5/31/2019

EGB 240: ELECTRONIC DESIGN

ASSESSEMNT 2: DVR DESIGN REPORT

ARIK INTENAM MIR

N9637567

Executive Summary:

The aim of the DVR project is to develop and configure a digital voice recorder system. This is to be accomplished using both analogue circuitry and firmware, to produce the optimal audio system. The project is aided with prescribed components, equipment, and software tools to optimally, design and implement the system. The main components provided includes:

- LMC6484 multi rail operational amplifier
- TeensyBOBv2 with embedded ATmega32U4 micro-controller
- Development/ breadboard
- Electret microphone
- Headphone interface kit

Also, equipment such as oscilloscopes and signal generators, and software tools such as LTspice and MATLAB were thoroughly used to conduct the project to an optimum standard.

The entire project consists of both hardware and firmware aspects. The hardware aspect of the project was implemented on the provided breadboard, which consisted of microphone, microphone amplification circuitry and a 4th order Chebyshev filter for functionality of anti-aliasing. The software side of things were produced using C programming language on IDE Atmel Studio 7.0 for operating the micro-controller.

All the outcomes and results from numerous experiments, design plans and simulations were produced through the process of continuous trial and error for achieving the best outcome possible.

Contents

List of Figures and Tables

1.0 Introduction

The report for the DVR project, aims to provide in-depth and step-by-step analysis of the design, planning, conceptual knowledge, and methodologies used to develop, prototype and demonstrate a digital voice recorder (DVR).

The entirety of the report has been divided into several sections. Each section consisting of relevant information, analysis and ideas. For understanding the major contents better, images and tables has been attached throughout the report.

A section has been dedicated completely, detailing the prerequisite conceptual knowledge required for different aspects of this project. Flowchart has been added to understand the contents more clearly. The Appendices section contains information and data which were too large to be attached to the body of the report.

Furthermore, the body of the repot consists of systematic explanation of the project and expectations from the functionality, productivity and technical aspects. Experimental results, scope of the project, limitations and future applications are displayed throughout the entirety of the report. It has been made sure that the report contains correct citations to the literatures and conceptual knowledge utilized throughout the report.

2.0 Literature Review

Understanding and being knowledgeable about the relevant key topics such as the theorems, analysis techniques and the conceptual ideas, played a significant role in the execution of the design and the documentation of this report. As the conceptual topics supported taking the technical decisions during the project.

2.1 Signal Sampling Theorem

Signal sampling theorem is the idea of converting a continuous time signal to a discrete-time signal. To explain more elaborately regarding practical examples, signal sampling can be used to convert analogue signals to their digital counter-parts. The conversion is done, so that the digital counter-part can processed via electrical mediums. Sampling signals are measured over a period of time to reproduce the signal in the digital spectrum. According to Nyquist sampling theorem, the sampling frequency must be at least twice the size of the bandwidth of the analogue signal, so no data is lost in the discrete spectrum, where $Fsamp \geq 2Fm$ [1]. If the Nyquist sampling conditions are met, only then the signal can be perfectly reconstructed from the digital spectrum.

The sampling frequency required for the DVR project was set at 15.625kHz. So to implement this required frequency, an understanding of how signal sampling works, provided with a better knowledge about analyzing and converting the samples from analogue to digital.

2.2 Aliasing

Interference seen in a reconstructed signal is known to be as Aliasing. Aliasing disrupts the reconstructed signal with noises. Aliasing occurs whenever the signal contains any frequency greater than the Nyquist requisite, which is half of the sampling rate [2]. This problem is solved using a special type of filters, known as anti-aliasing filters. The job of an anti-aliasing filter is to attenuate and cut-off frequencies lower or greater than a specific cut-off frequency.

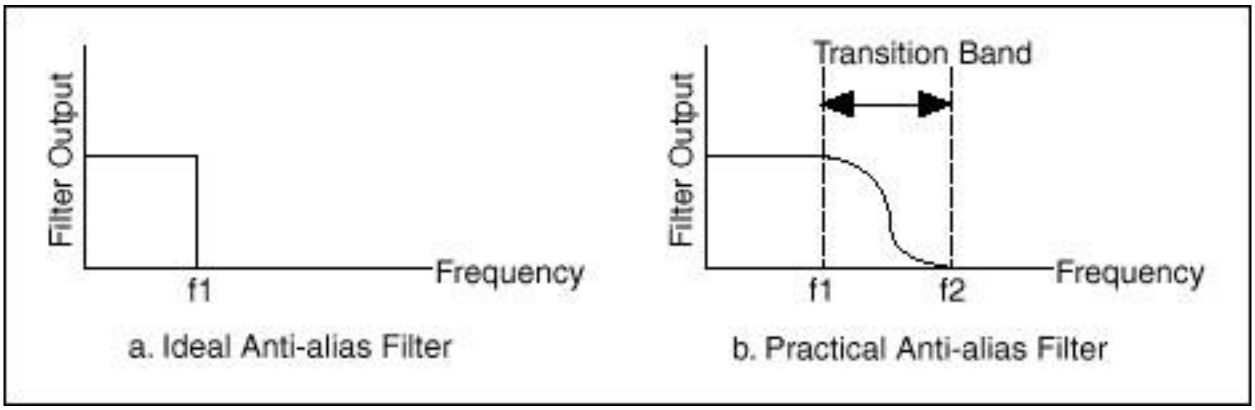


Figure 01: Anti- Aliasing Filter

For this DVR Design, to make sure the original signal doesn't get distorted, the anti-aliasing filters were set up to retard signals to -48dB for all frequencies greater than the half of the sampling frequency 15.625kHz.

2.3 Chebyshev Filter

Chebyshev filters are used for differentiating one band of frequencies from another [3]. This class of filters can be applied in various applications as it can be both analogue and digital filter. This filter shows a steeper roll-off compared to that of Butterworth filters by allowing ripple in the pass-band [4]. There are 2 types of Chebyshev filters- i) Type-I

ii) Type-II

The figure below shows the behavior of Type-I and Type-II Chebyshev filters:

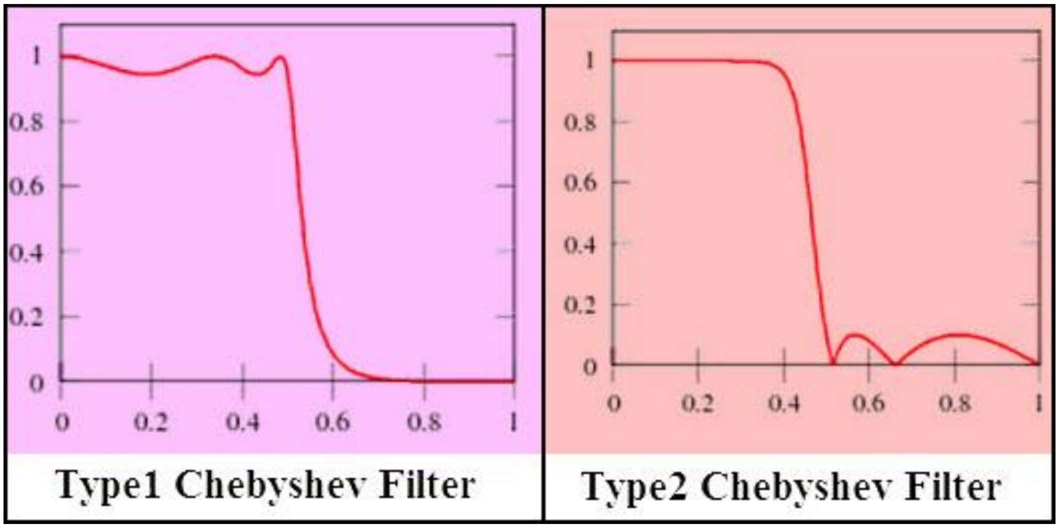


Figure 02: Type-I and Type-II Chebyshev Filter

For Type-I Chebyshev filters, it shows equiripple behavior in the pass-band and monotonic characteristic in the stop-band respectively [4]. And meanwhile for Type-II, stop-band shows equiripple behavior, while the pass-band shows monotonic characteristics. As there is no cut-off frequency defined at -3dB for Type-I, it is defined, where the value of the gain is equal to that of the pass-band ripple [4].

The order of a Chebyshev figure is found out using the formula:

$$nCHEBYSHEV = ceil\left(\frac{cosh^{-1}\sqrt{\frac{10^{0.1A_{min}-1}}{10^{0.1A_{max}-1}}}}{cosh^{-1}\left(\frac{\omega_s}{\omega_p}\right)}\right), where\ \omega_p = 2\pi f_p$$

2.4 Butterworth Filter

Like Chebyshev, Butterworth filter is used in signal processing appliance. Properties of a butterworth filter are quite unique to chebyshev. It is designed to have a frequency response with a more flat pass band and also consists of steadier roll-off in comparison to chebyshev. The figure below shows the behavior of a Butterworth filter:

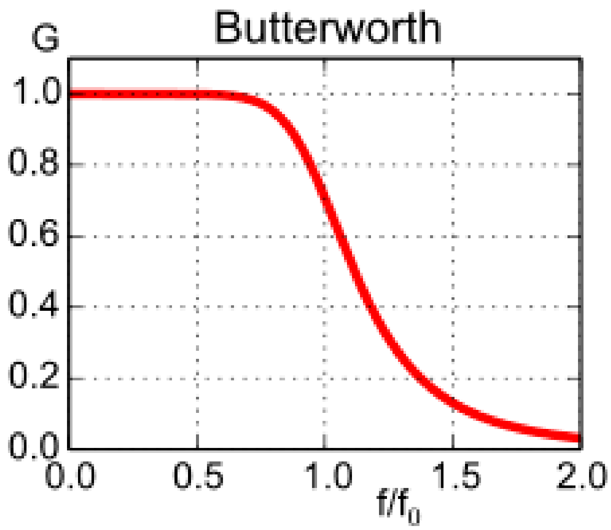


Figure 03: Butterworth Filter

The order of a butterworth filter is measured using the formula below:

$$nBUTTERWORTH = ceil\left(\frac{log_{10}\sqrt{\frac{10^{0.1A_{min}-1}}{10^{0.1A_{max}-1}}}}{2log_{10}\left(\frac{\omega_s}{\omega_p}\right)}\right), where\ \omega_p = 2\pi f_p$$

2.5 Human Speech

The energy produced from human energy is between the range of 300Hz to 3kHz. Humans can only detect sounds around 20Hz- 20kHz. Whereas, most sensitivity for human hearing is found to be around 300Hz-10Hz. Thus, frequency range between 20Hz to 2600 HZ was deemed to be the most appropriate range for ensuring clarity in speech recognition [6].

3.0 Design

3.1 Project Description

The goal of this project is to prototype, design, implement and demonstrate a Digital Voice Recorder. Some physical components were provided in regard to implementing. The table below shows the list of components provided.

Quantity	Name	Footprint
1	Development board w/ breadboard and USB cable	TeensyBOBv2 & ATmega32u4
1	Electret Microphone	CMA-6542TF-K
1	Quad rail-rail operational amplifier IC	LMC6484
1	Headphone interface kit	-

Table 01: List of components provided

The software tools that were used for this project are LTSpice, MATLAB, Teensy Loader and Atmel Studio 7.0. Atmel Studio 7.0 was used for generating the embedded C code, used for operating the micro-controller and for converting the C codes into a .hex file. The Teensy loader is used to load the .hex file to the micro-controller. Alongside the provided components and software tools, wave generators and oscilloscopes were also useful in the implementation of this project. The conditioning circuitry of this project was implemented on the micro-controller via Analogue to Digital Converter (ADC) and Pulse Width Modulation (PWM) peripherals. And the analogue functionality of the DVR was implemented on the breadboard, which also contains the prototype circuit for this project.

A Flash USB with a SD card was also provided, which was used to store audio samples from the micro-controller. In short, the main goal of the digital voice recorder is to record, stop, playback and store audio samples with optimum quality.

3.2 Design Goals & Specifications

As mentioned earlier, the main goal of this project is implementing a digital voice recorder, which operates according to some given constraints. It was required for the project to meet some specifications too.

The analogue input from the microphone was first amplified using the input operational amplifier. An anti-aliasing filter was required to design according to the sampling theorem (cited: 2.1 Signal Sampling Theorem) for restricting the bandwidth of analogue signals from the microphone. The sampling frequency is set at 15.625 kHz.

Embedded code for the micro-controller was written using C programming language. The required task of the embedded code was to implement the functionality of playback, record and stop of audio samples using pushbuttons S1, S2 and S3 respectively, from the TeensyBOBv2 board.

The list of major design specifications is given below.

- Sampling frequency = 15.625kHz.
- Maximum attenuation of the input filter = -1 to -3 dB
- Minimum attenuation = -48 dB
- ADC = 10-bit (Maximum)
- Clock frequency for Micro-controller = 16MHz.

3.3 Scope

The goal for this project was to design and build a digital voice recorder, for recording human speech at an average volume and then storing it and with other mentioned functionalities. So, the achievable objectives for this project includes:

- Building an input amplifier circuit
- Building a low-pass active analogue filter for sampling input signals to the ADC by means of Sallen-key topology, which was chosen to be 4th order Type I Chebyshev filter divided into two stages.
- Conceptual design was to be made using the software LTSpice.
- Circuit was to be prototyped on a breadboard and tested using oscilloscopes and wave generators for checking functionality against results from the simulations of LTSpice.
- Developing the embedded code using C language using Atmel Studio 7.0 for operating the pushbuttons and LEDs on the ATmega32u4 micro-controller.
- Storing the converted digital signals to a USB drive using a SD card.

While setting these goals according to the specifications it was already noted that some factors might limit the quality of the digital voice recorder. Some of these factors includes, not being able to design the DVR design for a PCB design, not being able to implement the prototype with components of higher quality and a major factor was the time constraint.

3.4 Methodology

The DVR project was done by following an efficient guideline, which included using both software resources and provided hardware components.

3.4.1 Microphone Amplification and DC removal circuitry

The initial input conditioning circuitry consisted of the microphone amplification and DC removal circuitry and Anti-Aliasing filters. The microphone amplification and DC removal circuit was chosen and built according to the provided requirements from the design specifications.

3.4.2 Anti-aliasing Filters

For designing and implementing the anti-aliasing filter, the concept and knowledge about Butterworth and Chebyshev filter, was explored [5]. Then pre-defined MATLAB codes was used to test these two types of filters using the buttord and cheb1ord functions respectively.

After testing both types of filters thoroughly with altering values of max attenuation and sampling frequency it was concluded that 4^{th} order Chebyshev filter met the requirements for the DVR project more efficient and productively than the Butterworth filters. Another reason the Chebyshev filter was chosen because, it required lesser components than the Butterworth filters. The MATLAB codes for testing the performance and choosing the order of the anti-aliasing filters are provided in the Appendix 01.

3.4.3 Circuitry Design and Simulation

The design of the circuit schematic and configuration of the amplifier and anti-aliasing circuitry was conducted using the software LTSpice. The designs were configured using the conceptual knowledge mentioned in the literature review. The component values required for the input circuitry was calculated by hand with some help from MATLAB and also with the help of the conceptual knowledge. The performance of the circuitry was also simulated using LTSpace by using defined LTSpice parameters. The appropriate component values were chosen finally by several tests using the trial and error method. The circuit schematic is given in Figure 8; while the LTSpice parameters for the simulations are given in the Appendix 03.

3.4.4 Prototyping

A breadboard was used to prototype the designed circuitry from LTSpice, using the required provided components. The prototype was beneficial for conducting the experimental tests via oscilloscopes. These experimental results were useful for comparing against the simulated results, acquired from LTSpice.

3.4.5 DVR Codes

The firmware codes required for the software part of this project was implemented on a pre-defined set of codes called “SkeletonDVR”, outlining the basic functionalities required for operating the micro-controller.

3.5 Technical Design

The flowchart given below shows the modus operandi of the technical designs of the DVR Design.

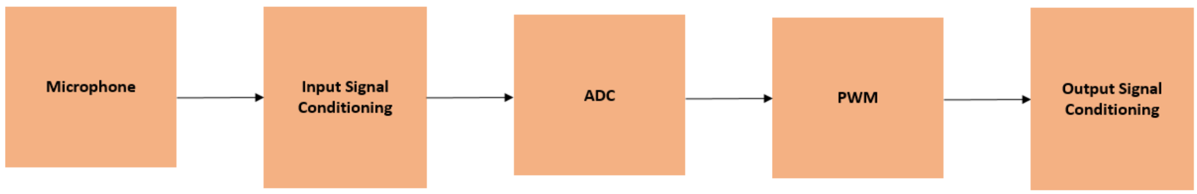


Figure 04: Flowchart of the steps required for the technical design

3.5.1 Microphone Amplification Circuitry

As per the design requirements, the average peak-peak voltage from the microphone has to be 50mV. So, the voltage running through was rescaled to a suitable signal level for operating in the micro-controller. 5V is the initial voltage for the micro-controller, thus making it a requirement for the DC bias to be at a reference voltage of 2.5V. A voltage divider circuitry was used for setting up the DC bias voltage at 2.5V. So, the amplifier circuitry was set up using the Sallen-key topology to rescale the output of the microphone. The design of the circuit is given below in Figure 05.

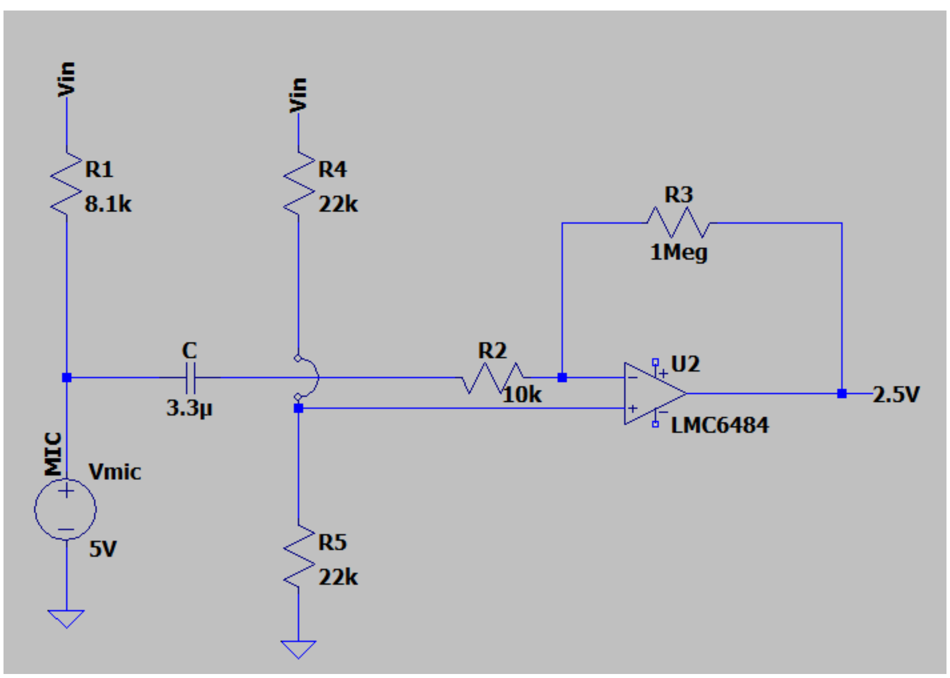


Figure 05: Amplifier circuit with a output reference voltage of 2.5V

3.5.1.1 Calculation & Estimation for the Component Values

Gain = $\frac{V_{OUT}}{V_{IN}}$ = $\frac{5V}{50mV}$ = 100.

According to conceptual knowledge for this specific circuit,

Gain = $\frac{V_{OUT}}{V_{IN}} = \frac{R_3}{R_2}$

Now,

Let $R_2 = 10k\Omega$,

So, $R_3 = Gain \times R_2$

$R_3 = 100 \times 10k\Omega$

$R_3 = 1M\Omega$.

The $R_2 = 10k\Omega$, operates as a high pass filter, but as it can have a very low cut-off frequency, thus making it unable for preventing the required frequency which needs to be attenuated, thus a capacitor was used for eliminating the DC component of the input and also for passing the AC signals from the microphone. The cut-off frequency f_c is chosen to be 5Hz. The calculation required for choosing the capacitor value is given below:

C = $\frac{1}{2\pi \times f_c \times R_2}$

C = $\frac{1}{2\pi \times 5 \times 10k\Omega}$

C = 3.2μF ≈ 3.3μF

3.5.2 Anti-Aliasing Filter

The design of the anti-aliasing filter was conducted using the Sallen-key topology by implementing a 4th order Chebyshev filter. The anti-aliasing filter was designed with the objective getting the required attenuation goal of 7.8125 kHz at -48dB. The 4th order Chebyshev filter was further divided into two 2nd order stages.

The filter was constructed such that, the 1st stage of the anti-aliasing filter has lower quality factor than the 2nd stage. The quality factors were 0.8576 and 4.0918 for 1st and 2nd stage respectively. Chebyshev filter was chosen instead of Butterworth filters, because it was observed that implementing a Chebyshev filter was simpler, as it requires lesser orders of filters to perform the same functionality of that of the Butterworth filter.

A set of MATLAB codes were generated and used to figure out the component values and quality factors, which are dependent on the sampling frequency, pass band frequency and the maximum stop-band attenuation respectively. The MATLAB codes are referenced in Appendix 01. As some of the component values from the calculations were not found practically, so the closest values are chosen. The output of the anti-aliasing filter connects to the J_{IN} port of the TeensyBOBv2.

3.5.2.1 Specification verification & Schematic

The schematic of the anti-aliasing filter is given in the figure below:

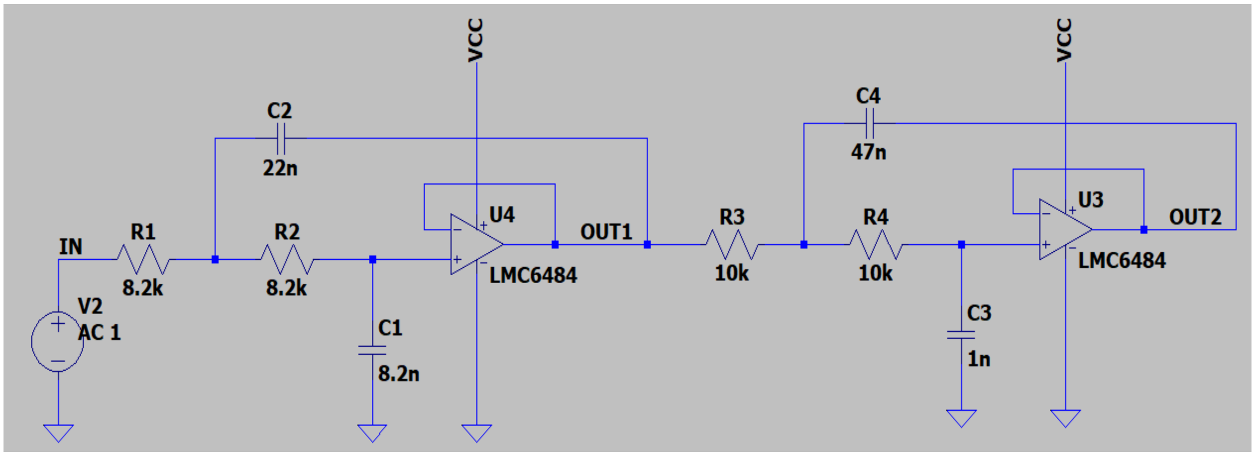


Figure 06: Chebyshev 4th order Anti-aliasing filter with the specific component values.

A Bode-plot was constructed from the specification to observe if the minimum attenuation of -48 dB was truly met. With some approximation it was seen that the expected result was found. The bode-plot is in the following figure.

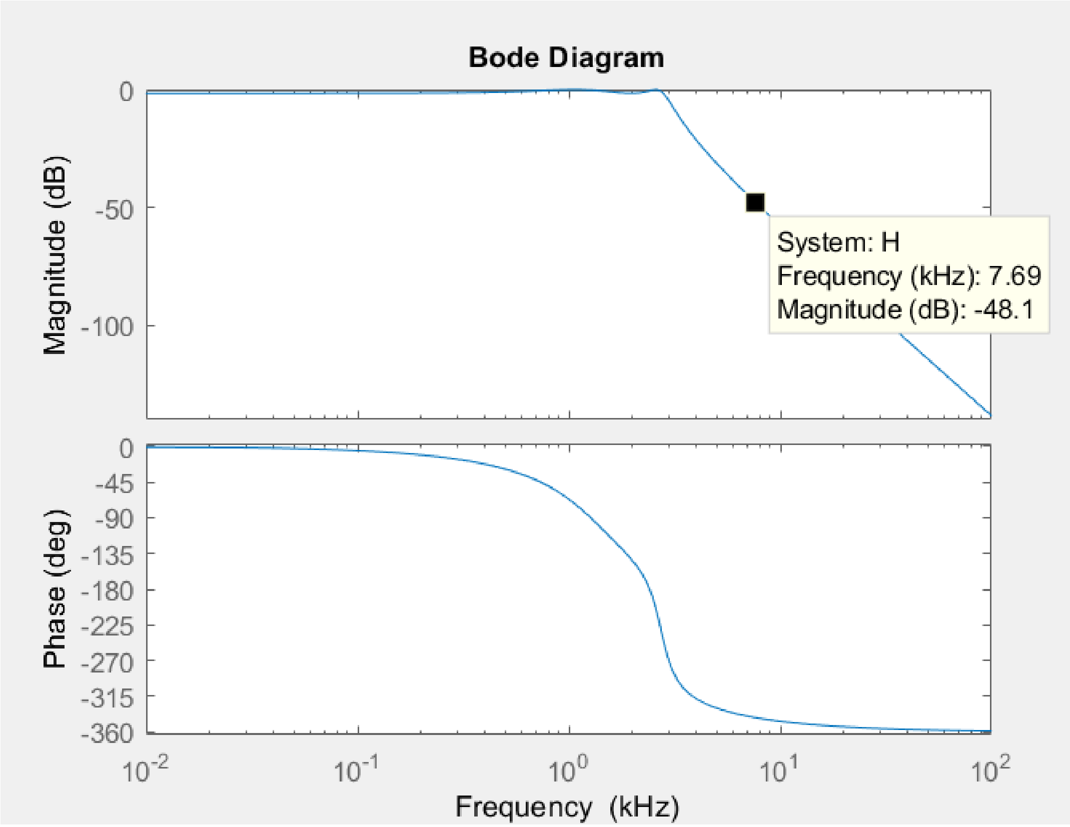


Figure 07: Bode-plot for the anti-aliasing figure with specifications met.

3.5.3 Firmware Development

The firmware required for the DVR project was implanted on the ATMega32u4 microncontroller using C programming language. Several pre-defined libraries and modules with their own user functions were provided for this task. These modules are-

- **adc.c**
- **buffer.c**
- **serial.h**
- **timer.c**
- **wave.c**

The ADC was configured at 15.625 kHz, 8-bit sampling and the PWM was configured for an output frequency of 62.5 kHz. The pushbuttons 1-4 and LED 1,2 and 3 were manipulated to carry out the functionality of the digital voice recorder. While pushbuttons 1-3 were used for the functionality of the DVR, pushbutton 4 was used only for resetting the firmware. The functionality of the pushbuttons and what they do are given below:

- **Pushbutton S1: PLAY- LED1(green)**

- **Pushbutton S2: RECORD- LED2(red)**
- **Pushbutton S3: STOP- LED3(blue)**

A state diagram is attached to the **Appendix 02**, showing the functionality of the DVR.

3.6 Complete Schematic

The complete schematic of the entire DVR circuitry is given below. The netlist from LTSpice for entire circuitry is attached in **Appendix 03**

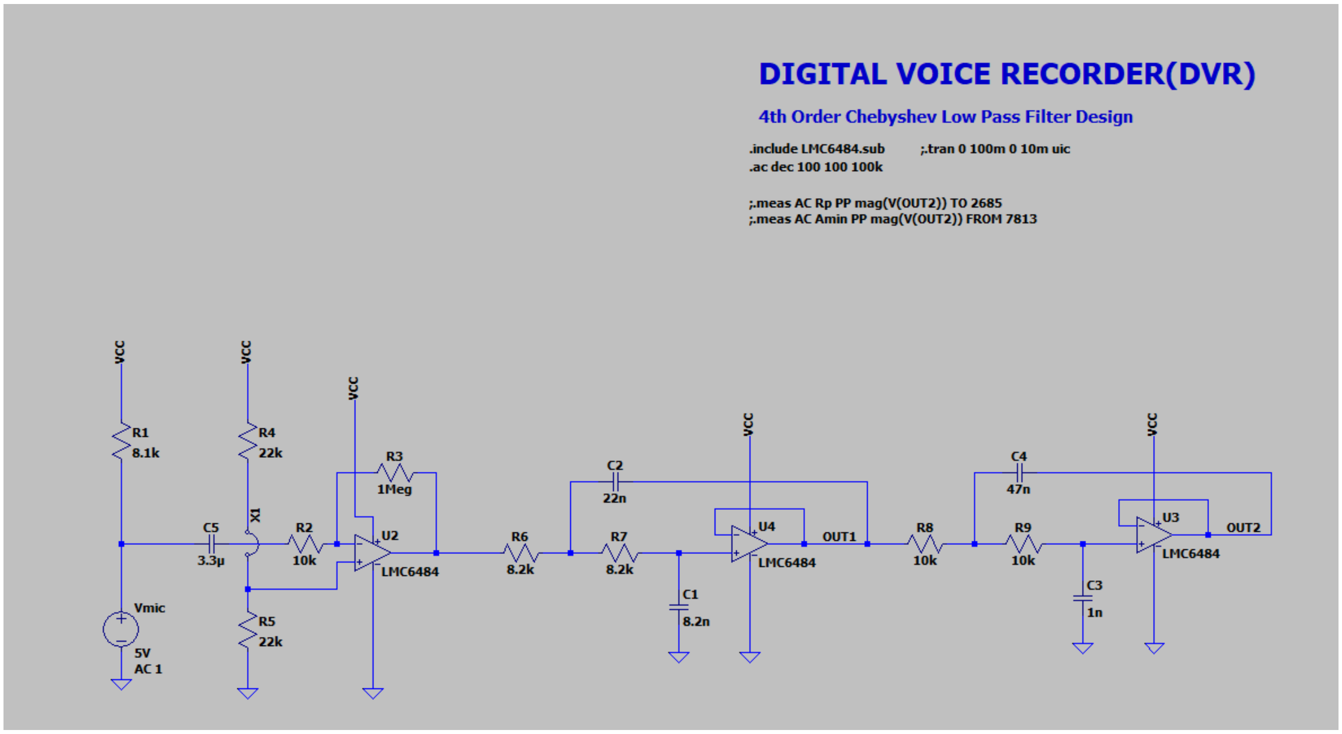


Figure 08: Schematic of the Digital Voice Recorder

4.0 Implementation

4.1 Hardware Implementation

As per the schematic in mentioned in figure 08 provided in the section 3.6, the physical components needed for implementing the analogue circuitry was sourced. The microphone amplification circuitry and the anti-aliasing filter was implemented using 3 gates of the provided LMC6484 op-amp IC. Gates 2,3 and 4 of the IC was used for implementing the analogue circuitry. A figure of the pin configuration of the LMC6484 is given below for a better understanding of the matter.

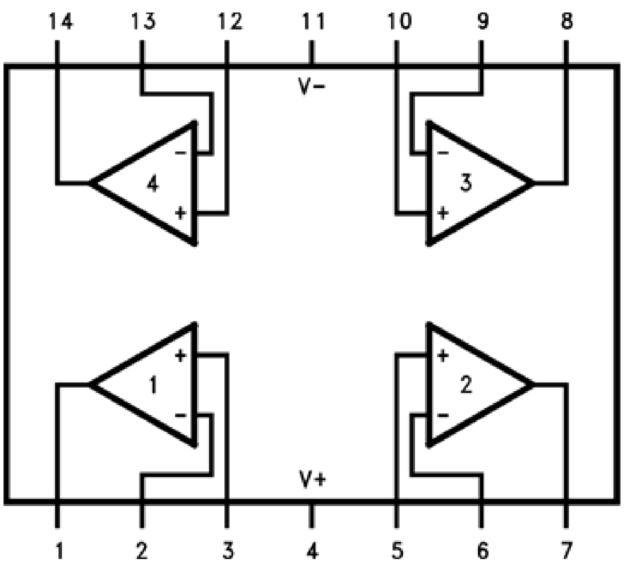


Figure 09: Pin configuration of LMC6484

- The microphone was implemented on the provided breadboard, which consequently connected to the microphone amplification circuit. The microphone amplification circuitry was implemented on the gate 2 of the IC. Two parallel resistors, each of 22kΩ was used for getting the reference voltage of 2.5V. The output of the amplification circuit then connects to the input of the 1st stage of the anti-aliasing filter.
- The 1st stage of the anti-aliasing filter was configured on the gate 4 of the LMC6484 IC. The output of this stage then connects to the input of the 2nd stage anti-aliasing filter.
- The 2nd stage of the filter circuitry was configured on the 3rd gate of the IC. Which finally connects to the J_{IN} port of the TeensyBOBv2.
- The provided headphone interface kit, soldered earlier, is connected to the J_{OUT} port of the TeensyBOBv2. And finally the J_{OUT} port is grounded, and finishing the hardware implementation of the DVR circuitry. A figure showing the Headphone interface kit is given below:

Figure 10: Headphone Interface Kit.

4.2 Software Implementation

The software part of this project was implemented and configured on the ATMega32u4 microcontroller, which is placed on the TeensyBOBv2.

- As mentioned earlier, programming language C was used for configuring the micro-controller. The codes required for operating was processed in the provided C file "EGB240DVR_Skeleton". AtmelStudio 7.0 was the IDE used for producing and configuring the codes.
- After building the solution from the code, a ".**hex**", was created, which was finally used by loading it on the micro-controller via Teensy loader.
- Another set of source code was produced in MATLAB for configuring the component values of the analogue circuitry. The MATLAB code is provided in the Appendix 01.

4.3 Prototype

Images showing the prototype of the DVR project from different angles are given below:

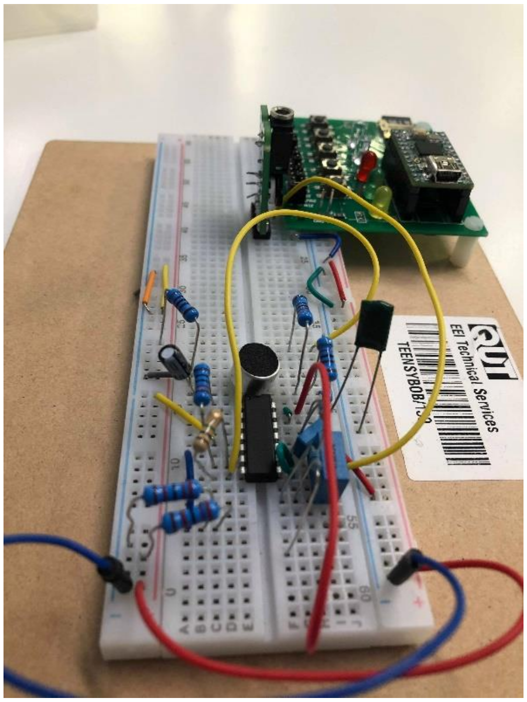


Figure 11: Implementation of the DVR prototype

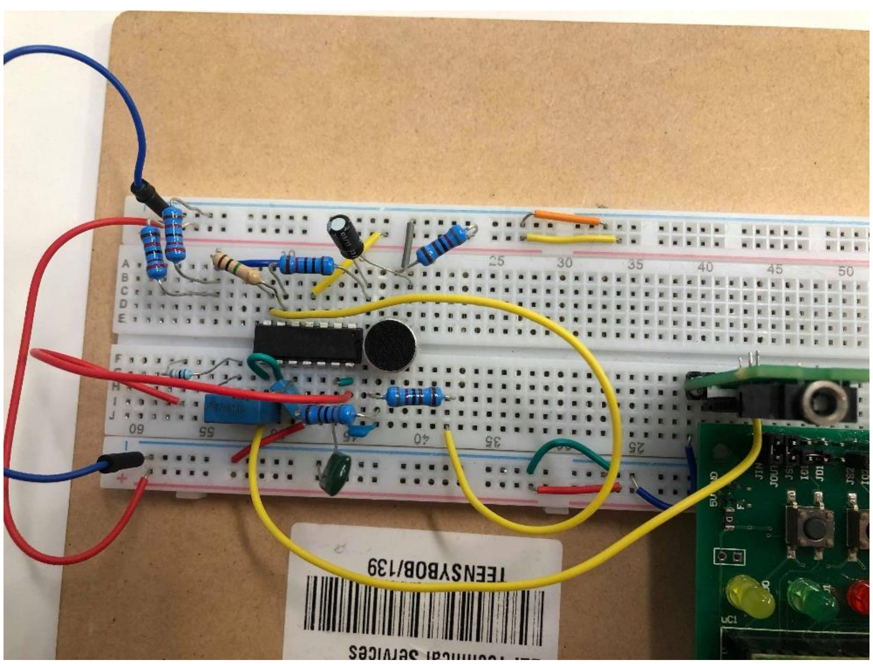


Figure 12: Prototype from a bird’s eye view

5.0 Experimental Results

5.1 LTSpice Simulation

The figure below shows the behavior of the amplifier circuit with a reference voltage of 2.5 as output.

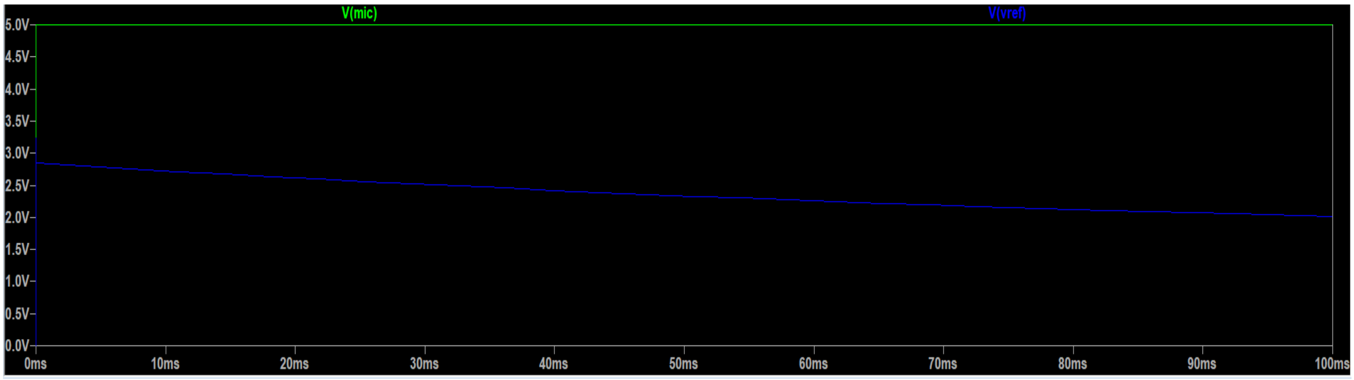


Figure 13: Behavior of amplifier circuit with a reference voltage of 2.5V.

The figure below shows the behavior of the anti-aliasing filter, showing a lower cut-off frequency of 2.6 kHz and a higher cut-off frequency of approximately 7.8125 kHz at -48dB.

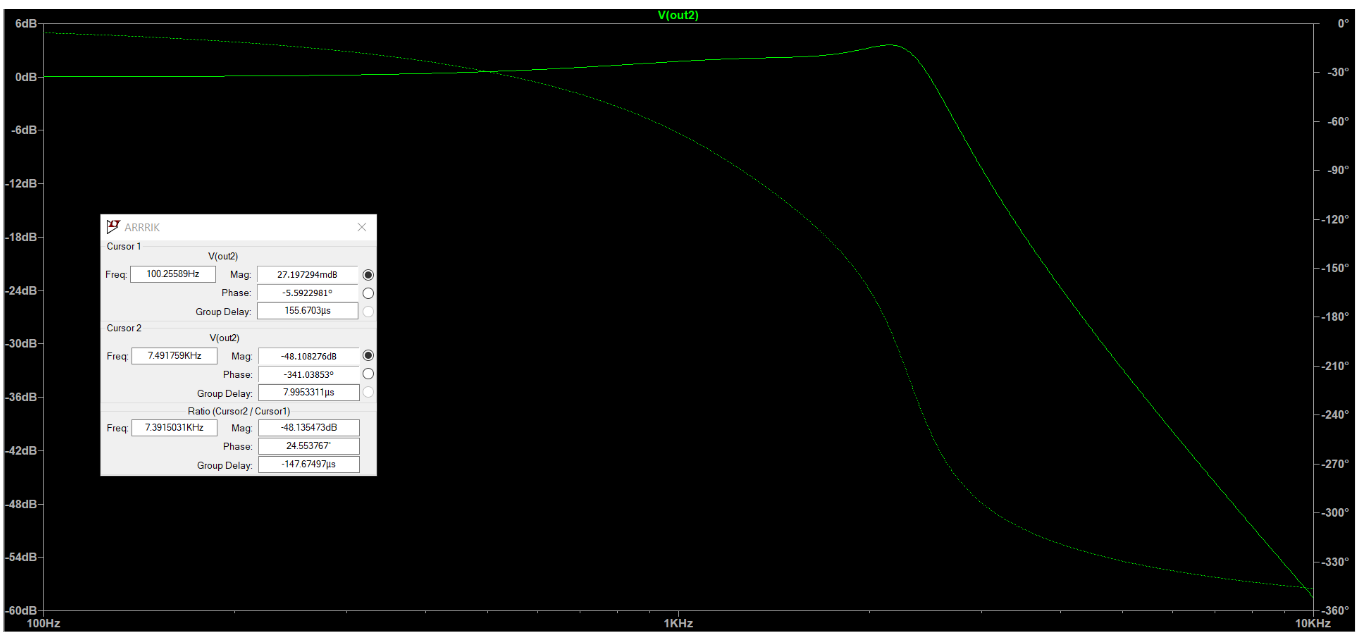


Figure 14: Behavior of the anti-aliasing filter

5.2 Oscilloscope Simulation

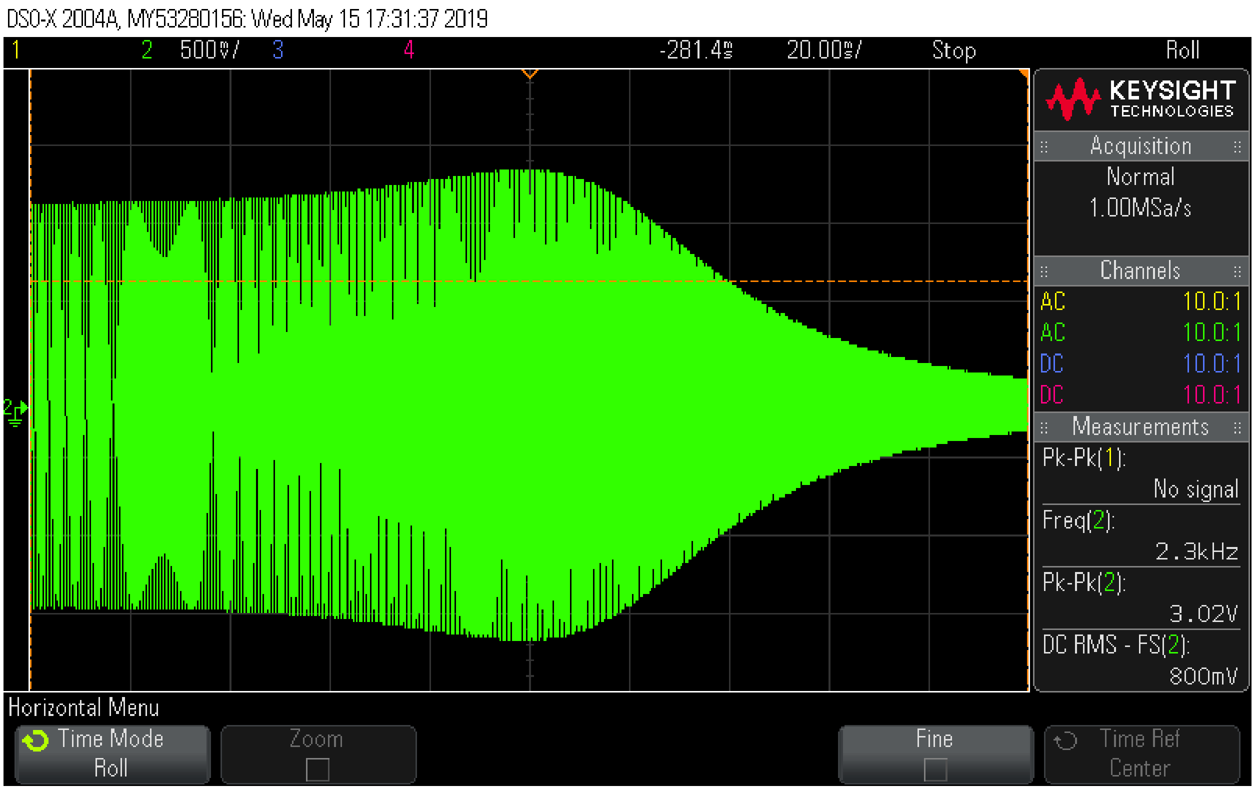


Figure 15: Output from the oscilloscope with a peak-peak voltage of 3.02V and frequency of 2.3kHz.

5.3 Analysis

Through thorough observations, it was noticed that the experimental results matched with the expected simulation results. Though there were limitations, but still the expected results were found.

6.0 Future Works

There were limitations and compromises along the way, in regard to available time period and preliminary conceptual knowledge. Even though the implementation and designing using the Sallen-key topology active filter was successful, still various factors of the project can be improved.

6.1 Implementation of Components

The DVR project could have been improved a lot more by using components of higher qualities. Because, random impedance and default errors could have been avoided by using higher quality components, which in return would also provide optimal performance from the DVR prototype. For example, ceramic capacitors and better resistors would have increase the reliability and presentation of the entire analogue circuitry. Lastly, superior components weren't being implemented due to financial and time constraints.

6.2 More Experiments and Simulations

As much as it was desired, more experiments and simulations of the circuitry of the project would have provided an even better output. Time constraint was the major reason behind this. For future work, it will be necessary, and more priority should be put towards undertaking more experiments and simulating with varying factors.

6.3 Cleaner and Tidier Prototype

For future work, developing a tidier prototype should be a key-factor. Even though the prototype would work just fine, the way it was assembled for this project, but in terms of presentation and understanding the circuitry just by viewing, it should have been tidier which includes trimming the components too.

7.0 Conclusion

The Digital Voice Recorder was designed and implemented in the most optimal and efficient way possible, by using the ideal components and configuration. It consisted of utilizing the provided components and brainstorming innovative ideas for designing. Both the analogue and the software side of this project has been tested, debugged, simulated numerous times through trial and error for producing the best possible outcome. All of the conceptual knowledge utilized and details of the modus operandi of the project has been outlined through out the report.

The prototype of the project starts with a microphone along with its amplification circuitry, connected in inverting op-amp fashion. The output from the amplifier circuit incorporates with the input of a 4^{th} order low-pass Chebyshev filter to sample input signals and for the functionality of anti-aliasing. The LMC6484 IC was utilized for implementing, both the amplification circuitry and the anti-aliasing filter. The output from the anti-aliasing is incorporated with the provided headphone interface kit via the ATMega32u4 micro-controller.

The micro-controller was functionalized for operating by developing C code. The code was required for converting the analogue input signals through the initial circuitry to a digital format. Codes were also implemented for manipulating the LEDs and pushbuttons, used for conducting operations in the micro-controller.

Reference

[1] L. Lovisolo, E. da Silva, *Signal Sampling.* Academic Press Library in Signal Processing. 2014.

Retrieved from: https://www.sciencedirect.com/topics/engineering/signal-sampling

obtained at: 24/05/19

[2] *Anti-Aliasing Filters and Their Usage Explained.* National Instruments. 2018.

Retrieved from: http://www.ni.com/en-au/innovations/white-papers/18/anti-aliasing-filters-and-their-usage-explained.html

Obtained at: 24/05/19

[3] S. Smith. *Chebyshev Filters.* Digital Signal Processing: A Practical Guide for Engineers and Scientists. 1st ed. 2013.

Retrieved from: https://www.sciencedirect.com/topics/engineering/chebyshev-filters

Obtained at: 25/05/19

[4] P. Horowitz and W. Hill, *The Art of Electronics 3rd Edition*, 3rd ed. Cambridge University Press, 2015.

Retrieved from: https://artofelectronics.net/wp-content/uploads/2016/02/AoE3_chapter9.pdf

Obtained at 25/05/19

[5] Lecture 7: Filter Design. EGB 240 Electronic Design, QUT. n.d.

Retrieved from: https://blackboard.qut.edu.au/bbcswebdav/pid-7877126-dt-content-rid-23769844_1/courses/EGB240_19se1/EGB240_Lecture_Week7_FilterDesign.pdf

Obtained at: 25/05/19

[6] B.Munir. *Voice Fundamentals – Human Speech Frequency.* 2012.

Retrieved from: http://www.uoverip.com/voice-fundamentals-human-speech-frequency/

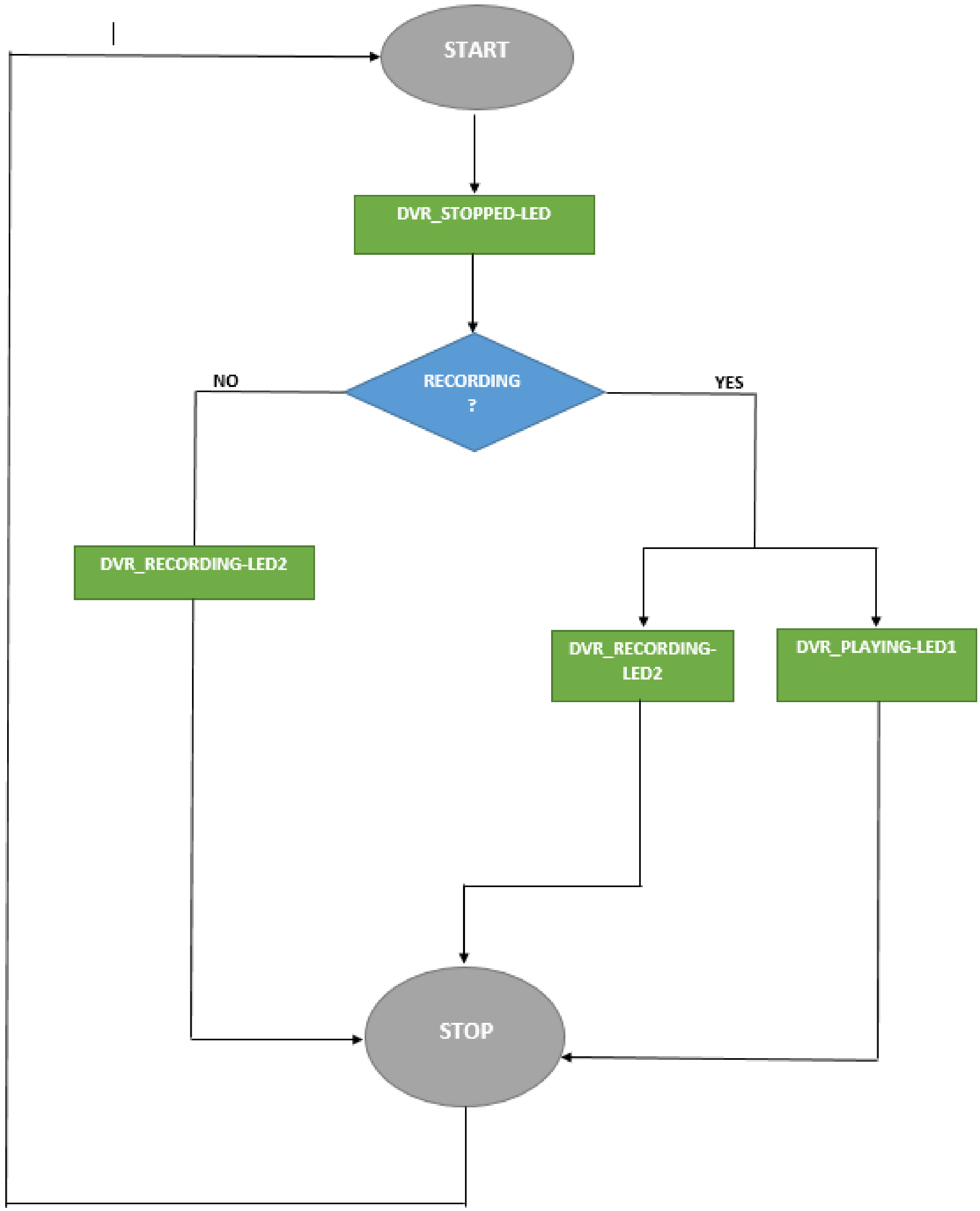
Obtained at: 26/05/19

Appendices

Appendix 01: MATLAB Source Codes.

```
%%EGB 240 Assessment 3
%ARIK INTENAM MIR- n9637567

%system specs-
fsamp = 15.625e3;
fs = fsamp/2; % stop band frequency,Hz
fp = 2800;% pass band freq, Hz

wp = 2*pi*fp;    %in radians
ws = 2*pi*fs; %in radians

Amin = 20*log10(1/2^8); %min. stopband attenuation
Amax = 1.5; %max stopband attenuation

%chebyshev requirements
[n_cheb, wn_cheb] = cheb1ord(wp,ws,Amax,Amin,'s');

%4th order chebyshev designing
[b, a] = cheby1(n_cheb,Amax, wn_cheb, 'low', 's');
H = tf(b,a)
bode(H)
 %converting from transfer functions to zeros and poles
[z,p,k]= tf2zpk(b,a)

%co-efficients for the first stage
c1_1st =2*(-real(p(1)));
c2_1st = real(p(1))*real(p(1)) + imag(p(1))*imag(p(1));

%co-efficients for the second stage
c1_2nd =    2*(-real(p(3)));
c2_2nd =real(p(3))*real(p(3)) + imag(p(3))*imag(p(3));

%NATURAL FREQUENCY AND QUALITY FACTOR FOR FIRST STAGE
wn2 = sqrt(c2_2nd)
Q2 = wn2/(c1_2nd)

%NATURAL FREQUENCY AND QUALITY FACTOR FOR SECOND STAGE
wn1 = sqrt(c2_1st)
Q1 = wn1/c1_1st

%%Calculation for Component Values of both stages
%STAGE 01
m = 1;
n= 2.9419;
R = 8.2e3; % resistor values for both resistors in 1st stage
C1 = 1/(wn2*R*(sqrt(n)))
C2 = n*C1

%STAGE 02
m = 1;
n= 2.9419;
R = 10e3; % resistor values for both resistors in 1st stage
C3 = 1/(wn1*R*(sqrt(n)))
C4 = n*C3
```

Appendix 02: State Diagram



Appendix 03: LTSpice Netlist

```
XU4 N009 OUT1 VCC 0 OUT1 LMC6484
R6 N004 N002 8.2k
R7 N009 N004 8.2k
C1 N009 0 8.2n
C2 OUT1 N004 22n
XU3 N008 OUT2 VCC 0 OUT2 LMC6484
R8 N003 OUT1 10k
R9 N008 N003 10k
C3 N008 0 1n
C4 OUT2 N003 47n
XU2 N005 N001 VCC 0 N002 LMC6484
R1 VCC N006 8.1k
C5 N007 N006 3.3µ
R2 N001 N007 10k
R4 VCC N005 22k
R5 N005 0 22k
R3 N002 N001 1Meg
Vmic N006 0 5V AC 1
.include LMC6484.sub
;.tran 0 100m 0 10m uic
.ac dec 100 100 100k
;.meas AC Rp PP mag(V(OUT2)) TO 2685
;.meas AC Amin PP mag(V(OUT2)) FROM 7813
* DIGITAL VOICE RECORDER(DVR)
* 4th Order Chebyshev Low Pass Filter Design
.backanno
.end
```