DIGITAL VOICE RECORDER

EGB240 Electronic Design

Group 314
Daniel Pearson - N9711023
Matthew Fagg – N9447334
Siddhant Mahapatra - N9840371

Executive Summary

This technical report documents design and construction of a Digital Voice Recorder. The primary aim of the task was to record voice segment through a microphone at a reasonable distance and playback those segment via a loudspeaker so that the voice is heard easily.

The design is constructed with the following components.

- 1 x Development board w/ breadboard and USB cable
- 1 x Electret microphone
- 1 x Miniature 8Ω loudspeaker
- 1 x LMC6484 quad rail-rail operational amplifier IC
- 1 x LM386 audio amplifier IC

Designing of the Digital Voice Recorder was in accordance with certain literature that provided in depth design aspects that needed to be included in the project. Since the use of a sampling frequency, Sampling Theorem is among the literature that provided in depth resources.

After initial research, theoretical analysis and simulation, an input via the microphone through an anti-aliasing filter with a cut-off frequency of 2.5 kHz and roll-off rate of 80 dB/decade with amplification through, was chosen to act as the input conditioning. Both theoretical calculations and simulation are offered to justify the decision for this input conditioning configuration.

Also through research, theoretical analysis and simulation, the output of a PWM from the microcontrol a reconstruction filter with a cut-off frequency of 5 kHz with a roll-off rate of 20 dB/decade, was chosen to act as a digital to analogue converter. Amplification was added to the end of the reconstruction filter to amplify the output to a speaker to a respectable level that can be heard easily and clearly.

Designing the input and output circuitry was done using a simulation environment. Following the design and simulation of components, components were constructed on a breadboard. Measurement devices were used to test the functionality of individual components and the overall input and output sections.

Code was written in C to be the firmware that implements the functionality of recording, stopping and playback of 10 second voice segment stored on the microcontroller's onboard SD card. The code accepts the input continuous time wave through the header Jin, the code processes it with the sampling frequency of 15.625 kHz, before outputting a PWM through the header Jout.

Experimental results show that the Digital Voice Recorder is able to achieve a level of amplification that is desirable.

The final project proved to be a success and the voice through the microphone was clearly able to be heard through the use of amplification through the circuitry and was clear with no distortion through the use of filtering and decoupling capacitors.

Table of Contents

List of tables and figures

Introduction

The purpose of this project is to design, prototype, and demonstrate a digital voice recorder based around a QUT-designed development board containing an ATmega23U4 microcontroller.

The circuit can be broken into two key stages;

Input

Sound is first registered by a microphone, the waveform requiring both amplification and filtering before it can be appropriately recorded.

Filtering is a crucial step to ensure the recording is of high fidelity and clarity, the choice of filter and its specifications heavily impacting the performance of this key stage. Amplification is required to raise the amplitude of the signal, ensuring suitable playback volume, however excessive amplification will result in additional noise or signal clipping, requiring a balance to be found.

Output

The recording is output as a pulse width modulated signal. Using a reconstruction filter, the jaggard digital signal can be smoothed into an analogue waveform, which can then be amplified using an audio amp and played back on a speaker.

The output amplification is based around the LM386 IC, however, to achieve the desired volume output and fidelity, multiple alterations are required.

This report outlines the specific technical aspects of the project and design process, and the final application and performance.

Literature Review

Sampling Theorem

Sampling theorem is a fundamental bridge between continuous time signal (aka analog signals) and discrete time signals (aka digital signals).

Signals generally are analog in nature, this includes temperature, sound, voltage and many more.

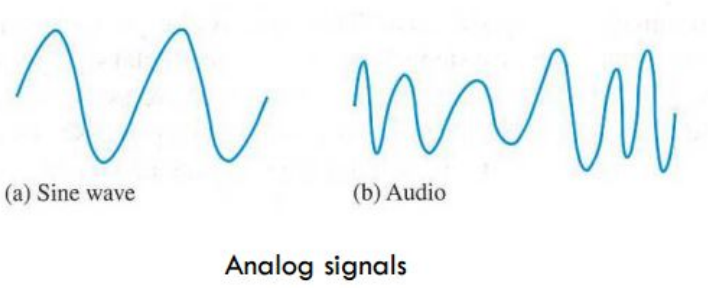

Figure 1 - Analog signal comparison

Were as digital signals have a discrete set of values. Digital signals are seen within computers and digital system such as mobile phones, tv's hard drives. The digital signal is represented by two binary digits, 1 and 0. 1 is called high, 0 is called low.

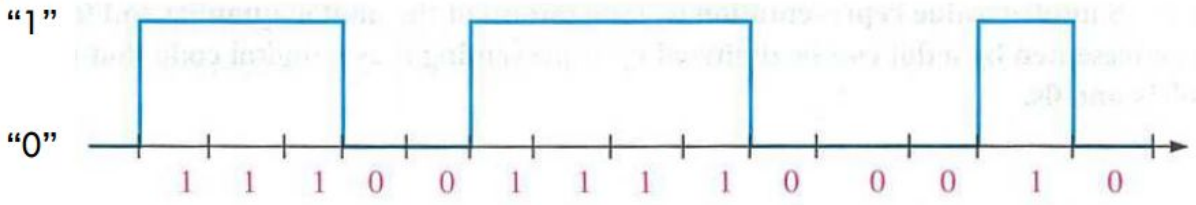

Figure 2 - Digital signal representation

To convert an analog signal to a digital signal involves using an Analog to Digital Convert. By doing so, we can begin to use electronics to interact with the analog signals around us. Using an Analog to Digital converter involves three steps: 1. Sampling 2. Quantization 3. Coding.

During the sampling process, an analogue signal is converted into digital signal by taking samples of the continuous-time signal at discrete time intervals. In order to convert the analogue signal to a digital signal to an exact match, the sampling frequency must be greater than twice its highest frequency component.

$$f_s \geq 2f_m$$

Pulse Width Modulation (PWM)

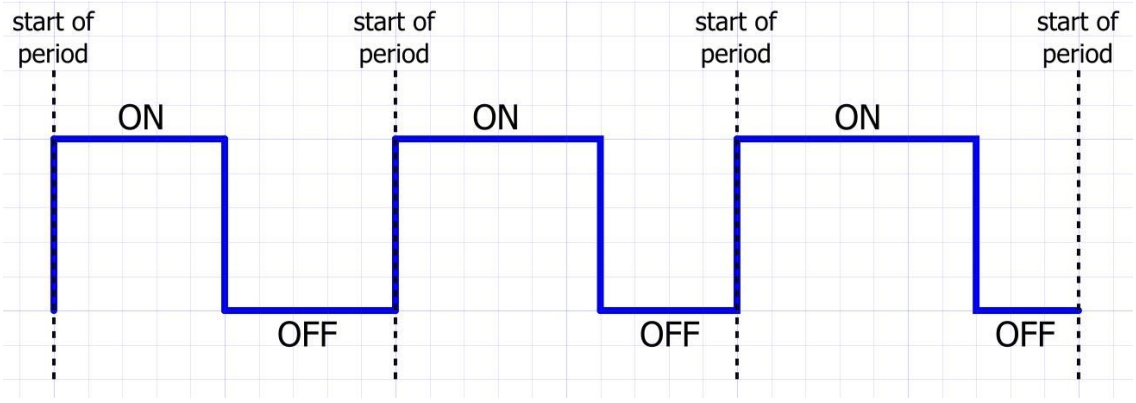

Figure 3 - Pulse width modulated waveform

The duration of the logic-low voltage is equal to the duration of the logic-high voltage in a typical digital clock signal which is a sequence of periods. In contrast, according to external conditions , the duration of the logic-high / logic-low voltage varies in a PWM signal which is also a sequence of periods , and these variations can be used to transmit information. Sinusoidal signals is the means of transmitting information in radio circuitry, to which some type of modulation is applied . We have pulse-width modulation instead of amplitude or frequency modulation and this situation is analogous to PWM functionality . We all know that an analog audio signal can be transmitted from an antenna to a car radio by first modulating a carrier wave and then processing the received signal in a way that removes the carrier and recovers the original audio information. Likewise, we can generate a programmable analog voltage by pulse-width modulating a digital carrier wave then "transmitting" this modulated signal to a low-pass filter.

In the above diagram, logic high is identified as the "ON" or active state, and logic low is the "OFF" or inactive state. In the first period, the duration of the active state is equal to the duration of the inactive state. Then, for the next two periods, the active-state duration increases by one grid width; this means that the inactive-state duration must decrease by one grid width, because the PWM carrier frequency (and thus the PWM period) is constant. In the context of our PWM DAC, we don't really need to know the absolute active and inactive durations; what matters is the ratio between the durations, which we discuss in terms of the PWM duty cycle:

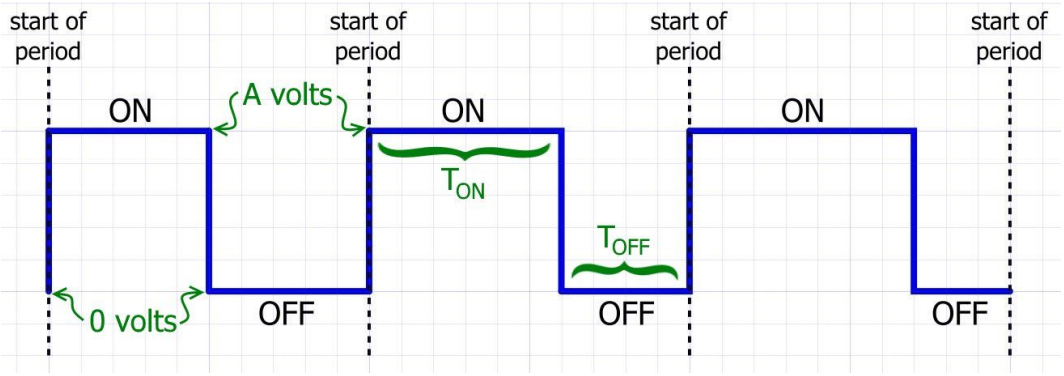

Figure 4 - Pulse width modulation variation

$$duty\ cycle = \frac{T_{ON}}{T_{ON} + T_{OFF}}$$

From Duty Cycle to Analog Voltage:

The nominal DAC voltage observed at the output of the low-pass filter is determined by just two parameters, namely, the duty cycle and the PWM signal's logic-high voltage; in the diagram, this logic-high voltage is denoted by A for "amplitude." The relationship between duty cycle, amplitude, and nominal DAC voltage is fairly intuitive: In the frequency domain, a low-pass filter suppresses higher-frequency components of an input signal. The time-domain equivalent of this effect is smoothing, or averaging—thus, by low-pass filtering a PWM signal we are extracting its average value. Let's assume the duty cycle is 50% (i.e., active duration equals inactive duration) and we are working with 3.3 V logic. You can probably guess what the nominal DAC voltage will be: 1.65 V, because the signal spends half of its time at 3.3 V and half at 0 V, and thus the smoothed-out version will end up right in the middle.

$$desired\ DAC\ voltage = A \times (duty\ cycle)$$

Resolution:

One of the first specs you look at when choosing a DAC is the "resolution," which is a somewhat vague term expressed in the somewhat vague unit of "bits." The number of "bits" refers to the data register that controls the digital-to-analog circuitry, such that a 10-bit DAC can generate 2^{10} = 1024 distinct output voltages.

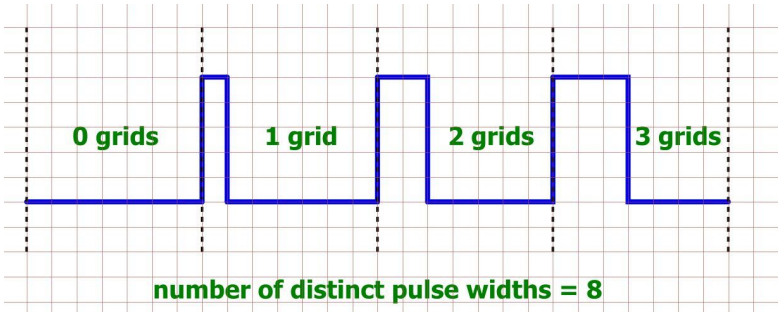

Figure 5 - Pulse width modulation resolution

Let's assume that the PWM signal shown in the diagram is restricted to pulse widths that are a multiple of one grid. This means that the duty cycle can assume 8 distinct values: 0%, ~14%, ~29%, ~43%, ~57%, ~71%, ~86%, and 100%. Each duty cycle corresponds to a particular output voltage, so what we have here is a 3-bit DAC, because 2^3 = 8.

The central element in a standard PWM hardware block is an N-bit counter that controls the width of the pulse, meaning that the equivalent DAC resolution is 2^N.

PWM in the Frequency Domain:

The smoothing is accomplished by a simple low-pass filter. Thus, we can achieve digital-to-analog conversion by using firmware or hardware to vary the PWM duty cycle.

$$desired\ DAC\ voltage = A \times duty\ cycle$$

where A (for “amplitude”) is the logic-high voltage.

Let’s begin our more-thorough exploration of the PWM DAC by looking at the frequency-domain representation of a PWM signal. Here is the LTspice schematic:

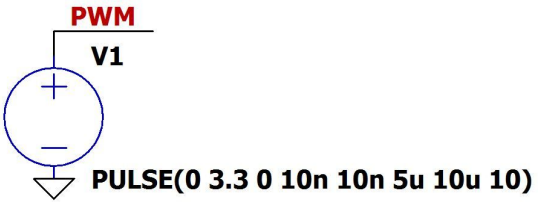

Figure 6 - Pulse signal generator in LTspice

In the PULSE characteristics, the pulse width is 5 µs and the period is 10 µs. Thus, the duty cycle is 50% and the PWM carrier frequency is 100 kHz. Note also that A = 3.3 V and the rise and fall times are both 10 ns. Here is the time-domain signal:

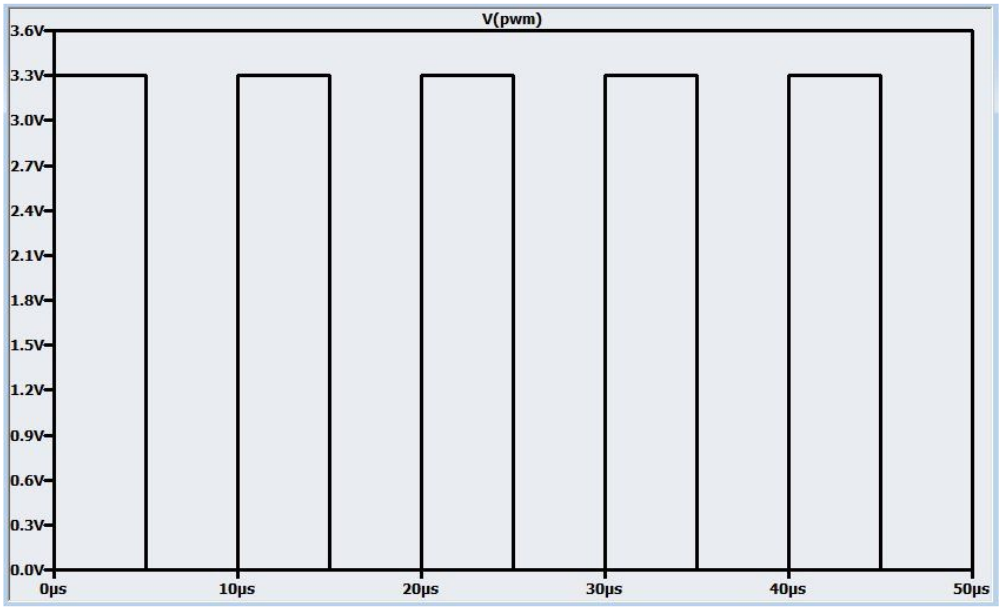

Figure 7 - Pulse width modulated signal in time domain

And here is the FFT:

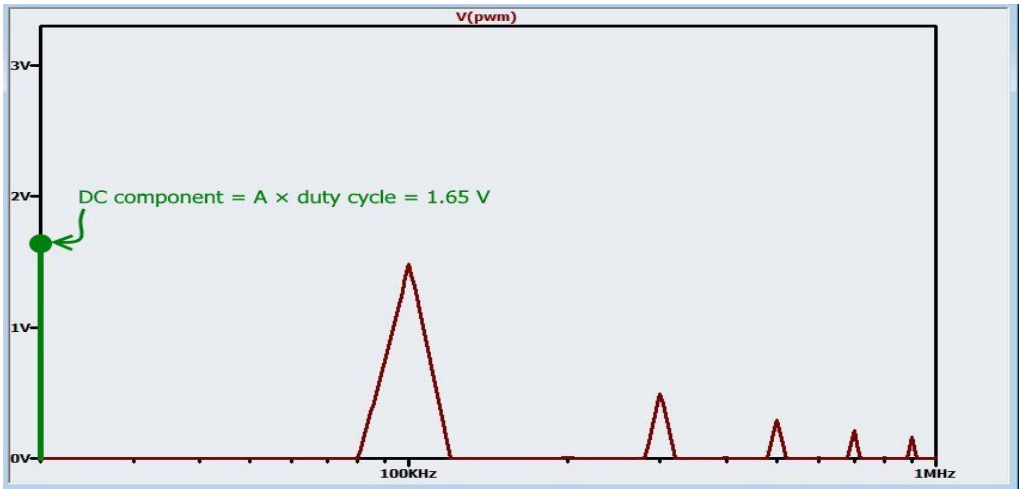

Figure 8 - Pulse width modulated signal in frequency domain

So what we want is that stable 1.65 V over on the far left, and what we don't want is that troublesome spike at 100 kHz (as well as all the higher-frequency spikes). At this point you can probably see why we use a low-pass filter in a PWM DAC: the filter retains the DC component while suppressing everything else. If we had a perfect filter, we would have a perfectly stable DAC voltage—just look back at the previous plot and imagine a "brick-wall" filter that transitions from no attenuation to complete attenuation at 50 kHz. All the non-DC components of the signal would be eliminated, and we would be left with a DC voltage at 1.65 V.

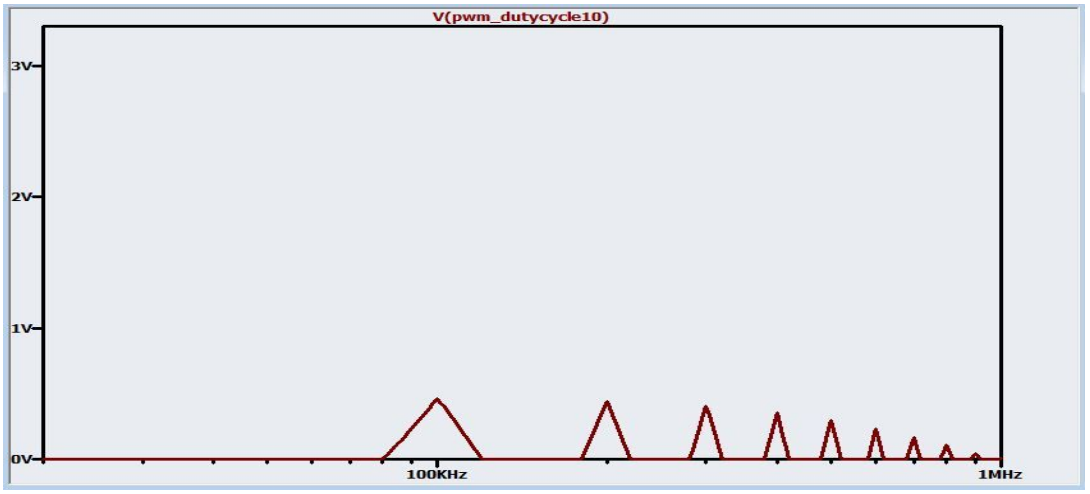

Figure 9 - PWM of 10% Duty Cycle

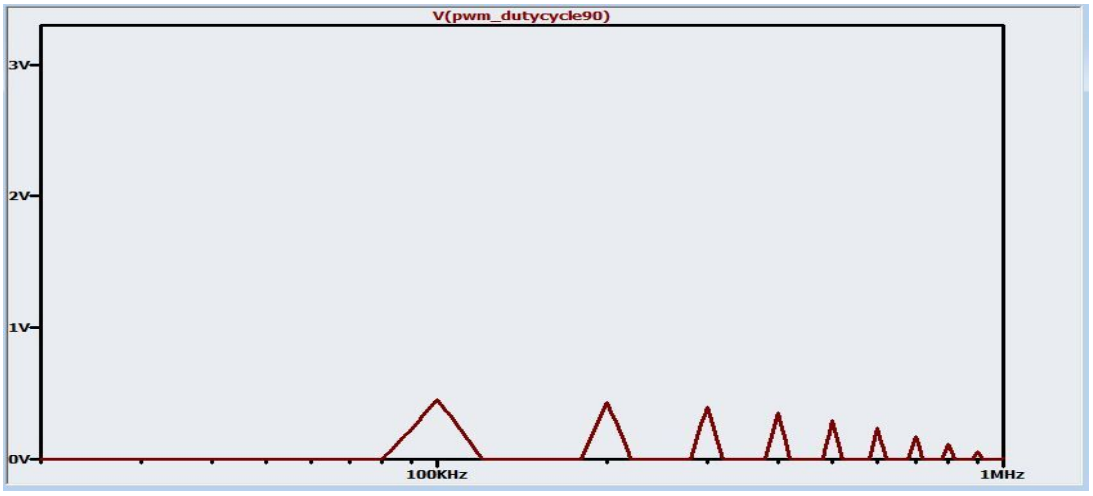

Figure 10 - PWM of 90% Duty Cycle

The spectrum certainly changes relative to 50% duty cycle, but one thing doesn't change: the first spike is at the carrier frequency. So regardless of the duty cycle, we have a fairly large frequency band—in this case, from DC to 100 kHz—in which the low-pass filter can transition from no attenuation to significant attenuation.

A lower cutoff frequency means less ripple and longer settling time; a higher cutoff frequency means more ripple and shorter settling time.

Project Description

The aim of this project, is to design, prototype and develop a digital voice recorder (DVR) using a QUT-designed development board that contains a Teensy microcontroller board. The digital voice recorder is to have input and output conditioning that interact with certain aspects of the Teensy microcontroller board.The development board additionally supplies 5 V to the breadboard from the USB power supply of the Teensy. An ATmega32U4, 8 bit microcontroller is integrated within the Teensy board.

The input circuitry implemented is to have a microphone used to speak into. Filters are added to reduce noise and amplify the input signal or voice of the user. This input circuitry will interface with the microcontroller where the microcontroller acts as an analogue to digital converter (ADC). To get the Teensy and on board microcontroller to act as an ADC, embedded code, within in C is to be loaded onto the microcontroller.

The embedded code, takes care of converting the input signal to a digital signal. The code also takes care of user inputs and storing of the recorded voice samples onto an SD memory card onboard the Teensy. Pushbuttons on the development board are used as inputs through the embedded code. Three pushbuttons are to be used as inputs. Pushbutton S1 is used to play any recording that is stored onto the SD card. Pushbutton S2's function is to record a new segment of audio and to store it onto the SD card when the maximum record time is reached. The function of the final button, Pushbutton S3, it to stop the playback of any audio playing or to stop recording a new audio segment.

From the embedded code converting the input signal to a digital signal, an output is in the form of a Pulse Width Modulation. A filter is used to reconstruct the output to as close as possible to the input signal. Further amplification to the output is added to the signal. A loudspeaker is used to listen to the reconstructed signal.

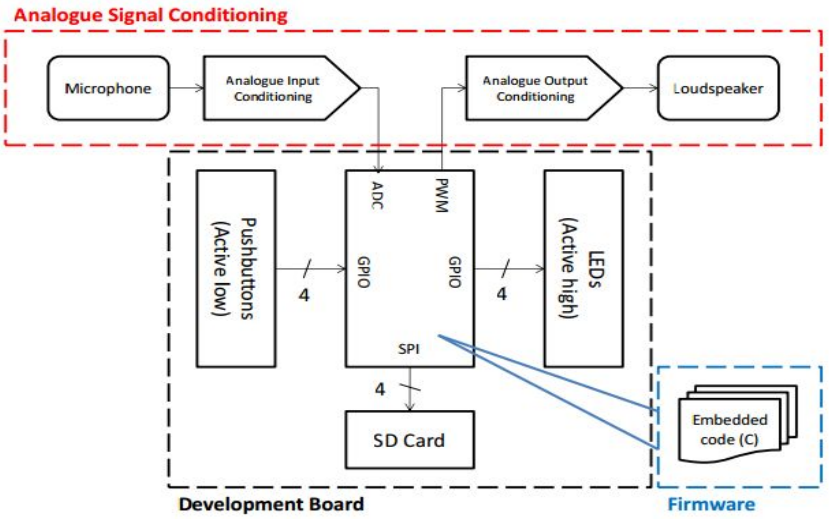

Figure 11 - Digital voice recorder designed according to block diagram.

Design Goals and Specifications

The overall goal of this project is to record audio through a microphone and play it back through a loudspeaker, at as high quality as possible and distortion free. In between the microphone and loudspeaker, a microcontroller interacts with the signal, converting it from an analogue signal to a digital.

An anti-aliasing filter is to be design to restrict the bandwidth of the signal from the microphone. The anti-aliasing filter is to be designed according to sampling theorem mentioned in the literature review. The specified sampling frequency is set at 15.625 kHz and from the formula in the literature review the cut-off frequency should be around 7 - 8 kHz to reconstruct the input signal.

Embedded code is to be written in C. The specifications of the code is to implement the function of playback,recording and stopping of audio segments via the input of three pushbuttons on the Teensy development board. The code implements the function of PWM with a sampling frequency of 15.625 kHz.

A reconstruction filter is also to be design to reconstruct the PWM wave to a sine wave. This sine wave is to be a close as possible to the input sine wave.

The implementation of the project is to be prototype on a breadboard or veroboard or PCB if desired. The layout and construction are to be of a high standard.

As the Teensy is powered by 5 volts via its connection to a USB port. The development board is connected to the top rails of the breadboard it sits on. From this, no outside power supplies are to be used other than the 5 volts supplied by the Teensy.

Scope

The aim of the project is to construct a digital voice recorded specifically designed for recording speech of a normal speaking volume at reasonable distance, and playing it back with sufficient fidelity and volume to distinguish the recorded message with ease.

These goals were set with the knowledge that several factors would limit the quality with which the DVR could perform. These limitations include:

- Ineluctable breadboard noise due to inconsistent connections and interference between rails
- Inexorable component tolerances existing in all practical applications outside of simulation
- Limited speaker size, specifically the diaphragm, restricting the possible volume output
- Obligatory microphone size, capping its possible dynamic range and sensitivity

These difficulties are however unlikely to preclude the previously stated goal of the project.

Methodology

The process of producing this project onto a breadboard included designing , simulation and prototyping.

In the designing phase of the project, research was conducted in accordance to the required specifications of the project. This research was to expand knowledge of applications of how digital voice recorders work. Research about sampling theorem and how PWM work was conducted to design the input and output stages of the project.

For the input stage, research about the use of a microphone and its required specifications via the datasheet. Specification of the LMC6484 were also looked upon to determine how to implement and use this certain operational amplifier. As for the output stage, specifications of the LM368 power amplifier was researched as to how to implement it in the project.

Since the use of filters was known to have to be used within the project, Sampling theorem was used to decide upon what cut-off frequency of what the filters were to be. Also with the use filters, the different types of topologies and variations of filters was researched to find out the most applicable filter for the use within the project.

With the knowledge of all the topics conducted in the researching, the next stage was to design the aspects. In designing the filters and overall circuitry of the project, calculations were used to further justify the use of certain values of components within the circuit. A simulation environment within LTspice was used to simulate the calculations. LTspice progressed the project in a way that could be seen in the real world.

Once all simulation was done, prototyping the whole project was implemented onto a breadboard. Certain values from the calculations and simulations stage of the project were off by 10%. This meant that some aspects of the circuit did not line up correctly with simulation.

The firmware of the microcontroller was written in an IDE called Atmel Studio. The was written along side supporting libraries to assist in coding functionalities required. Recording, playback, stopping and storing functionalities were implemented. Since there was a test code that could be used, coding was the last thing to do on the list.

In the prototyping stage of the project, oscilloscopes were used to measure the workings of the individual aspects of the project, such as the amplifiers.

Once all circuitry was prototyped on the breadboard and the code was finished, talking into the microphone was used to validate the overall functionality of the project.

Technical Design

Voltage Divider

This can be achieved easily through voltage division.

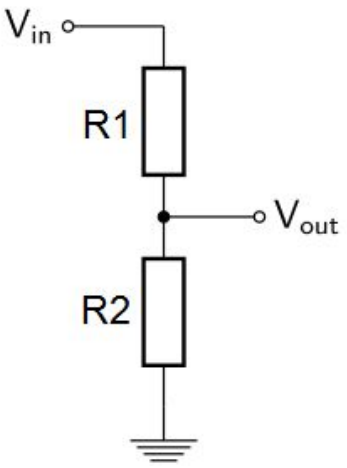

Figure 12 - Voltage Divider

Using resistors, a simple voltage divider can be implemented that obeys the general rule:

$$V_{out} = \frac{R2}{R1+R2} \cdot V_{in}$$

CMA-6542TF-K Microphone

The CMA-6542TF-K is an omnidirectional electret condenser microphone, and is used to form the basis of the recording circuitry.

A schematic is provided to allow measurement of the microphone characteristics.

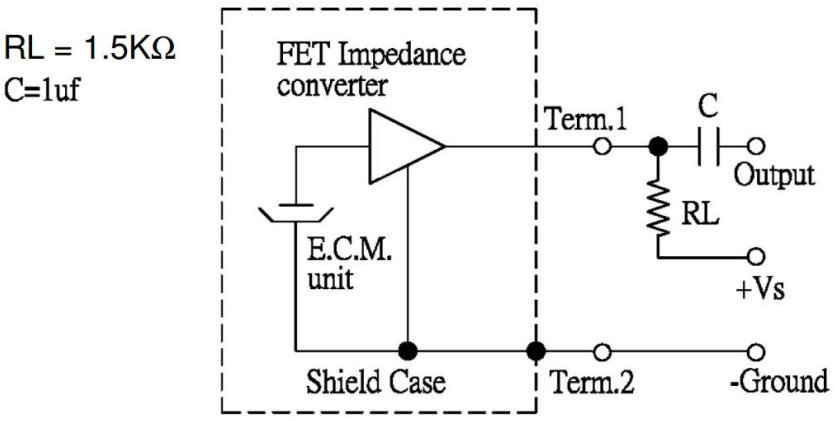

Figure 13 - Measurement Circuit schematic

The circuit acts as a voltage divider, the resistance of the mic altering the amplitude of the voltage passing through to the output, while the capacitor removes any unwanted voltage offset. A frequency response curve is also provided based on findings from the measurement circuit.

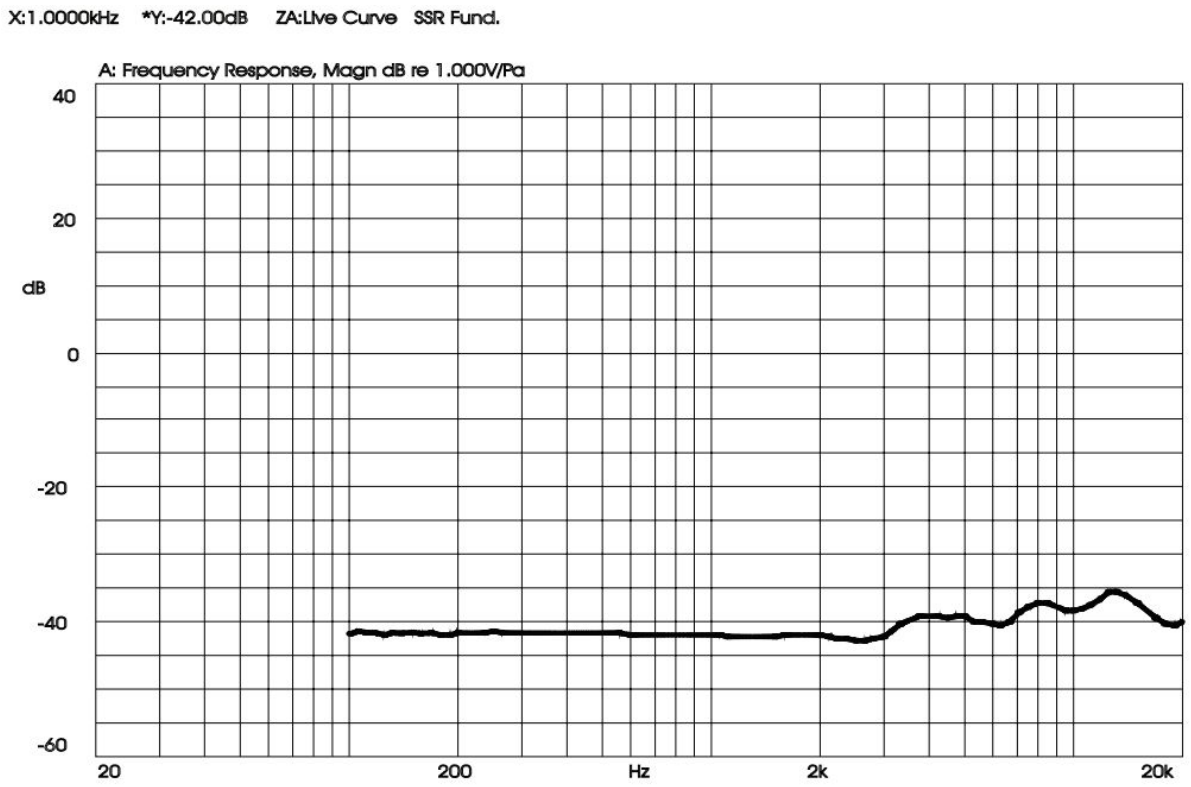

Figure 14 - Frequency Response curve

The microphone's sensitivity increases at high frequencies, indicating low pass filtering may be necessary to avoid high frequency interference noise.

Filters Topologies

Sallen Key

Sallen–Key is an electronic filter topology to build 2nd order active filters that is well known for its simplicity. Sallen–Key filters are often implemented using an opamp as a voltage follower. Getting a high Q factor might need extreme component value spread / high amplifier gain although it possesses a resilient to component tolerance. Two or more stages can be cascaded to increase the order of the filter. Using a unity-gain opamp, a generic Sallen Key topology is used in the figure below.

An opamp is implemented as the buffer in a typical low pass and is also called a +ve feedback filter as the output is fed back into the +ve terminal of the op amp. This type is used the most as it makes use of only a single operational amplifier, making it quite inexpensive.

Drawbacks - 1) The max Q value is very limited and its the reason why it is not used in applications that require a high Q. Q refers to the "quality" factor, the energy ratio of stored component to dissipated counterpart at resonance. Filters with high Q can produce filter roll-offs that are quite sharp. Have their Poles are placed closer to jw axis in filters with high Q values. The design is stable theoretically, as long as the poles are on the left-hand side of the s plane . The stability reduces as the poles get closer to the imaginary axis .

The damping factor is Q inverse, and more applicable to high-pass and low-pass applications.

Q is typically around 5, in a single opamp Sallen Key filter.

2) Min required open loop gain($90Q^2$) is high compared to the gain of this circuit (-3Q). The max cutoff frequency must be significantly lower than GBW product of the amplifier . This results in a better/high performance amplifier so that the filter response does not get adversely affected.

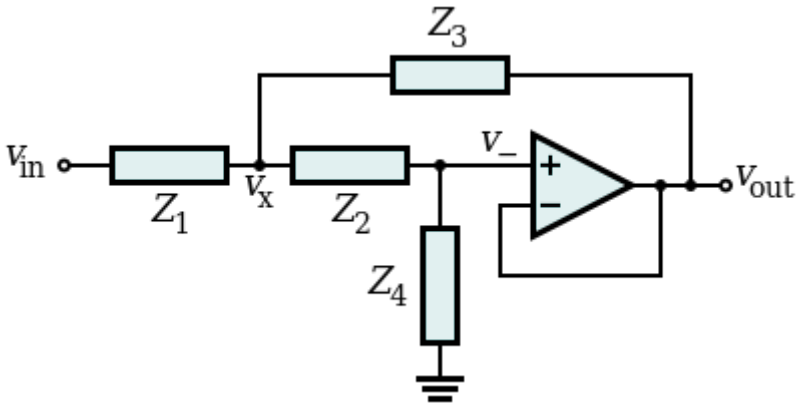

Figure 15 - Sallen key filter topology

Infinite-impulse, multiple feedback low pass filter

This delivers Q's in the range of 25 by using 1 amplifier. Although it is not as low as the Sallen-Key, the amplifier's GBW product ($20Q^2$ at resonance) is still relatively high than the gain ($-2Q^2$).

The gain and Q value and the Sallen key are inextricably related. It inverts the signal . The Multiple feedback architectures and Sallen Key are sensitive to variation in the external components.

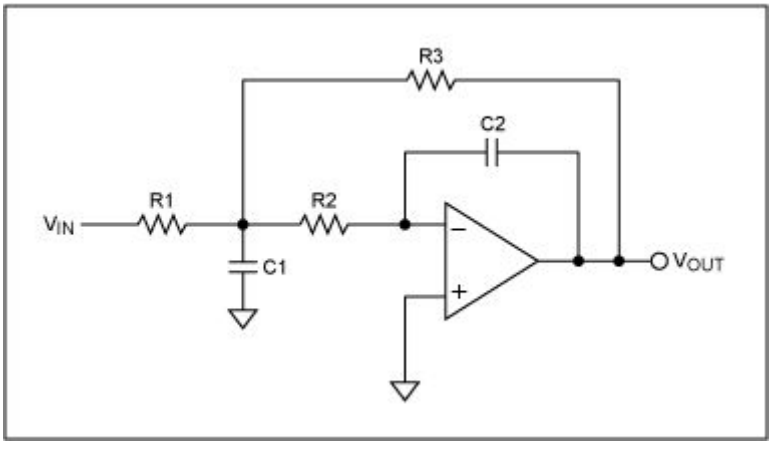

Figure 16 - Multiple feedback low pass filter topology

State Variable

Two integrators are preceded by a summing node. The fc, and the Q can be independently controlled and it also produces a high-pass, band-pass and low-pass o/p, making the architecture very versatile.

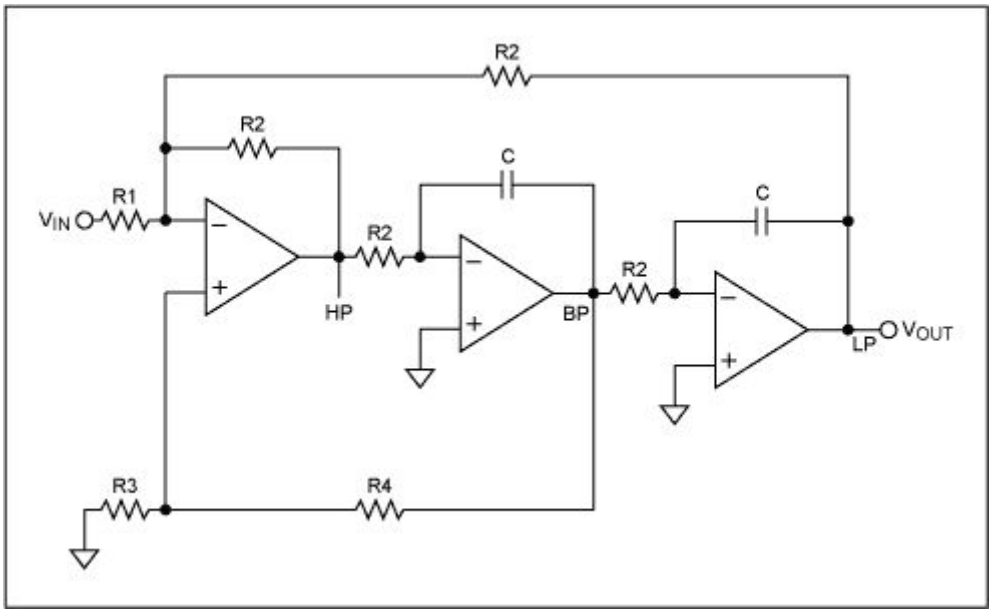

Figure 17 - State variable filter topology

The independent control of the Q and gain can be realized (**below**), by annexing a 4th amplifier. For high Q circuitry, state variable is ideal. With proper filter design, Q's of 500 or more are easily attainable. The filter's output gain (Q) is slightly lower than the open loop gain (3Q), and the low-pass gain is Q, unlike the single op amp architectures, and it causes a reduction in the op amps GBW requirements. Q of 500 would be 22.5MHz for a open loop gain of a minimum of $90Q^2$, for a Sleen Key .

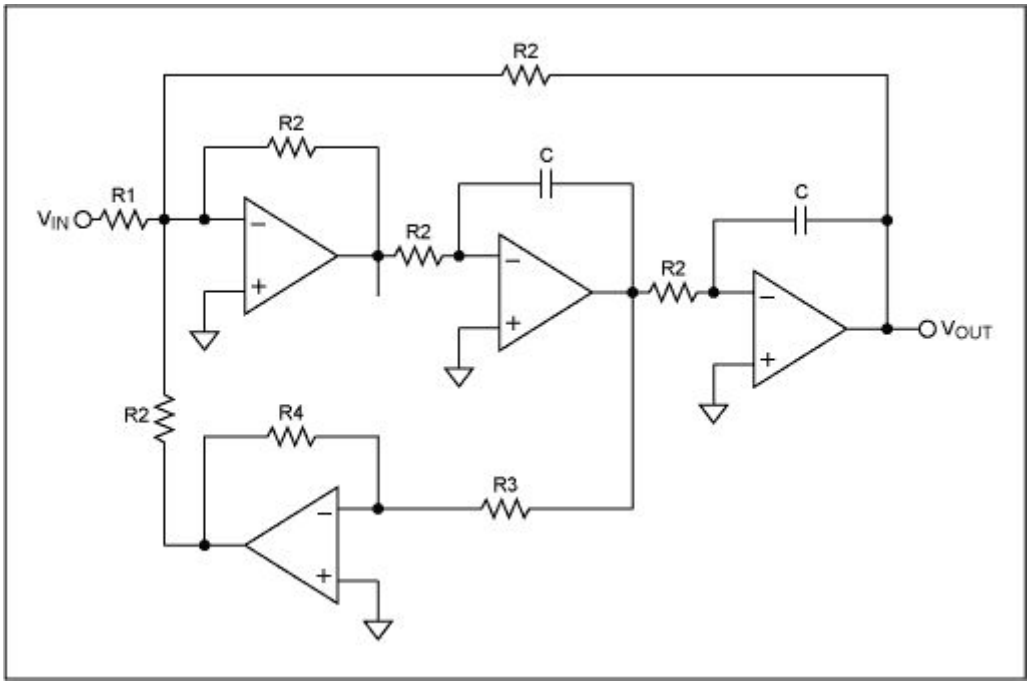

Figure 18 - State variable filter variation

Common to all topologies so far, the Q and percentage bandwidth remains constant as the frequency fc changes, and the start variable is the least sensitive to component variations. The Q value remains what is was as fc is shifted in the frequency domain, but with increase in fc and with decrease in fc the bandwidth of the filter decreases and increases respectively .

Drawback: Use of 3 or 4 amplifiers especially for power sensitive applications. The plethora of filter software and filter design cookbooks makes the design very straightforward. One must be very cautious with layout and component selection when working with high Q's as there exists a tendency to exhibit instability when working with high Q's, with little mismatched components and due to this instability there are chances that they might oscillate.

The Biquad Filter

Even though it exhibits similarity to the state-variable type, it comprises of an inverter preceded by an integrator and then followed by another integrator. The state variable filter behaves differently than this circuit due to this subtle change .

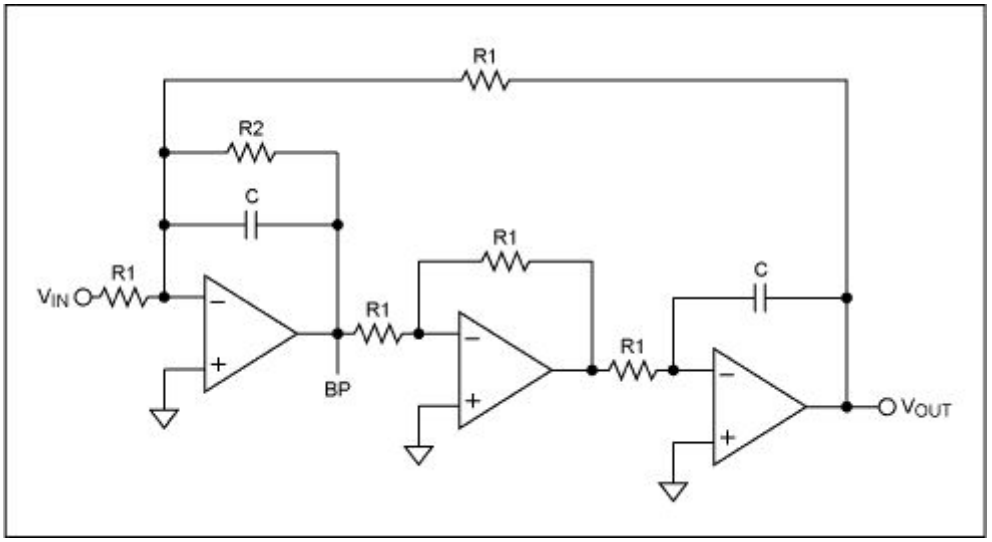

Figure 19 - Biquad filter topology

For a biquad, the Q value changes as fc changes, but the bandwidth stays constant. The Q value decreases as fc decreases, and as fc increases, the Q value increases. The biquad behaves like the state variable except for the above mentioned clause. It is less sensitive to external component variations, allows very high Q values, and a 3 or 4 amplifier configuration can be obtained. To obtain a steeper filter roll-off response when multiple stages are cascaded, the 3 and 4 amplifier circuits require more design time, and draw more power. Since a single amplifier is cheaper than a quad, it is more expensive. But a trade-off must be made, as it is much better in the performance point of view.

Variations

Butterworth

The Butterworth or maximally-flat response is the best-known filter approximation. It produces no ripple with a nearly flat passband. A low-pass or highpass rolloff rate of 20 dB/decade (6 dB/octave) for every pole is obtained with the rolloff smooth and monotonic. Hence, beyond the cutoff frequency for every factor of ten increase in frequency, an attenuation rate of 100 dB is witnessed typically for a 5th-order Butterworth low-pass. It has a reasonably good phase response.

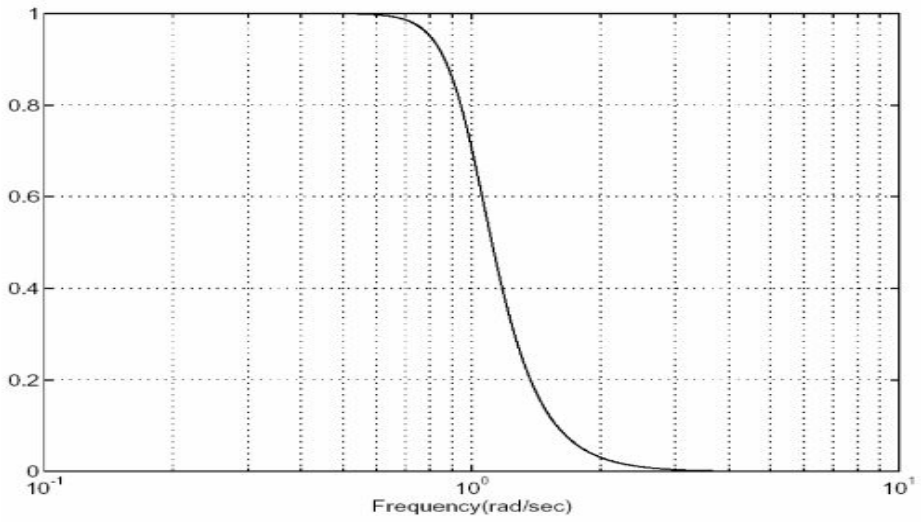

Figure 20 - Butterworth filter Bode plot

Chebyshev

By allowing ripple in the frequency response it acts as a clever strategy for achieving a faster roll-off. The roll-off becomes sharper (good) as the ripple increases (bad) . They have a poor phase response. An 8th order Chebyshev filter will be required against a 19th order Butterworth filter for a stopband attenuation of 20dB and a passband flatness within 0.1dB. More importantly when using a lower specification processor. Type 1 filters are Chebyshev filters where the ripple is only allowed in the passband . Although seldom used Type 2 filters are Chebyshev filters that have ripple only in the stopband.

The frequency response of a lowpass Chebyshev filter is shown in the following figure.

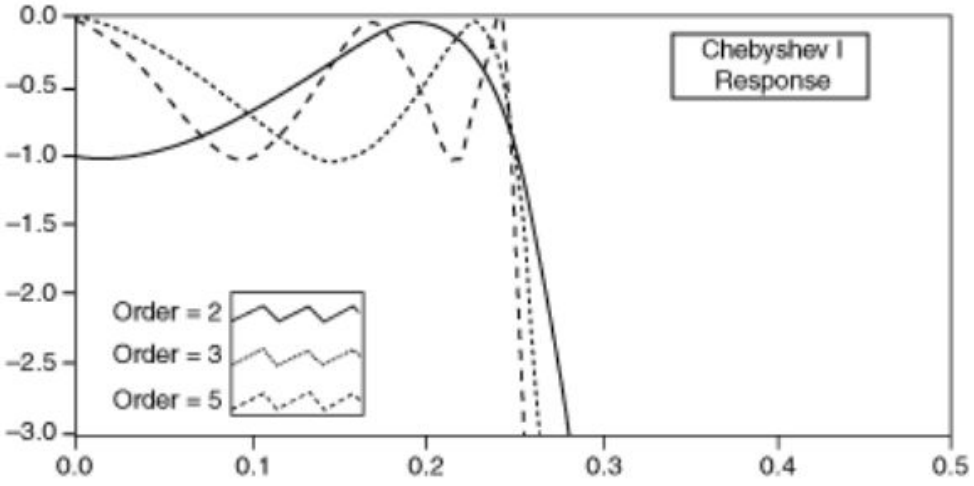

Figure 21 - Chebyshev type 1 magnitude plot

The Frequency response of a lowpass Chebyshev II filter is shown in the following figure.

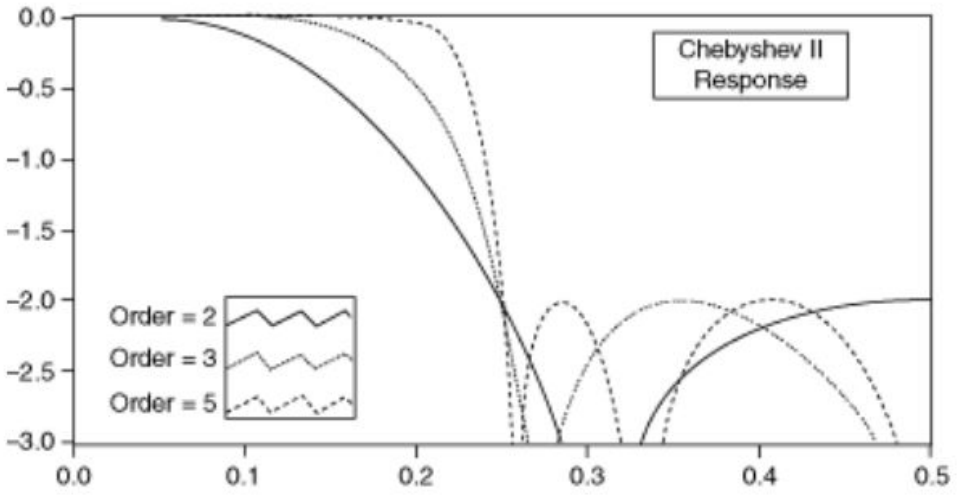

Figure 22 - Chebyshev type 2 magnitude plot

Both the Chebyshev filters have the same advantage over Butterworth filters with a lower order filter, which is a smaller absolute error and faster execution speed due to a sharper transition between the passband and the stopband.

Bessel

• Maximally flat response in both magnitude and phase

• Nearly linear-phase response in the passband

Nonlinear-phase distortion which is very inherent in all IIR filters, is reduced by Bessel filters. Especially in the transition parts of the filters, a nonlinear-phase distortion, is seen in high-order IIR filters and IIR filters with a steep roll-off. With FIR filters, Linear-phase response can also be obtained.

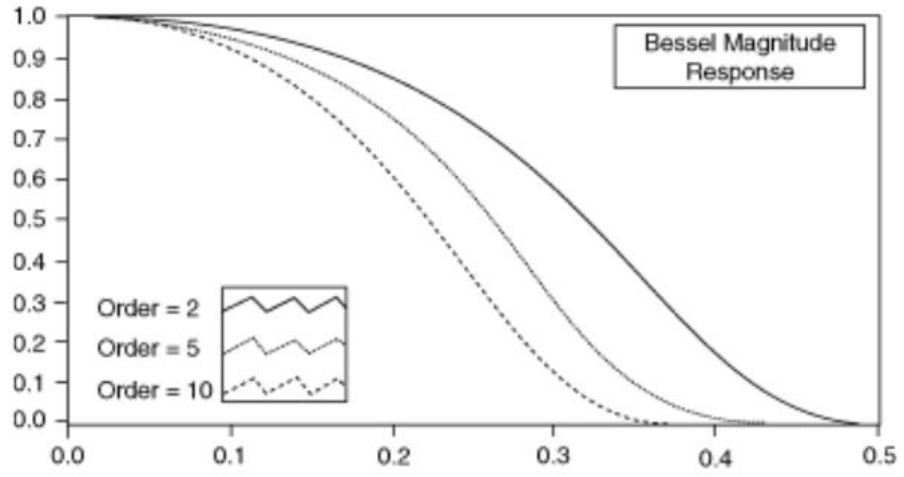

Figure 23 - Bessel magnitude response

Elliptic

The phase response is very nonlinear and both the passband and the stopband possess ripples in the amplitude response, although the cut-off slope of an elliptic filter is steeper than that of a Butterworth, Chebyshev, or Bessel. The elliptic response will reject frequencies outside a certain

frequency band and pass frequencies falling within that band, with the lowest-order filter, regardless of phase shifts or ringing .

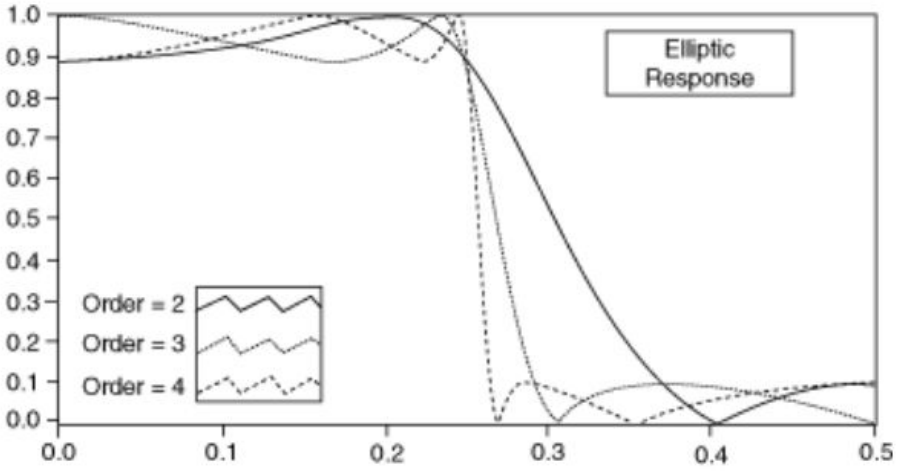

Figure 24 - Elliptic filter magnitude response

The elliptic filters provide the sharpest transition between the passband and the stopband, when compared with the same order Butterworth or Chebyshev filters.

Input Conditioning

1) **Topology:** Sallen Key. Because it is active, makes use of only one opamp, making it relatively inexpensive and enhancing its simplicity.

2) **Filter** : Type 1, 4th Order Chebyshev . Without compromising too much on the frequency response by allowing ripples in the pass band, an optimal roll off rate can be achieved. An optimal trade off between these two parameters in the response can be obtained by a Chebyshev . 4th order is optimal as increasing the order further will cause more ripples in its frequency response (bad), and decreasing the order will affect its roll off rate in an adverse way.

A Chebyshev filter with a lower order filter can achieve a sharper transition between the passband and the stopband when compared to a Butterworth filter. A Chebyshev filter exhibits faster execution speeds and smaller absolute errors than a Butterworth filter owing to its sharp transition between the passband and the stopband. Chebyshev has a decent roll-off rate and the lowest amount of ripples in its pass band on comparison to Elliptic and Bessel .

Output Conditioning

Passive RC FIlter: To produce a true DAC, a PWM output has to be converted to an analog voltage level, which can be accomplished using a simple low-pass filter built using a resistor and a ceramic capacitor.

Using a simple low-pass filter, a PWM signal can be converted into a fairly stable voltage ranging from ground to logic high (e.g., 3.3 V). By using firmware and hardware to vary the PWM duty cycle according to the following relationship, we can achieve a DAC implementation:

$$\text{desired DAC voltage} = A \times \text{duty cycle}.$$

Where A => The voltage for logic high.

Embedded C Code

Overview of Functionality-

- Pushbutton S1, Play: Begins playback of any recorded audio. If the play button is held on beyond the end of the recording, playback will cease and a new playback cycle will not be initiated until the play button is released and then pressed again. Record functionality will be disabled during playback. LED1 (green) will be on while playing.
- Pushbutton S2, Record: Begins recording. Recording will continue until the stop button is pressed, or the maximum record time is reached. If the record button is held on beyond the maximum recording time, recording will cease and a new record cycle will not be initiated until the record button is released and then pressed again. Playback functionality will be disabled while recording. LED2 (red) will be on while recording.
- Pushbutton S3, Stop: Immediately ceases any playback or recording cycle. LED3 (blue) will be on while stopped.

Code Analysis -

The microcontroller's inbuilt ADC uses Timer0 for its operation and selects Timer0 CMPA as a trigger.

Initially, when the program starts, it disables interrupts, configures clocks, configures PLL (used by Timer4 and USB serial), initialises USB serial interface (for debug purposes), initialises timer (used by FatFs library), initialises circular buffer, initialises ADC, enables interrupts, sets pushbuttons 1 to 4 PORTF 6-4 as inputs, PORTD 7-4 as outputs (LEDs) and PORTB 6 as an output (JOUT) . After all the after interrupts are enabled, WAVE file interface is initialised.

When the Record button is pushed, the state machine's current state is changed to recording mode after resetting the buffer, setting the maximum record time, clearing 2 flags, creating new WAVE file with read/write access (forces overwrite if file exists) and starting the ADC sampling along with performing the necessary LED operation.

The microcontroller's inbuilt ADC uses Timer0 for its operation and select Timer0 CMPA as a trigger. When it starts its ADC process, the clock is set to 250 kHz clock and interrupts enabled. The ADC Interrupt service routine is executed on completion of ADC conversion and the result is passed onto the buffer_queue function which stores the result onto the buffer (2 page buffer of 512 8-bit samples capacity each). It writes a page of the buffer of audio samples into the open WAVE file when a buffer page is full. When the stop button is pressed or when the last page has been recorded, Stop is flagged. When the stop is flagged, writes final page, finalises WAVE file, stop ADC

sampling, prints status to console, and the state returns to stopped mode.

When the Playback button is pushed, buffer resets, number of pages to be played is initialized to the number of pages recorded, opens an existing WAVE file with read only access, reads 1024 8-bit audio samples from the open WAVE file, initializes 4 flags, initializes and enables the debounce function and starts the PWM process.

When the PWM process starts, system clock is configured for 16 MHz, initializes ports for use and initialises USB serial interface. Timer 4 is initialized for the 8-bit PWM, with OCR4B connected to B6 (JOUT) and PWM frequency set at 15.625 kHz, top is set to 0xFF (255), initialized to DC duty cycle of 50%, Timer/Counter4 Overflow Interrupt is enabled and the timer is reset. When the timer count hits the TOP, the overflow interrupt is called, it is reset and the count starts again. Every time the timer count hits the top twice in a row, that is at a frequency of 31.25 kHz, it removes and returns a sample from the head of the queue (buffer) which is then diverted to OCR4B connected to JOUT. Every time a page has been emptied, a new page is ready to read to buffer. Every time a buffer page is full, it allows the code to write a full page to the buffer. When the last page is played or the stop button is pressed, it finalizes and closes an open WAVE file, stop the PWM sampling and the state undergoes a transition to stopped state. The timer 4 is deactivated to disable the PWM.

An additional feature is included in the code design that facilitates the 4th pushbutton to act as a fast-forward button. When the pushbutton is pressed, the output frequency changes from 31.25 kHz to 15.625 kHz. To return to the original frequency of 31.25 kHz, the pushbutton has to be pushed again. The fast-forward rate is 2x. The pushbutton also has a debouncing feature that avoids glitches in pressing, especially button bouncing. Finds transitions of PB4 from 0->1 and if last 3 PB readings the same, it considers this stable.

LMC6484

The Texas Instruments LMC6484 is a quad rail-to-rail input and output operational amplifier (op-amp). All four op-amps of the LMC6484AI variation were used for construction of the input conditioning, including amplification and filtering. To ensure the device operates in the desired manner, it is imperative to take into account the recommended operating specifications.

Recommended Operating Conditions and Characteristics

	Min	Max	Unit
Supply Voltage, V+	3	15.5	V
Junction temperature, T_j	-40	85	°C

Power for the device is sourced directly from the TeensyBOB's 5 volt output, which is comfortably within the recommended operating conditions.

Assuming an operating voltage of 5V and a T_j of 25°C, the device operates with a typical input current around 0.02 pA. This low operating current that the device is expected to run at also results in a very low thermal design power (TDP) that must be dissipated, negating the need for any passive or active cooling of the chip.

The device is divided into four separate op-amps, each with individual outputs, inverting inputs, and

non inverting inputs.

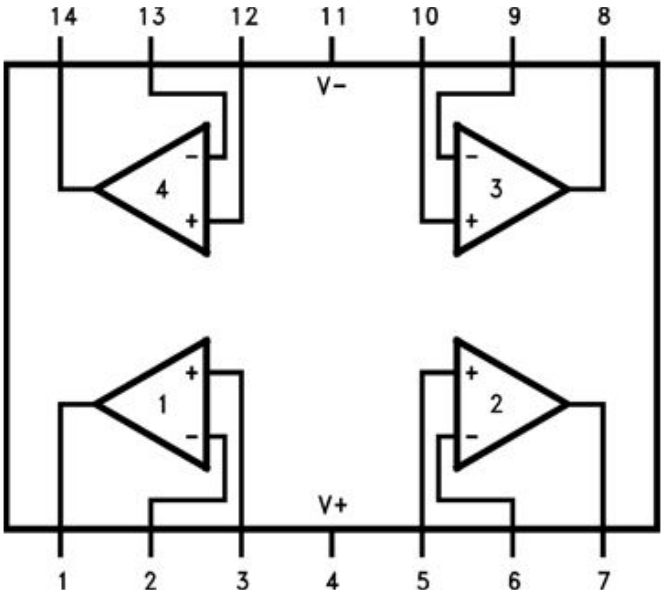

Figure 25 - LMC6484 Pinout top view

As stated, all four op-amps will be used for conditioning the input signal, with gates 1 and 2 used for amplification, while 3 and 4 were reserved for each of our individual second order filters.

Amplifier

For a low noise amplifier, the desired input impedance is as low as possible. Thus, the use of inverting amplifiers are optimal, as due to their feedback resistors, they have less input impedance than the non-inverting alternative. A schematic for a basic inverting amplifier is shown below, where gain is given by: $\frac{-R2}{R1}$

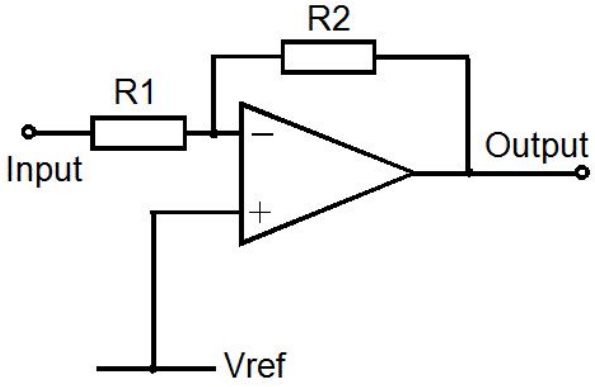

Figure 26 - Inverting amplifier topology

The non-inverting input is connected to Vref, which to achieve the desired offset, will be connected to 2.5V. The maximum peak to peak voltage for the LMC6484 is limited to that of the supply voltage, which in this use case is 5V.

LM386

The Texas Instruments LM386 is a low voltage audio power amplifier, designed for use in low voltage consumer applications. Capable of driving loads from $4\,\Omega$ to $32\,\Omega$, the gain is internally set to 20, but is adjustable by modifying the component values in the surrounding circuitry.

Recommended Operating Conditions and Characteristics

	Min	Max	Unit
Supply Voltage, V_{CC}	4	12	V
Junction temperature, T_A	0	70	°C

The device will also be powered by the 5V supply from the TeensyBOB, inside the reccomended supply voltage range.

When operating at 5V, it has an expected current draw of approximately 3.7mA.

Gain and Volume

While increasing the volume and gain have a similar effect in that they both increase the intensity of the sound produced by the speaker, both perform separate roles in terms of the performance of the LM386. Gain is a specific characteristic of the amplifier, and refers to the amplification of the input, whereas volume allows for adjustments to the sound within that specified range of amplification.

Gain is specifically controlled through pins 1 and 8, capable of reaching a theoretical high of 200.

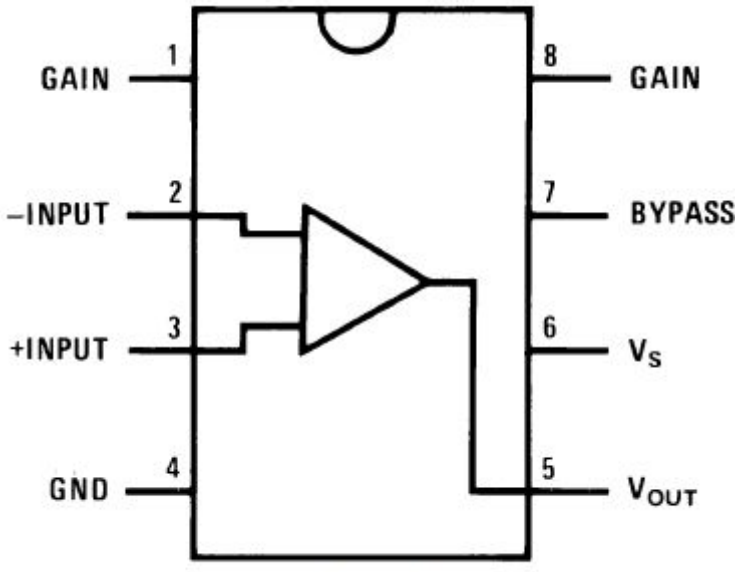

Figure 27 - LM386 Pinout top view

Provided in the data sheet is a schematic for operating the LM386 with a maximum gain of 200.

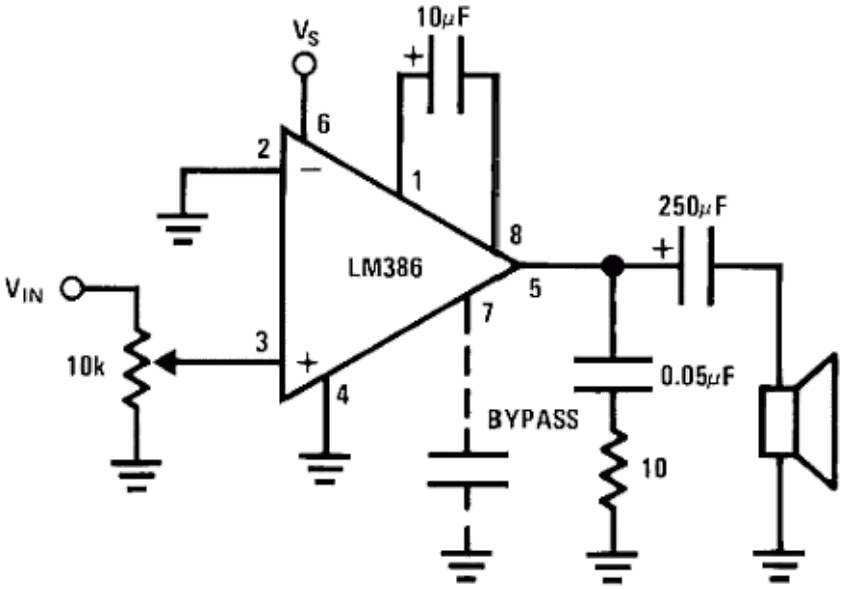

Figure 28 - Schematic for LM386 with 200 gain

Other than component values, the one key difference that must be considered when using the LM386 with higher gains, is that to prevent instability or reduction of gain, the bypass pin 7 must be connected to ground, using either a 0.1µF capacitor or a simple short circuit.

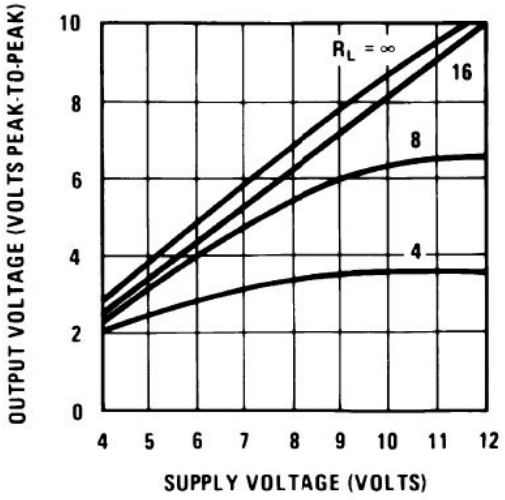

Figure 29 - Output Voltage vs Supply Voltage

Due to the use of a 5V supply voltage, the maximum output voltage is limited to approximately 3V peak to peak, despite the theoretical 200 gain.

Resistors

A resistor, measured in ohms, is a passive electrical component and used in almost all circuits.

Figure 30 - Resistor Symbol

Larger resistors often have higher tolerances, with lower variance between components, thus, their use is preferable in most circumstances to provide accurate and reliable performance.

Capacitors

A capacitor, whose capacitance is measured in Farads, is a passive electrical component that stores electrical energy.

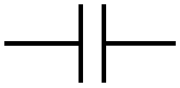

Figure 31 - Capacitor Symbol

Coupling Capacitors

A coupling capacitor is used to remove DC offset, while allowing the AC component to pass.

Decoupling Capacitors

A decoupling or bypass capacitor is used to reduce noise caused by other circuit elements. By shorting the AC signal to ground, a cleaner DC signal can be achieved.

Implementation

Microphone

The measurement circuit provided in the microphone is appropriate for use in the final DVR circuit. This circuit uses a 1.5kΩ resistor and a 1uF capacitor.

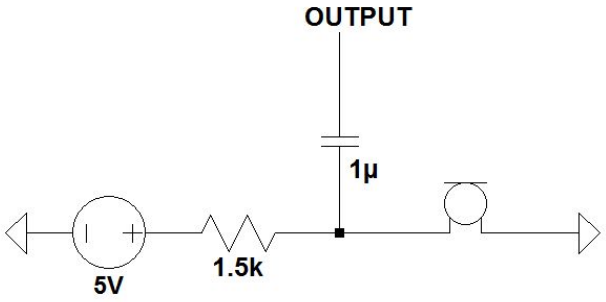

Figure 32 - Microphone schematic

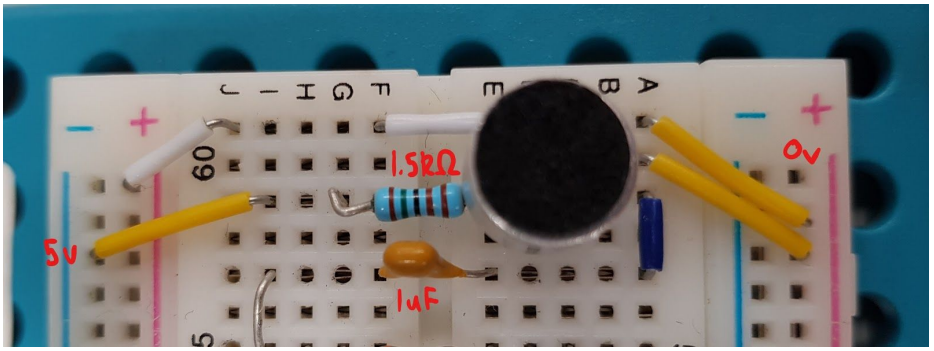

Figure 33 - Microphone design on breadboard

The circuit was tested and the microphones performance evaluated with an oscilloscope. A loud click produced a waveform of peak to peak amplitude approximately 120mV. This performance is as expected, and the portion of the circuit deemed functional and appropriate.

Voltage divider

The TeensyBOBV2 provides 5 volts to the circuit, suitable for driving both the LMC6484 and LM368. However, as the input into the ADC on the ATmega32U4 is required to have an offset of 2.5V, a 2.5V reference voltage is required as reference for parts of the input circuit conditioning. Using two equal sized resistors, a voltage divider could be constructed with an output of 2.5V. A decoupling capacitor was also deemed necessary to minimize voltage ripple and noise.

The circuit was designed in the following manner.

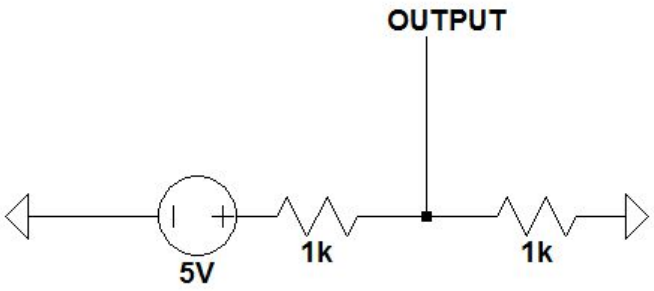

Figure 34 - Voltage divider schematic

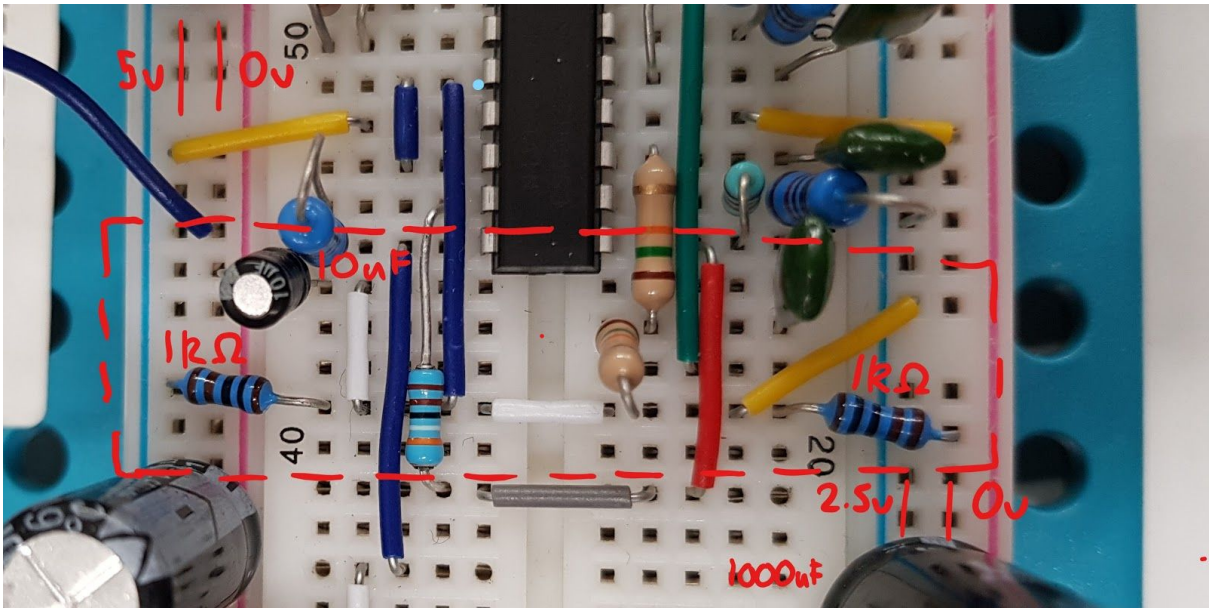

Figure 35 - Voltage Divider constructed on breadboard

It was found in testing that the single 10uF was not sufficient for removing the entirety of the interference noise, thus, an additional 1000uF capacitor was added in the final design.

Input amplification

Amplification is a crucial component in the input conditioning stages of the circuit. While increased the volume is obviously a key desire for a digital voice recorder, because of the voltage limitations implied by the LMC6484, the maximum voltage amplitude is restricted to 5V peak to peak, and so to avoid clipping a balance must be found.

To optimize the signal to noise ratio, two dedicated amps will be used, with additional gain also provided in the filter stage. This allows for the signal to be amplified before the filter, increasing the amount of unwanted noise that will be removed from the signal and then again after the filter when the noise is removed. When using multiple amplifiers in series, the overall gain is the product of their individual gains.

Both amplifiers are based on the LMC6484 op-amp, in an inverting configuration to reduce impedance and subsequent noise.

It was determined that a loud click right in front of the microphone generated a waveform with peak

to peak amplitude of approximately 100mV. It can be assumed that the amplitude of normal speaking volume, which the DVR is designed for will be significantly lower, thus, for testing purposes, a signal with amplitude 5mV was used.

Stage 1

The first amp is responsible for the majority of the gain in the input conditioning, however, ensuring its exact gain is not necessary, as any discrepancies can be rectified in the final gain stage. Using a 3.8kΩ and a 120kΩ resistor, a theoretical gain of 32 can be achieved which would result in an amplitude of approximately 160 mV.

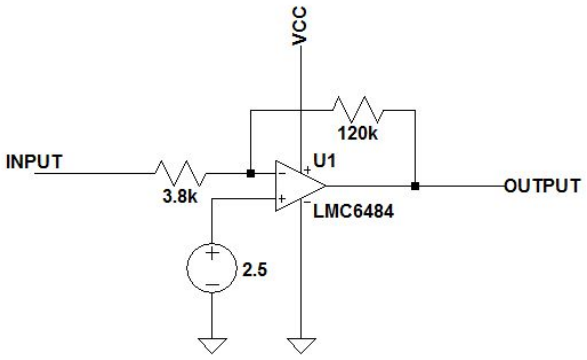

Figure 36 - Stage 1 amp schematic

This stage produces an unexpected gain almost double that expected, approximately 64. While this gain is unexpected, it is still suitable for our needs, and can be accounted for in the final stage of amplification.

Stage 2 (Anti-aliasing filter)

The gain provided by the filter is simply a byproduct of the Sallen Key topology. Unlike Butterworth designs where the gain component can be calculated and changed independently to the cut off frequency, the gain is incorporated into the key design calculations of the Chebyshev.

The design used in the DVR results in a gain of approximately 4.7, which is replicated almost perfectly in the practical circuit.

Stage 3

The third and final amp is tuned specifically to bring the amplitude up close to the 5V peak to peak, before it moves into the ADC in the ATmega. Because the total gain is the product of the individual amps, this third stage is only required to add minimal gain to the system to reach the desired output amplitude. When testing with a 5mV signal, the signal leaving the filter has an amplitude of approximately 1.5V. To bring this up close to the desired 5v, a 3.8kΩ and a 10kΩ are used for a theoretical gain of 2.6, resulting in a final amplitude of 3.9V, appropriate for our needs while minimising any risks of the signal peaking.

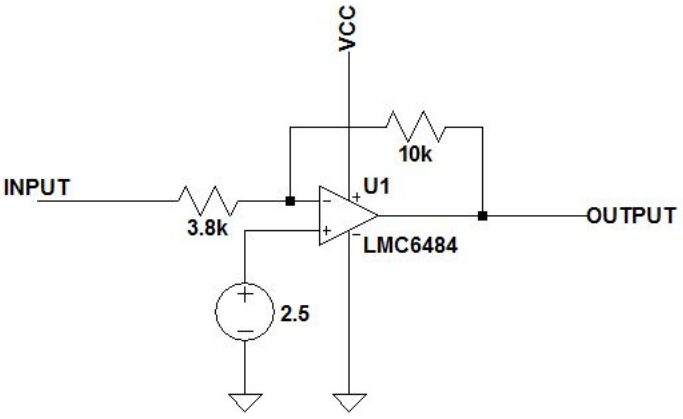

Figure 37 - Stage 2 amp schematic

In practice, the amp produces a gain of approximately 2.5, just shy of the theoretical 2.6.

Total input amplification

The total gain over the input stage is the product of each individual gain, which is calculated to be approximately 752. This results with an input with amplitude 0.005mV to an output peak to peak voltage of approximately 3.8V.

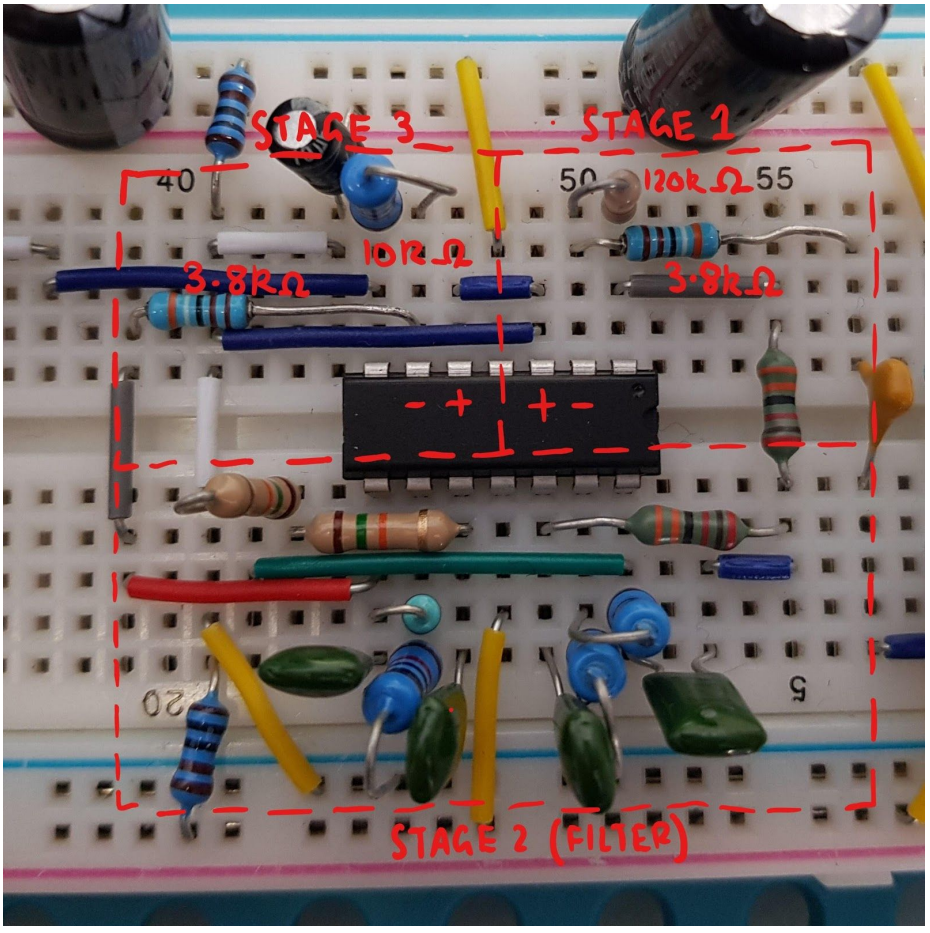

Figure 38 - All three amplification stages constructed on breadboard

Input filtering

As stated in the literature review above, the sampling frequency must be at least twice the highest frequency of the analog signal. The sampling frequency is set at 15.625 kHz by the PMW waveform from the Jout header on the development board. From the sampling frequency of 15.625 kHz, the highest frequency for the analog signal would be 15.625 divided 2 which equals 7.8 kHz.

This 7.8 kHz is used to be the cutoff frequency for the low pass filter to limit the analog signal. This low pass filter acts as an anti-aliasing filter. An ideal filter, cuts off every thing after the cutoff frequency as seen in the below figure.

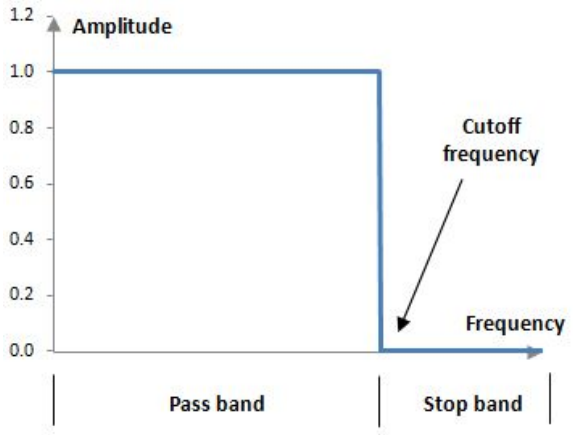

Figure 39 - Ideal filter frequency response

Since an ideal filter can not be implemented in the real world, a more realistic filter had to be calculated in order to filter out the 15.625 kHz. A lower cut-off frequency had to be used instead of 7.625 kHz to account for the roll off rate of the 4th order chebyshev that was decided to be used. Also, the frequency of talking is between 85Hz and 255Hz, so incorporating a lower cut-off frequency won't cut-off the analog signals generated by talking into the microphone.

A cut-off frequency of 2.5 kHz was decided upon for the low pass 4th order chebyshev. Between a filter with a cut-off frequency of 7 - 8 kHz and a filter with 2.5 kHz, there was no difference on overall function of the project. A filter with 2.5 kHz cut-off frequency provided better noise reduction from the microphone. The roll off rate off a 4th order chebyshev is -80 dB/decade. With this roll off rate, the low pass attenuates higher frequencies more steeply compared to a 2nd order low pass, which only has a roll-off rate of -40 dB/decade. The 4th order filter reduces the signal amplitude to one sixteenth of its original level every time the frequency double. So, power decreases by 24 dB/octave or 80 dB/decade. This causes the sampling frequency of 15.625 kHz to be at a small amplitude that doesn't have an impact on the recording, making no background noises while recording.

Matlab Calculations

Using Matlab to calculate the transfer function for a 4th order Chebyshev low pass filter.

```
fp = 2.5e3;      % passband freq (Hz)
Amax = 1;        % passband ripple (dB)
n = 4;           % filter order

[b a] = cheby1(n,Amax,2*pi*fp,'low','s');
t = tf(b, a);

t =

                          1.496e16
  ------------------------------------------------------------
  s^4 + 1.497e04 s^3 + 3.587e08 s^2 + 2.878e12 s + 1.678e16

Continuous-time transfer function.
```

This transfer function is used to find the capacitor and resistor values that are need to produce a cut-off frequency. To do so, the transfer function has to be split up for the first 2nd order filter and the second 2nd order filter. This is done by finding the poles using the following script.

[z p k] = cheby1(n,Amax,2*pi*fp,'low','s');

```
>> p

p =

      -5291.5 +     6398.3i
      -5291.5 -     6398.3i
      -2191.8 +      15447i
      -2191.8 -      15447i

>> k

k =

   1.4956e+16
```

$$H(S) = \frac{1.4956\times10^{16}}{(s+5291.5s-j6398.3)\,(s+5291.5s+j6398.3)\,(s+2191.8-j15447)\,(s+2191.8+j15447)}$$

To get the transfer function to be 2 separate transfer functions for each filter stage, multiply out the

complex conjugate poles. This is done by the following script.

```
a1A = 2*(-real(p(1)));
a2A = real(p(1))*real(p(1)) + imag(p(1))*imag(p(1));

a1B = 2*(-real(p(3)));
a2B = real(p(3))*real(p(3)) + imag(p(3))*imag(p(3));
```

The two transfer functions are in the form:

$$H(s) = 0.89125 \times \frac{a_{2A}}{\left(s^2 + a_{1A}s + a_{2A}\right)} \times \frac{a_{2B}}{\left(s^2 + a_{1B}s + a_{2B}\right)}$$

Substituting values into a_{1A} , a_{2A}, a_{1B}, a_{2B} gives:

$$H(S) = 0.89125 \times \frac{6.8939\times10^{7}}{(s^{2}+10583s+6.8939\times10^{7})} \times \frac{2.4341\times10^{8}}{(s^{2}+4383.7s+2.4341\times10^{8})}$$

Using this transfer function it can be broken down into two.

$$H_1(S) = \frac{6.8939\times10^{7}}{(s^{2}+10583s+6.8939\times10^{7})} \quad \text{and} \quad H_2(S) = \frac{2.4341\times10^{8}}{(s^{2}+4383.7s+2.4341\times10^{8})}$$

Each filter will be in the following configuration.

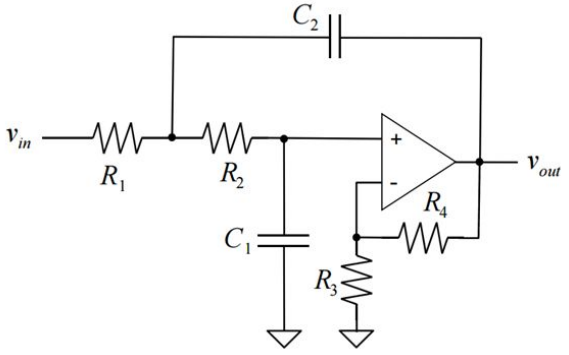

Figure 40 - Sallen key topology

From those two transfer function above, values for each capacitor and resistor can be found. To make it simpler, R_1 & R_2 , C_1 & C_2 will be equal to each other on both stages. Each of the transfer

functions are in the form of the following equation.

$$\frac{v_{out}}{v_{in}} = \frac{\frac{K}{(RC)^2}}{s^2 + s\left(\frac{3-K}{RC}\right) + \frac{1}{(RC)^2}}; \qquad \left(K = 1 + \frac{R_4}{R_3}\right)$$

For a rough estimate of C:

$C = \frac{10}{f_p}\,\mu F$ f_p = 2.5kHz

$C = \frac{10}{2500}\mu F = 0.004\mu F = 4nF$

Using the above equation for the transfer function, the value for R can be found. Since $\frac{v_{out}}{v_{in}}$ is the same as H(s), values from $H_1(s)$ & $H_2(s)$ are equal to the respective constants from $\frac{v_{out}}{v_{in}}$.
So for stage A $\frac{1}{(R_A C)^2} = 6.8939\times10^{7}$ and stage B $\frac{1}{(R_B C)^2} = 2.4341\times10^{8}$, where C = 4nF.

$R_A = 30.1\ k\Omega\ and\ R_B = 16\ k\Omega$.

To find the remaining resistors, which are associated with the gain of the filter, the gain constant needs to be calculated. Using the standard form of a transfer function the gain for stage A is found to be K_A = 1.725 and K_B = 2.719 for stage B.

Stage A gain equals 1.725, using $K = \frac{R_4}{R_3}$ and choosing R_3 = 10kΩ to make it simple, R_4 = 7.25kΩ
Stage B gain equals 2.719, using $K = \frac{R_4}{R_3}$ and choosing R_3 = 10kΩ to make it simple, R_4 = 17.2kΩ

From those calculations:

Stage A - R_{1A} = 30.1 kΩ
R_{2A} = 30.1 kΩ
R_{3A} = 10 kΩ
R_{4A} = 7.25 kΩ
C_{1A} = 4nF
C_{1B} = 4nF

Stage B - R_{1B} = 16 kΩ
R_{2B} = 16 kΩ
R_{3B} = 10 kΩ
R_{4B} = 17.2 kΩ
C_{1B} = 4nF
C_{2B} = 4nF

Display the bode plot.

```
figure (1);
h = bodeplot(t);                    %display bode plot
setoptions(h,'FreqUnits','Hz');
grid on;
```

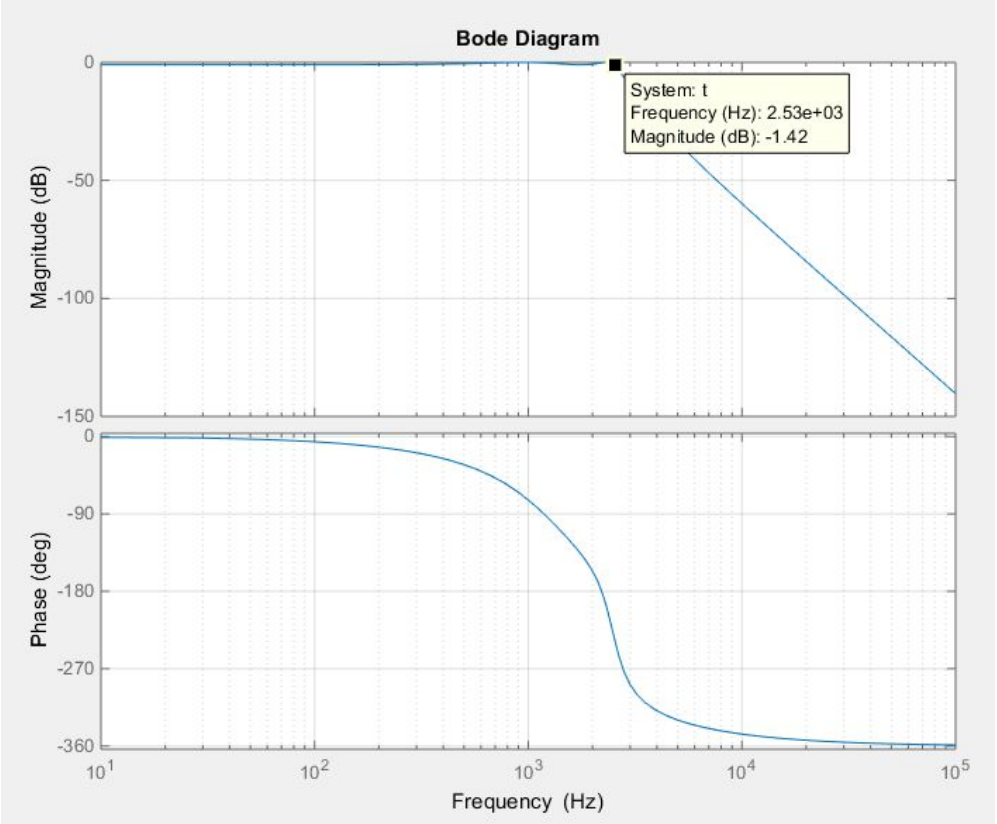

Figure 41 - Frequency response of the anti-aliasing filter

As seen in the bode plot, the -1 dB point is at 2.5 kHz.

Simulation

Using the values calculated above and simulating them into LTspice.

Please refer to Appendix 1 and 2

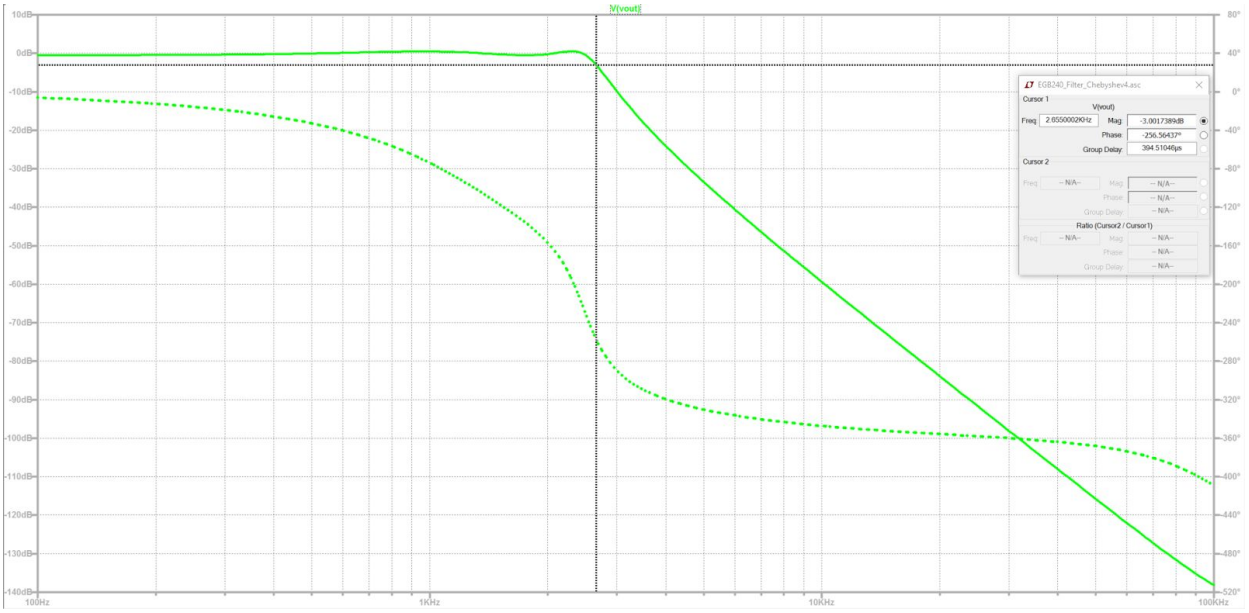

Figure 42 - Simulated filter design in LTspice

Final design and implementation

The final design of the 4th order low pass chebyshev filter to into account the real values of capacitors and resistors there was. Which meant changing the 30.1 kΩ to 33 kΩ, 4nF to 3.9nF, 7.25 kΩ to 6.8kΩ, 16 kΩ to 15kΩ and 17.2 kΩ to 18 kΩ. This had an effect on the overall outcome of the filter.

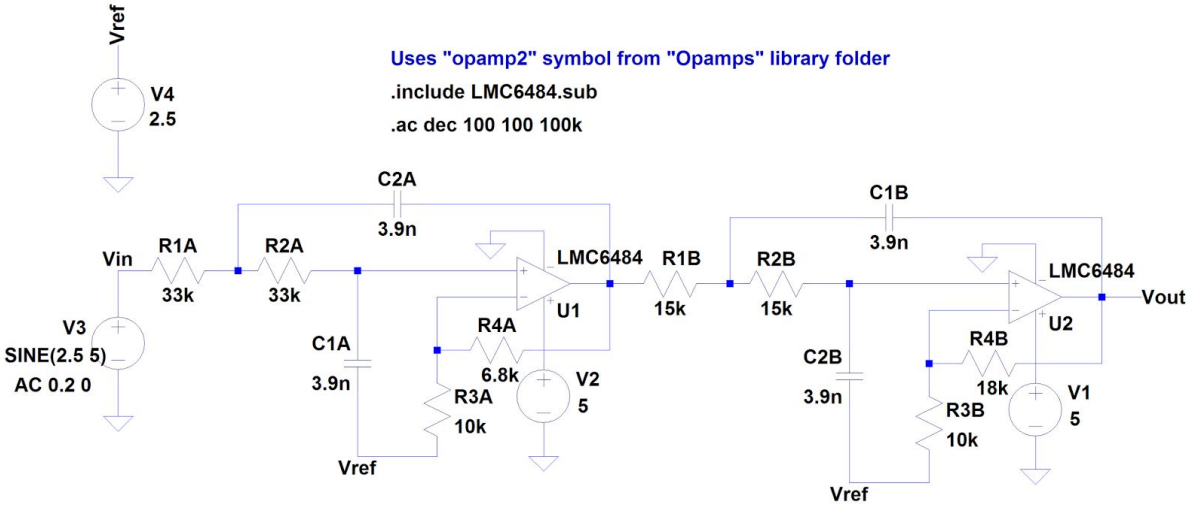

Figure 43 - Antialiasing filter in LTspice

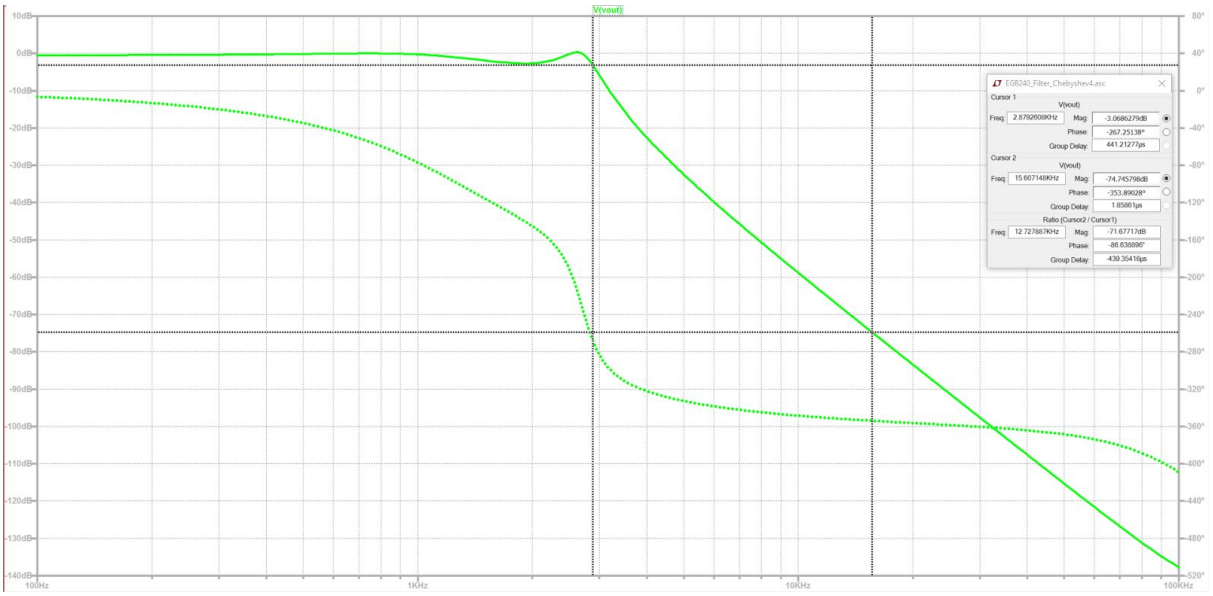

Figure 44 - Practical filter application in LTspice

Comparing the simulation circuit to the circuit with roughly correct values, (only roughly correct due to unincorporated tolerances of capacitors and resistors) it can be seen that the cut-off frequency is closer to 3 kHz. This increase would be due to the significant increase from 30 kΩ to 33 kΩ, there would be a decrease in stage B due to the fact that smaller resistors were used than calculated.

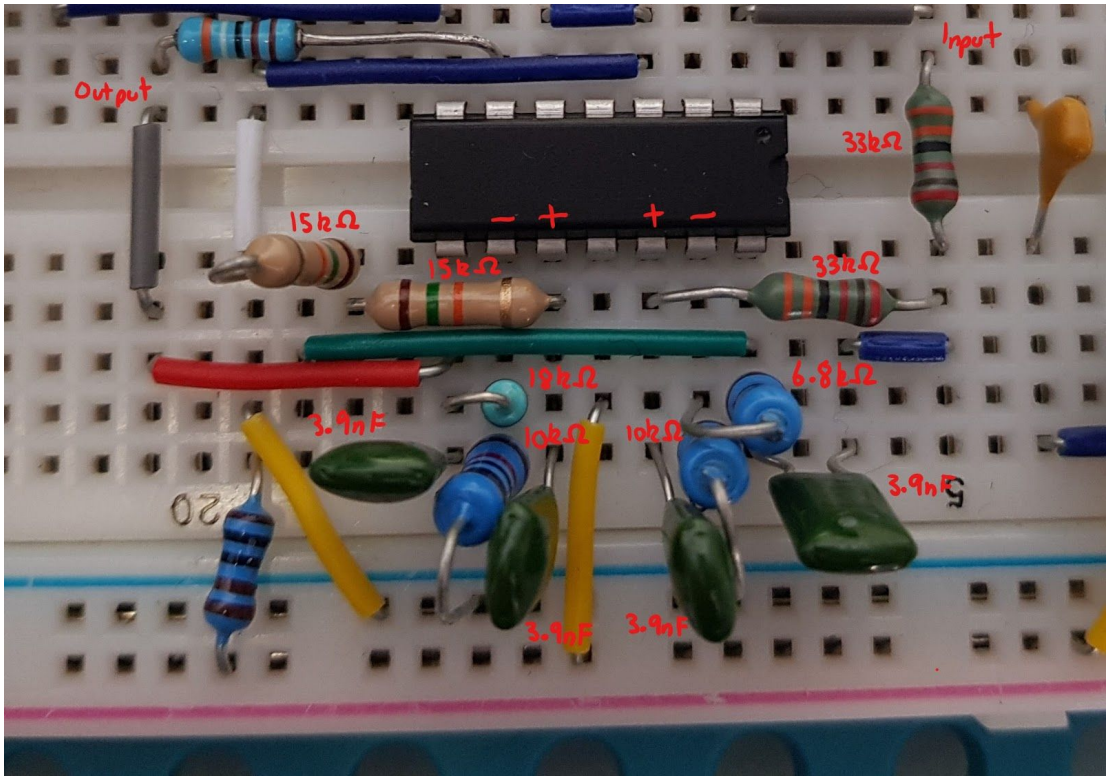

Figure 45 - Filter constructed on breadboard

Output filtering

From the output of the PWM wave being a digital wave, a reconstruction filter needs to be implemented to convert the PWM back to a continuous time wave so that the input wave can be heard through the loudspeaker. This reconstruction filter just needs to be a simple low pass RC filter to work as intended. The cut-off frequency needs to coincide with the sampling theorem. From the sampling theorem, the sampling frequency f_s must be greater than twice the maximum frequency present in the signal. From this specification, the reconstruction filter must have a cut-off frequency of 7 -8 kHz.

With the implementation of this filter into the output stage of the project, it was found that there was a tiny bit of noise that wasn't being attenuated from the low pass RC circuit. A lower cut-off frequency was used instead. At 5 kHz, the reconstruction filter made desirable effect to the output conditioning of the circuit.

Output amplification

An LM386 audio power amplifier was used to increase output volume of the playback. While the schematic from the data sheet was configured to have a gain of 200, several alterations were made to optimise the performance of the circuit.

A potentiometer in series with the capacitor between pins 1 and 8 allows for the gain to be finely controlled, with which a balance can be found between audio fidelity and volume. Due to the large gain, it was found appropriate to use a bypass capacitor between pin 7 to ground to reduce distortion and noise.

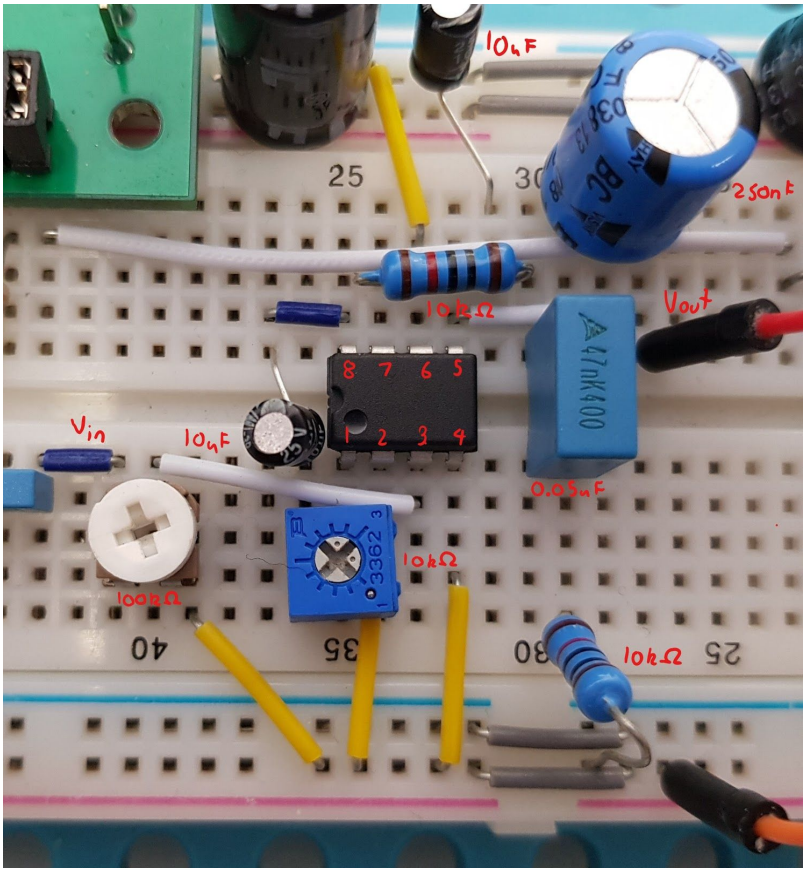

Figure 46 - Audio amp constructed on breadboard

At gain of 200, while the the output is loud, it also exhibits some distortion indicating some clipping due to the 5v maximum amplitude set by the input voltage. By tuning the gain however, a balance can be found between volume and clarity. If a higher voltage could be supplied to the LM386, a far louder output would be achieved.

Experimental Results

Amplification

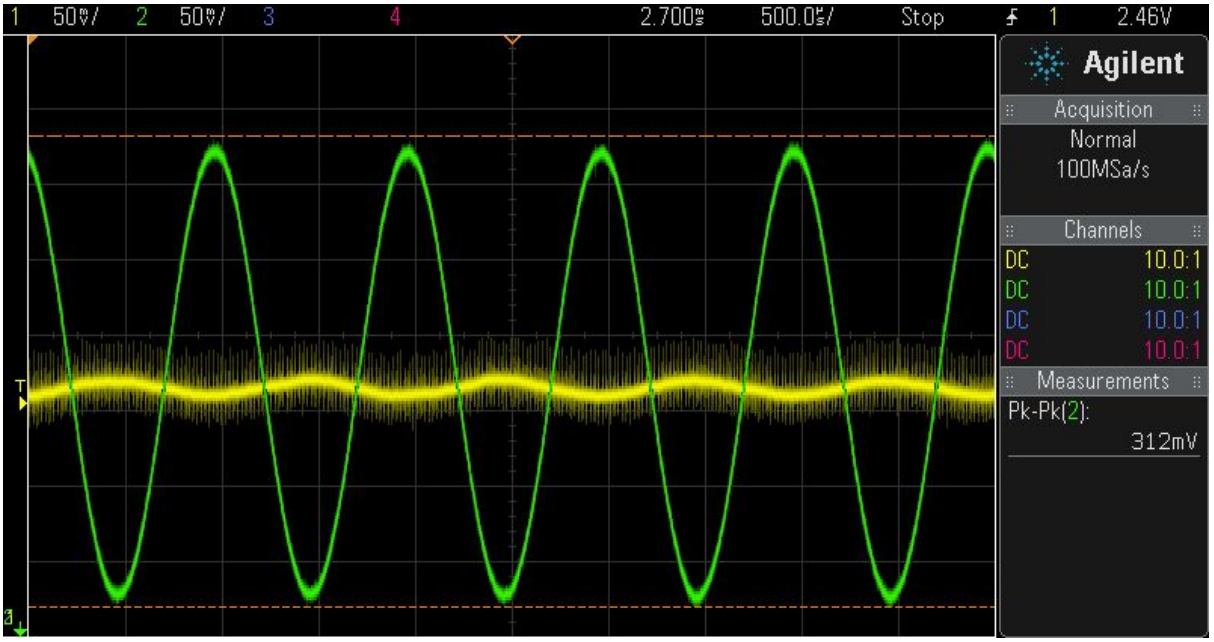

Figure 47 - Stage 1 amplification with 5mV input

Figure 48 clearly shows the gain present in the first amplification stage, where the amplitude of the original signal is 5mV. It makes clear the inverting nature of the amp.

Figure 48 - Stage 2 amplification

Figure 49 clearly shows the gain present in the second amplification stage.

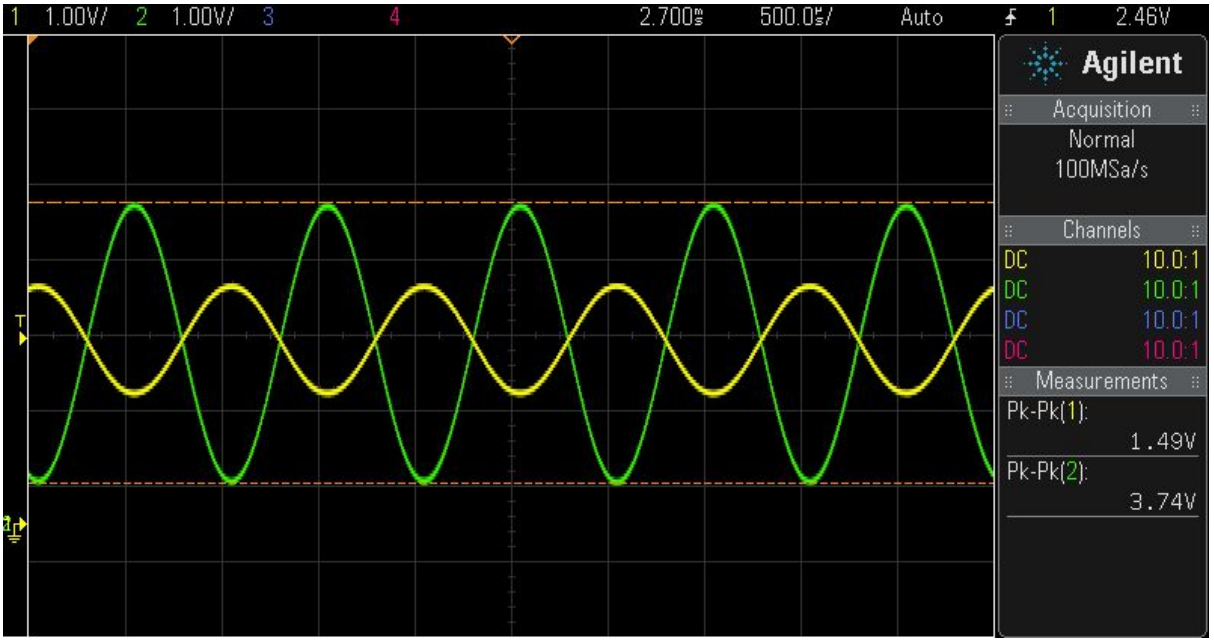

Figure 49 - Stage 3 amplification

Figure 50 shows the final stage of the input amplification, the final amplitude at 3.74V, indicating a total gain over the input stage of 748, just shy of the 752 calculated.

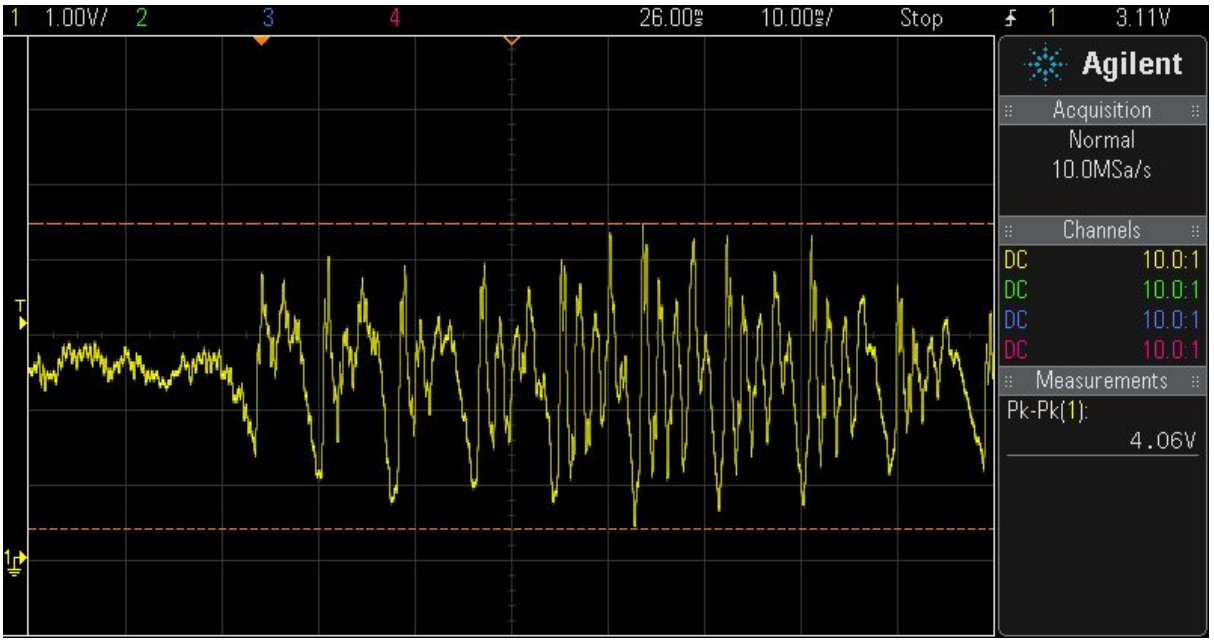

Figure 50 - Final speech waveform

Figure 51 shows the final output of the input conditioning when speaking into the microphone.

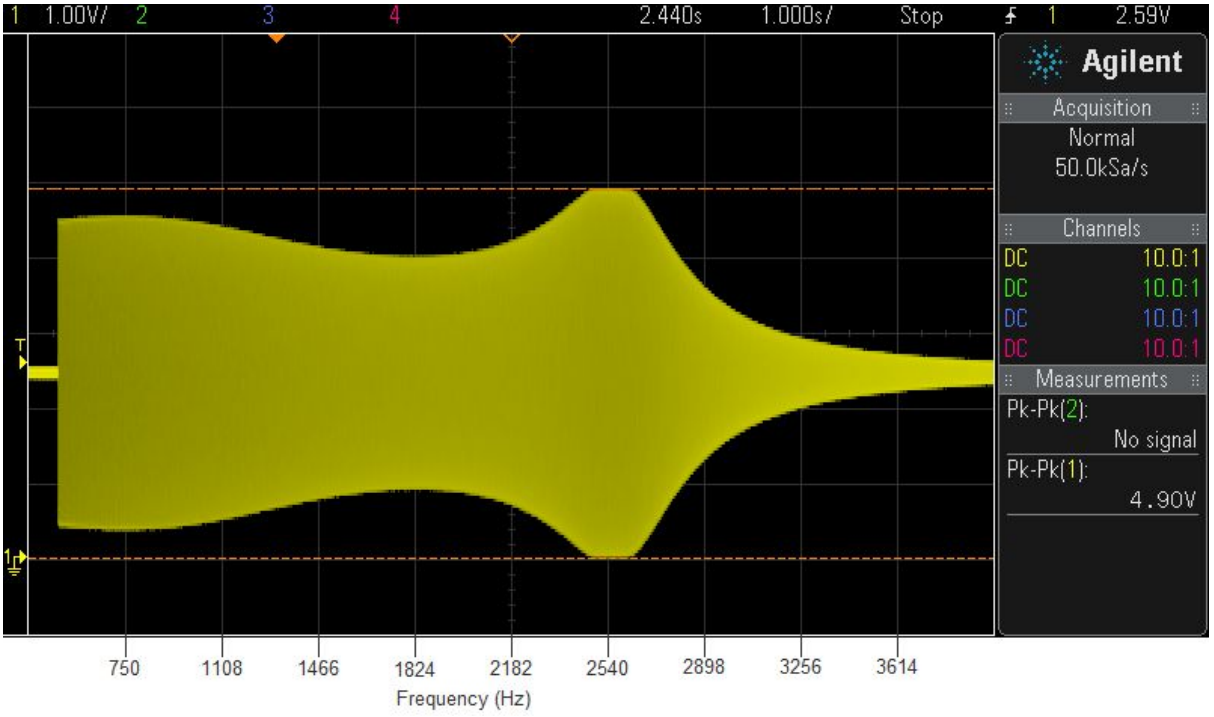

Figure 51 - Filter frequency response curve

Using the sweep functionality on a signal generator, the filters response was tested between 500 Hz up to 4kHz, which was captured using an oscilloscope and can be seen in figure []. From the plot, it can be seen that the amplitude drops rapidly at approximately 2600Hz, indicating the cutoff frequency. The plot also closely resembles that obtained from simulation of the filter.

Future Work

While the DVR's performance was adequate for the required task, multiple variations and changes could be made to the design to enhance or alter its performance.

Recording

The key source of noise in the input conditioning and recording circuitry is due to the poor 5V source provided by the TeensyBob development board, which is rectified using large decoupling capacitors between the 5V and ground rails. This solution, while functioning as intended, is inelegant and bulky. In future revisions of the circuit, finding an alternative to this would both allow a reduction in the size and complexity of the circuit. In doing this, if a higher voltage could be provided to the LMC6484, the imposed voltage limitation could be lifted, allowing a higher amplitude signal to be recorded. Designing and producing a printed circuit board (PCB) would also remove interference created by inconsistencies existing in the breadboard connections.

Code

The code has commented parts filled with code that enhances the current functionality of the project. On uncommenting the commented, and annexing in a few snippets of code, we aim to add a feature that allows the device to record and store multiple files, by writing the name of the WAVE file, number of pages recorded, and the duration in seconds , into a .dat file every time a recording takes place, and using those information stored in the .dat file for a smooth playback.

Playback

The reconstruction filter that changes the PWM signal output by the Teensy into a waveform appropriate for playback on the speaker could be marginally improved by changing it from a passive low pass to an active low pass. This was deemed inappropriate for this first prototype as it required an additional LMC6484 IC to be added to the circuit for only a marginal improvement in the simulated performance and even more negligible practical benefits. Similarly to the input conditioning, the clarity and performance of the audio power amplifier based around the LM386 can be improved using shorter and more reliable connections between components achievable through construction on a more stable platform like a PCB. Providing a higher voltage to the IC would also allow clipping to be reduced and a higher output volume achieved from the current speaker, however the most effective way of increasing the system's output voltage would be to simply modify the existing speaker by adding a larger diaphragm or replace it outright with a more capable unit.

Conclusion

By applying specific concepts and knowledge regarding both electronic and software design, a digital voice recorder was designed, built and tested to a required specification.

The input circuit conditioning takes a signal from a CMA-6542TF-K Microphone, and passes it through two dedicated amplifiers and a filter, constructed around the LMC6484 operational amplifier IC. The filter, based on the Sallen key topology, is a fourth order ChebyShev, chosen for its steep rolloff rate. To avoid aliasing, the cutoff frequency is required to be below half of the sampling rate, however it was found appropriate to limit it to approximately 2.6kHz to reduce any high frequency interference existing in the circuit. The resultant gain of the two dedicated amplifiers and the gain component of the filter is approximately 752, bringing the peak to peak amplitude of a speech waveform just over 4V. This is an appropriate balance between volume and avoiding any peaking imposed by the 5V supply limitation.

The output conditioning receives the PWM signal from the Teensy, converts it into a waveform suitable for playback, and amplifies it for increased listening volume. The reconstruction filter is a passive low pass with a cutoff frequency of 5kHz, which was found to be suitable for smoothing out the PWM waveform. An LM386 with gain 200's circuitry was customised to allow greater control of both gain and volume, and avoid any distortion that may exist due to running at high gain. The resultant output is both loud and sufficiently clear.

The project was considered a success as the desired performance outlined in the scope was completed successfully, however, multiple alterations would ultimately benefit multiple characteristics of the circuit, including its performance as an effective digital voice recorder.

References

www.circuitbasics.com/build-a-great-sounding-audio-amplifier-with-bass-boost-from-the-lm386/#comments