Roll No. 23110027, 23110028, 23110029

ES116:ELECTRICAL PROJECT

DSP SIMULATOR AND TRAINER

Animesh Behera, Anirudh, Anirudh Mittal
IIT Gandhinagar

Abstract— Our project in the course - "ES116: Principles and Applications of Electrical Engineering" involves creating a Digital Signal Processing (DSP) Simulator, which is based on the Arduino platform and aims to help learners understand the complex field of DSP. This project results from our dedication to combining theoretical understanding with practical application, enabling learners to directly explore and experiment with DSP algorithms. Learners will thoroughly understand filters, echoes, and Fourier transforms, among other concepts, by tinkering with real-time audio signals. We aim to provide a fully immersive learning experience where these essential DSP concepts are understandable and engaging. With this project, we strive to improve further the instructional environment in the field of electrical engineering.

INTRODUCTION

A necessary part of present-day electronic and communication systems is digital signal processing. The commitment and real-world experiences required to completely understand complicated DSP topics are sometimes absent from theoretical knowledge. This project presents an Arduino-based DSP Trainer that aims to fill these gaps in education by giving students a valuable tool for interactive learning that will increase their understanding and engagement with DSP topics.

MOTIVATION

The necessity to improve the practical learning elements of DSP education served as the primary incentive for the project. The Arduino-based DSP Trainer was designed to give students a hands-on platform to actively experiment with DSP algorithms and observe the impacts in real-time on audio signals, leading to a greater comprehension of the subject matter and an encouragement of creative thinking and exploration.

COMPONENTS

- Arduino Uno
- Condenser Microphone
- LM386 Audio Amplifier Module
- LM358 Audio Amplifier Module
- 4 ohm Speaker
- Breadboard
- Jumper Wires
- Push Buttons
- 10kΩ Potentiometer
- Assorted Resistors and Capacitors
- DC Adapter

METHODOLOGY

1. Design and Planning: Detailed sketches were drawn to depict the circuit and component interactions.
2. Assembly of Components: Components were placed on the breadboard using the design diagrams as a guide.
3. Programming and Configuration: Arduino manages DSP algorithms and user inputs from the push buttons through programming.
4. System Testing and Validation: The constructed system underwent a thorough testing process to ensure that all parts worked as they should and that the DSP algorithms handled the audio signals appropriately.

CIRCUIT ASSEMBLY

1. Power Configuration: The Arduino receives power from a 12-volt DC adapter and delivers a controlled voltage of 5-volt to other parts.
2. Audio Input Configuration: The output of the condenser microphone is fed into the Arduino's analog input through a pre-amplifier circuit. This pre-amplifier circuit uses particular resistors and capacitors of 1 kilo-ohm and ten nanofarads to match the Arduino's input parameters with the microphone's output impedance and voltage level.
3. User Interface: An individual digital pin on the Arduino is connected to each of the three push buttons. The potentiometer attached to these buttons enhances the user's ability to smoothly transition between various DSP filters by providing changeable resistance.
4. Signal Processing and Output: A PWM signal's output allows the processed signal to leave the Arduino. It then travels through a passive low-pass filter to convert it to an analog signal. It is then amplified by the LM386 module before output through speaker

THEORY

The DSP algorithms used are essential to audio signal filtering:

1. A low-pass filter (LPF) decreases frequencies over a predetermined cutoff while maintaining lower frequencies using an easy-to-understand first-order filter. The following formula determines the cutoff frequency and efficiency:

$$y[n]=\alpha \cdot x[n]+(1-\alpha)\cdot y[n-1].$$

The equation of the coming signal and output relation.

For first-order filters like this, a more accessible approximation relates alpha to the sampling period (T = 1/Fs) and the RC time constant (τ) of an analog filter:

$$\alpha = \frac{T}{T+\tau}$$

For a simple IIR filter like this, the relationship between alpha and the cutoff frequency (Fc) in Hertz can be approximated by considering the sampling rate (Fs) of the input signal. The formula to approximate the cutoff frequency is derived from the discrete-time implementation of an RC low-pass filter,

$$Fc = \frac{Fs}{2\pi} \times \tan^{-1}(2\pi \frac{Fc}{Fs})$$

And since the cutoff frequency of the analog low-pass filter is given by:

$$Fc = \frac{1}{2\pi\tau}$$

Therefore:

$$Fc = \frac{Fs\,\alpha}{2\pi(1-\alpha)}$$

Where the sample rate and theoretical cutoff frequency are used to calculate the required frequency response, which determines the value of α.

2. High-Pass Filter (HPF): This filter has a construction similar to the LPF but is designed to attenuate lower frequencies and permit high frequencies to pass. It is essential for identifying edges or unexpected shifts (noise magnifiers) in the audio stream.
3. Band-Pass Filter (BPF): This filter, intended to let only a particular band of frequencies pass, is essentially a combination of LPF and HPF. It is especially helpful for extracting frequencies from a complex audio source that falls into a specific band.

PROCEDURE

1. Circuit Assembly: Precisely arranging and connecting parts on the breadboard.
2. Programming: Implementing DSP algorithms on the Arduino and setting them up to react to user inputs for filter selection. The code that was written and debugged is attached here. [code file.txt].
3. Operational testing is performed to verify functioning and is adjusted in response to performance data.

CIRCUIT DIAGRAM

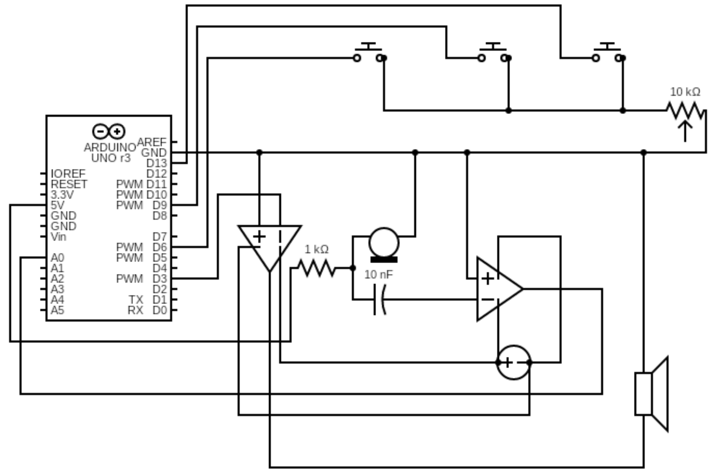

Fig 2: The circuit diagram

RESULTS AND DISCUSSIONS

Testing showed that audio recordings are successfully modified by the DSP Trainer according to the selected DSP filters. The audio outputs validated the system's efficiency, revealing each filter's effects.

OUTCOMES

The DSP Trainer effectively achieved its teaching objectives by augmenting students' practical knowledge of DSP and igniting their curiosity for additional research in the domain.

CONCLUSION

The DSP Trainer, which is built on Arduino, has been shown to be a useful teaching tool for explaining the practical applications of DSP. It makes abstract theoretical ideas more approachable and understandable by simplifying them.

ACKNOWLEDGEMENT

We express our gratitude to Prof. Arup Lal Chakraborty, Department of Electrical Engineering and Electrical Lab, along with our TA, Mr. Somen Sarkar, for their unwavering support and guidance throughout this project.

FINAL PROJECT PICTURE

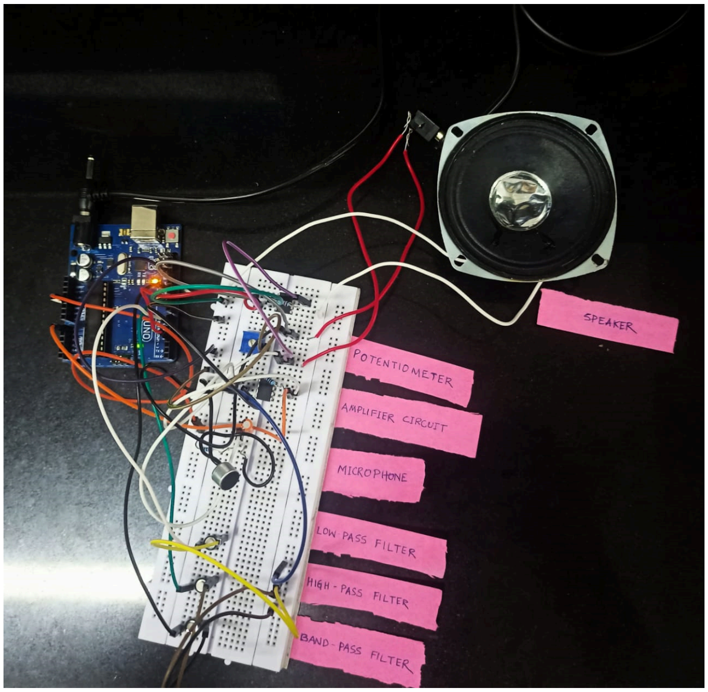

Fig 1: The project picture of the final prototype

REFERENCES

1. "Introduction to Digital Filters" by Julius O. Smith III, available online at https://ccrma.stanford.edu/~jos/filters/
2. Arduino Official Documentation: https://www.arduino.cc/reference/en

***