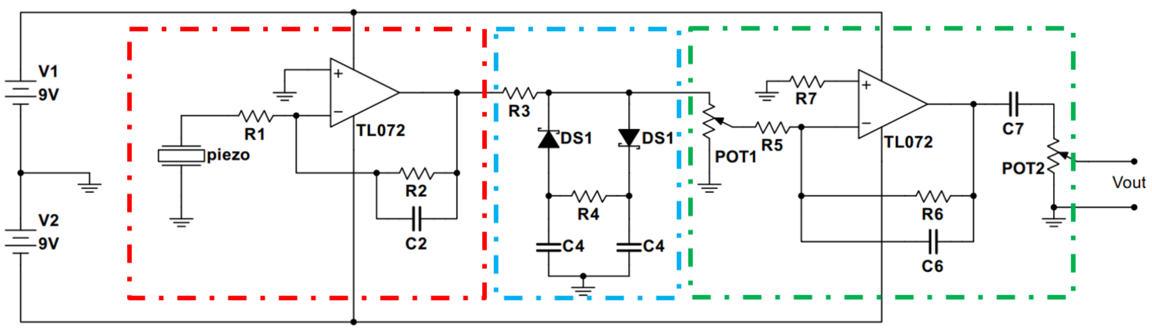


Figure 65: Idealization of a piezo Circuit Adaptation for an Audio Implementation

8.2.1 Initial Signal Conditioning

The initial signal from the piezo input should be amplified by a charge amplifier. The charge amplifier behaviour was already described in section 7.2.4 for the active MIDI implementation. The gain can be controlled by the capacitor C2 has shown in equation (6). As the typical final audio signal is kept within a 2.2 Vpp margin the initial gain should match about 2 times the maximum output. This guarantees that losses further down the circuit can still maintain a 2.2Vpp at the final output. For this application the capacitor chosen was of 15nF resulting in an attenuation of 3 times the input voltage.

For this application frequency response gets more important. The band pass filter response of the charge amplifier was already described in equation 7. After defining the gain, resistance R2, together with C2 will define the low pass filter frequency cut-off. Opposite to the MIDI implementation, where low frequencies should pass on to the system, in the audio implementation these need to be attenuated in order to keep the signal centered around 0 Volts with little oscilation regarding DC level. This filter should not attenuate too much or else the fundamental frequency from the string gets attenuated as well. For the harp, the lowest fundamental frequency can get as low as $20H_z$ for extreme low notes which means that the R2 value should be defined from string to string. The result of a attenuated and non attenuated wave can be seen in Figure 66. When non attenuated, represented in yellow, It is clear that the signal oscillates and tends to zero as it stops vibrating. The attenuated signal, represented in green, shows a more centralized signal by reducing resistance R2.

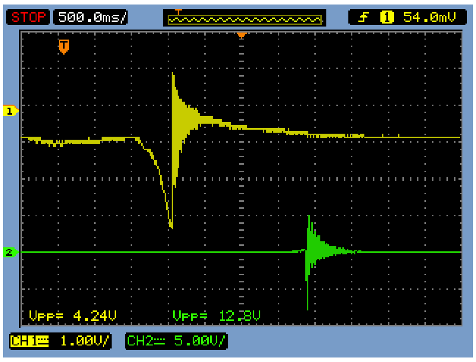


Figure 66: Attenuated DC (green) compared to non attenuated DC (yellow)

For an average, the feedback resistance was set to 220kΩ resulting in a cutoff frequency of $F_{hp} = 48H_z$. However, as already stated, this resistor needs to be defined individually for each piezo.

For the low pass filter, it is important to keep in mind the influence of the resonance frequency as already shown in section 6.2. For the piezos in use, the signal starts getting further amplified from 1kH_z frequency upwards. The input resistance, together with the piezo capacitance, will define the low pass cutoff frequency. By varying the input resistance it is very clear by looking at Figures 67, 68 and 69 that higher frequencies get more attenuated the higher the resistance increases. Especially for 55kΩ the results from the resonance get highly attenuated establishing the typical gain proportion of a vibrating string. However, the higher the input resistance the lower the signal gets in voltage due to the higher voltage drop across the resistor. This behaviour can be compensated with higher gain.

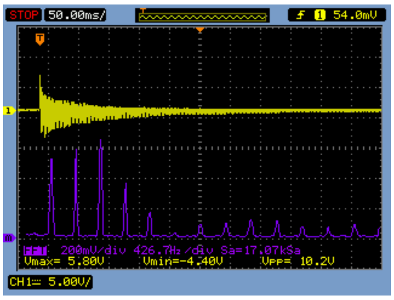


Figure 67: Frequency Response using R1 = 100Ω

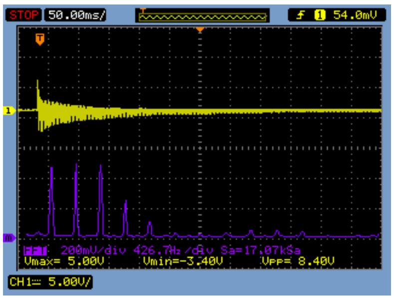


Figure 68: Frequency Response using R1 = 22kΩ

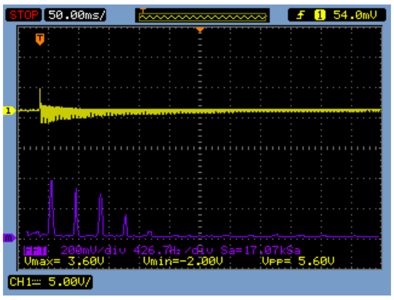


Figure 69: Frequency Response using R1 = 55kΩ

For this application a 55kΩ resistor was used with a resulting cutoff frequency of $F_{lp} = 516H_z$. Even though this decision blocks most of the harmonics characteristics of the string it highlights the fundamental frequency. In the final design potentiometers should be used here and allow the user to regulate the cutoff frequency. For a final product though, having two potentiometer for each frequency cutoff in each string can be a desired feature. However, this implementation would come at a higher price.

In the guide of the Logos Foundation it is stated that the initial amplifier works as a current amplifier. This is not entirely true even though, by making the conversion from current to voltage or, more correctly, from charge to voltage it is in fact a current amplifier. Despite this, it is not a good definition for a charge amplifier.

8.2.2 Filtering and Effects

Regarding filtering and effects, represented in Figure 66 in blue, the design from the Logos Foundation opted for a smoothing limiter. In this part of the circuit it would be possible to add any number or variety of choices, from equalizers, delay modules, noise generators among many others. Another possible design would be to leave the output of the initial amplifier connected to a jack output allowing performers to plug it to any audio device or straight into the final gain module. Once again, this would not only come at a higher price as it could prove effortless to provide such feature for each individual string. For this project, the limiter will be implemented as it is a good choice to smooth the sound from the piezo.

The two schottky diodes DS1 allow the signal to pass in both directions. However, due to their low forward voltage drop, the resulting signal voltage will decrease from about 0.45V to 0.75V according to the datasheet. This value is dependent on the current from the output of the amplifier and will increase the higher the output current. The resistance R3 together with the two capacitances C4 results in a typical low pass filter. In this application the resistance was left at 1kΩ, to minimize voltage drop, and the capacitors set to 100nF resulting in a cutoff frequency of 1.6kH_z.

As stated in the guide, resistance R4 will only serve to determine the response time τ together with capacitors C4. The resulting response time should be about 100ms. The results between a clipped and non clipped signal are clearly visible in Figures 70 and 71. Notice in Figure 71 that even though of lower amplitude, the edges are smother due to the high frequency cutoff. This results in a more "round" sound output, ideal for amplification of string outputs.

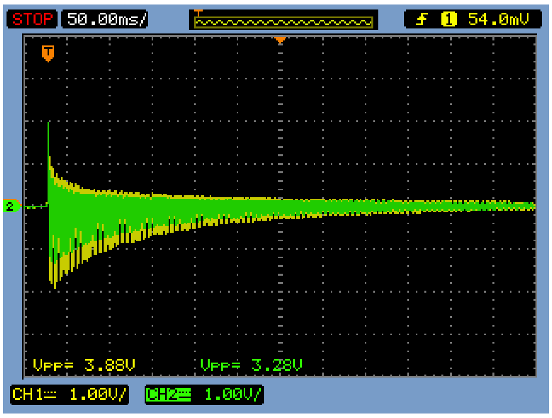


Figure 70: Comparison between a clipped and non clipped signal

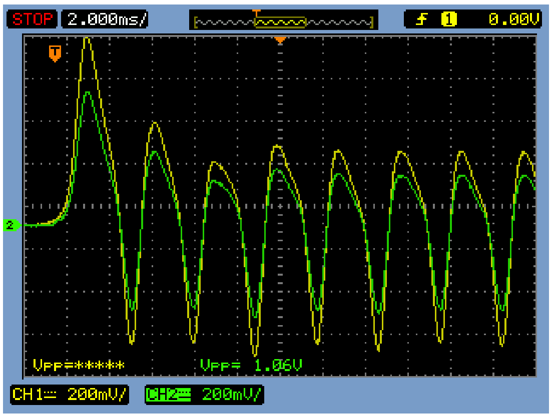


Figure 71: Comparison between a clipped and non clipped signal amplified

8.2.3 Final Gain

The final amplifier consists of a simple inverting adder circuit. The gain can be defined by the inverted relation between the feedback and input resistance. If three inputs were connected to the inverting input the gain can be set by the equation presented in (8). More inputs can be added to the equation repeating the same pattern. The resistance R7 compensates the input error caused by the bias current and should be dimensioned according to equation (8).

$$V_o = -\frac{R6}{R5_1} * V_1 - \frac{R6}{R5_2} * V_2 - \frac{R6}{R5_3} * V_3 \qquad R7 = R5||R6 \tag{8}$$

The gain of the final output highly depends on the average voltage input of each piezo. For this application a gain of 1.5 will be used which allows for some amplification at the output while restraining high values at the input of the amplifier. In the implemented circuit the signal at the input of the final amplifier is around 1.3 Vpp which with amplification reaches about 2Vpp. If several piezos are played simultaneously the signal can reach as high as 2.5Vpp.

However, even though the recommended signal is about 2.2Vpp these 2.5Vpp are only very small peaks and should not pose any damage to the actual amplifier. The capacitor C will act as a final low pass filter in the output in order to block any high frequency noise that may come from internal circuit or possible connections. As the values for R5 and R6 were defined as 50k and 150k respectively this capacitor was set to 200pF resulting in a cutoff frequency of $F_{lp} = 5.3kH_z$. Finally the output capacitor C7 will block any DC output stabilizing the signal within a 0V and the two potentiometers will allow for individual piezo control (POT1) and output voltage control (POT2).

9 Conclusion and Result Analysis

This section contains a final overview on the project with some final considerations on possible optimizations and result analysis.

9.1 Brief Market Evaluation

In other to compare the implementation price for a final application table 4 shows an estimation of the final fabrication price of the project.

	Celtic Harp (Passive) ($)	Celtic Harp (Active)($)	Pedal Harp (Passive) ($)	Pedal Harp (Active) ($)
Piezos	400	400	580	580
Resistors	6	7	9	10
Capacitors	4	5	7	8
OpAmps	-	17	-	24
Microcontroller	40	20	40	20
Multiplexer	-	2	-	2
Total MIDI	450	451	636	644
Resistors	7	7	9	9
Capacitors	7	7	9	9
OpAmps	18	18	25	25
Total MIDI + Audio	482	483	679	687

Table 4: Rough estimation of Final Fabrication Price for a final Implementation in euros

By observing table 4 the final price estimation is about 500$ for the Celtic Harp and 700$ for the pedal harp. This estimative is considering single production prices instead of bulk ones which would lower this value even further. Also, there is no consideration for logistics evaluation which could result in a higher final price, depending on demand and size of the team involved in the fabrication process.

As a reminder, typical prices for implementations from companies such as Kortier and Mountain Glen round from 3000$ to 4500$ with MIDI and audio amplification excluding ports and logistics transportations. This shows a large amount of profit or unoptimized implementations revealing the full potential of this project. Besides, larger companies such as Salvi, which do not present products with such complete applications, could enter the market with a more competitive product from the ground up.

For this reason, even though there is a space for optimizations, this final implementation is a success in regards to its application. Also, without much investigation, by personal contact with harp players and general musicians there is a certain demand for this type of product in the market.

Finally, this product even though made especially for an harp, could also be implemented to any type of string musical instruments. Even if guitars have been made with MIDI and audio amplification for several years, instruments such as cellos or violins still lack competitiveness in the market. It would also be possible to send the product in separate and allow the payer to place the piezo in any string for any kind of artistic purpose.

9.2 Result Analysis

Regarding the MIDI application both implementations have revealed a success even when tested by some musicians on the field. Also, all objectives stated in section 7.1 were accomplished. However, this implementation (both passive and active), was not fully tested and implemented on an harp due to price restrictions and piezo availability. This project was developed using 4 piezos working in parallel and tested in both harps, guitars and ukuleles. A final test using all 34 or 47 piezos, for celtic or pedal harps respectively, would still need to be implemented and validated once all the materials could be obtained.

For this reason, even though this thesis project provides an evaluation and study of the MIDI implementation it does not consist of a final project study and validation.

Regarding audio amplification all objectives stated in section 8.1 were also accomplished even though this project adapted an existing idea for an application. Ideally, a final market product would contain the initial amplification and leave the output completely free for an already made and validated pre-amplifier to be connected, such as the Kortier product.

A custom model, such as implemented, has been proven a good choice, but there are several already developed pre-amplifier capable of delivering similar results with lower power consumption. For this reason, even though there is a study done for the audio application this does not serve at all as final application but as a suggestion for a possible implementation and validation on the project. Also, when discussing a final implementation there is a debate on where to let, or not, the musician control gain and frequency cutoff for each individual string. Ideally this would be a good implementation but in practice having to define cutoff frequency and gain for each of the 47 strings would be too much for a plug and play implementation. In a final project this detail would have to be considered by collecting as many opinions from players and musicians.

Finally, regarding power consumption two batteries were used for this application as one would not suffice in terms of autonomy for a final project. A final implementation would likely contain an option for connecting the harp to an external outlet in order to provide power, instead of depending on external batteries. However, having an option to power the device with batteries is a requirement for most musicians allowing more flexibility.

Regarding both implementations, a final test and implementation would still needed to be done using a full size version, however as a preliminary version both audio and midi implementations serve their purpose and fulfil all proposed objectives.

9.3 Optimizations

Even though many details of this project were studied in detail there is still a certain room for optimizations regarding both final implementation and consideration.

First of all the piezos chosen for this project were highly dependent on availability and price. The whole project was developed around the response from the piezos available instead of the other way around. After having a clear understanding of the specifications needed for the piezo it is clear that a more careful choice should be taken when buying the piezo elements. Also, even though different materials for the piezo were not compared in depth, a final choice should mostly depend on its frequency response according to the section 6.1.

For this product to be complete, according to the author's intentions, the final product should be able to amplify and work with MIDI simultaneously. With the current implementation this is not viable option as both circuits were developed in separate for each application, instead of developed in parallel from scratch. The final solution is not as simple as connecting both in parallel as that would reduce gain for both sides and degrade MIDI and sound output quality. A possible implementation would be to connect both circuits in cascade or to connect the piezo to a first charge amplifier and then design the circuit to work using voltage gain with inverting configurations for the operation amplifiers. However, this implementation was not validated due to time restrictions.

Also, even though the piezos were tested using string of both materials (metal and nylon) results were not constant throughout experimentation for metal strings. Regarding MIDI, there is no problem associated with different materials however regarding audio the sound for metal strings resulted in very harsh outputs. Even when filtered, this behaviour in the resulting sound was still unnatural. It is clear that a more adequate choice for the piezo would have to be considered and adopted however monetary restrictions pose an impediment for further testing.

Finally, to present a final product, the final implementation should be be validated by musicians in the field. As the project, due to money resctrictions, was not implemented in an harp this was unfortunately not possible to do.

10 References

[1] *Camac Harps, France*. https://www.camac-harps.com/en/. Accessed: Feb. 2019.

[2] *Salvi Harps*. https://www.salviharps.com/. Accessed: Feb. 2019.

[3] David Kortier. *Kortier Harpmaker*. http://kortier.com//. Accessed: Feb. 2019.

[4] *Mountain Glen Harps*. http://mountainglenharps.com/. Accessed: Feb. 2019.

[5] *Fishman*. https://www.fishman.com/. Accessed: Feb. 2019.

[6] Jeff Schnellinger. "https://www.sensorsmag.com/components/principles-piezoelectric-accelerometers". In: *Sensors Magazine* (2004). Accessed: 02-2019. URL: http://kortier.com//.

[7] Andrew J. Fleming S.O. Reza Moheimani. *Piezoelectric Transducers for Vibration Control and Damping*. 1st ed. Springer-Verlag London, 2006. ISBN: 978-1-84628-332-1.

[8] Gustav Gautschi. "Piezoelectric Materials for Sensors". In: *Piezoelectric Sensorics: Force Strain Pressure Acceleration and Acoustic Emission Sensors Materials and Amplifiers*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 13–50. ISBN: 978-3-662-04732-3. DOI: 10.1007/978-3-662-04732-3_3. URL: https://doi.org/10.1007/978-3-662-04732-3_3.

[9] Metra Mess- und Frequenztechnik in Radebeul e.K. *Piezoelectric Principle*. https://www.mmf.de/piezoelectric_principle.htm. Accessed: Feb. 2019.

[10] *Piezoelectric Ceramic Products*. PI Ceramic GmbH, Lindenstrasse, 07589 Lederhose, Germany. PI Ceramics. 2015. URL: https://www.nxp.com/docs/en/data-sheet/K64P144M120SF5.pdf.

[11] *Atmega328P 8-bit microcontroller*. rev: 8161. Atmel. Oct. 2013. URL: https://www.sparkfun.com/datasheets/Components/SMD/ATMega328.pdf.

[12] *MK64FX512VMD12 Cortex-M4F 32-bit microncontroller*. rev: 7. NXP Semiconductors. 2014. URL: https://www.nxp.com/docs/en/data-sheet/K64P144M120SF5.pdf.

[13] The MIDI Manufactures Association. *The Complete MIDI 1.0 Detailed Specification*. 3rd ed. rev: 96.1. The MIDI Manufactures Association, 1996.

[14] Scott Helmke. *Mint Box Piezo Buffer*. http://www.scotthelmke.com/Mint-box-buffer.html. Accessed: Feb. 2019.

[15] Logos Fundation. *5-channel mixer and preamplifier for piezo contact microphones*. https://logosfoundation.org/elektron/mixers/piezo_mixer.html. Accessed: Feb. 2019.

[16] Nelson Pass. "Practical Mosfet Testing for Audio". In: *Pass Laboratories* (2003). Accessed: 02-2019.

[17] *PI Ceramics*. https://www.piceramic.com/en/. Accessed: Feb. 2019.

[18] Keysight Technologies. *Oscilloscope DSO1012B Datasheet*. http://www.farnell.com/datasheets/2345279.pdf. Accessed: May 2019.

[19] National Instruments. *Multisim*. https://www.multisim.com/. Accessed: May 2019.

[20] João Beleza. *Python Osciloscope in Real Time from a Serial Interface*. https://github.com/Mafiosi/Midi_Harp.git. Accessed: June 2019.

[21] *Teensy 3.5 Development Board*. PJRC. Nov. 2016. URL: https://www.pjrc.com/store/teensy35.html.

[22] pedvide. *Teensy 3.x ADC implementation*. https://github.com/pedvide/ADC. Accessed: May 2019.

[23] Julius O. Smith. *Spectral Audio Signal Processing*. online book, 2011 edition, Accessed: http://ccrma.stanford.edu/~jos/sasp/, May 2019.

[24] PaulStoffregen. *Audio*. https://github.com/PaulStoffregen/Audio. Accessed: May 2019.

[25] *Piezoelectric Disks 7BB-12-9 and 7BB-15-9 Datasheet*. Accessed: muRata Manufacturing Co. May 2019. URL: https://www.sparkfun.com/datasheets/Sensors/Flex/p37e.pdf.

[26] *Schottky Diode IN5817*. Accessed: Diodes Incorporated. June 2019. URL: https://www.diodes.com/assets/Datasheets/ds23001.pdf.

[27] *Schottky Diode IN5817*. Accessed: multicomb. June 2019. URL: https://components101.com/sites/default/files/component_datasheet/3.3V-Zener-Diode-1N4728A-datasheet.pdf.

[28] *ADC Sizing Guide*. Accessed: Texas Instruments. June 2019. URL: http://www.ti.com/lit/an/spna061/spna061.pdf.

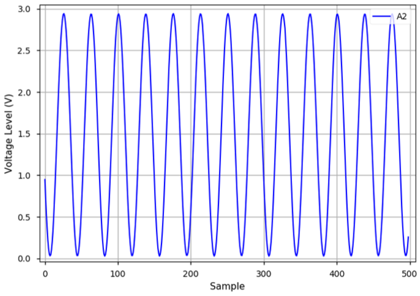
[29] *TL072 Operation Amplifier*. Accessed: Texas Instruments. June 2019. URL: http://www.ti.com/lit/ds/symlink/tl072.pdf.

[30] *TL061 Operation Amplifier*. Accessed: Texas Instruments. June 2019. URL: http://www.ti.com/lit/ds/symlink/tl061.pdf.

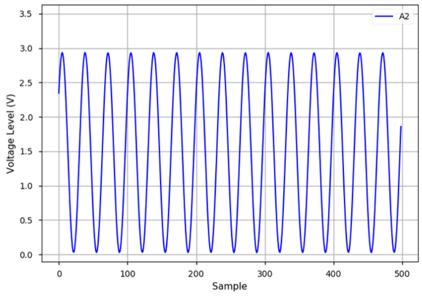
A Teensy and Python Response Times

	Sampling Speed	Conversion Speed	Avg. Sample Collect Time (ms)	Avg. Total Sample Send Time (ms)	Avg. Single Sample Send Time (μs)	Avg. Single Sample Receive Time (μs)
Test A 10 bits	Very Low	Very Low	26.322	7.527	4	8
Test A 16 bits	Very Low	Very Low	29.983	9.810	5	8
Test B 10 bits	Very Low	Medium	15.660	7.538	3	7
Test B 16 bits	Very Low	Medium	16.989	9.955	4	9
Test C 10 bits	Very Low	Very High	6.203	7.540	3	8
Test C 16 bits	Very Low	Very High	9.134	9.902	5	8
Test D 10 bits	Medium	Very Low	19.858	7.421	4	9
Test D 16 bits	Medium	Very Low	22.526	9.200	5	9
Test E 10 bits	Medium	Medium	11.428	7.538	4	8
Test E 16 bits	Medium	Medium	12.261	9.100	4	8
Test F 10 bits	Medium	Very High	5.470	7.530	4	8
Test F 16 bits	Medium	Very High	5.804	9.061	4	9
Test G 10 bits	Very High	Very Low	17.063	7.320	4	8
Test G 16 bits	Very High	Very Low	18.426	9.890	5	9
Test H 10 bits	Very High	Medium	10.332	7.538	4	8
Test H 16 bits	Very High	Medium	11.662	9.860	5	9
Test I 10 bits	Very High	Very High	5.070	7.538	4	9
Test I 16 bits	Very High	Very High	5.606	9.748	5	9

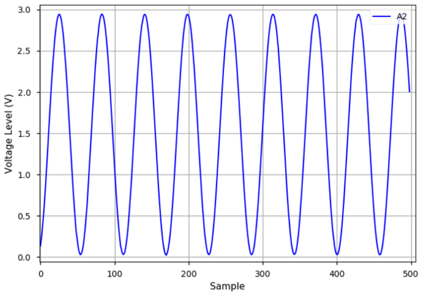
Table 5: Teensy ADC and Serial Interface Response Times



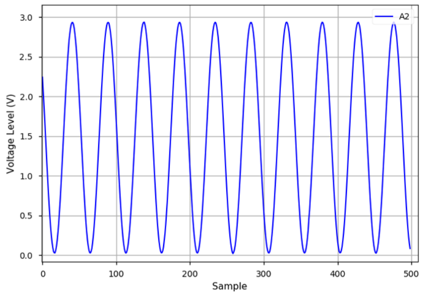
(a) Test B 10 bits at 3 Vpp



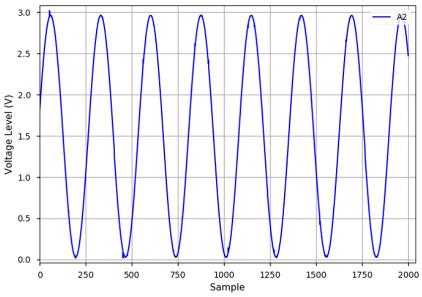
(b) Test B 16 bits at 3 Vpp



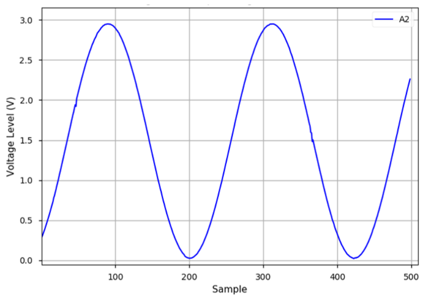
(c) Test E 10 bits at 3 Vpp



(d) Test E 16 bits at 3 Vpp

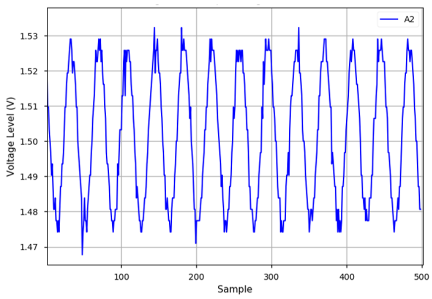


(e) Test I 10 bits at 3 Vpp

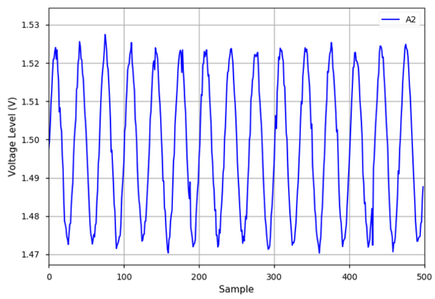


(f) Test I 16 bits at 3 Vpp

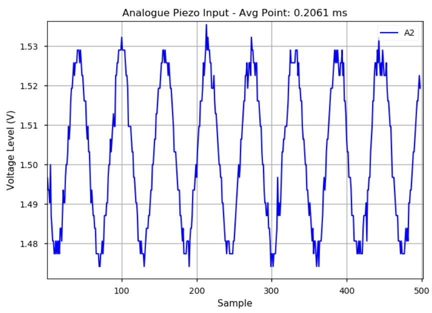
Figure 72: ADC Results for a 3 Volts Peak to Peak Wave at $5kHz$



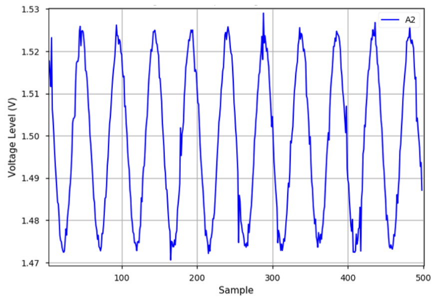
(a) Test B 10 bits at 50 mVpp



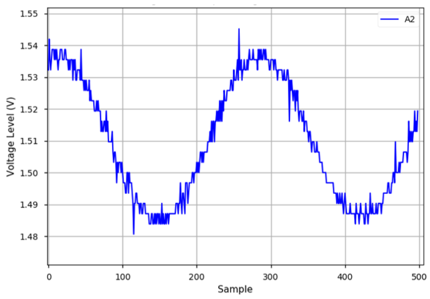
(b) Test B 16 bits at 50 mVpp



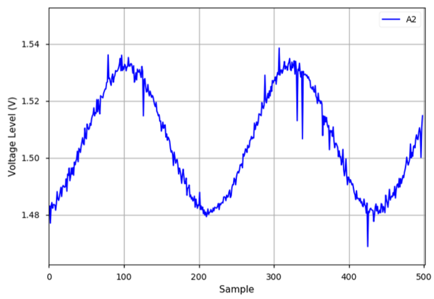
(c) Test E 10 bits at 50 mVpp



(d) Test E 16 bits at 50 mVpp



(e) Test I 10 bits at 50 mVpp

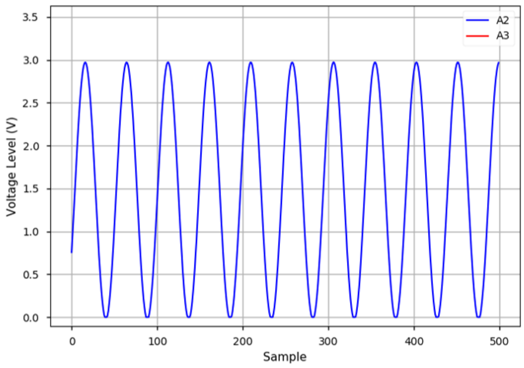


(f) Test I 16 bits at 50 mVpp

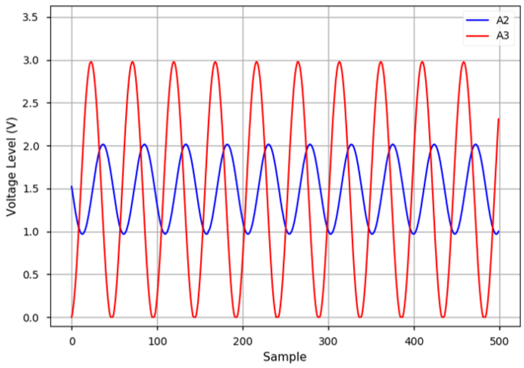
Figure 73: ADC results for a 50 mVolts Peak to Peak Wave at 5kHz

	Test (a)	Test (b)	Test (c)	Test (d)
Avg. Collect Time (ms)	11.663	11.443	25.254	26.005
Avg. Send Time (ms)	9019	19481	19470	29635

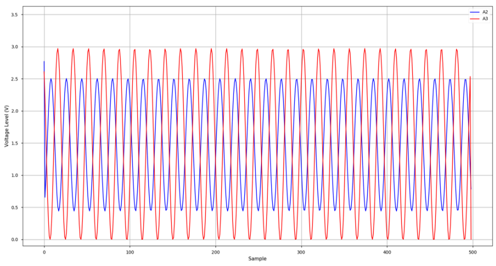
Table 6: Average ADC and Serial Interface Response Time for concurrent tests realized in Figure 74



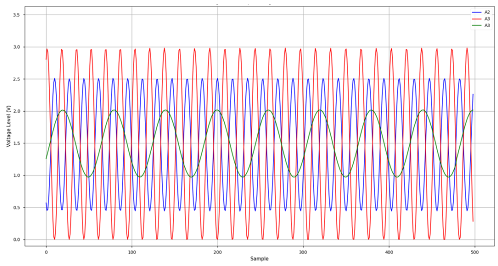
(a) One ADC Reading one Input



(b) Two ADCs Reading one Input each



(c) One ADC Reading two Inputs



(d) Two ADCs reading one and two inputs each

Figure 74: ADC results for a 1, 2 and 3 Volts Peak to Peak Wave at $5kHz$ using 16 bit Resolution, Very High Sampling Speed and Medium Conversion Speed

	Sampling Speed	Conversion Speed	Total Time Taken (μs)
Test 1_12	Very Low	Very Low	13.5
Test 2_12	Very Low	Low	13.0
Test 3_12	Very Low	Medium	7.6
Test 4_12	Very Low	High	4.8
Test 5_12	Very Low	Very High	3.5
Test 6_12	Low	Very Low	11.2
Test 7_12	Low	Low	11
Test 8_12	Low	Medium	6.7
Test 9_12	Low	High	4.4
Test 10_12	Low	Very High	3.0
Test 11_12	Medium	Very Low	9.8
Test 12_12	Medium	Low	9.5
Test 13_12	Medium	Medium	5.8
Test 14_12	Medium	High	3.8
Test 15_12	Medium	Very High	2.8
Test 16_12	High	Very Low	8.7
Test 17_12	High	Low	8.5
Test 18_12	High	Medium	5.2
Test 19_12	High	High	3.7
Test 20_12	High	Very High	2.6
Test 21_12	Very High	Very Low	8.5
Test 22_12	Very High	Low	8.3
Test 23_12	Very High	Medium	5.1
Test 24_12	Very High	High	3.6
Test 25_12	Very High	Very High	2.5

Table 7: Average ADC Response Time for Single Measurements using no Averaging and 12 bit Resolution

	Sampling Speed	Conversion Speed	Total Time Taken (μs)
Test 1_16	Very Low	Very Low	14.7
Test 2_16	Very Low	Low	14.2
Test 3_16	Very Low	Medium	8.2
Test 4_16	Very Low	High	5.0
Test 5_16	Very Low	Very High	4.8
Test 6_16	Low	Very Low	12.8
Test 7_16	Low	Low	12.5
Test 8_16	Low	Medium	7.1
Test 9_16	Low	High	4.6
Test 10_16	Low	Very High	3.3
Test 11_16	Medium	Very Low	11.4
Test 12_16	Medium	Low	10.6
Test 13_16	Medium	Medium	6.2
Test 14_16	Medium	High	4.1
Test 15_16	Medium	Very High	2.9
Test 16_16	High	Very Low	9.3
Test 17_16	High	Low	9.1
Test 18_16	High	Medium	5.8
Test 19_16	High	High	3.8
Test 20_16	High	Very High	2.7
Test 21_16	Very High	Very Low	9.2
Test 22_16	Very High	Low	9.1
Test 23_16	Very High	Medium	5.8
Test 24_16	Very High	High	3.8
Test 25_16	Very High	Very High	2.8

Table 8: Average ADC Response Time for Single Measurements using no Averaging and 16 bit Resolution

	Sampling Speed	Conversion Speed	Resolution	Averaging	Total Time Taken (μs)
Test A	Very Low	Very Low	10	0	13
Test B	Very Low	Very Low	10	4	41
Test C	Very Low	Very Low	10	8	90
Test D	Very Low	Very Low	10	16	180
Test E	Very Low	Very Low	10	32	360

Table 9: Average ADC Response Time for Single Test with an increasing averaging

•

B Timings and State Diagram

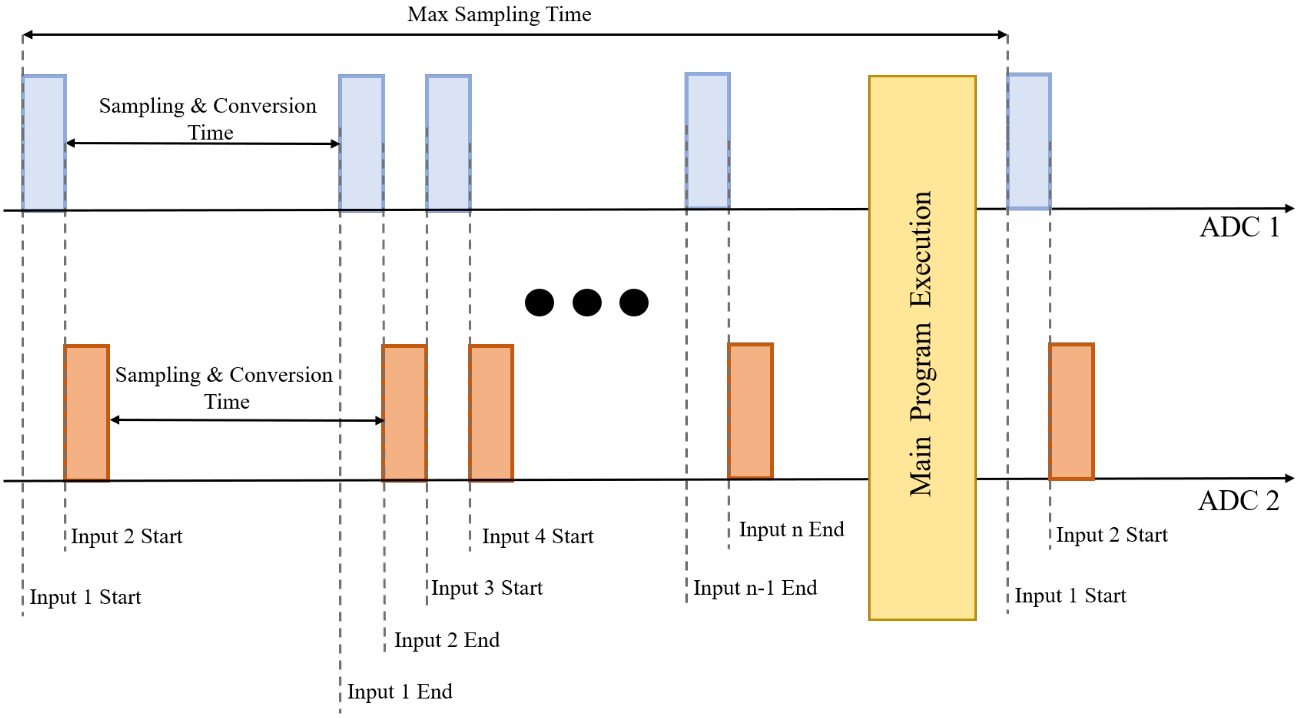


Figure 75: Timings schematic for MIDI implementation

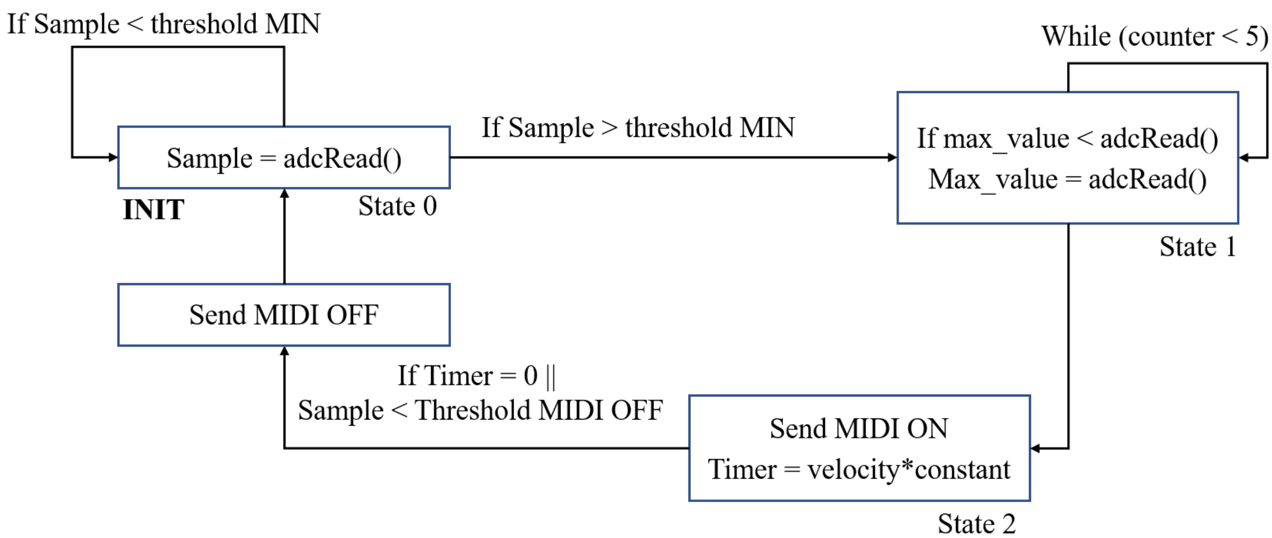


Figure 76: State Diagram Exemplification of Code Implementation