

EXPERIMENT NO. 4: PLAYING MELODY WITH BUZZER USING ARDUINO

**COURSE CODE: CSE 460
COURSE TITLE: IoT LABORATORY**

Submitted by

RAIAN RASHID (403)

Submitted to

**Dr. Md. Ezhariul Islam, Professor
Samsun Nahar Khandakar, Lecturer**



**Computer Science and Engineering
Jahangirnagar University**

Dhaka, Bangladesh

February 23, 2025

Contents

1	INTRODUCTION	1
2	OBJECTIVE	1
3	MATERIALS AND EQUIPMENT	2
4	PROCEDURE & CIRCUIT DIAGRAM	2
5	ARDUINO CODE	4
6	RESULT & DISCUSSION	6

1. INTRODUCTION

In this experiment, we will explore how to generate sound and play melodies using a buzzer and an Arduino. Buzzers are simple electronic devices that produce sound when an electrical current is passed through them, and by controlling the frequency of the signal sent to the buzzer, we can create different pitches. Using the Arduino's digital output pins and its built-in programming functions, we will learn how to program the buzzer to play specific musical notes in sequence. This experiment introduces basic principles of sound generation and is a foundational skill for creating sound-based alerts, alarms, or musical applications in embedded systems and IoT projects.

2. OBJECTIVE

The objective of this experiment is to understand how to interface a buzzer with an Arduino to generate sound and play melodies. By controlling the frequency and duration of the signals sent to the buzzer, the goal is to develop a program that can produce different musical notes and sequences. This experiment aims to familiarize students with the basics of sound synthesis, programming techniques for tone generation, and the practical application of digital outputs for creating auditory signals in embedded systems.

3. MATERIALS AND EQUIPMENT

The following materials and equipment were used to complete the experiment on Arduino Basics and LED Blinking:

- Arduino Uno Board
- Buzzer
- Connecting Wires
- Breadboard
- Arduino IDE (Software)

4. PROCEDURE & CIRCUIT DIAGRAM

The following steps were followed to complete the experiment and toggle the LED using the Arduino board and the button:

1. Connect the Arduino Uno board to the computer using a USB cable.
2. Insert one end of the Buzzer to the pin 9 of the Arduino and the other end on the ground (GND).
3. Write the code for initiating the buzzer, upload the code to the Arduino board using the "Upload" button in the IDE and observe the buzzer if it makes any sound or noise.

Circuit Diagram:

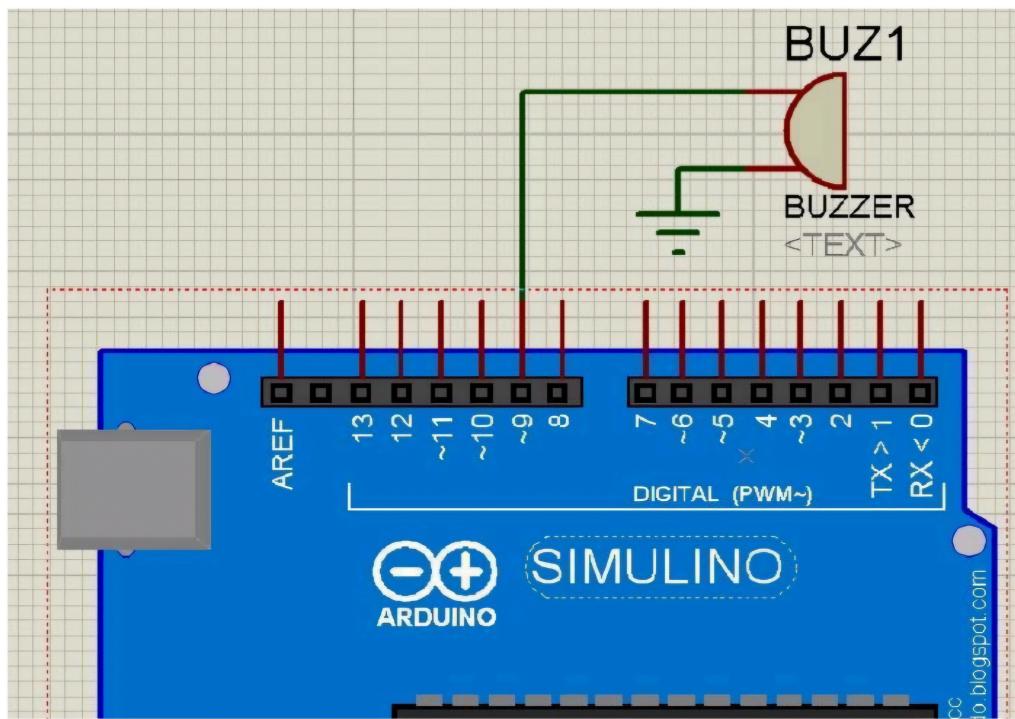


Figure 4.1: Circuit Diagram

5. ARDUINO CODE

The Arduino code for playing a buzzer:

```
1 const int buzzerPin = 9;
2 const int length = 15;
3 char notes[] = "ccggaagffeeddc";
4 int beats[] = {1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 2, 4};
5 int tempo = 300;
6 void playTone(int period, int duration) {
7     long cycles = (duration * 1000L) / (period * 2);
8     for (long i = 0; i < cycles; i++) {
9         digitalWrite(buzzerPin, HIGH);
10        delayMicroseconds(period);
11        digitalWrite(buzzerPin, LOW);
12        delayMicroseconds(period); }
13    }
14 void playNote(char note, int duration) {
15     char names[] = {'c', 'd', 'e', 'f', 'g', 'a', 'b', 'c'};
16     int tones[] = {1915, 1700, 1519, 1432, 1275, 1136, 1014, 956};
17     for (int i = 0; i < 8; i++) {
18         if (names[i] == note) {
19             playTone(tones[i], duration); } }
20    }
21 void setup() {
22     pinMode(buzzerPin, OUTPUT); }
23 void loop() {
24     for (int i = 0; i < length; i++) {
25         if (notes[i] == ' ') {
26             delay(beats[i] * tempo);
27         } else {
28             playNote(notes[i], beats[i] * tempo);
29         }
30         delay(tempo / 2);
31     }
32     delay(2000);
33 }
```

In this code:

1. *playTone()* generates a square wave signal on the buzzer by toggling its state at a specific frequency for a given duration
2. *digitalWrite()* sets a digital pin to either HIGH (ON) or LOW (OFF).
3. *delayMicroseconds()* pauses execution for a specified number of microseconds.
4. *setup()*: runs once at startup to initialize configurations like setting the buzzer pin as an output.
5. *loop()* repeats indefinitely, playing the melody in sequence.
6. *delay()* pauses execution for a specified number of milliseconds.

6. RESULT & DISCUSSION

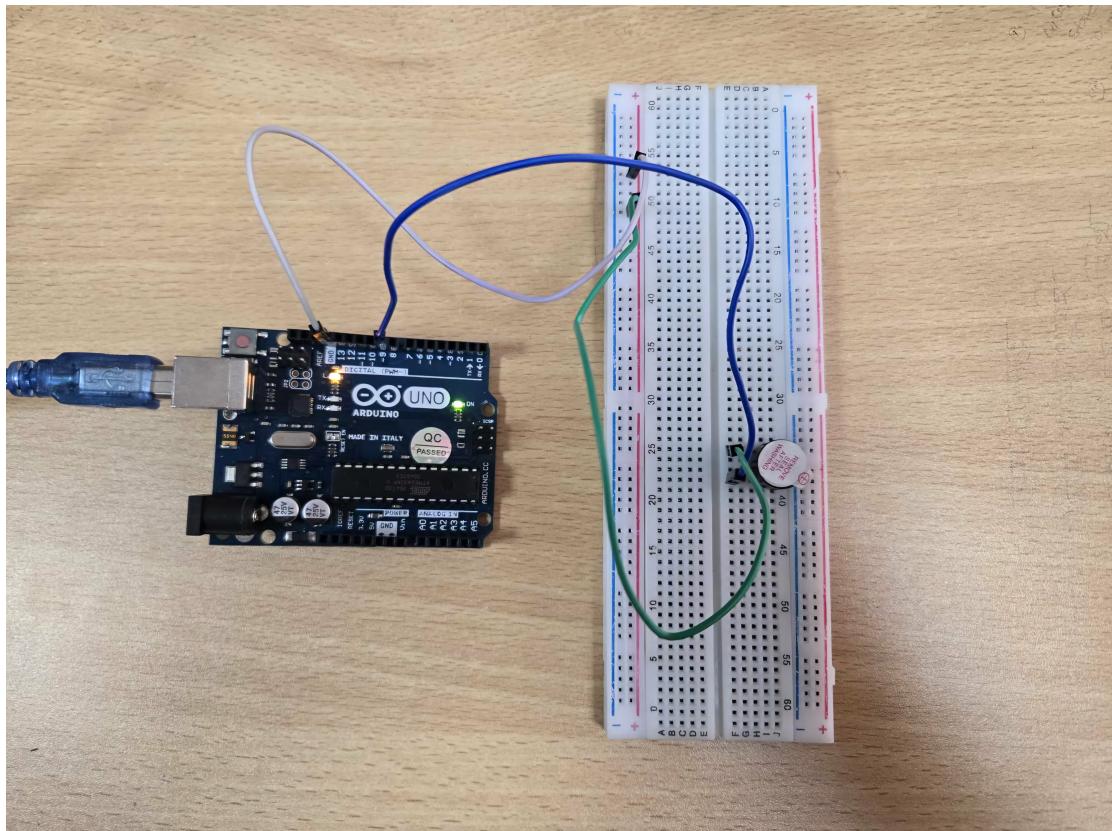


Figure 6.1: Playing Buzzer using Arduino

The implemented circuit successfully plays the predefined melody on the buzzer, following the programmed sequence of notes. The buzzer produces distinguishable tones corresponding to musical notes, with appropriate pauses between them. The circuit operates as expected, continuously looping the melody after a short delay.

The circuit uses an Arduino to generate sound by toggling the buzzer pin at specific frequencies, creating musical notes. *The playTone()* function produces square wave signals, while *digitalWrite()* and *delayMicroseconds()* control the pitch and duration. This experiment demonstrates basic sound synthesis with Arduino, and future improvements could include user-controlled melody selection or dynamic tempo adjustment.