Jurnal NASIONAL SAINS & TEKNIK

Jurnal Nasional SAINS dan TEKNIK
Vol. 2, Issue 1, pp. 1–8 (2024)
doi: http:/doi.org/10.25124/jnst.v2i1.7452

RESEARCH ARTICLE

Design and Implementation of Sound Synthesizer for Arduino

Wayan Richie A.W., Nyoman Bogi Aditya Karna* and Arif Indra Irawan

Fakultas Teknik Elektro, Universitas Telkom, Bandung, 40257, Jawa Barat, Indonesia
*Corresponding author: aditya@telkomuniversity.ac.id
Received on 16 April 2024; accepted on 19 May 2024

Abstrak

Signal instruments are musical instruments used not only for music but also to provide signals as a form of auditive communication. Along with the development of civilization,signal instruments are now digitized, and many instruments are popping up, for example, megaphones, one of which is in the future. Arduino is a micro-controller device often asked to be able to communicate with speakers and become a signal instrument that can output sound signals. However, because the signal generated by Arduino is digital, it must be converted into an analog signal first by converting the digital signal generated by Arduino into an analog signal using a DAC circuit. After converting the signal into a digital signal, it will be forwarded to the amplifier, which uses a transistor BC337 as a switch to make Arduino a signal instrument. The system design is based on Arduino nano as a data processing tool, where the data stored in Arduino nano will be forwarded to the DAC circuit and then will be returned to the amplifier before finally being channeled to the speakers to produce a sound output that matches the data stored on the Arduino nano

Key words: Arduino, Signal Instrument, DAC, Amplifier

Introduction

Science and technology are developing very rapidly, creating innovations that evolve towards more perfect products. In this modern era, technology is inseparable from everyday life. Technological developments can be integrated into several aspects, one of which is technology in the electronic field. The use of electronic objects is currently a necessity for humans. One of these electronic devices is the synthesizer.

The synthesizer is concerned with designing a system that can read and produce artificial sounds. There are three processes involved in the synthesizer: the reading process, the sound production process, and designing a system that can read and emit artificial sounds [1]. Synthesizers can create various types of sounds, including the sound of musical instruments and even sounds that cannot be created by conventional musical instruments. Historically, synthesizers have been an important part of the development of electronic music, giving musicians the ability to produce more complex and innovative sounds and songs. In the 1970s through to the 1980s, some new instrument designs used newly introduced techniques, some incorporating analog sounds. Some of these digital synthesis technologies were more cryptic than analog. In an effort to master them, many musicians were forced to ignore analog sounds, leading to a generation of musicians by the mid-1980s who had as much difficulty mastering analog sound synthesis as those two years earlier [2].

Synthesizers work by generating electronic sound waves controlled by parameters or settings, such as frequency, amplitude, and waveform. These waves are then converted into sounds heard through loudspeakers or speakers. Synthesizers generate electronic sound waves through complex electronic circuits, which can be modified, combined, and adjusted to produce the desired sound [3].

There are several types of synthesizers, namely analog synths, digital synths, and software synths. Analog synths use analog circuits to produce sound, while digital synths use digital circuits. Software synths, as the name suggests, run on a computer device through special software, for example using a microcontroller called Arduino. Arduino is an open-source physical computing platform based on a simple input and output (I/O) circuit and development that implements a processing language. Arduino can be used for independent interactive object development or connected to software on a computer [4]. One of Arduino's advantages is that its software is open-source, meaning anyone can use it without needing to pay. Therefore, in this final project, I designed a synthesizer for Arduino that aims to create artificial sounds.

Basic Concept

Arduino Nano

Arduino is an open-source platform used for electronic construction and programming. Arduino can receive and send information to most devices, even over the internet, to control certain electronic devices. Arduino uses a circuit board called Arduino Uno and a software program (Simplified C++) to program the board [5]. On Arduino hardware,

Gambar 1. Digital to Analog Converter

there are a number of pins that function as digital and analog input/output (I/O) that can be programmed to interact with sensors, actuators, and other electronic components. The software on Arduino includes a development environment (IDE) for writing programs and drivers used for connection to computer devices [6].

Currently, Arduino is widely used in microcontroller programming due to its easy-to-use settings. Arduino is a circuit board with a chip that can be programmed to perform many instructions. Arduino sends information from the computer program to the microcontroller and ends up at a specific circuit to execute the instructions. Arduino can help to read information from input devices such as sensors, antennas, potentiometers, etc., and can also send information to output devices such as LEDs, speakers, LCD screens, DC motors, and more [5].

Digital to Analog Converter

A Digital to Analog Converter (DAC) is an electronic device that converts digital signals into analog signals. A digital signal is a series of bits with a binary value of 0 or 1, while an analog signal is a continuous signal with a magnitude that varies within a certain range [7]. For example, in sound signal processing, the sound information generated from the microphone is initially analog, but in digital processing, this sound needs to be converted into a digital signal first. After processing in digital form, the sound signal needs to be converted back into an analog signal before it is output as sound through the speakers.

A DAC takes a series of digital inputs (a series of 1s and 0s, e.g., numbers like 10011001) and converts them into analog outputs. The DAC in any digital audio device (MP3 player, CD player) is crucial since they all store sound in digital form but need to drive the speakers with analog signals. Therefore, it is necessary to convert digital data into analog signals.

Amplifier

An amplifier is an electronic component composed of resistors, diodes, and transistors. The arrangement of the amplifier is made in an integrated circuit or what is called an Integrated Circuit (IC). In its application, the amplifier functions as a signal amplifier [8]. In the amplifier circuit, there are two inputs: inverting and non-inverting inputs. There are also two input sources as the power source of the amplifier, namely positive voltage (+Vcc) and negative voltage (-Vee). Amplifiers are widely used as signal amplifiers, both for linear and non-linear signals, especially in control systems, instrumentation, and analog computing [9].

Amplifiers can change the signal from one level to another, where the signal could be a voltage or current signal [10]. Many amplifiers on the market are in the form of integrated circuits (IC). Applications of amplifiers include inverting circuits, integrators, and differentiators. Amplifiers are also called differential amplifiers; they include a gain stage, a level shifter circuit, and a final amplifier.

Properties of Amplifiers

The following are the properties of an amplifier:

1. Input voltage difference (Vdm) = 0
2. Amplifier input current (ia) = 0
3. Open loop amplifier (AVOL) is infinite
4. Open loop output resistance (Ro,ol) = 0
5. Open loop input resistance (Ri,ol) is infinite
6. Common Mode Rejection Ratio (CMRR) is infinite

Amplifiers are used for preamps, mixers, buffers, or oscillators, depending on the circuit configuration. LM386 type amplifiers are usually used with additional capacitors and resistors to improve performance and reduce noise and interference.

Speaker

Speakers are transducers that convert electrical signals to audio frequencies by vibrating components to vibrate the air around them, producing sound waves that can be heard by the human ear. Speakers carry electrical signals and convert them back into vibrations to create sound waves [11]. Speakers are essential components of other electronic devices. Active speakers are widely used in homes, houses of worship, offices, and shops. Currently, active speakers come with several wireless communication features to facilitate the process of sending the sound to be generated [12].

Principle of Sound Signal Amplifier for Arduino

The basic principle of a sound signal amplifier for Arduino is to amplify audio signals using an operational amplifier or op-amp. An op-amp is an electronic circuit that has the ability to amplify electrical signals with adjustable gain. To build a sound signal amplifier with Arduino, an op-amp can be used as an audio signal amplifier. The amplifier has three terminals: inverting input (-), non-inverting input (+), and output.

With the audio amplifier circuit, the audio signal will be amplified; the audio signal coming from a small signal source can vibrate the speaker membrane to a certain level as needed by adjusting the potentiometer [13].

MATLAB

MATLAB is a high-level programming language and interactive environment that is widely used in the fields of science, engineering, and mathematics. MATLAB is an acronym that stands for "Matrix Laboratory," indicating the program's primary concentration on matrix computations and linear algebra. MATLAB, created by MathWorks, is a complete suite of tools for data analysis, visualization, and algorithm development. MATLAB's capacity to efficiently manipulate matrices and arrays is one of its core strengths, making it appropriate for applications involving massive volumes of numerical data. It includes a large number of built-in mathematical functions and operators, allowing users to easily do complex calculations. MATLAB also allows you to create and manipulate multidimensional arrays, which allows you to handle massive datasets and image processing jobs more efficiently.

The MATLAB environment includes an interactive command-line interface known as the MATLAB Command Window, which allows users to perform commands and evaluate expressions in real time. It also includes the MATLAB Editor, a robust integrated programming environment (IDE) with features such as syntax highlighting, debugging tools, and code profiling.

Toolboxes, which are add-on modules that provide specialized capabilities for certain applications, can be used to extend the functionality of MATLAB. These toolboxes span a wide range of fields, including signal processing, control systems, image and video processing, optimization, and other topics. They improve MATLAB's capabilities and enable users to more successfully tackle challenging challenges in their respective professions.

MATLAB, with its simple syntax and abundant documentation, provides a user-friendly environment for both novice and professional programmers. It is commonly used in academia, industry, and research institutes for data analysis, simulation, algorithm development, and prototyping. The variety and robustness of MATLAB make it a popular choice among scientists, engineers, and researchers all over the world.

Gambar 2. Quality of Experience

Quality of Experience (QoE)

Quality is the level of excellence of something. In the network communication ecosystem, there are many approaches to ending "quality." However, in the vast majority of digital multimedia entertainment applications, the main concern and really focus on the overall QoE (Quality of Experience) perceived by the end user. QoE represents the quality of the video, the audio quality of the audio, the quality of the combined audiovisual content, or the quality of interaction with a particular audiovisual service. Audiovisual services are for people. Today, the traffic of audiovisual content over the Internet as well as the number of TV channels appearing in recent years is very large, not to mention the OTT (Over the Top) service and media platforms that offer a large catalog of series, films and documentaries and their business model (in most of the following cases) is conditional on a monthly subscription. As a result, end users expect more of the quality of these content and this indicates the success or failure of a given service.

How to measure the quality of an audiovisual content is not simple, mainly because it is governed not only by an objective measurement but also by subjective judgment, and this is difficult to determine. In any audiovisual sequence, video and audio are degraded during acquisition, compression, transmission, processing, and viewing. Distortion directly affects the final quality perceived by the end user. Typical degradation, in the case of video, is contrast or color issues due to the nature of the scene, blurring, blocking, loss of bitrate due to encoding, packet loss, and internal transmission delay. dung, etc. Regarding QoE, understanding Quality of Experience as a subjective quantitative metric of end-user satisfaction with a particular AV service, there are two ways to measure it: subjective quality method or objective quality method 2.

Here in this project author will use Objective Quality Method to assess the Quality of Experience of the audio using RMSE and PSNR as parameter 3.

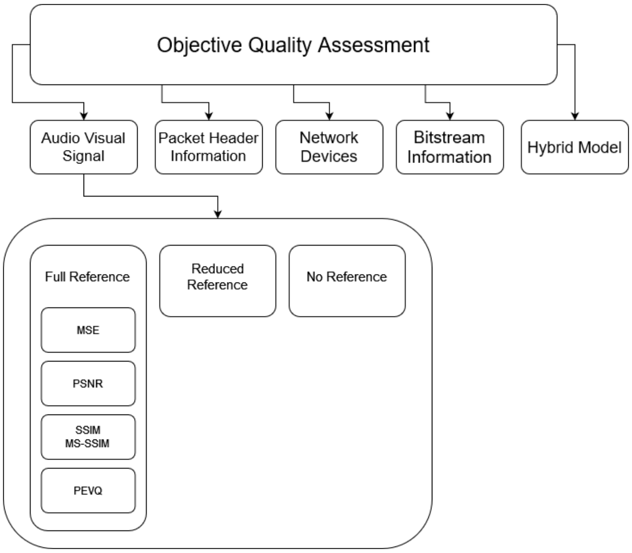
Root Mean Square Error (RMSE)

The Root Mean Square Error (RMSE) is a statistical measure that quantifies the average difference between anticipated and observed values to assess the accuracy of a prediction or estimator. It is widely used in domains such as statistics, data analysis, machine learning, and signal processing. The RMSE formula is shown below in Equation **??**.

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\text{Predicted}_i - \text{Actual}_i)^2} \qquad (1)$$

The square root of the mean of the squared discrepancies between the anticipated and observed values is used to calculate the RMSE. The following steps are involved in this process:

1. Calculate the difference between each expected value and the observed value.
2. Each of these discrepancies must be squared.
3. Make an average of the squared differences.



Gambar 3. Objective Quality Assessment

4. To get the RMSE, take the square root of the average.

The RMSE measure returns a single numerical value that represents the overall difference between anticipated and observed values. Due to the squaring process, it gauges the size of the prediction mistakes, emphasizing larger errors over smaller ones. A lower RMSE number suggests more accuracy, while a value of 0 denotes perfect forecast alignment.

The root mean square error (RMSE) is a popular evaluation metric for regression models in which the goal is to estimate a continuous variable based on input characteristics. It enables model developers to compare and evaluate different models or techniques. The RMSE metric can be used to obtain insight into the average magnitude of prediction mistakes and make informed decisions about the model's predictive capabilities.

It should be noted that RMSE is sensitive to outliers, as their squared differences might have a disproportionate impact on the overall error. As a result, when analyzing the results, it is advisable to assess the RMSE value in conjunction with other evaluation metrics and to consider the specific characteristics of the situation at hand.

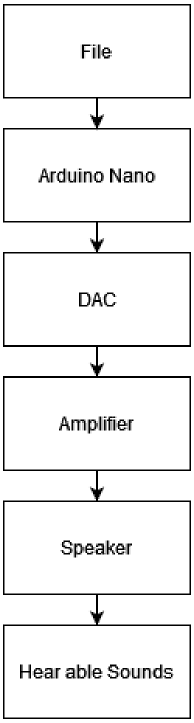
The RMSE measure returns a single numerical value that represents the overall difference between anticipated and observed values. It emphasizes larger errors due to the squaring process. A lower RMSE indicates more accuracy, with a value of 0 denoting perfect alignment.

RMSE is commonly used for evaluating regression models and helps in comparing different models or techniques. It is sensitive to outliers, and their squared differences can disproportionately impact the overall error. Therefore, it's advisable to assess RMSE alongside other metrics and consider the specific characteristics of the data.

Peak Signal-to-Noise Ratio (PSNR)

Peak Signal-to-Noise Ratio (PSNR) is a widely used metric in image and video processing to evaluate the quality of a reconstructed or compressed signal compared to the original uncompressed version. PSNR is expressed in decibels (dB) and is calculated using the Mean Squared Error (MSE) between the original and reconstructed (or compressed) signals. The PSNR formula is shown in Equation 2.

$$\text{PSNR} = 20 \cdot \log_{10}\left(\frac{MAXValue}{MSE}\right) \qquad (2)$$



Gambar 4. General Design Block Diagram

Where:

- MAX is the maximum possible pixel value of the signal.
- MSE is the Mean Squared Error between the original and reconstructed (or compressed) signal.

PSNR measures how much distortion or error is introduced during compression or reconstruction. A higher PSNR value indicates a better quality representation, as it signifies less distortion compared to the original signal. However, PSNR has limitations and may not always correlate well with perceived image quality, especially for complex images or videos. Other measures like the Structural Similarity Index (SSI) are often used in conjunction with PSNR to provide a more comprehensive assessment of image quality.

Model and System Design

General Design

This project aims to use Arduino to produce sound by inputting files into Arduino using an amplifier to forward the data to a speaker. The process is as follows: sound data in the form of a file is sent to the Arduino Nano. The signal is then processed and forwarded to the Digital to Analog Converter (DAC). The sound signal is amplified by an amplifier and output through the speaker. If the sound is audible, the project is considered successful. The sound generated by Arduino can be customized, allowing for future development for various purposes.

Specification of the Tools

Arduino Nano
The Arduino Nano specifications are listed in Table I.

LM 386 Amplifier
The LM 386 amplifier specifications are listed in Table II.

Table 1. Arduino Nano Specification

Microcontroller	ATmega328P
Operating voltage	5V
Input voltage (recommendation)	7-12V
Input voltage (limit)	6-20V
Digital I/O pins	14
Analog I/O pins	8
Maximum current (each I/O pin)	40 mA
Maximum current (through 3.3V pin)	50 mA
Flash Memory	32 KB
SRAM	2 KB
EEPROM	1 KB
Clock frequency	16 MHz

Table 2. LM 386 Amplifier Specification

Operating voltage	4-12V
Output power	0.25-1W, depending on the 9V input model
Load impedance	8 Ohms
Frequency range	300 Hz - 100 kHz
Gain	20 to 200 times
THD	$<$0.2% at 125 mW power and 8 Ohm impedance
Number of pins	8 (dual in-line package)
Operating current	$\pm$10 mA

Speaker
An 8-ohm speaker specifications are listed in Table III.

Test Parameters

The project focuses on sound produced by Arduino Nano for various purposes. The Arduino Nano acts as a file storage and processes the data, converting it into digital signals. With the DAC, the digital signal is converted to something akin to an analog signal, which can then be processed by the speaker. An amplifier is also essential to ensure clear sound output. Successful completion of the project is confirmed if the inputted sound can be heard from the speaker.

System Design

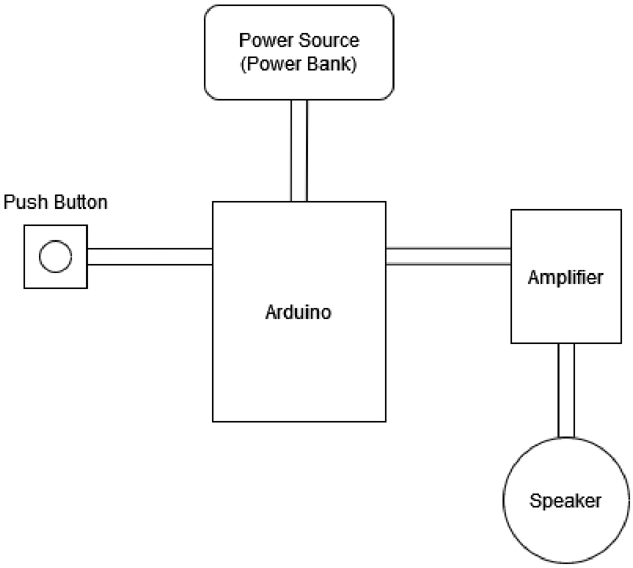
The system design involves creating an instrument based on Arduino, connecting it with several other devices: a push button to trigger the loop, an amplifier to amplify the signal, and a speaker for output. When the push button is pressed, the instrument runs its sequence, culminating in the speaker producing an output. The system design is shown in Figure 5.

Program and Flowchart

Based on the program attached in the appendix, the flowchart of the program is shown in Figure 6. The program utilizes libraries such as ‘XTDACAudio.h‘, ‘OneButton.h‘, and ‘Sound1.h‘, where ‘Sound1.h‘

Table 3. 8 Ohm Speaker Specification

Impedance	8 ohms
Power handling	1 Watt
Frequency response	20 Hz (minimum)
Sensitivity	85 dB
Cone diameter	2 inches
Number of drivers	Multiple drivers with 8 ohm impedance



Gambar 5. System Design

represents the audio file. All libraries used in this project are included in the appendix.

Supported Audio File Type

This project uses .wav audio file format, as it allows the audio file to be converted to unassigned 8-bit PCM. This format facilitates copying the audio variable figure into C code to be moved into the Arduino IDE.

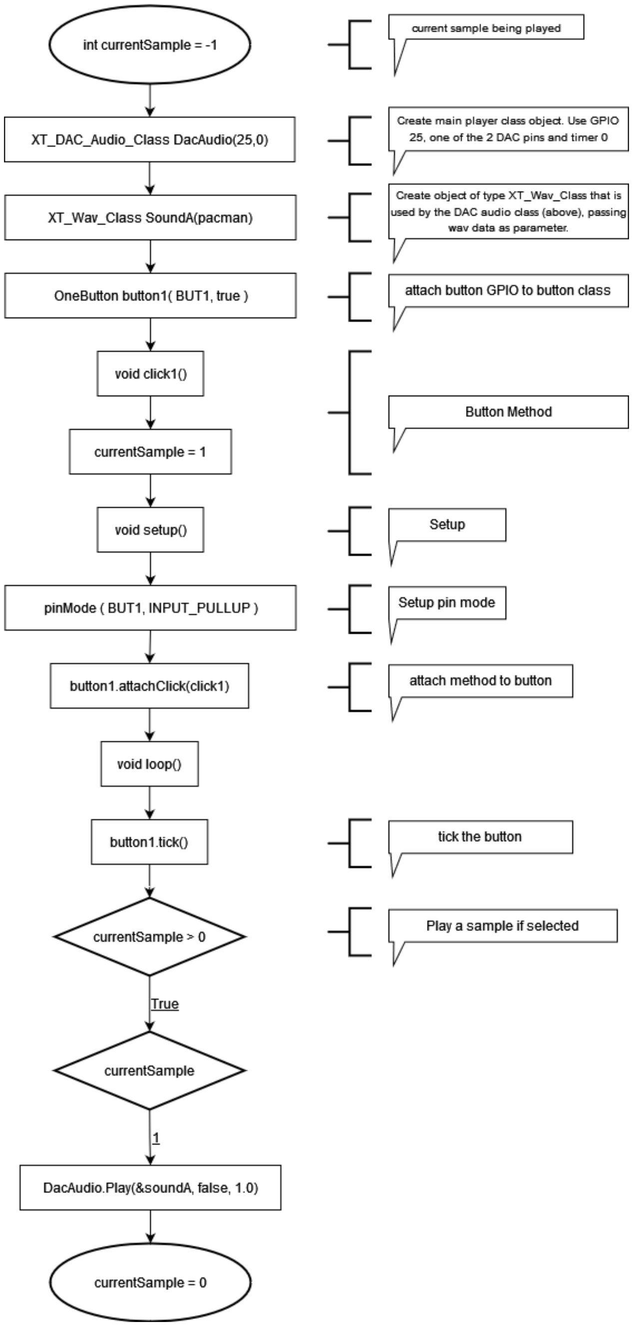
Test Scenario

The test scenario involves the following steps:

1. Prepare the necessary tools.
2. Assemble the Arduino Nano along with the DAC circuit, Amplifier, and Speaker.
3. Input the audio file into the Arduino Nano.
4. Run the system; successful generation of audio indicates completion of the test.

Instrument Audio Record

After the instrument produces audio by converting the original audio, the processed audio can be recorded and compared with the original audio using RMSE. Both the processed and original audio must have the same length to ensure they are comparable when compiled into number sequences. The recorded sound is shown in Figure 7.

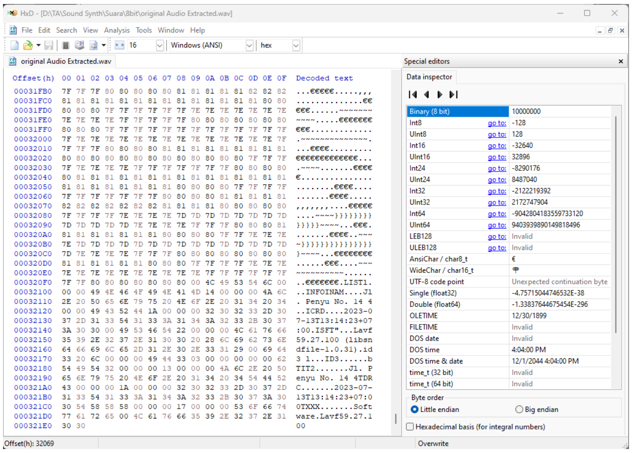


Gambar 6. Program Flowchart

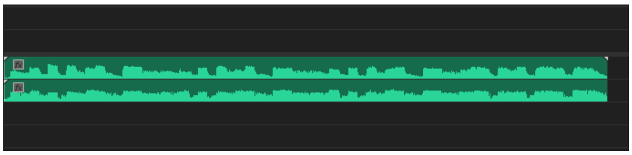
RESULTS AND ANALYSIS

Comparing Original Sound with Processed Sound

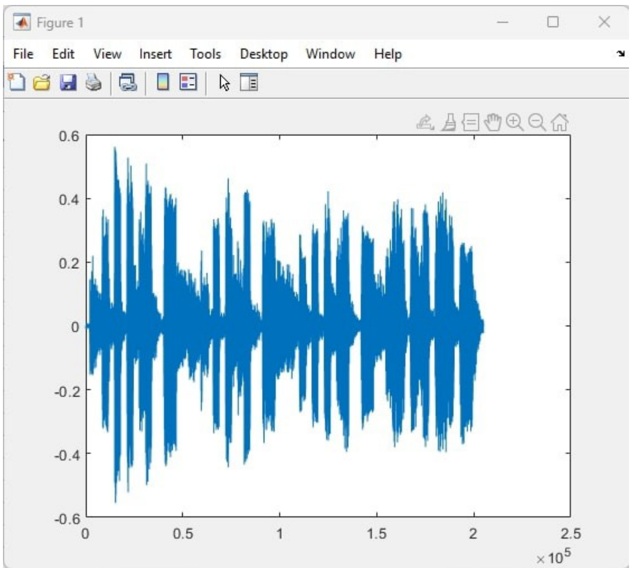
After processing the input file into the instrument, we now have the instrument producing sound that we can compare to the original sound. We determine the similarity between the two sounds using the RMSE method. We compile the sound variables using MATLAB. After compiling both sounds, we compute the RMSE to evaluate the accuracy of the instrument. If the RMSE value is under 1, it means the instrument successfully transmits the sound without distortion that can alter the sound phase.



Gambar 7. Audio File Variable



Gambar 8. Captured Sound with Original Sound (Figure 8).

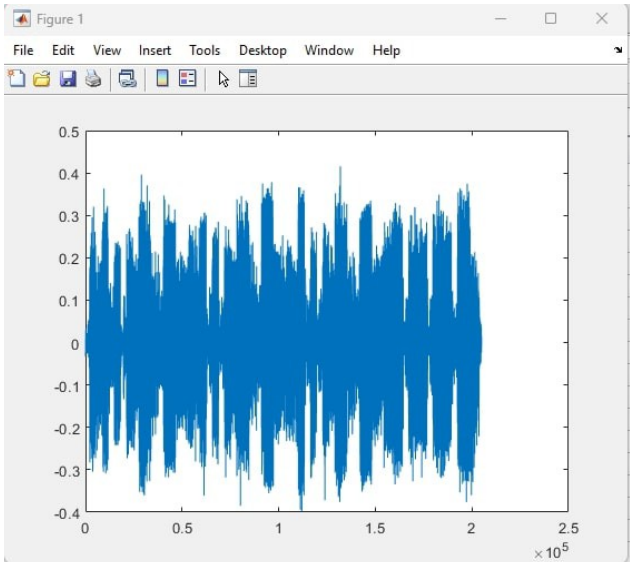


Gambar 9. Original Sound Plotting (Figure 9).

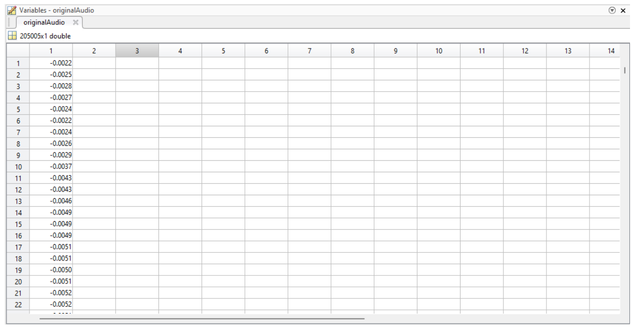
Plotting and Compiling Variables

Figure 9 shows the original sound plotting, which we compare to the processed sound plotting shown in Figure 10. From the plots, it appears that the wave shapes are quite similar because these two sounds are essentially the same. The difference lies in one sound being processed into a synthesizer instrument where there is a DAC to convert the digital signal to analog, resulting in slight distortion, while the other sound is not.

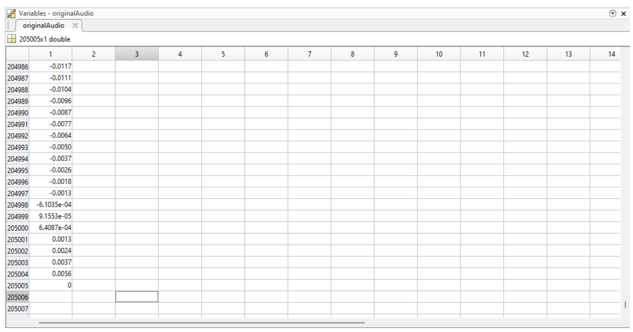
After plotting the sound signals, we examine their variables. The sound variables, shown in Figures 11 to 14, are processed using MATLAB and each sound has 205005 lines in its variable, indicating they are comparable.



Gambar 10. Processed Sound Plotting (Figure 10).



Gambar 11. Original Sound Variable 1 (Figure 11).



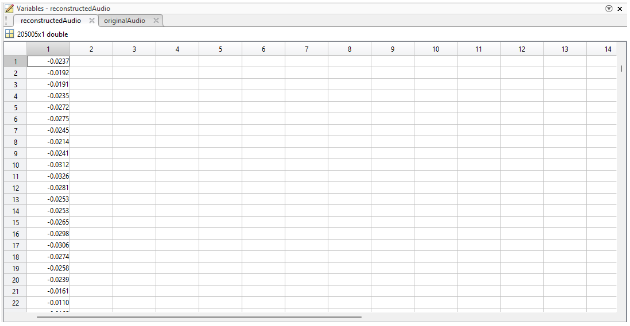
Gambar 12. Original Sound Variable 2 (Figure 12).

Quality of Experience

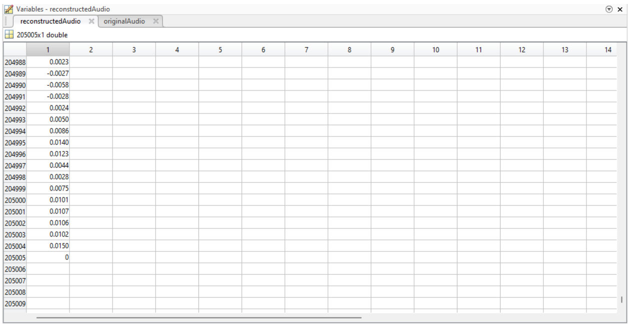
The Quality of Experience (QoE) is assessed objectively using methods such as RMSE and PSNR. This ensures that all results are data-driven.

RMSE Calculation

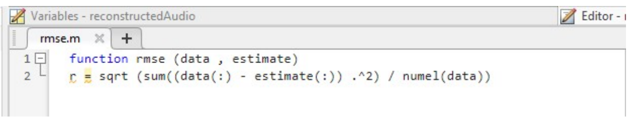
Based on variables obtained from both sound signals, we can now use the RMSE formula to compare the sounds. A smaller RMSE value indicates better similarity, with a good RMSE value typically being under 1.



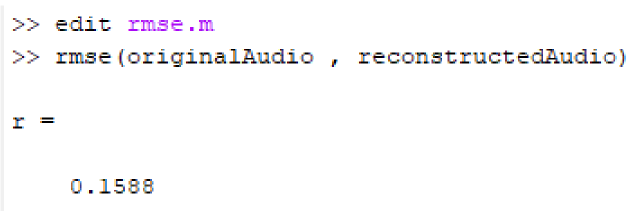
Gambar 13. Processed Sound Variable 1 (Figure 13).



Gambar 14. Processed Sound Variable 2 (Figure 14).



Gambar 15. RMSE Formula (Figure 15).



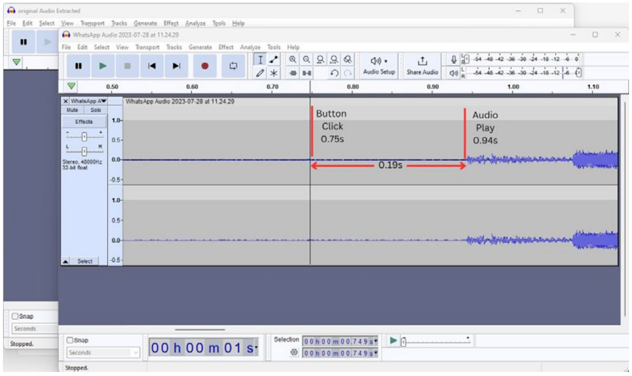
Gambar 16. Calculation RMSE Result using MATLAB (Figure 16).

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{205005}(\text{OriginalSound}_i - \text{ProcessedSound}_i)^2}{205005}} = 0.158 \tag{3}$$

Calculation results using MATLAB are shown in Figures 15 and 16.

PSNR Calculation

After finding RMSE value now we can find PSNR value 5, using RMSE we can find MSE4, here's also we will use max value as 255, 255 as representative of 8- bit data where earlier we compile the audio into unassigned 8- bit PCM audio type data.



Gambar 17. Audio Delay Time Stamp (Figure 17).

$$\begin{aligned} RMSE &= \sqrt{MSE} \\ 0.1588 &= \overline{MSE} \\ (0.1588)^2 &= MSE \\ MSE &= 0.02521744 \end{aligned} \tag{4}$$

$$PSNR = 20.\log 10\left(\frac{255}{0.02521744}\right) = 80.1dB \tag{5}$$

Audio Delay

To determine audio delay, additional tools and programs are usually required. In this experiment, a simple approach using timestamps on the audio was used. Figure 17 shows the timestamp indicating a 0.19 second delay between the click and the audio playback.

On the timestamp, the click is heard at 0.75 seconds, and the audio starts playing at 0.94 seconds, indicating a 0.19 second delay.

CONCLUSION AND SUGGESTION

Conclusion

Based on the test that has been carried out, it is concluded that the Synthesizer based on Arduino can process audio, converting it from digital to analog signal using DAC, and then produce audio based on the audio inputted. If the produced audio convert into variable form we will get the same variable length between original audio and produced audio because before we compile those audio we had to make sure they had the same length in time. Comparing both of the audio give us accuracy of the instrument, in this case the instrument giving a good performance by resulting RMSE of 0.1588 indicating the audio produce is identical with the original audio, the instrument also did a marvelous job converting the Digital Signal to Analog Signal without distorted the audio that can make audio change from its original form. Therefore, the project concludes as a success.

Suggestion

Based on the result of the final project, which still has many flaws in its demonstration due to limited time and ability of the author, it is advised that in the future researchers add various features to improve this project with similar or improved goals. Therefore, the author can give a few recommendations as follows:

1. Use a mic condenser to get better audio quality when doing recording.
2. Improve the variable of input audio so researchers have more material to find the accuracy of the instrument.

3. Use better components as it is quite fragile based on what the author is using now.

Daftar Pustaka

1. Ichwan MG. Implementasi Metoda Unit Selection Synthesizer Dalam Pembuatan Speech Synthesizer Suara Suling Recorder. MIND (Multimedia Artificial Intelligent Networking Database) Journal. 2020:65.
2. Jenkins M. Analog Synthesizers: Understanding, Performing, Buying–From the Legacy of Moog to Software Synthesis. Routledge: Charon Tec Ltd (A Macmillan Company); 2009.
3. Sudirman M. Text To Speech Bahasa Indonesia Menggunakan Synthesizer Concatenation Berbasis Fonem. Jurnal Cosphi. 2018:32-3.
4. Agum Anantama AA. ALAT PANTAU JUMLAH PEMAKAIAN DAYA LISTRIK PADA ALAT. JTST. 2020 Dec:30.
5. Badamasi YA. The Working Principle Of An Arduino. IEEE; 2014.
6. Royhan M. Pengukuran Tegangan Baterai Mobil Dengan Arduino Uno. Jurnal Teknik Informatika UNIS. 2018:30.
7. Sagita SM. Pengkonversian Data Analog Menjadi Data Digital Dan Data Digital Menjadi Data Analog Menggunakan Interface PPI 8255 Dengan Bahasa Pemrograman Borland Delphi 5.0. Faktor Exacta. 2015:168-9.
8. Nuryanto LE. PENERAPAN DARI OP-AMP (OPERATIONAL AMPLIFIER). ORBITH. 2017;13:43.
9. Tuwaidan YA. Rancang Bangun Alat Ukur Desibel (dB) Meter Berbasis Mikrokontroler Arduino Uno R3. Jurnal Teknik Elektro Dan Komputer. 2015:38.
10. Rizzoni G. Principles and Applications of Electrical Engineering 5th Edition. McGraw-Hill Companies; 2005.
11. Agung Yanuar Wirapraja MM. Emisi Radiasi Speaker Aktif pada Frekuensi 30 Mhz - 1 Ghz dan 1 - 6 Ghz. JURNAL TEKNOLOGI PROSES DAN INOVASI INDUSTRI. 2017;4:79.
12. Palending IW. Rancang Bangun Sistem Audio Nirkabel Menggunakan Gelombang Radio FM; 2012.
13. Fransisko P. Media Promosi Elektronik untuk Produk-Produk di Supermarket Menggunakan Arduino Nano. Jurnal Sistem Cerdas dan Rekayasa (JSCR). 2012:73.