`

TAYLOR'S	SCHOOL OF ENGINEERING

ELECTRICAL AND ELECTRONIC ENGINEERING PROGRAMME

PROGRAMMING TECHNIQUES
(EEE 60104)

Group No.	**1**	**Total Marks**	
Lab No.	**3**		
Title of Lab	**Using a Piezo to Produce Different Tones**		

Lecturer	Dr. Chew Wei Jen	
Date	**Lab Date: November 1st, 2023**	**Submitted Date: November 15th, 2023**

Lab Report Item/total marks	**Marks**
Format/10	
Abstract and Introduction/15	
Materials and Method/15	
General Results and Discussions/50	
Conclusion and References/Appendix /10	
Total /100	

No.	**Members ID**	**Name of Group Members**
1	0360231	Muhammad Rafif Edie Wijaya
2	0364144	Law Shing Yi
3	0361508	Wong Kai Tian
4	0349907	Syazani Zikry Zahari
5	0353217	Surryaraj A/L Poobalan

`

TAYLOR'S	SCHOOL OF ENGINEERING

Table of Contents:

1. Abstract.....1
2. Introduction.....1
3. Methodology.....1
4. Results.....3
5. Discussion.....4
6. Conclusion.....4
7. Reference.....5
8. Appendix.....5

`

1. Abstract

In this experiment, the objective was to create a C program that would use an Arduino Uno and a Piezo to produce three different note based on the three push button that was integrated into the circuit. Most of the time Piezo or known as a buzzer, was used in the Arduino project to produce sound. With just a few components like resistors, Piezo and pushbuttons were able to complete the lab experiment with ease. The lab report describes how to program Arduino Uno to make various sound notes in Piezo coupled with pushbuttons.

2. Introduction

This lab session's main objective is to give students practical experience writing a C program for the Arduino Uno, with a particular emphasis on integrating a piezo buzzer to generate three different tones. Three pushbuttons are connected to the Arduino Uno in an interactive learning environment, and each one controls and generates a different tone. In this instance, the piezo buzzer serves as an audio indicator and is important for the translation of the coding into sound signals.

The piezoelectric effect is a concept that controls the operation of the piezo buzzer, a crucial electronic component used in many Arduino sound-generating applications. According to this basic idea, piezoelectric crystals physically distort when an electric field is applied, resulting in vibrations produced. This fundamental feature of piezoelectric materials serves as the basis for the piezo buzzer's sound-producing system.

In the context of the lab, the students delve into the intricacies of programming the Arduino Uno to harness the piezoelectric effect effectively. Three different tones are produced by the Arduino Uno's programmed logic, which changes the electrical signals transmitted to the piezo buzzer in response to pushbuttons being pressed. In addition to improving the students' grasp of C programming for microcontrollers, this exercise offers valuable knowledge about the practical uses of the buzzer in C programming.

3. Methodology

a. Materials:

i. Arduino Uno

ii. Piezo

iii. 10 kΩ resistors

iv. 100 Ω resistors

v. Pushbuttons

b. Methods:

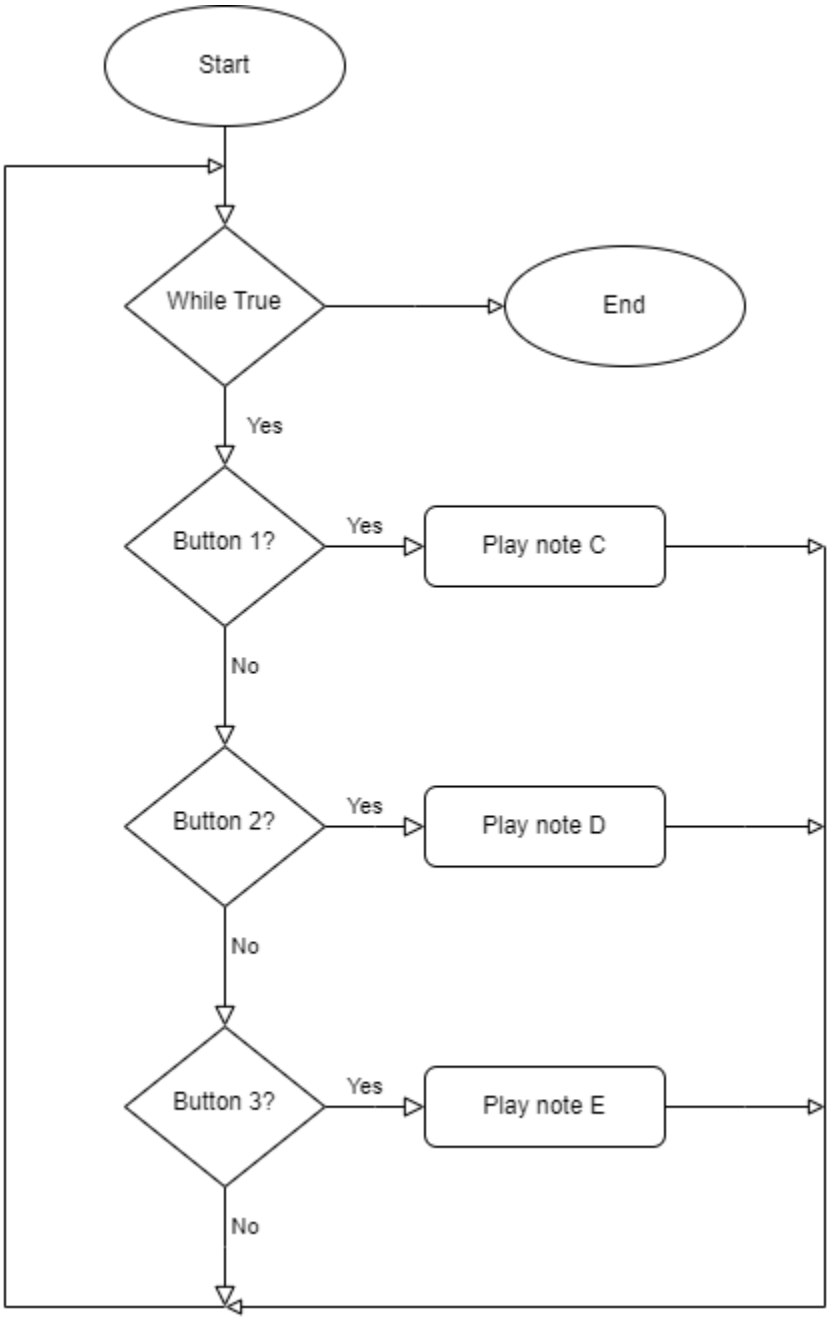

Figure 1. Flowchart of the program

c. Procedure:

i. The piezo is connected to the digital pin 7 on the Arduino through a 100Ω resistor.

ii. 3 pushbuttons are connected to the analog pin A0-A2 on the Arduino through a 10kΩ resistor.

iii. A C program that allows the circuit to produce 3 different frequencies (Table 1) of sound is written.

Table 1: Note to frequency chart

Note	Frequency (Hz)
C	523.25
D	587.33
E	659.26

4. Results

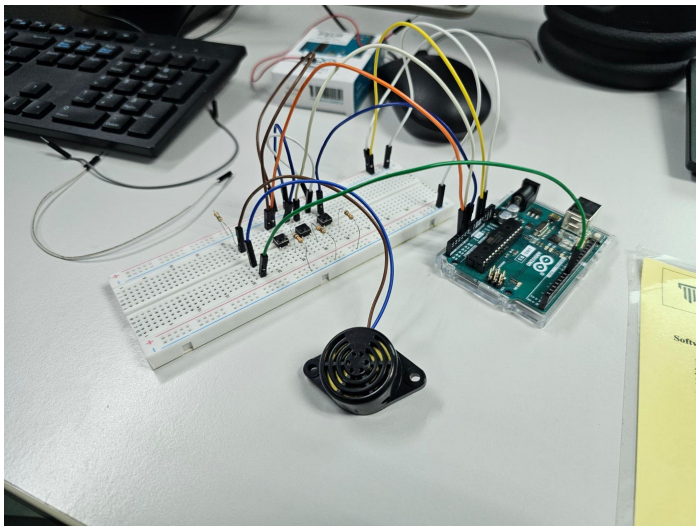

Figure 2. The circuit design of the piezo buzzer.

5. Discussion

a. In this lab, the objective was to turn on the piezo buzzer using 3 pushbuttons. This code started by setting the Arduino pins for inputs and outputs. In this case, the output is a piezo buzzer which was connected to the D-side of the Arduino. For the inputs, 3 analog pins from the Arduino were used, which was equivalent to *DDRC*. In order to obtain the input signals, a pushbutton was connected to each analog input. These pushbuttons were responsible for turning on the piezo buzzer to 3 different frequencies. The main part of the code was put inside a *while(1)* loop, this was done to make sure that the Arduino could keep getting the input signals and turning on the piezo buzzer.

b. To turn on the piezo buzzer, a void function named *sound* was created. This function takes 2 inputs, the intended frequency, and the amount of time that the function runs. The frequency input of the function will be converted to a period, $T = \frac{1}{f}$. Half of that period is dedicated to turning on the buzzer, while the other half is allocated to turning the buzzer off. The command *delayMicroseconds* was used to do this objective, making sure that the function generates the appropriate square wave. Essentially, this function created a square wave that was used to control the piezo buzzer. The second input *t,* represents the runtime of the function. The square wave produced was located inside the while loop that will run as long as $t > 0$. After each iteration of the square wave, the value of *t* will be reduced.

6. Conclusion

In conclusion, this experiment was successful in achieving its objective of using Arduino Uno and C software to operate a Piezo to generate three different sound tones/frequencies based on the pushbuttons that were pressed. The interaction between Piezo and Arduino Uno was a success, all of the three different sound frequencies managed to be produced using the C programming in Arduino platform.

7. Reference

[1] Arduino. "Port Manipulation," [Online], Available: https://docs.arduino.cc/hacking/software/PortManipulation, Accessed: November 1, 2023.

[2] Arduino. "Pin Mapping for Arduino Uno," [Online], Available: https://docs.arduino.cc/hacking/hardware/PinMapping168, Accessed: November 1, 2023.

8. Appendix

```
#include <avr/io.h> // header file for input output pins
#include <util/delay.h> // header file for delay

void sound(long a,int t);

int main(void){
  DDRD = B01000000;
  DDRC = B00000000;
  while(1)
  {
    if(PINC == 0b00000001)//input A0
    {
      sound(523.25,100);
    }
    else if(PINC == 0b00000010)// input A1
    {
      sound(587.33,100);
    }
    else if(PINC == 0b00000100)// input A2
    {
      sound(659.26,100);
    }
  }
}

void sound(long frequency, int t)
{
  while (t>0)
  {
   PORTD = 0b01000000;
   delayMicroseconds(1000000L/frequency/2);
   PORTD = 0b00000000;
   delayMicroseconds(1000000L/frequency/2);
```

`


```
    t = t - 1000/frequency;
  }
}
```