CENTRE FOR NANO SCIENCE AND ENGINEERING (CeNSE)

Indian Institute of Science

# INTERNSHIP REPORT

## Hearing Aid Device using the Teensy 3.6 Microcontroller

Submitted to:

**Prof. M M Nayak**
**Centre for Nano Science and Engineering**
**Indian Institute of Science, Bangalore**

Submitted by:

Jayanth S
ID: 01FB16EEC111

Date: 11th July 2019

# ACKNOWLEDGEMENTS

The internship opportunity I had at the Centre for Nano Science and Engineering (CeNSE), Indian Institute of Science (IISc) was a great chance for learning and personal development.

I would like to express my sincere gratitude to Prof. Navakanta Bhat and Prof. M M Nayak for giving me this opportunity and for their valuable guidance.

I owe my gratitude to my mentor Mr. Rakhesh for his guidance and support which not only made this project a success, but also a memorable experience.

My sincere thanks to all members of the MEMS Packaging Lab, CeNSE, IISc for their continuous support and encouragement.

I will strive to use the skills and knowledge I have gained, and work hard for their improvement.

Jayanth S

# TABLE OF CONTENTS

# ABSTRACT

The Teensy 3.6 is a USB-based microcontroller development system from the Teensy range of microcontrollers developed by Paul Stoffregen. It is completely compatible with the Arduino software and libraries and can thus be used just like an Arduino microcontroller. However, a bootloader application called the Teensy Loader has to be installed to make it compatible with the Arduino Integrated Development Environment (IDE). Teensyduino is a software add-on in the Arduino IDE that is used to run sketches on the Teensy. The Teensy can be configured as many types of devices such as a speaker, keyboard etc. All communication is performed at full native 12 Mbit/sec USB speed. For more information, please visit the website:

https://www.pjrc.com/store/teensy36.html

The Audio System Design tool for the Teensy is used to create code for the circuit connections. It converts the circuit into code directly.

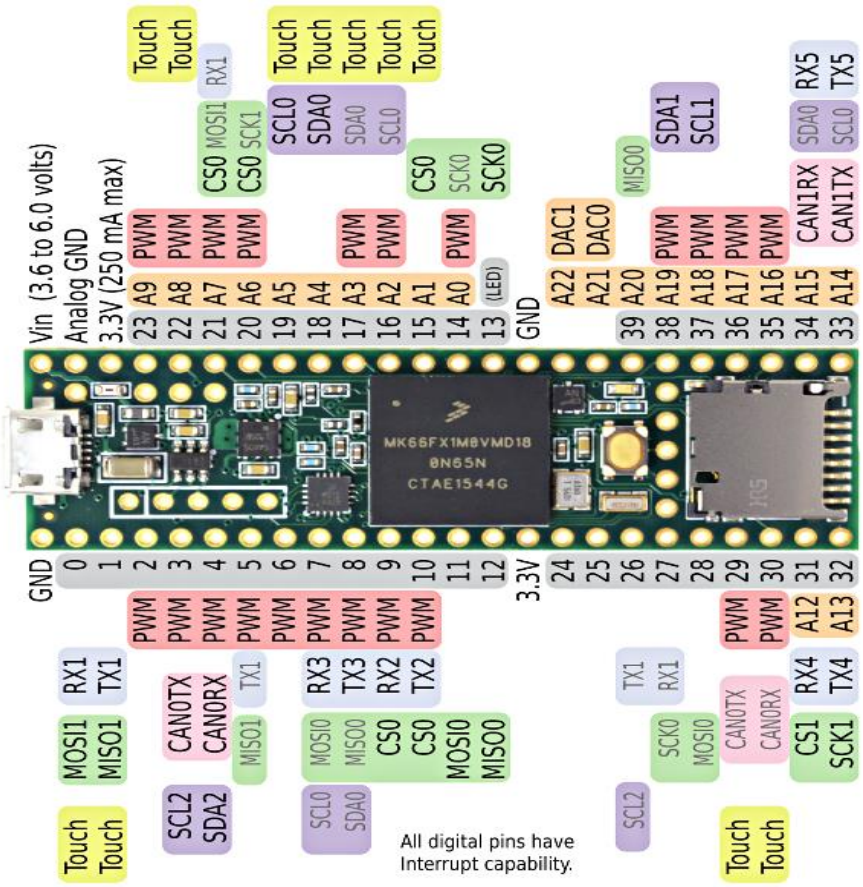



Figure 1: Teensy 3.6 Pinout

# INTRODUCTION

A hearing aid device is used to improve hearing for people suffering from hearing loss. They are common medical devices that use amplification and filtering methods to achieve improved hearing. Modern hearing aids also use digital signal processing methods like feedback, noise reduction and compression techniques. In my project, I incorporate adaptive gain to improve comfort for the user.

A basic hearing aid device consists of an amplifier and some filters. Below is a block diagram of the circuit:

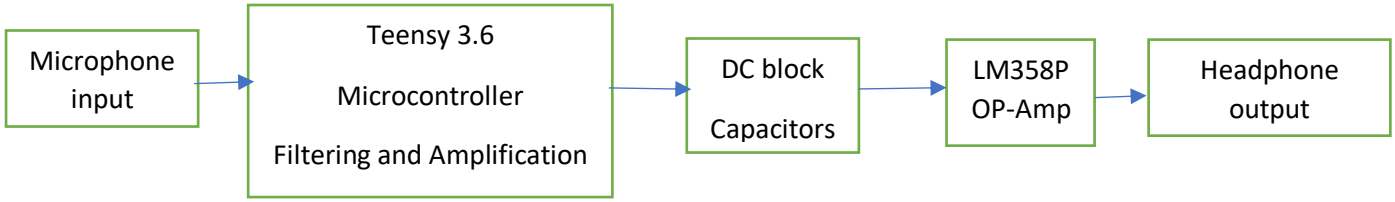



Figure 2: Block Diagram of the circuit

Both an Adafruit PDM MEMS microphone and an Adafruit electret microphone were tested for taking in audio input.

| PIN | Electret Microphone | PDM MEMS Microphone |
|---|---|---|
| VCC | 3.3V(250mA max) | 3.3V(250mA max) |
| GROUND | GND | GND |
| OUT/DAT | 16 A2 PWM SCL0 Touch | 13(LED) SCK0 |
| CLK | - | 9 PWM RX2 CS0 |
| SEL | - | GND |

Table 1: Connecting the microphones to the Teensy 3.6

Capacitors of 10 micro farads were used to block DC voltage. They prevent the flow of DC and minimise the flow of low frequency audio currents while minimum impedance to RF signals.

The LM358P op-amp is a general-purpose dual op-amp from Texas Instruments. It is an 8 pin DIP that has a common power source for both the op-amps. It was used to prevent any back current from damaging the Teensy. Short the inverting input terminals with their respective outputs.

A 24 ohms earphone (general purpose) was used as the output.

# CIRCUIT

The circuit is as shown below:


Figure 3: Circuit created using the Audio System Design Tool

This circuit was used to connect the PDM MEMS microphone. Replace the pdm1 component with adc1 and make the appropriate circuit connection changes to connect the electret microphone. The two connections leading to the DACS (Digital to Analog Converter Stereo) represent the left and right sides of the earphones respectively.

Connect DAC0 (A21) and DAC1 (A22) to a 10 micro farad capacitor each. Feed the capacitor outputs to the LM358P op-amp pins 3 and 5 which are the non-inverting pins of the op-amp and connect the output pins 1 and 7 to the left and right sides of the audio jack socket. Ground both the audio jack socket and the LM358P and provide VCC of 5V to the LM358P.

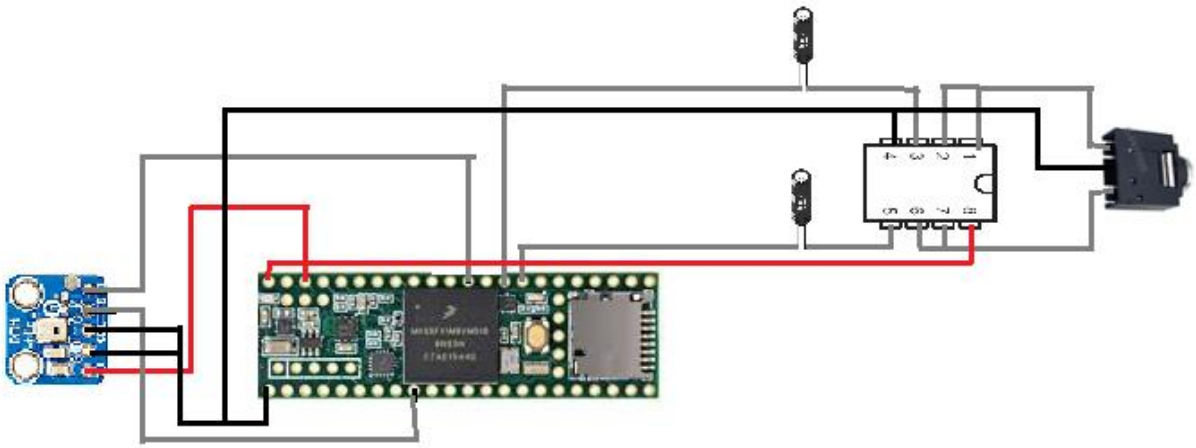


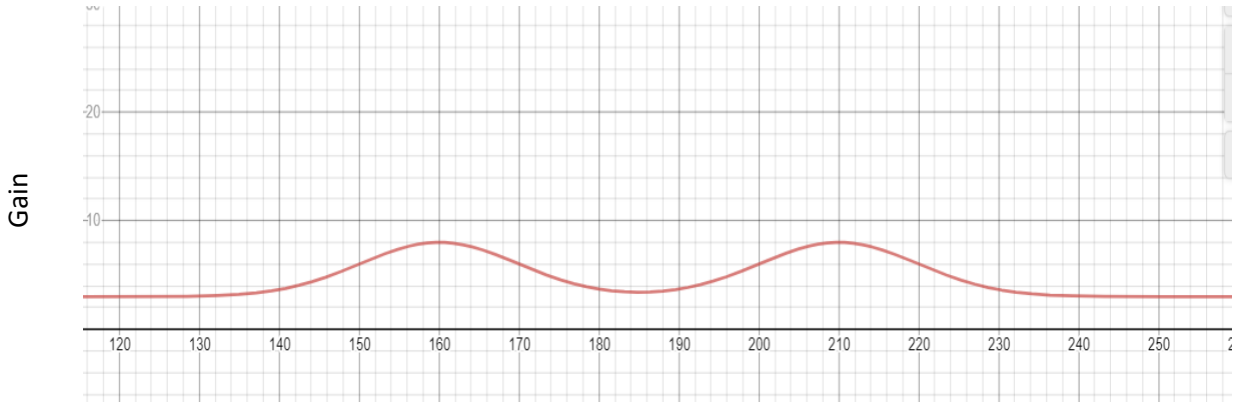
Figure 4: Circuit diagram

# ALGORITHM

- Use the audio library to make the connections and import the code.
- Declare audio memory in setup()
- In loop(), set the filter cut-off frequency.
- Read output at A22 and name it prev.
- After a delay of 1 ms, read A22 again and name it present.
- Find the difference between present and prev.
- If the difference is very low, set amplifier gain to 1 as the audio may be noise or a sound from far away.
- If not, feed the value obtained from the below formula as amplifier gain:

$$\mathbf{val} = \mathbf{5}e^{-(\frac{(\mathbf{prev}-\mathbf{210})}{\mathbf{14}})^2} + \mathbf{5}e^{-(\frac{(\mathbf{prev}-\mathbf{160})}{\mathbf{14}})^2} + \mathbf{3}$$

- The above formula implements adaptive gain.
- Finally, give a delay of 100 milli seconds.




DAC output value taken using analogRead()

Figure 5: Gaussian curve used for adaptive gain

The curve was created using the Desmos online calculator. Initially, a piecewise linear curve was used. However, it led to rapid changes in the gain at the points of non-differentiability. The gaussian curve had a better outcome because it is continuously differentiable. Also, the gaussian automatically takes care of the corner conditions. No special code is needed to define the gain when it is out of bounds.

Further, at normal speaking sound levels, the gain was recorded to be maximum at a distance of about 30 centimetres from the microphone.

# OBSERVATIONS

**PDM MEMS MICROPHONE** **ELECTRET MICROPHONE**

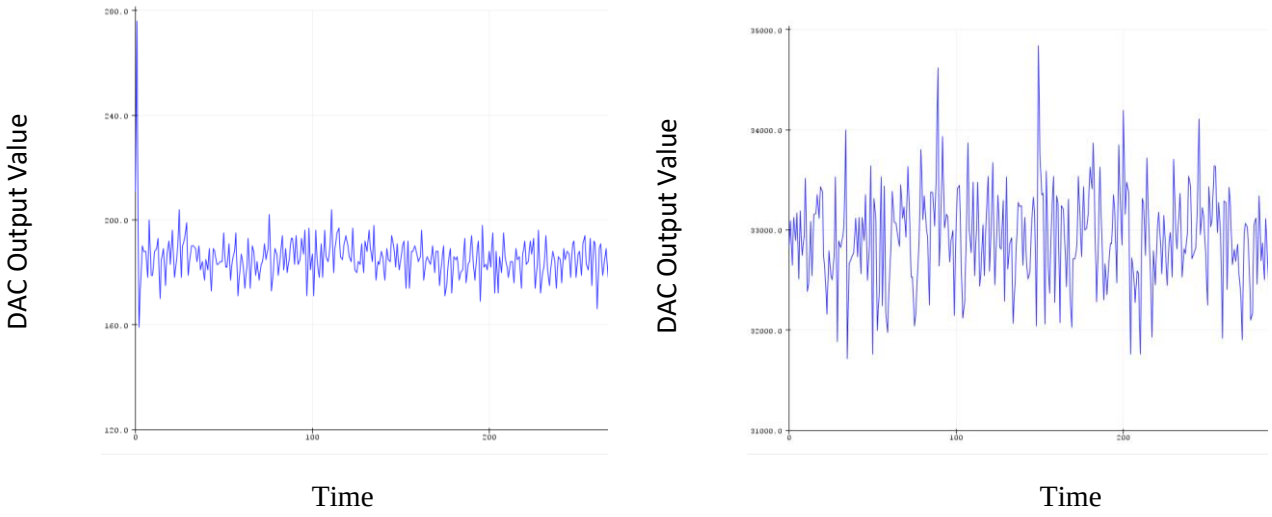



Figure 6: Noise levels in the microphones

Both the microphones were first tested without amplification and filtering. The PDM microphone has lesser noise values compared to the electret microphone. Thus, the PDM MEMS microphone was used for further testing.

Next, a 1000Hz signal was given to the PDM microphone without any amplification and filtering. The output was recorded as follows using the Arduino serial plotter.

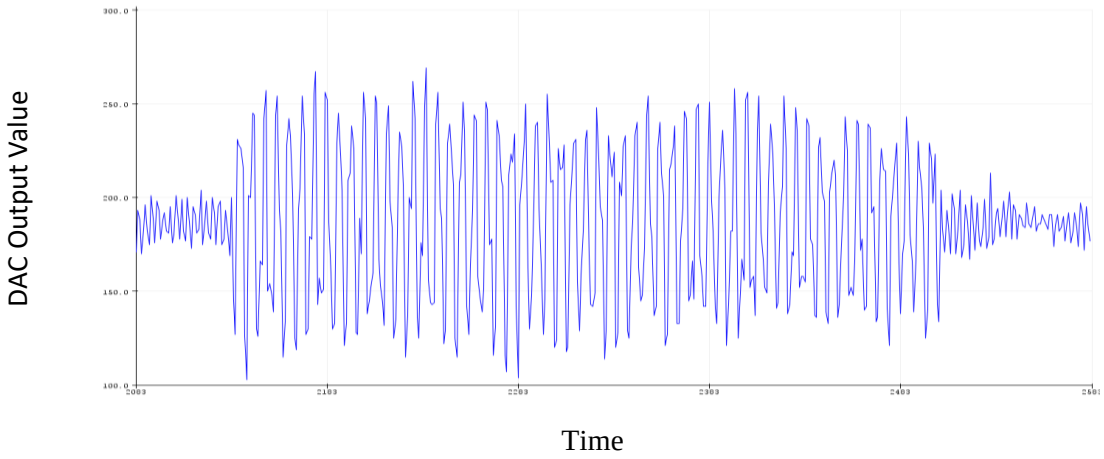


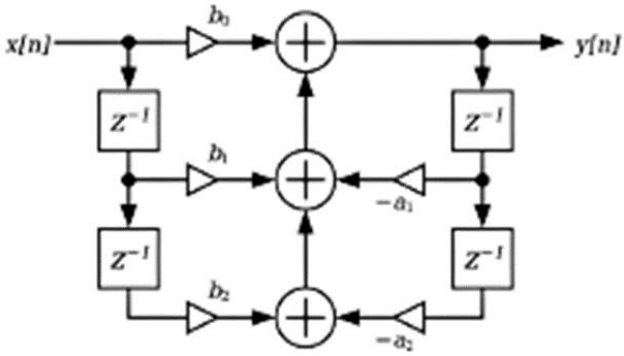

Figure 7: 1000Hz input without filter

Then, a low pass filter at 500 Hz and a high pass filter at 30 Hz (biquad1in the circuit), both with -3dB roll-off, were implemented using the bi-quad component in the audio system design library. A bi-quad is simply a two-pole IIR filter, where both the numerator and denominator of the transfer function are quadratic equations.

The bi-quad filter in the Teensy library can be cascaded up to 4 stages. The LPF and HPF were cascaded in the circuit.

The robustness of the bi-quad filter is a big consideration as the bi-quad structure is stable: higher order IIR filters are prone to instability in fixed point systems due to long feedback delay paths, causing poles to go outside of the unit circle. The absence of any feedback between the individual cascaded sections (decoupling the poles and zeros between sections) results in a more stable structure that is less sensitive to quantization.

Thus, a bi-quad filter was used instead of higher order filters in the circuit.




$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}$$

Figure 8: The Bi-quad Filter and its Transfer Function

A 1000Hz signal was passed through this circuit and the following was observed:

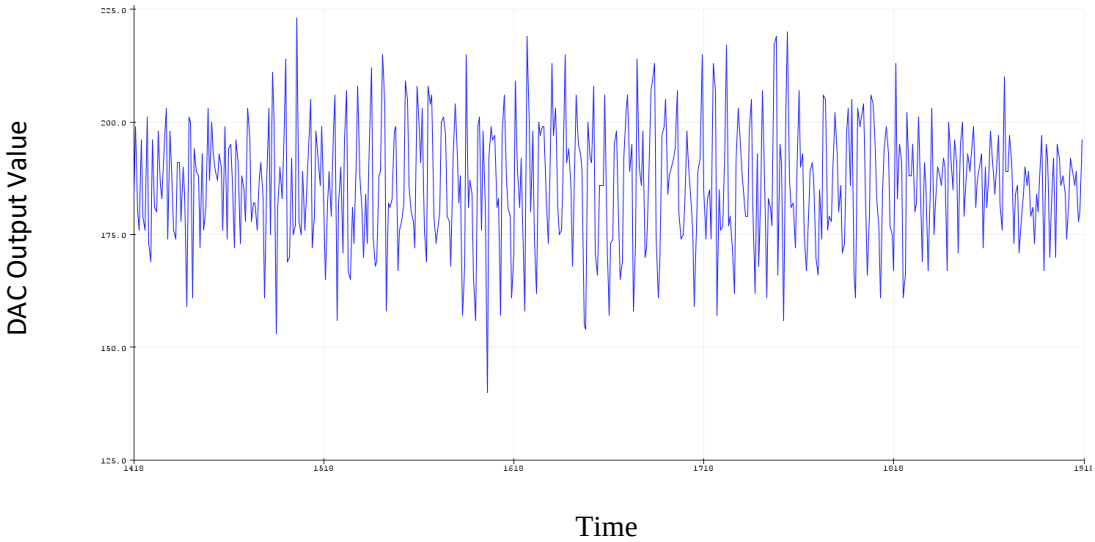



Figure 9: 1000Hz input with LPF cut-off frequency at 500 Hz and HPF cut-off frequency at 30 Hz

Then, a 500Hz signal was sent with the LPF cut-off at 1000Hz and HPF cut-off at 30Hz. An amplifier (amp1 in the circuit) was added with gain 3 and the following was observed:

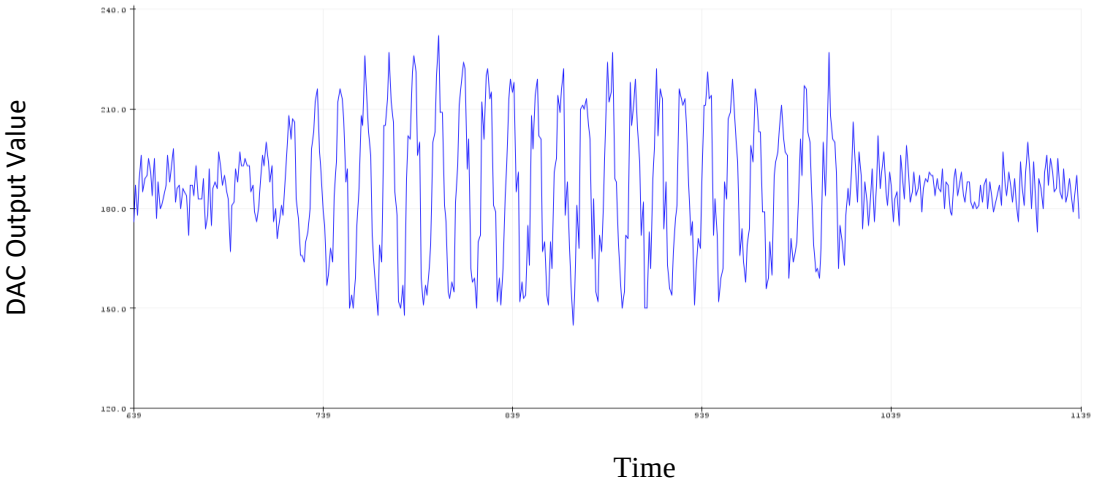



Figure 10: 500 Hz input with LPF cut-off at 1000 Hz and HPF cut-off at 30 Hz

Adaptive gain was implemented next. Adaptive gain is used to control the gain of the amplifier by responding to the environment. If the audio input is loud, it can be assumed that the audio source is nearby and thus lesser amplification is needed. If the input is feeble, it can be either noise or a valid audio input. In this case, it was considered noise and gain was reduced.

# RESULT

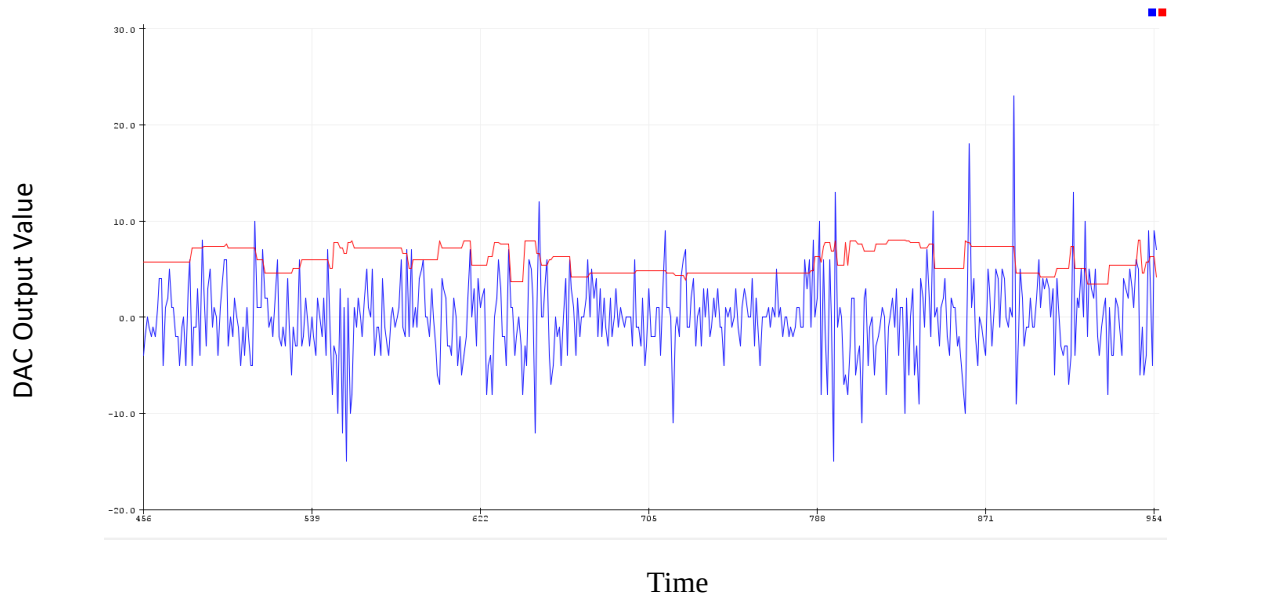



Figure 11: Adaptive gain

In the graph, the red line is adaptive gain and the blue line is the difference between the present value of DAC and the previous value of DAC.

According to the program, whenever there was a difference of less than 5 or more than -5, it was considered noise. Therefore, gain was decreased. If this condition is not met, the gain is changed accordingly using the gaussian curve. This can be seen in the above graph. This serves as a basic mechanism for noise reduction. However, the drawback is that sounds from afar are also attenuated.

The code for varying the gain using the gaussian curve indicates that the amplification will be highest when the audio input leads to a sample at 170 and 210 at the DAC output. It was also observed that changing the means of the gaussians can be used to amplify any audio input as long as the mean corresponds to that sampling value. For example, for a loud sound, the audio sampling may result in a DAC value as low 120 or as high as 260. If the gaussian means were kept at these points, then these sounds can also be amplified.

# REFERENCES


1. The Teensy audio tutorials:
   https://www.pjrc.com/store/audio_tutorial_kit.html
2. The PJRC Teensy forum:
   https://forum.pjrc.com/
3. Teensy Audio System Design Tool:
   https://www.pjrc.com/teensy/gui/index.html
4. The Adafruit PDM MEMS Microphone:
   https://www.adafruit.com/product/3492
5. LM358P IC datasheet:
   https://www.digchip.com/datasheets/parts/datasheet/477/LM358P.php
6. The Gaussian curve:
   https://ned.ipac.caltech.edu/level5/Leo/Stats2_3.html
7. Desmos graphing calculator:
   https://www.desmos.com/calculator

# APPENDIX

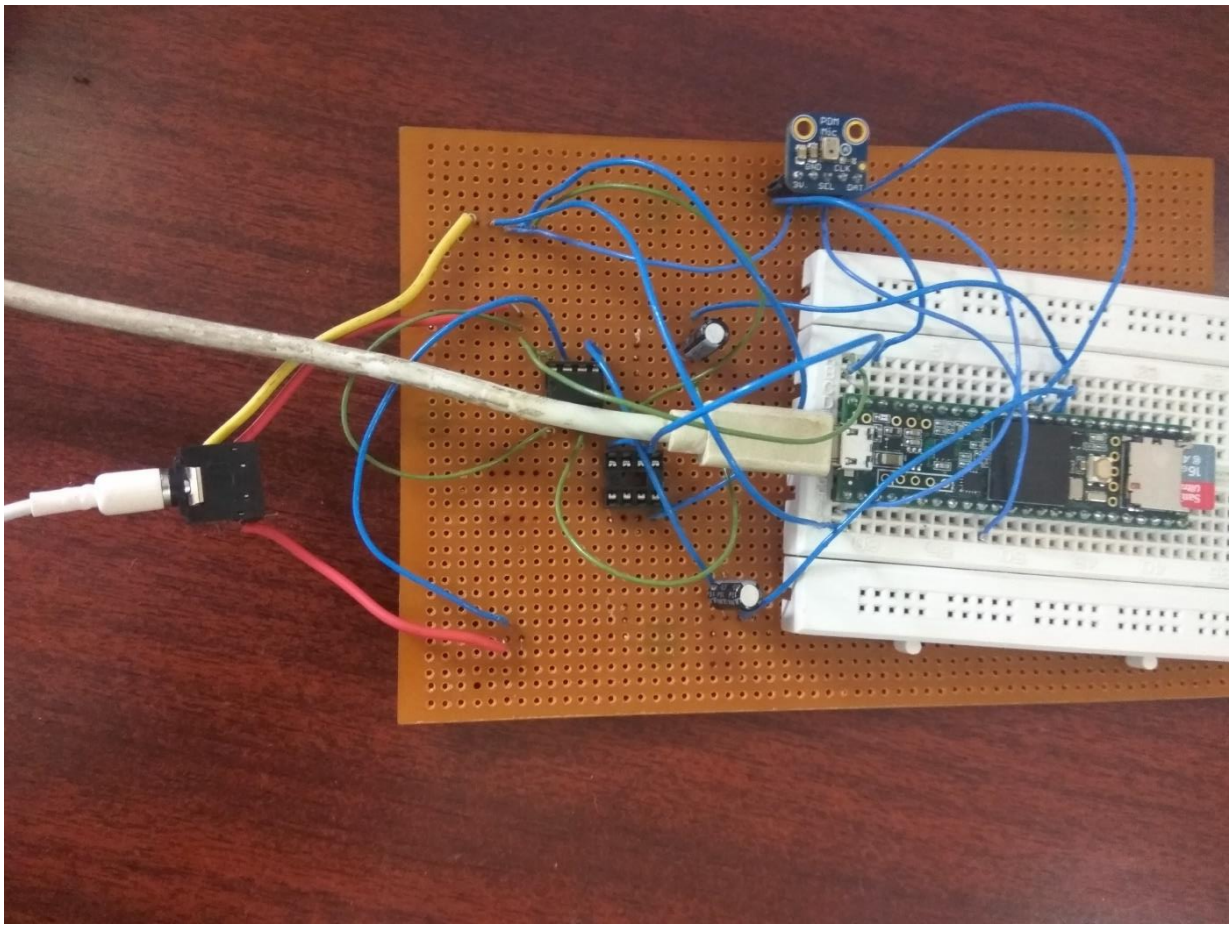


Figure 12: Picture of the circuit