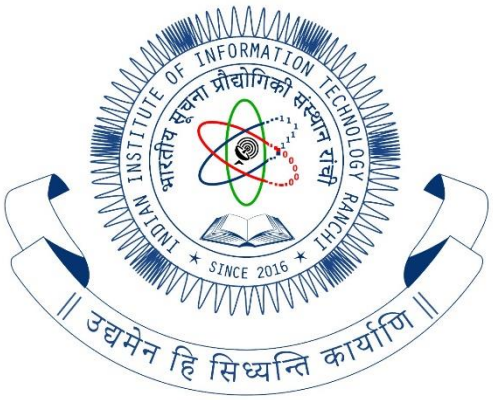
Indian Institute of Information Technology, Ranchi

(An Institution of National Importance under MHRD, Govt. of India)



Embedded Systems Lab

Project Report On

"**Digital Piano**"

Submitted By:

Prabhat Suman

Aim:

To Design a Digital Piano on Arduino (i.e. to generate different tones by pressing switches).

Requirements:

Hardware:

(i) Arduino UNO R3
(ii) Push Buttons
(iii) Small Piezo Buzzer (a small speaker)
(iv) Connecting Wires (Jumper Wires)
(v) Breadboard
(vi) Power Supply

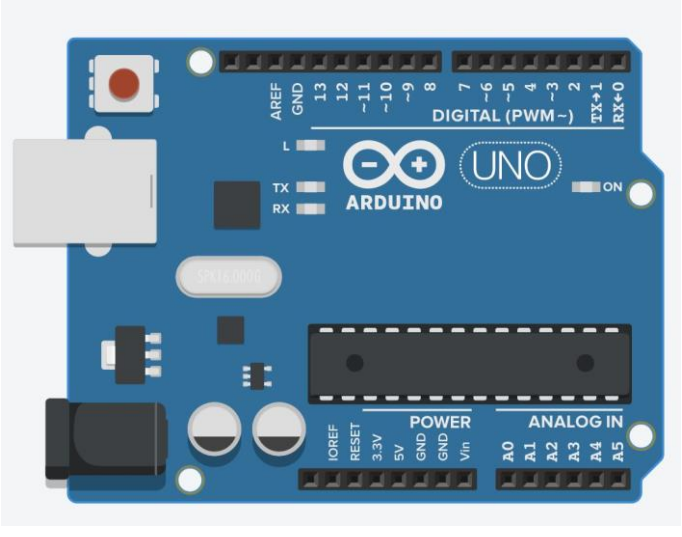
Software:

1. Arduino IDE
2. Tinkercard

Theory:

Arduino Uno:

It is a microcontroller board based on the ATmega328P (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator (CSTCE16M0V53-R0), a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Arduino Uno is programmed using the Arduino Software (IDE), our Integrated Development Environment common to all our boards and running both online and offline.



tone():

Generates a square wave of the specified frequency (50% duty cycle) on the pin. A duration can be specified, otherwise the wave continues until a call to notone(). The pin can be connected to a piezo buzzer or speaker to play tones.

Only one tone can be generated at a time. If the tone is already playing on different pin, call to tone() has no effect. If tone is playing on same pin, the call will set its frequency.

Use of tone() will interfere with PWM output on pins 3 and 11.

It is impossible to generate tone lower than 31Hz.

Syntax:

tone(pin,frequency)
tone(pin,frequency,duration)

Parameters:

pin: the Arduino pin on which to generate the tone.
frequency: the frequency of the tone in hertz. Allowed data types: unsigned int.
duration: the duration of the tone in milliseconds (optional).
Allowed data types: unsigned long.

Piezo Buzzer:

It is an electronic device that is used to produce tone, alarm or sound. It is light-weighted with simple construction and cost effective. It can be constructed over wide range of sizes that work across different frequencies and producing different tones.

So, when certain piezoelectric materials are subjected to an alternating field of electricity, the piezo buzzer element stretches and compresses in sequence with the frequency of the current. As a result, it produces an audible sound.

Piezo buzzers can typically operate anywhere between three and 250 volts. It normally consumes less than 30 milliamperes even at higher rate frequencies.

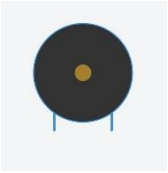
Other common features include:

Volume control; line-out audio connections; headphone output.

May include many more instrument sounds beyond piano samples.

May have a transposition feature.

Are often easily portable and low maintenance (do not need to be tuned).



Digitalpiano:

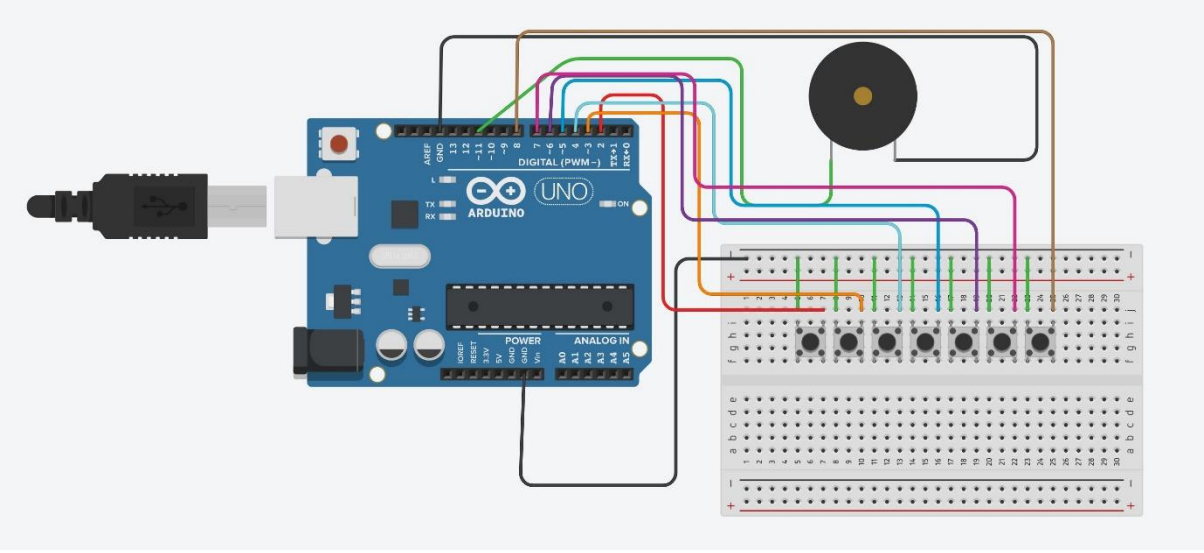
It is a type of electronic keyboard instrument designed to serve primarily as an alternative to the traditional acoustic piano, both in the way it feels to play and in the sound produced. Digital pianos use either synthesized emulation or recorded samples of an acoustic piano, which are then amplified through an internal loudspeaker. They also incorporate weighted keys, which recreate the feel of an acoustic piano. Some digital pianos are designed to also look like an upright or grand piano.

Digital pianos typically use analog sensors for their keyboard action, as opposed to digital sensors of a regular electronic keyboard and synthesizer. These sensors work in a similar way to those used in analog joysticks found on video game controllers, in which they the velocity input is converted from the key movement as well, not just the initial pressure of the key sensor.

Pushbutton:

The pushbutton is a component that connects two points in a circuit when you press it. The surface is usually flat or shaped to accommodate the human finger or hand, so as to be easily depressed or pushed. The "push-button" has been utilized in calculators, push-button telephones, kitchen appliances, and various other mechanical and electronic devices, home and commercial.

Circuit Diagram

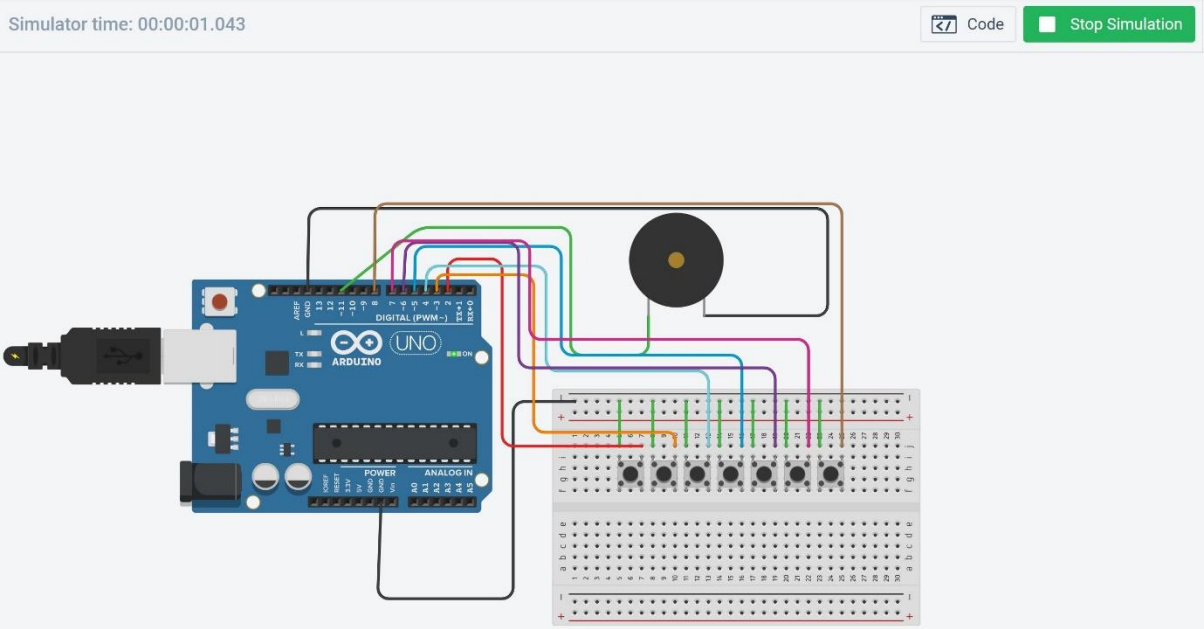


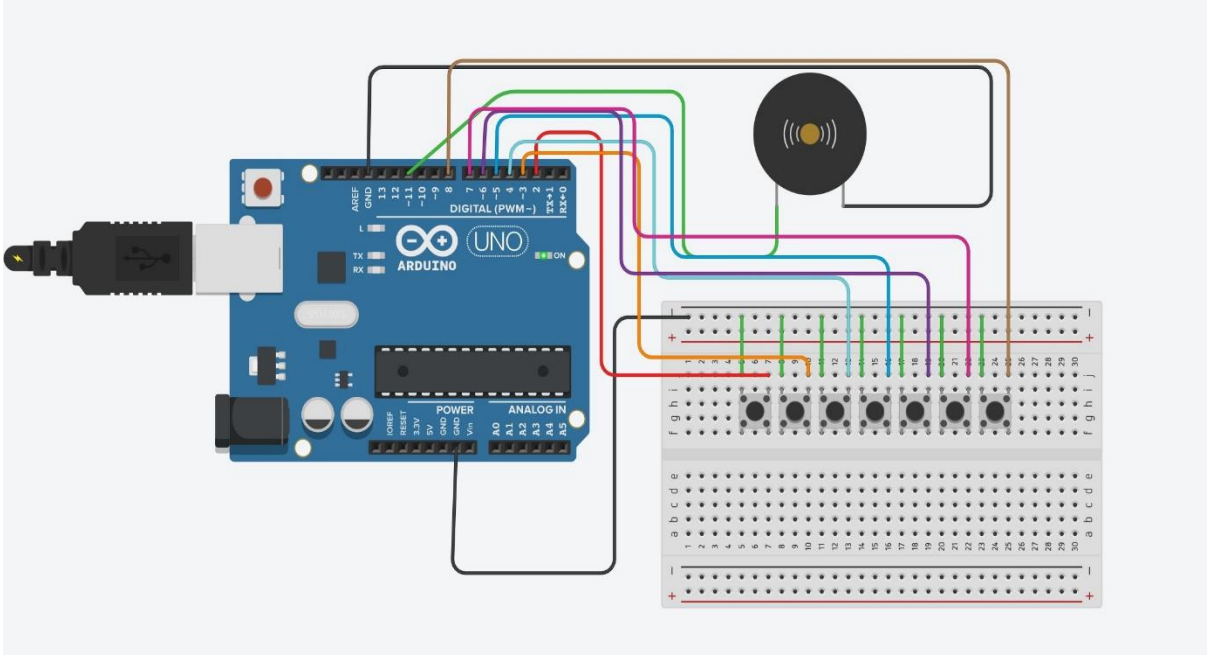
Code:

```
#define T_C 262
#define T_D 294
#define T_E 330
#define T_F 349
#define T_G 392
#define T_A 440
#define T_B 493

const int C = 8;
const int D = 7;
const int E = 6;
const int F = 5;
const int G = 4;
const int A = 3;
const int B = 2;
const int Buzz = 11;
const int LED = 13;

void setup()
{
  pinMode(C, INPUT);
  digitalWrite(C,HIGH);


  pinMode(D, INPUT);
  digitalWrite(D,HIGH);

  pinMode(E, INPUT);
  digitalWrite(E,HIGH);
  pinMode(F, INPUT);
  digitalWrite(F,HIGH);

  pinMode(G, INPUT);
  digitalWrite(G,HIGH);

  pinMode(A, INPUT);
  digitalWrite(A,HIGH);

  pinMode(B, INPUT);
  digitalWrite(B,HIGH);
}

void loop()
{
  while(digitalRead(C) == LOW)
  {
    tone(Buzz,T_C);
    digitalWrite(LED,HIGH);
  }

  while(digitalRead(D) == LOW)
  {
    tone(Buzz,T_D);
    digitalWrite(LED,HIGH);
  }


while(digitalRead(E) == LOW)
  {
    tone(Buzz,T_E);
    digitalWrite(LED,HIGH);
  }

  while(digitalRead(F) == LOW)
  {
    tone(Buzz,T_F);
    digitalWrite(LED,HIGH);
  }

  while(digitalRead(G) == LOW)
  {
    tone(Buzz,T_G);
    digitalWrite(LED,HIGH);
  }
while(digitalRead(A) == LOW)
  {
    tone(Buzz,T_A);
    digitalWrite(LED,HIGH);
  }

  while(digitalRead(B) == LOW)
  {
    tone(Buzz,T_B);
    digitalWrite(LED,HIGH);
  }

  noTone(Buzz);
  digitalWrite(LED,LOW);}
```

Output:

Start Simulation





After pressing different Pushbutton, we can generate different sound.

Working:

Make the connections as per the circuit diagram and upload the code to Arduino. Connect 8 Push Buttons to digital I/O pins from 2 to 8 of Arduino. These pins act as the tone input pins. All the other terminals of these push buttons are connected to GND. Then, connect the Piezo Buzzer to one of the PWM capable pins of Arduino. The other end of the Piezo Buzzer is connected to GND. Now, make the connections as per the circuit diagram and upload the code to Arduino. Once the power to the circuit is turned on, Arduino is ready to accept the input from the buttons.

Each button is associated with a PWM signal in the code. When a button is pushed, that particular PWM signal is generated through the Piezo Electric Buzzer.

Once you are done with the tone, you can push the Interrupt Button. As soon as the Arduino enters Interrupt Mode, all the previously pressed tones are played back through the Piezo Buzzer.

Conclusion:

A Digital Piano Project based on Arduino is implemented here. This project can clarify the sound capabilities of Arduino. If the note is not accurate, you can adjust the note value in the Arduino sketch to set what value that the pitch is achieved. You can also change the scale that or make your own scale.

Thank You