

Problem Statement: Understanding the connectivity of Arduino with Buzzer. Write an application to for buzzer.

Objectives:

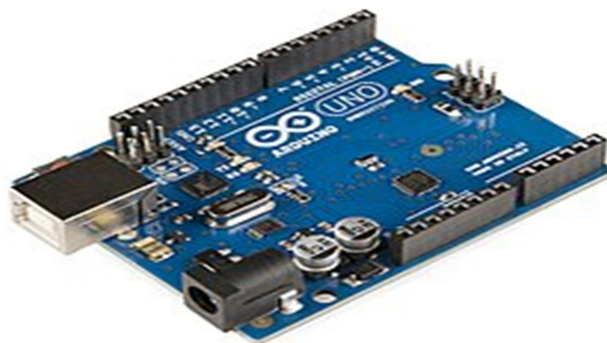
1. To study about Arduino kit and its component.
2. To Learn the interfacing of buzzer with Arduino UNO

Software & Hardware Requirements:

1. Operating System: Windows (XP/Vista/7/10)
2. Software: Arduino IDE 1.8.3
3. Hardware: Arduino UNO, Buzzer, Patch Cords, USB cable type A/B.

Theory: What is Arduino?

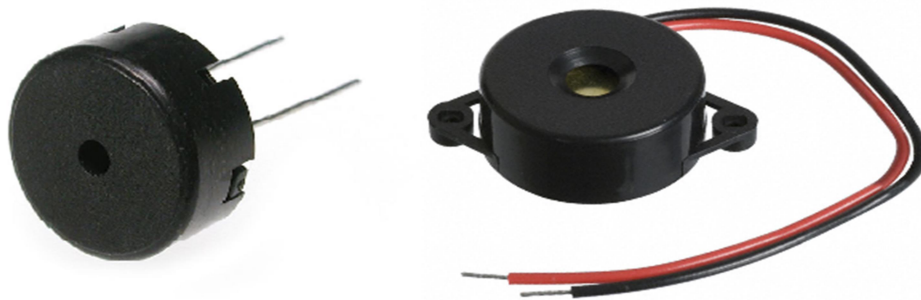
Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing.



Piezo Speaker / Buzzer:-

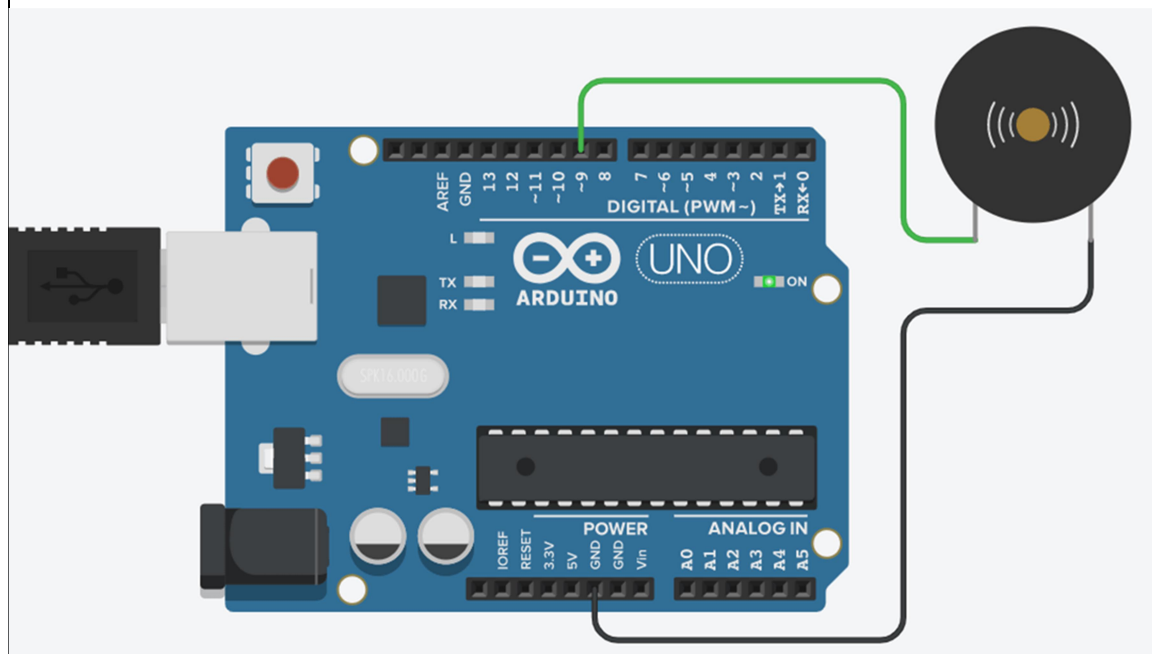
A piezoelectric speaker (also known as a piezo bender due to its mode of operation, and sometimes colloquially called a "piezo", buzzer, crystal loudspeaker or beep speaker) is a loudspeaker that uses the piezoelectric effect for generating sound. The initial mechanical motion is created by applying a voltage to a piezoelectric material, and this motion is typically converted into audible sound using diaphragms and resonators. The prefix piezo- is Greek for 'press' or 'squeeze'.

Compared to other speaker designs piezoelectric speakers are relatively easy to drive; for example they can be connected directly to TTL outputs, although more complex drivers can give greater sound intensity. Typically they operate well in the range of 1-5 kHz and up to 100 kHz in ultrasound applications.



Buzzer with Arduino UNO – Connection Diagram :-

Connection :- Pin 9 and Ground pin to be connected to Buzzer.



Source Code

```
const int buzzerPin = 9;
void setup(){
  pinMode(buzzerPin, OUTPUT);
}

void loop(){
  analogWrite(buzzerPin, 127);
  delay(500);
  analogWrite(buzzerPin,0);
  delay(2500);
}
```