

## Index

Sr No	Experiment List	Sign
1	Learn Basic electronic components and to get familiar with Arduino software/hardware and Arduino simulator.	
2	Write a program to blink Arduino onboard LED & to interface external LED with Arduino	
3	Write a program to blink Arduino onboard LED & to interface external LED with breadboard & Arduino	
4	WAP to blink LEDs with Arduino & to interface external 5 LEDs with breadboard & Arduino and write a program to turn ON/OFF LED.	
5	Write a program to blink 6 LEDs, one at a time, in back and forth formation.	
6	Write a program to Fade LED with Arduino Analog Output.	
7	Write a program to turn ON LED when push button is pressed.	
8	Write a program to turn ON LEDs when push button is pressed and gets OFF after second automatically like a door bell.	
9	WAP to interface Push Button with Arduino and write a program to turn ON LED when push button is pressed and gets OFF when push button is pressed again.	
10	WAP to interface Push Button with Arduino and write a program to turn ON LEDs when push button is pressed in 7 different patterns.	
11	WAP to interface RGB LED with Arduino and write a program to use the RGB LED with Arduino to obtain different colors.	
12	WAP to interface RGB LED with Arduino and write a program to use the RGB LED with Arduino to obtain multiple colors.	
13	WAP to interface ON/OFF LED with Arduino and write a program to use the Slide Button with Arduino to obtain LED ON & OFF.	
14	WAP to interface Piezo Speaker with Arduino and write a program to use the Piezo Speaker with Arduino to obtain continuous buzzer sound.	

Sr No	Experiment List
15	WAP to interface Piezo Speaker with Arduino and write a program to use the Piezo Speaker with Arduino to obtain on Push button switch ON/OFF.
16	WAP to interface Piezo Speaker with Arduino and write a program to use the Piezo Speaker LED ON/OFF with Arduino to obtain on Push button switch ON/OFF.
17	Write a program use to turn ON LED and generate a 2 different notes on two button keyboard.
18	WAP to interface Piezo Speaker & Multicolored LED with Arduino and write a program to create fast LED stream that goes back and forward while interfacing with buzzer.
19	WAP to interface a 7 segment Display with Arduino & WAP to turn ON Display for displaying values from 0 to 9 & A to F
20	WAP to interface 16*2 Character LCD module with Arduino & WAP to display running with LCD character on parallel interface LCD controller chip.
21	WAP to interface 16*2 Character LCD module with Arduino & WAP to display running with LCD to custom character generated based on parallel interface LCD controller chip.
22	WAP to interface 16x2 LCD, Push Button, Potentiometer with Arduino and WAP to display message on LCD when push button is pressed.
23	1. To upload data on Thingspeak and cloud manually. 2. To update readings to Thingspeak from Arduino using Tinkercad. 3. To interface Temperature sensor and ESP8266 with Arduino and update temperature reading to Thingspeak.
24	To interface LDR sensor, LED and ESP8266 with Arduino and update light intensity values to Thingspeak and tweet "LIGHT ON" message on tweeter when light intensity value is less than 300.
25	To interface servo motor / DC motor with Arduino and WAP to sweep a servo back and forth through its full range of motion/to control a DC motor.

**Practical: 1**

**Aim:** Introduction of Basic Components and Installation.

**1. Resistor: -**

In Tinkercad, a **resistor** is an electronic component used to limit or control the flow of current in a circuit. It is represented by a zig-zag symbol and can be customized with a specific resistance value, measured in ohms ( $\Omega$ ). Resistors are essential for protecting components like LEDs by reducing current flow, and they are often used in series or parallel configurations to manage voltage and current. You can easily add, adjust, and connect resistors in Tinkercad to simulate real-world electronic circuits.

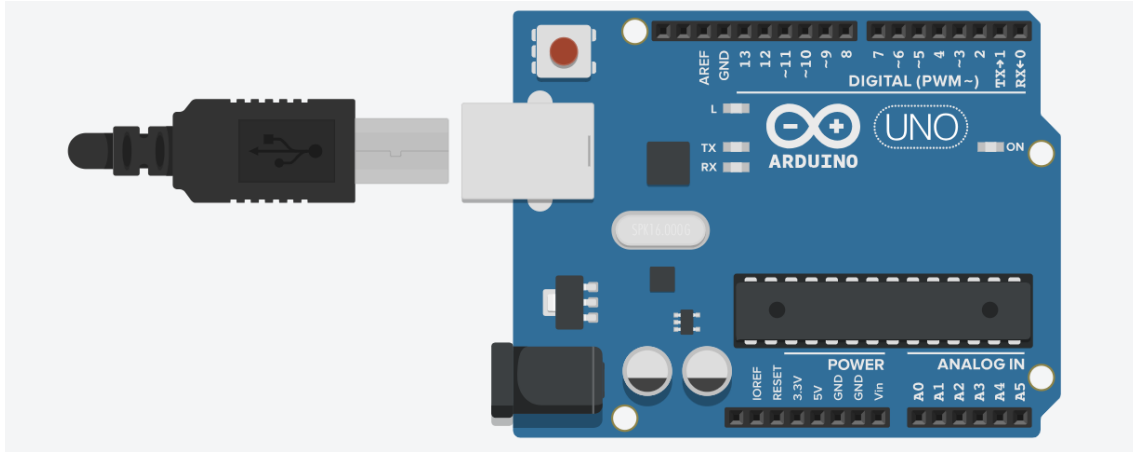
**2. LED: -**

In Tinkercad, an **LED** (Light Emitting Diode) is a semiconductor device that emits light when current flows through it. It is represented by a small triangle with two arrows pointing outwards, indicating the light emission. LEDs are polarized, meaning they have a positive (anode) and negative (cathode) leg, which must be connected correctly in the circuit. In Tinkercad, you can adjust the LED's properties, such as its color, and use resistors in series to prevent damage from excessive current. LEDs are commonly used in circuits to indicate status or display information.



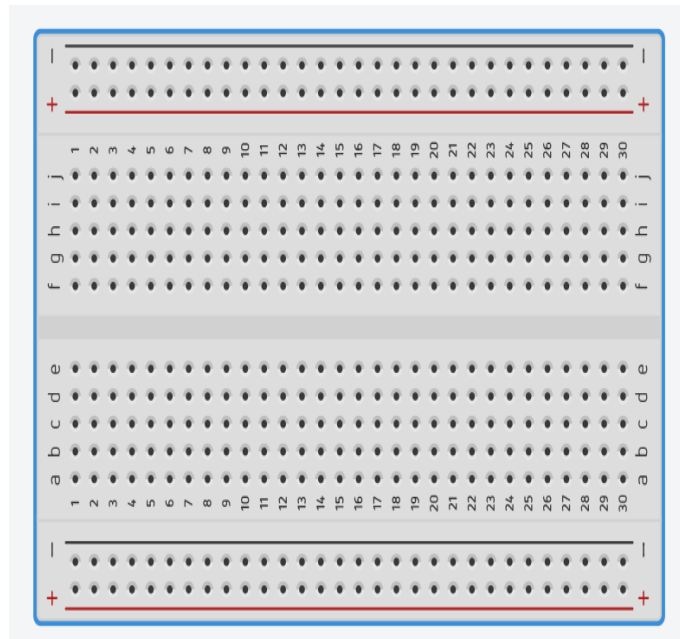
### 3. Arduino Uno R3: -

In Tinkercad, the **Arduino Uno R3** is a microcontroller board based on the ATmega328P. It features 14 digital input/output pins, 6 analog inputs, a USB connection for programming, and a power jack. It is widely used in electronics and robotics projects to control various components like LEDs, sensors, and motors. In Tinkercad, you can simulate and program the Arduino Uno R3 using a block-based or text-based interface, allowing you to test and prototype circuits before physical implementation.



### 4. Breadboard Small: -

In Tinkercad, the **Breadboard Small** is a virtual prototyping tool used for building and testing electronic circuits without soldering. It has a grid of holes where components like resistors, LEDs, and wires can be inserted and connected. The small breadboard has a limited number of rows and columns, making it ideal for simple circuits. It is commonly used in Tinkercad for assembling circuits in a compact, easy-to-manage space for testing before moving to a more permanent setup.



### 5. LED RGB: -

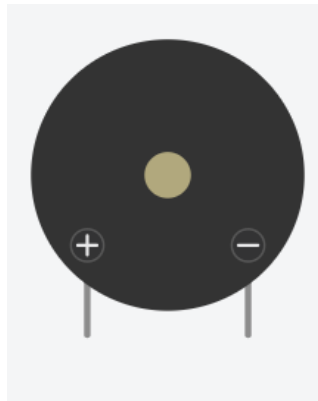
In Tinkercad, an **RGB LED** is a type of LED that can display a wide range of colors by combining red, green, and blue light. It has four pins: one for each color (red, green, blue) and a

common cathode or anode pin. By adjusting the voltage on each pin, you can control the intensity of each color, allowing the RGB LED to produce various colors. This component is commonly used in Tinkercad to create colorful displays and effects in circuits.



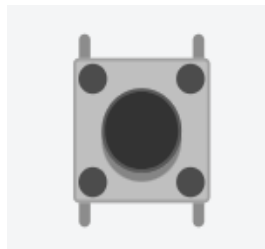
#### 6. Piezo: -

In Tinkercad, a **Piezo** is a small speaker or buzzer that produces sound when an electric current is applied to it. It works based on the piezoelectric effect, where mechanical stress produces an electric charge, which in turn generates sound waves. The Piezo component in Tinkercad is often used in circuits to create audio signals, alerts, or tones in projects like alarms or simple sound effects. It can be controlled by applying varying voltages or signals from a microcontroller like an Arduino.



#### 7. Push Button

The push buttons included in your Photon kit are classified as **momentary** switches, which means they detect when they are being pressed or pushed. For example, the keys on a computer keyboard are momentary switches: they are only "on" when you press them (and they turn "off" when you release them). The push button has 4 metal legs on its base (two legs on one side, and two legs on the opposite side). Unlike most other parts that connect to only one side of a breadboard, the push button has to connect to **both** sides of a breadboard.



## Practical: 2

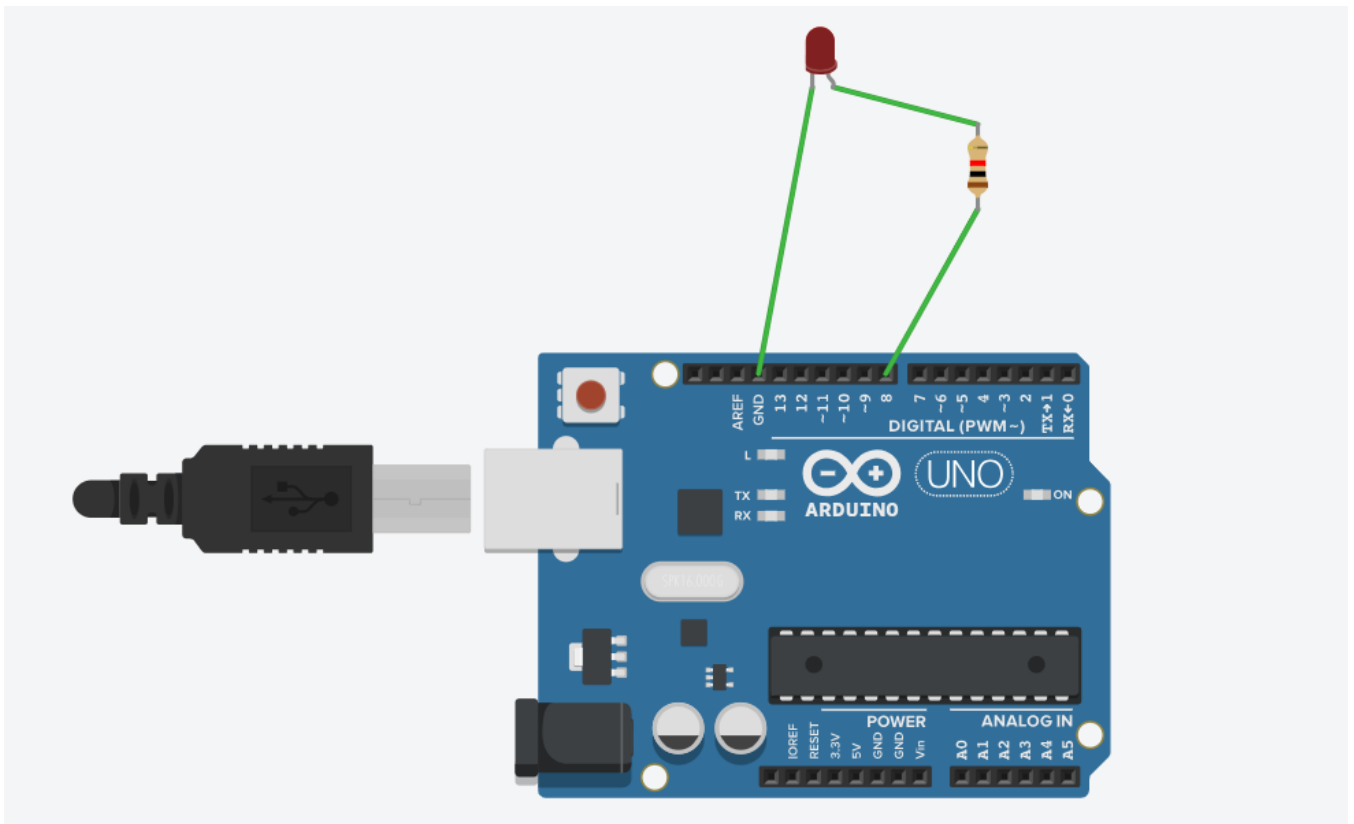
**Aim :** WAP to blink Arduino onboard LED & to interface external LED with Arduino and write a program to turn ON LED for 1 second after every 2 seconds.

**Objective:**

- To learn Arduino UNO basics.
- Write a program to blink Arduino onboard LED and to interface external LED with Arduino.
- Write a program to turn ON LED for 1 second after every 2 seconds

**Components:** LED, Resistor, Arduino Uno r3, wires

**Circuit:-**

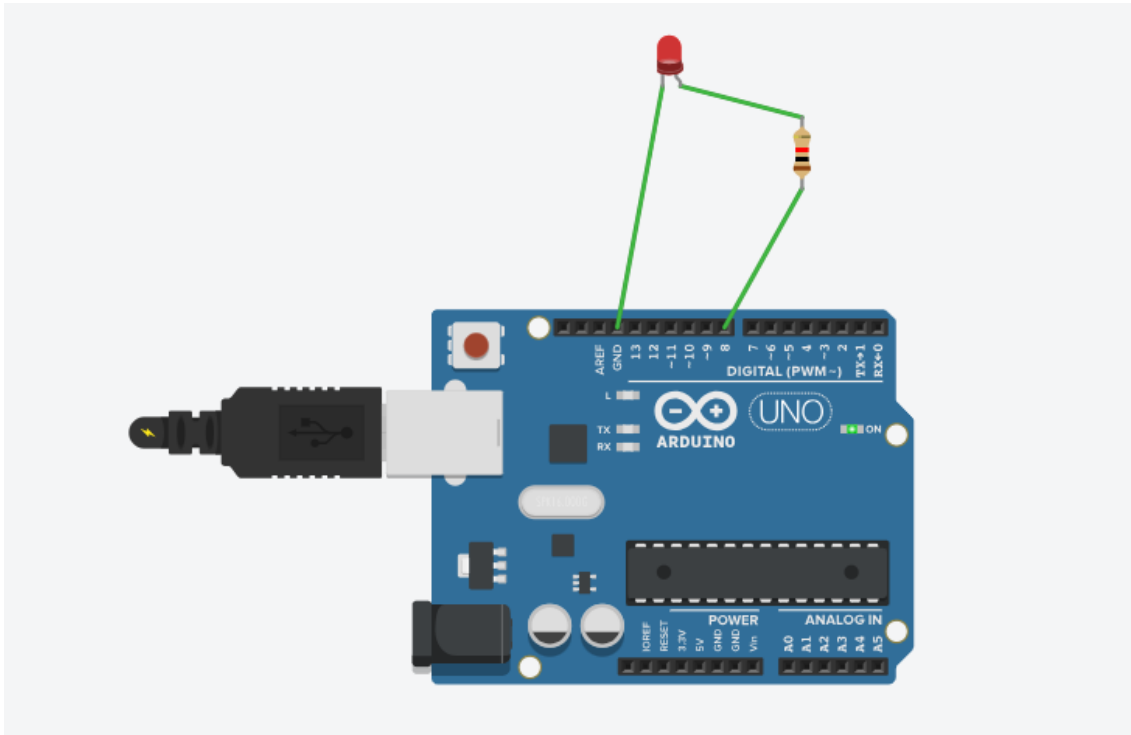


**Code:-**

```
int blink_led=8;
void setup()
{
  pinMode( blink_led=8, OUTPUT);
}

void loop()
{
  digitalWrite( blink_led=8, HIGH);
  delay(5000); // Wait for 1000 millisecond(s)
  digitalWrite( blink_led=8, LOW);
  delay(5000); // Wait for 1000 millisecond(s)
```

```
}
```

**Output:****Conclusion:**

Thus, learnt about basic components of IoT like Arduino UNO (blinking Arduino onboard LED), LED's and interfacing external LED and resistor with Arduino.

### Practical:3

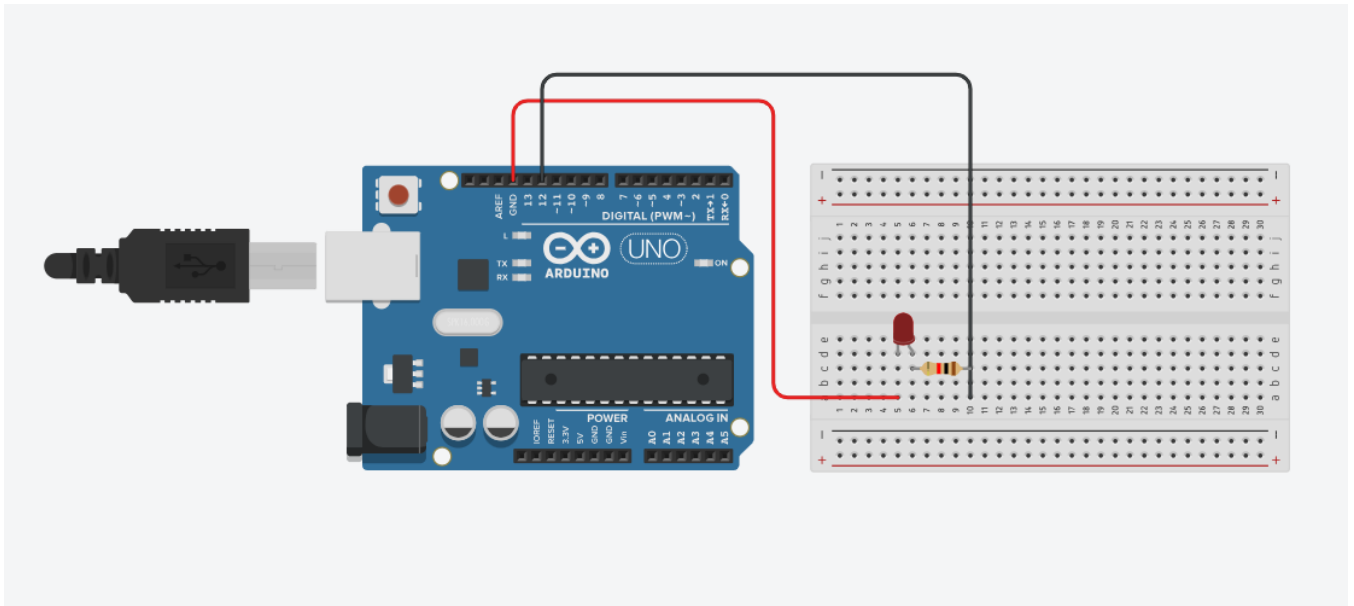
**Aim :**WAP to blink Arduino onboard LED & to interface external LED with Breadboard & Arduino and write a program to turn ON LED for 1 second after every 2 seconds.

**Objective:**

- To learn Arduino UNO basics
- To learn Breadboard basics
- Write a program to blink Arduino onboard LED and to interface external LED with Breadboard & Arduino.
- Write a program to turn ON LED for 1 second after every 2 seconds

**Components:** LED, Resistor, Arduino Uno r3, wires

**Circuit:**

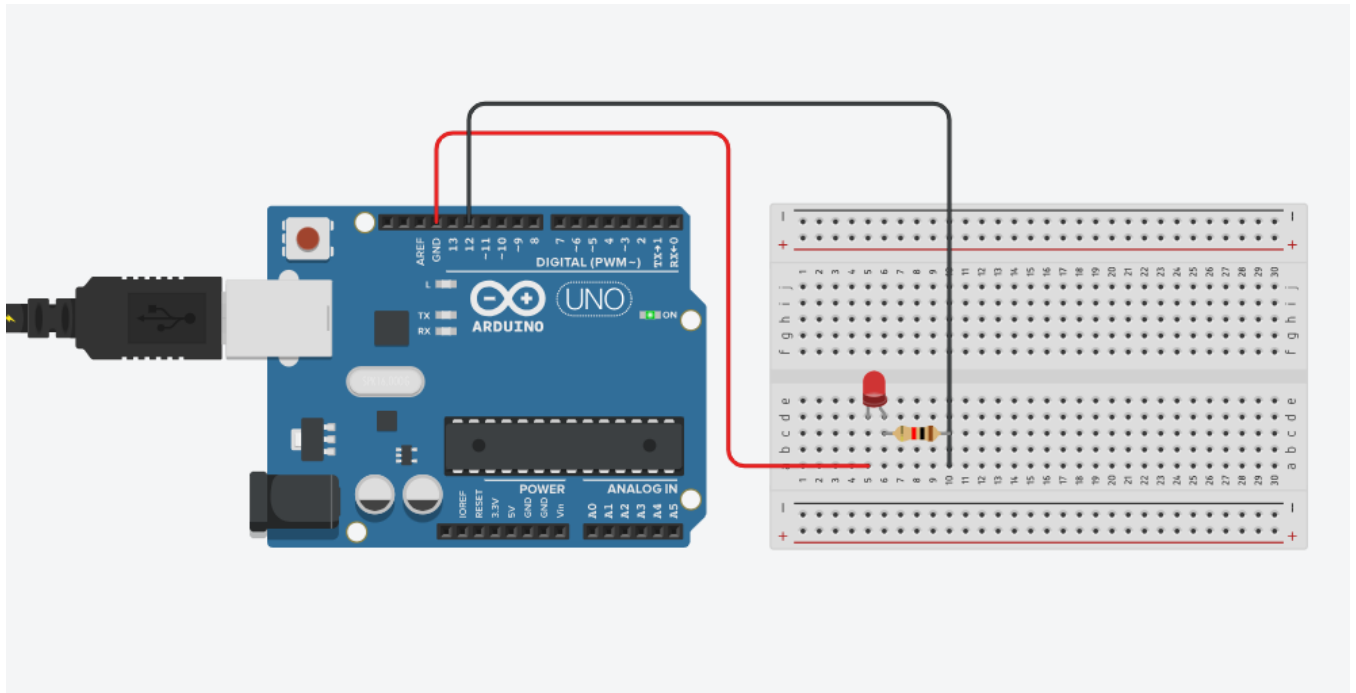


**Code:**

```
// C++ code
//
int Blink_led = 12;
void setup()
{
  pinMode(Blink_led, OUTPUT);
}

void loop()
{
  digitalWrite(Blink_led, HIGH);
  delay(1000); // Wait for 1000 millisecond(s)
  digitalWrite(Blink_led, LOW);
  delay(1000); // Wait for 1000 millisecond(s)
}
```



**Output:****Conclusion:**

Thus, learnt about basic components of IoT like Arduino UNO (blinking Arduino onboard LED), Breadboard, LED's and interfacing external LED and resistor with breadboard & Arduino.

### Practical: 4

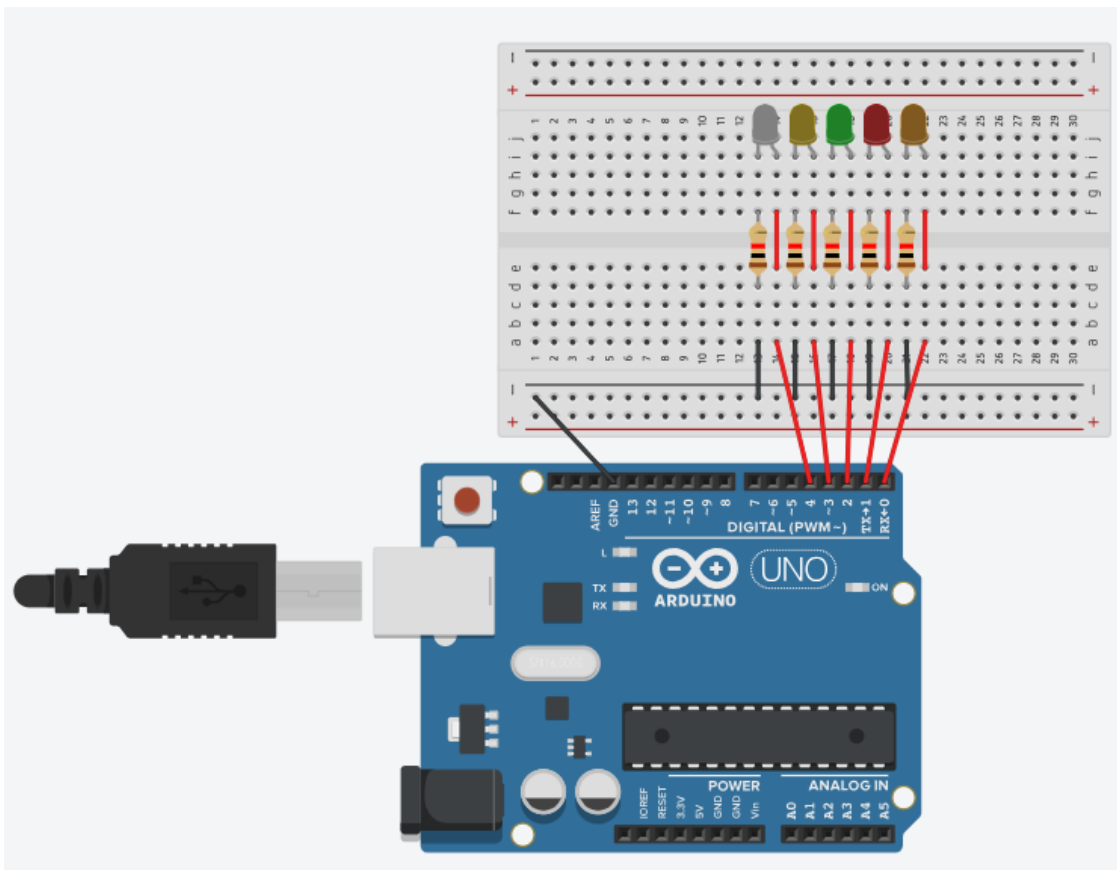
**Aim:** WAP to blink LEDs with Arduino & to interface external 5 LEDs with Breadboard & Arduino and write a program to turn ON/OFF LED.

**Objective:**

- To learn Arduino UNO basics
- To learn Breadboard basics
- Write a program to blink Arduino onboard LED and to interface external LED with Breadboard & Arduino.
- Write a program of blinking 5 LEDs – test program to run 5 LED sin a pattern

**Components:** 5 LEDs, 5 Resistors, Arduino Uno r3, wires

**Circuit:**



**Program Code:**

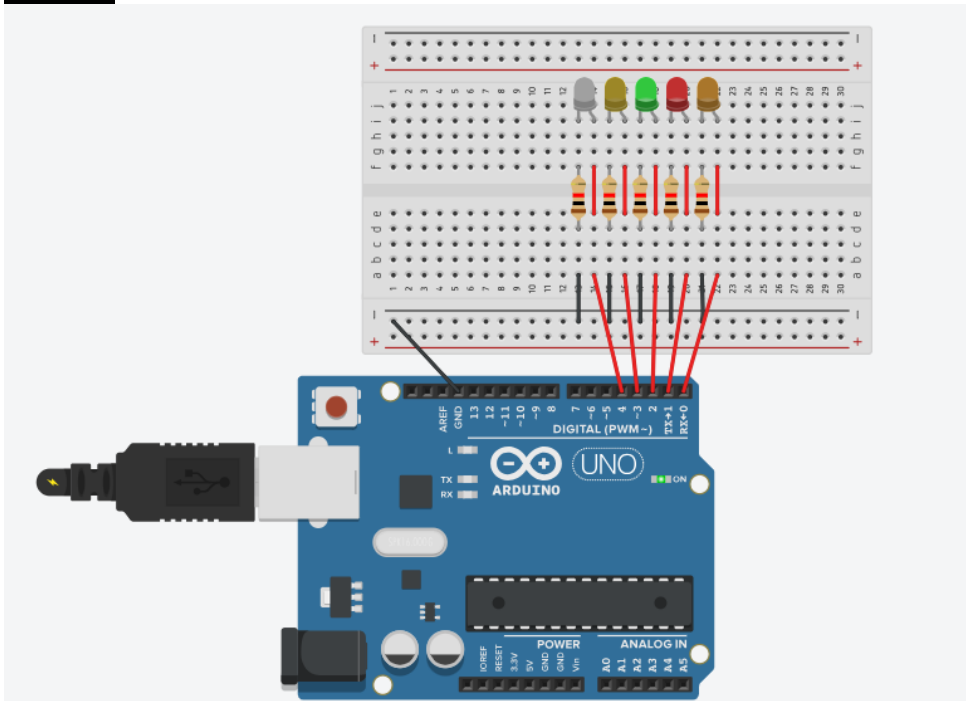
```
// C++ code
//
/* Blinking LEDs - test program to run 5 LEDs in a pattern of blinks

int led1 = 0;
int led2 = 1;
int led3 = 2;
int led4 = 3;
int led5 = 4;
```

```
void setup()
{
  pinMode(led1, OUTPUT);
  pinMode(led2, OUTPUT);
  pinMode(led3, OUTPUT);
  pinMode(led4, OUTPUT);
  pinMode(led5, OUTPUT);
}

void loop()
{
  digitalWrite(led1, HIGH);
  delay(80); // Wait for 1000 millisecond(s)
  digitalWrite(led1, LOW);
  digitalWrite(led2, HIGH);
  delay(80); // Wait for 1000 millisecond(s)
  digitalWrite(led2, LOW);
  digitalWrite(led3, HIGH);
  delay(80); // Wait for 1000 millisecond(s)
  digitalWrite(led3, LOW);
  digitalWrite(led4, HIGH);
  delay(80); // Wait for 1000 millisecond(s)
  digitalWrite(led4, LOW);
  digitalWrite(led5, HIGH);
  delay(80); // Wait for 1000 millisecond(s)
  digitalWrite(led5, LOW);
}
```

### Output:



### Conclusion:

Thus, learnt about basic components of IoT like Arduino UNO (blinking Arduino onboard LED), Breadboard, LED's and interfacing external 5 LEDs blinking in a pattern of blinks with connections of resistors, breadboard, Jumper wires & Arduino.

### Practical: 5

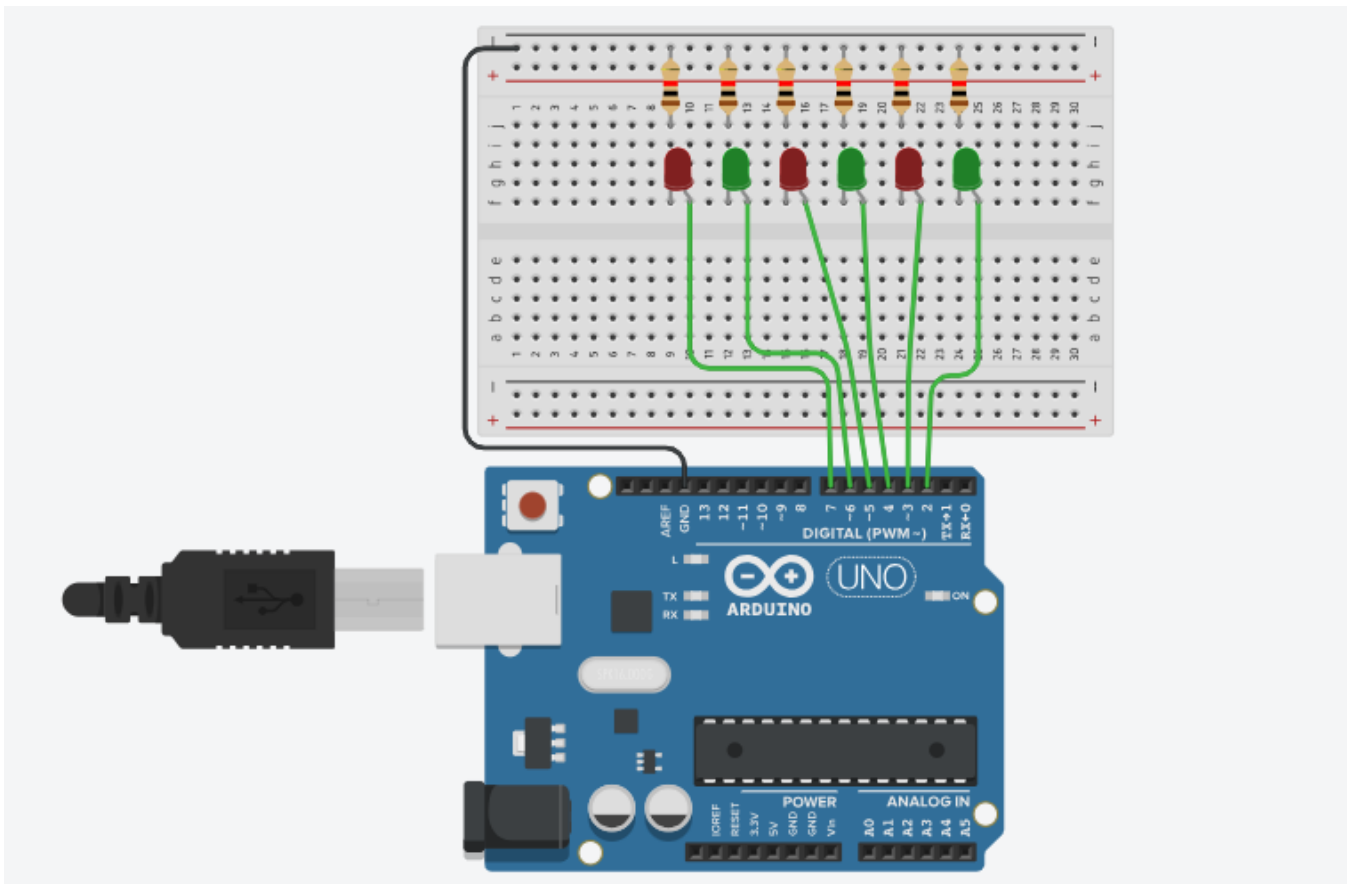
**Aim :**WAP to interface external 6 LED s with Breadboard & Arduino and write a program to blink 6 LEDs , one at a time, in a back and forth formation.

**Objective:**

- To learn Arduino UNO basics
- To learn Breadboard basics
- Write a program to blink Arduino onboard LED and to interface external LED with Breadboard & Arduino.
- Write a program of blinking 6 LEDs –one at a time in a back and forth formation.

**Components:**6 LEDs, 6 Resistors, Arduino Uno r3, wires

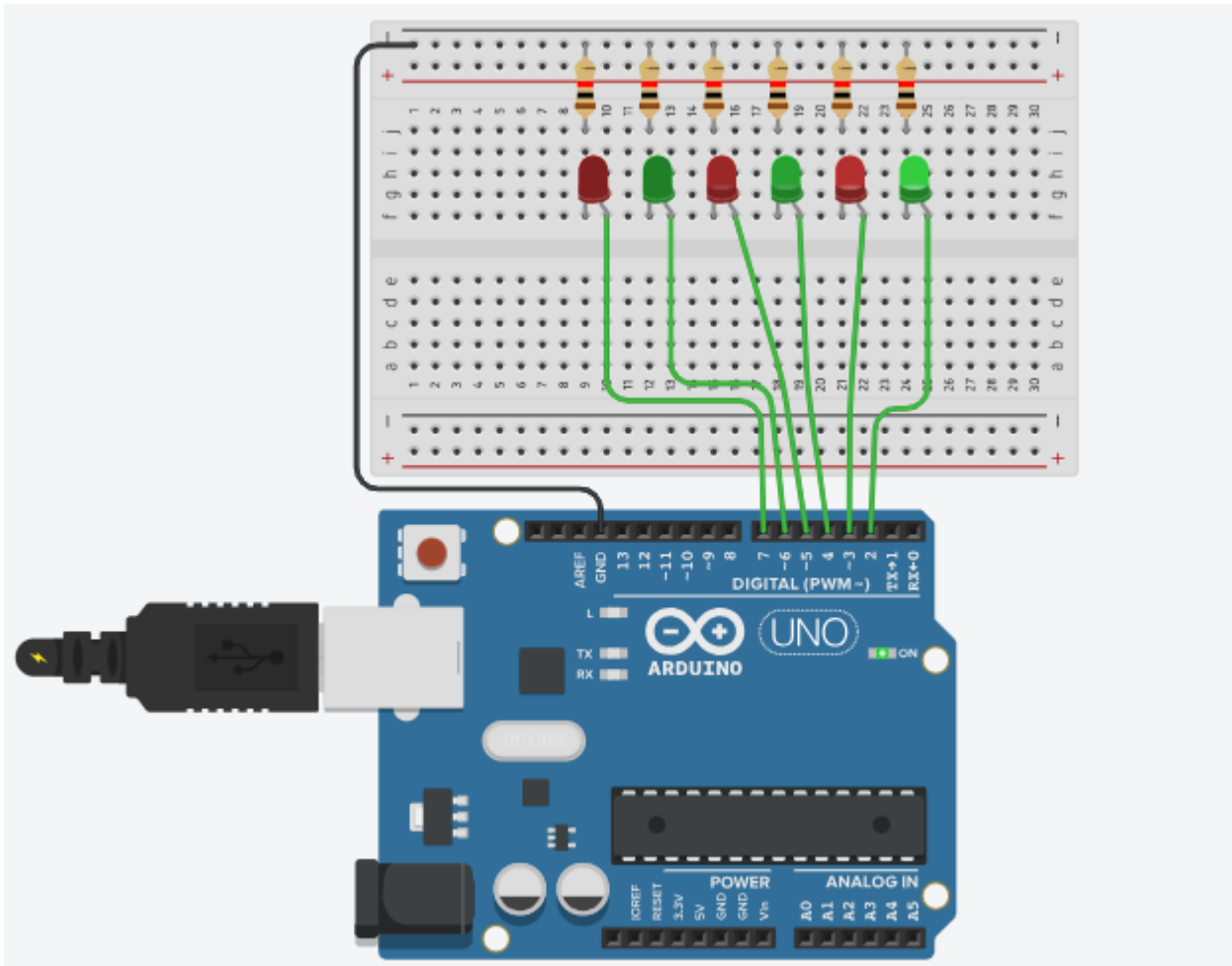
**Circuit:**



**Program Code:**

```
// C++ code
//
int timer = 100;
void setup()
{
  for (int pin = 2; pin < 8; pin++)
  {
    pinMode(pin, OUTPUT);
  }
}
```

```
void loop()
{
  for (int pin = 2; pin < 8; pin++)
  {
    digitalWrite(pin, HIGH);
    delay(timer);
    digitalWrite(pin, LOW);
  }
  for (int pin = 7; pin >= 2; pin--)
  {
    digitalWrite(pin, HIGH);
    delay(timer);
    digitalWrite(pin, LOW);
  }
}
```

**Output:****Conclusion:**

Thus, learnt about basic components of IoT like Arduino UNO (blinking Arduino onboard LED), Breadboard, LED's and interfacing external 6 LEDs Scrolling LEDs blinking one at a time ,in a back and forth formation with connections of resistors, breadboard, Jumper wires & Arduino.

## Practical : 6

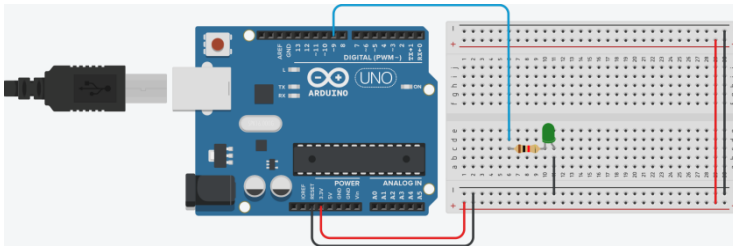
**Aim:** WAP to interface external LED with Breadboard & Arduino and write a program to Fade LED With Arduino Analog Output.

**Objectives:**

- To learn Arduino UNO basics.
- To Learn Breadboard basics.
- Write a program to blink Arduino onboard LED and to interface external LED with Breadboard & Arduino.
- Write a program to Fade LED With Arduino Analog Output.

**Components:** Arduino UNO R3, Breadboard, 1 Resistor, 1 LED, Wires

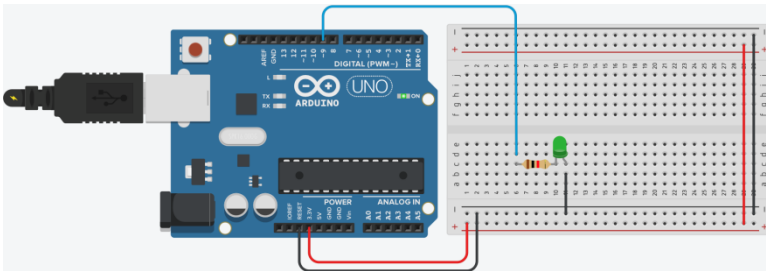
**Circuit:**



**Program Code:**

```
int brightness = 0;
void setup()
{
  pinMode(9, OUTPUT);
}
void loop()
{
  for (brightness = 0; brightness <= 1000; brightness += 5)
  {
    analogWrite(9, brightness);
    delay(50); // Wait for 30 millisecond(s)
  }
  for (brightness = 1000; brightness >= 0; brightness -= 5)
  {
    analogWrite(9, brightness);
    delay(50); // Wait for 30 millisecond(s)
  }
}
```

**Output:**



**Conclusion:**

Thus, learnt about basic components of IoT like Arduino UNO (blinking Arduino onboard LED), Breadboard, LED's and interfacing Fade LED With Arduino Analog Output with connections of resistors, breadboard, Jumper wires & Arduino.

## Practical: 7

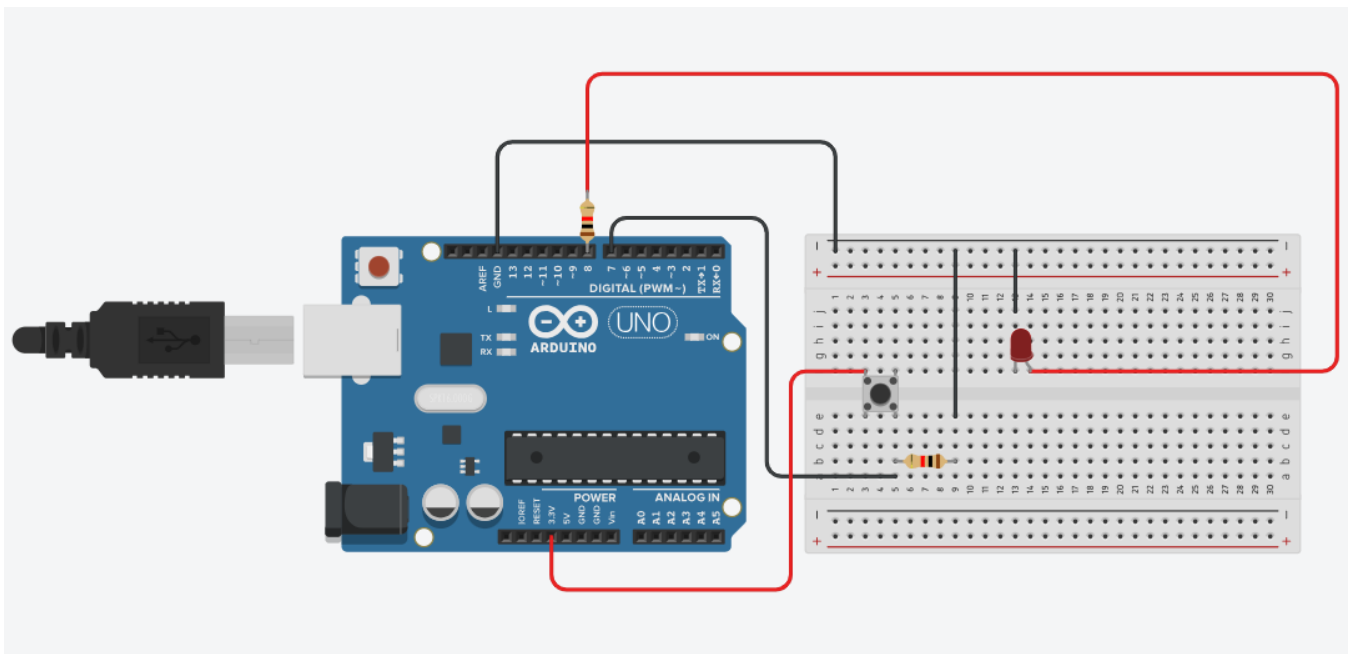
**Aim:** WAP to interface push Button with Arduino and write a program to ON LED when push button is pressed.

### Objective:

- To learn Arduino UNO basics
- To learn Breadboard basics
- To Learn Push buttons/switches
- Programming of interfacing Push Button with Arduino.
- Write a program to turn ON Led when push button is pressed

**Components:** 1 LED, 2 Resistor, Arduino Uno r3, 1 Push button, wires

### Circuit:

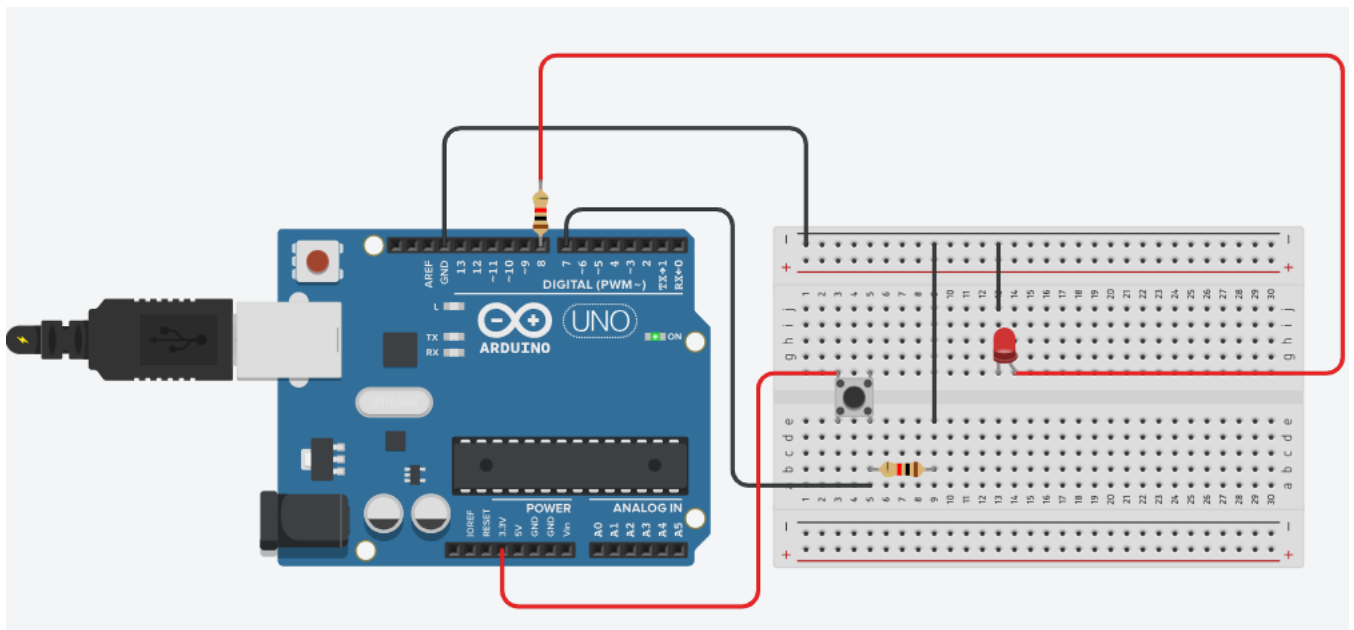


### Program Code:

```
// C++ code
//
unsigned const LED=8;
unsigned const BUTTON=7;
unsigned int buttonState=0;
void setup()
{
  pinMode(BUTTON, INPUT);
  pinMode(LED, OUTPUT);
}

void loop()
{
  if(digitalRead(BUTTON)==1)
  {
```

```
digitalWrite(LED, HIGH);  
buttonState+=1;  
  
if(buttonState%2==0){  
    digitalWrite(LED, LOW);  
    buttonState=0;  
}  
}  
}
```

**Output:****Conclusion:**

Thus, learnt about basic components of IoT like Arduino UNO (blinking Arduino onboard LED), Breadboard, LED's and interfacing push button with Arduino to turn on LED when push button is pressed with connections of resistors, breadboard, Jumper wires & Arduino



### Practical: 8

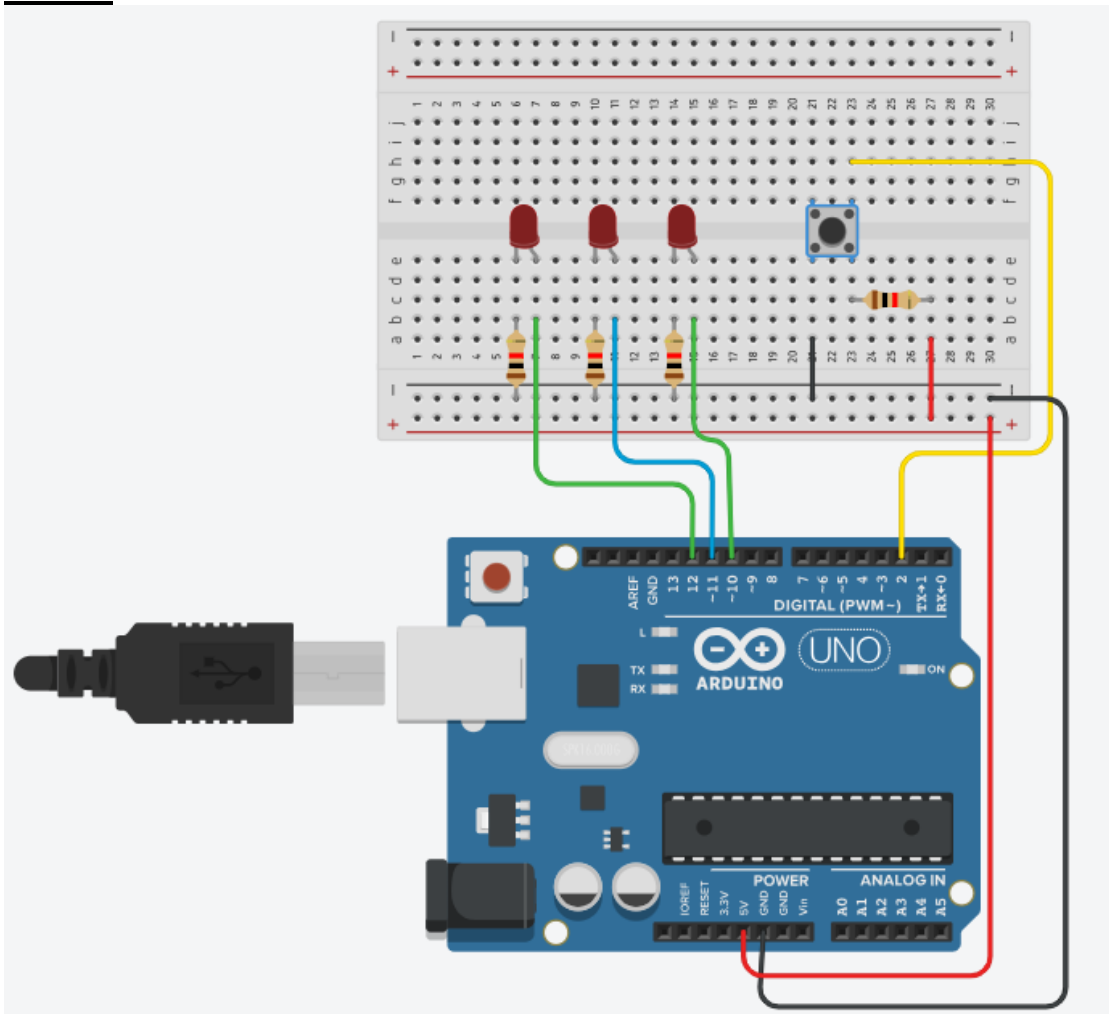
**Aim :** WAP to interface Push Button with Arduino and write a program to turn ON LEDs when push button is pressed and gets OFF after second automatically like a door bell.

**Objective:**

- To learn Arduino UNO basics
- To learn Breadboard basics
- Programming of interfacing Push Button with Arduino
- Write a program to turn ON LEDs when push button is pressed and gets OFF after second automatically.

**Components:** 3 LEDs, 4 Resistors, Arduino Uno r3, 1 Push button, wires

**Circuit:**

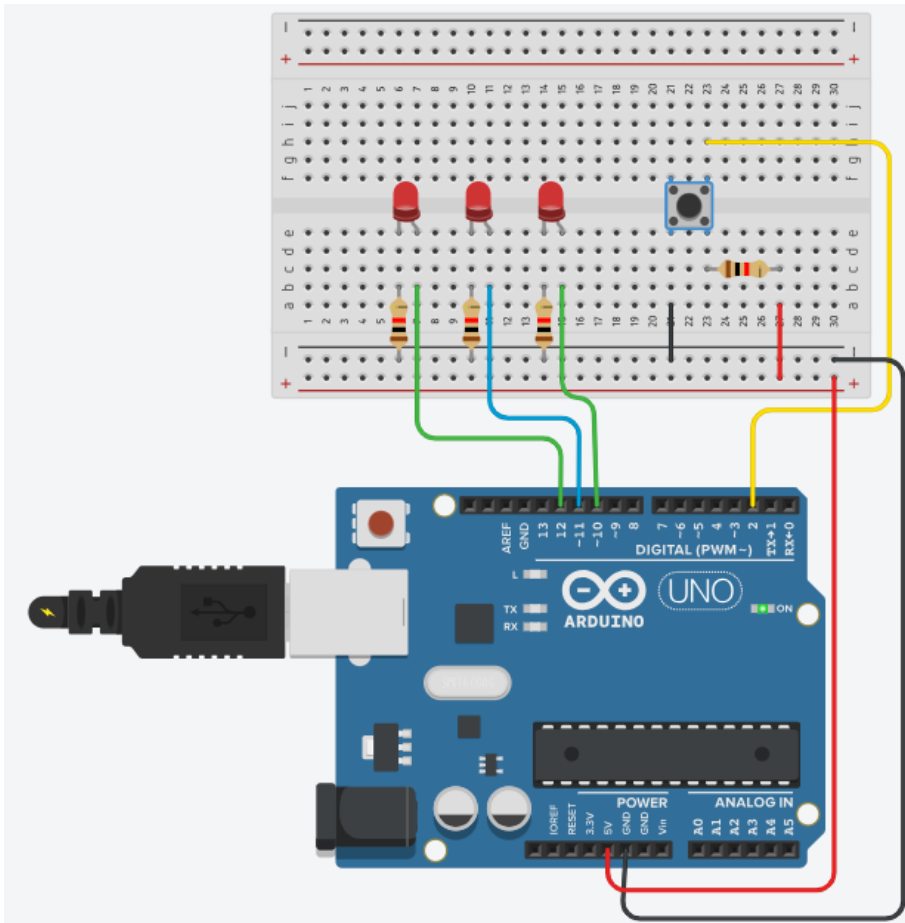


**Program Code:**

```
// C++ code
//
int ledPin1 =10;
int ledPin2 =11;
int ledPin3 =12;
int keyPin =2;
boolean ledOn = false;
void setup()
{
```

```
pinMode(ledPin1, OUTPUT);
pinMode(ledPin2, OUTPUT);
pinMode(ledPin3, OUTPUT);
pinMode(keyPin, INPUT);
}
void loop(){
  int keyState = digitalRead(keyPin);
  if(keyState==0)
  { ledOn = !ledOn;
    digitalWrite(ledPin1, ledOn);
    delay(500);
    digitalWrite(ledPin2, ledOn);
    delay(500);
    digitalWrite(ledPin3, ledOn);
    delay(500);
  }
}
```

### Output:



### Conclusion:

Thus, learnt about basic components of IoT like Arduino UNO (blinking Arduino onboard LED), Breadboard, LED's and interfacing push button with Arduino to turn ON/OFF LEDs when push button is pressed like a door Bell with connections of resistors, breadboard, Jumper wires & Arduino.

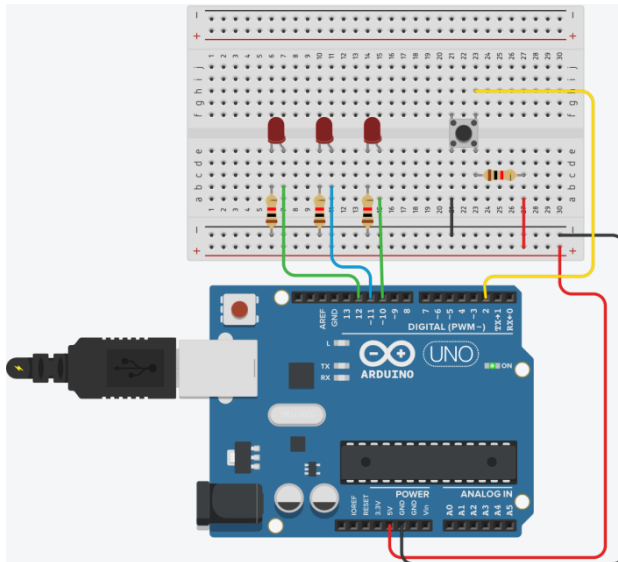
**Practical : 9**

**Aim:** WAP to interface Push Button with Arduino and write a program to turn ON LEDs when push button is pressed and gets OFF when push button is pressed again.

**Objectives:**

- To learn Arduino UNO basics.
- To Learn Breadboard basics.
- To Learn Push buttons / switches.
- Programming of interfacing Push Button with Arduino
- Write a program to turn ON LEDs when push button is pressed and gets OFF when push button is pressed again.

**Components:** Arduino UNO R3, Breadboard, 4 Resistor, 3 LED, 1 Push Button, Wires

**Circuit:****Program Code:**

```
int ledPin1 = 10;
int ledPin2 = 11;
int ledPin3 = 12;
int keyPin = 2;
```

```
bool ledState = false; // Track LED ON/OFF state
bool lastButtonState = HIGH; // Store previous button state
```

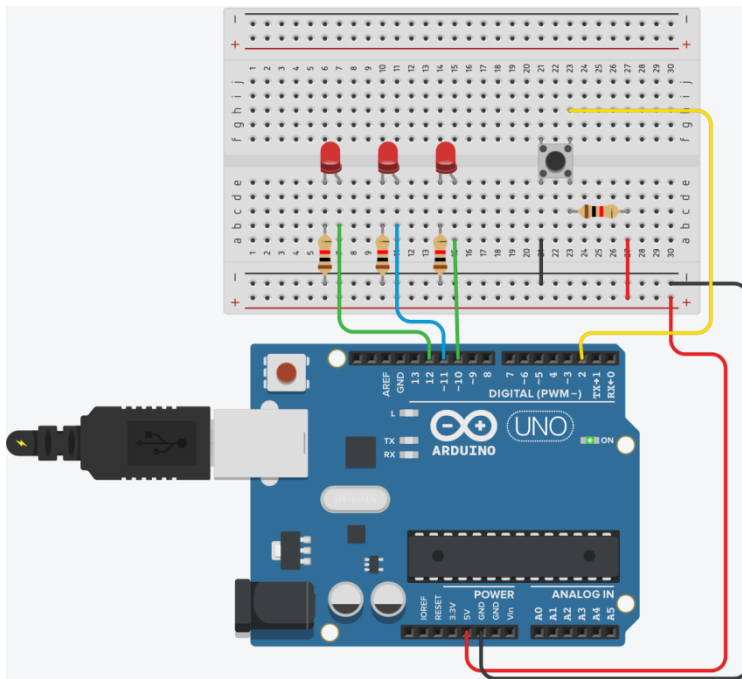
```
void setup()
{
  pinMode(ledPin1, OUTPUT);
  pinMode(ledPin2, OUTPUT);
  pinMode(ledPin3, OUTPUT);
  pinMode(keyPin, INPUT_PULLUP); // Enable internal pull-up resistor
}
```

```
void loop()
{
  bool currentButtonState = digitalRead(keyPin);
```

```
// Detect button press (falling edge: HIGH to LOW)
if (lastButtonState == HIGH && currentButtonState == LOW)
{
    ledState = !ledState; // Toggle LED state

    digitalWrite(ledPin1, ledState ? HIGH : LOW);
    delay(500);
    digitalWrite(ledPin2, ledState ? HIGH : LOW);
    delay(500);
    digitalWrite(ledPin3, ledState ? HIGH : LOW);
    delay(500);
}

lastButtonState = currentButtonState;
}
```

**Output:****Conclusion:**

Thus, learnt about basic components of IoT like Arduino UNO (blinking Arduino onboard LED), Breadboard, LED's and interfacing Fade LED With Arduino Analog Output with connections of resistors, breadboard, Jumper wires & Arduino.

## Practical: 10

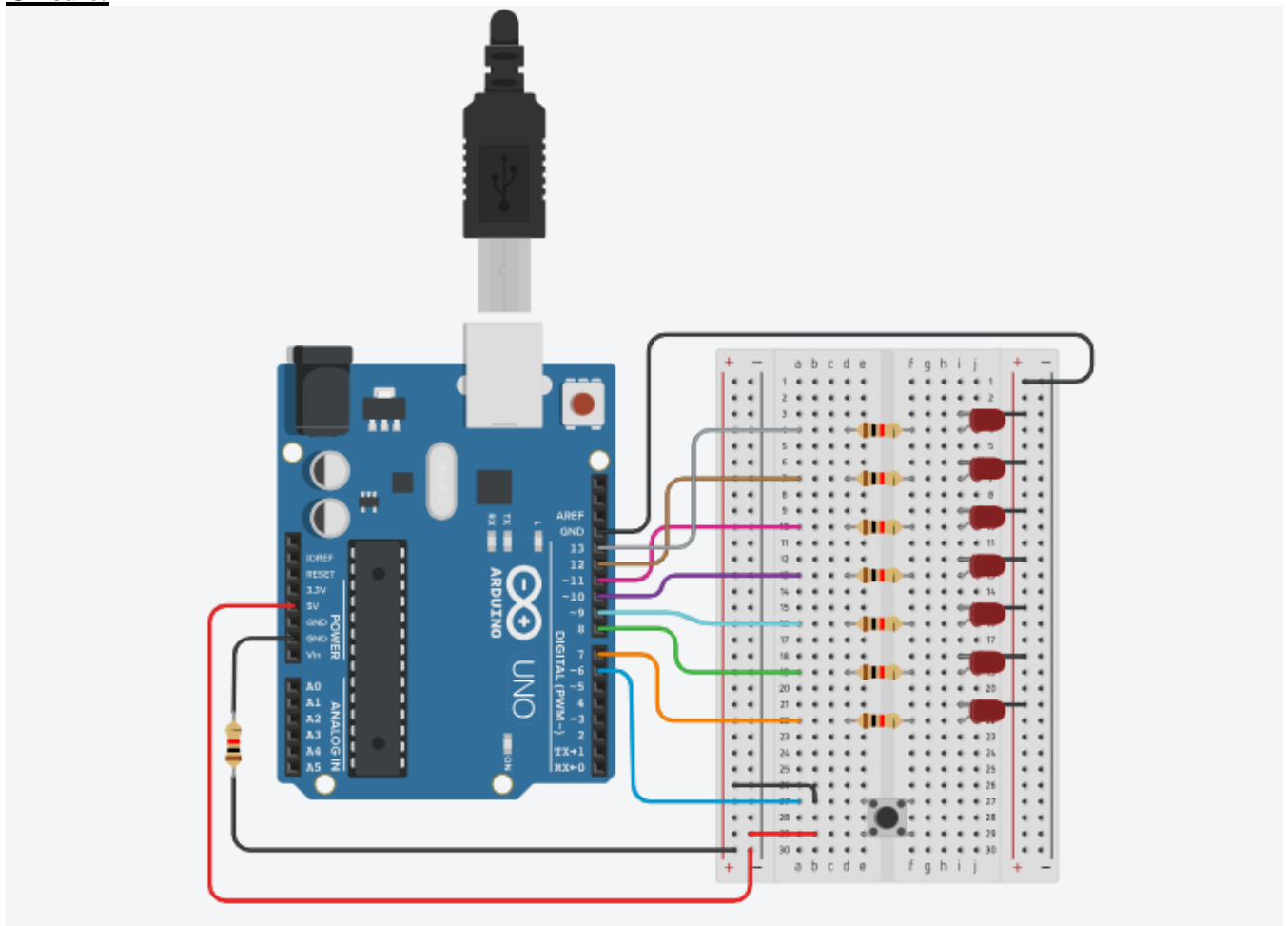
**Aim :** Write a program to interface push button with Arduino and write a program to turn on LEDS when push button is pressed in 7 different patterns.

**Objective:**

- To learn Arduino UNO basics
- To learn BreadBoard basics
- To learn push Buttons/Switches
- Programming of interface Push buttons with Arduino
- Write a program to turn on LEDS when push button is pressed in seven different patterns.

**Components:** 7 LEDs, 8 Resistors, Arduino Uno r3, 1 Push button, wires

**Circuit:**



**Program Code:**

```
// C++ code
//
int L1 = 13;
int L2 = 12;
int L3 = 11;
```

```
int L4 = 10;
int L5 = 9;
int L6 = 8;
int L7 = 7; //7 LED pin
int buttonPin = 6; //the number of the pushbutton pin
int de=50; // delay time
```

```

int p=0; // variable for pattern
int buttonState = 0; // variable for reading the
pushbutton status
void setup() {
  pinMode(L1, OUTPUT);
  pinMode(L2, OUTPUT);
  pinMode(L3, OUTPUT);
  pinMode(L4, OUTPUT);
  pinMode(L5, OUTPUT);
  pinMode(L6, OUTPUT);
  pinMode(L7, OUTPUT);

  pinMode(buttonPin, INPUT);
}
void loop()
{
  buttonState = digitalRead(buttonPin);
  if (buttonState == HIGH)
  {
    p++;
    delay(2000);
  }
  if(p==1)
  {
    digitalWrite(L1,1);
    digitalWrite(L2,0);
    digitalWrite(L3,0);
    digitalWrite(L4,0);
    digitalWrite(L5,0);
    digitalWrite(L6,0);
    digitalWrite(L7,0); //1
    delay(de);

    digitalWrite(L1,0);
    digitalWrite(L2,1);
    digitalWrite(L3,0);
    digitalWrite(L4,0);
    digitalWrite(L5,0);
    digitalWrite(L6,0);
    digitalWrite(L7,0); //2
    delay(de);

    digitalWrite(L1,0);
    digitalWrite(L2,0);
    digitalWrite(L3,1);
    digitalWrite(L4,0);
    digitalWrite(L5,0);
    digitalWrite(L6,0);
    digitalWrite(L7,0); //3
    delay(de);

    digitalWrite(L1,0);
    digitalWrite(L2,0);
    digitalWrite(L3,0);
    digitalWrite(L4,1);
    digitalWrite(L5,0);
    digitalWrite(L6,0);
    digitalWrite(L7,0); //4
    delay(de);
    digitalWrite(L1,0);
    digitalWrite(L2,0);
    digitalWrite(L3,0);

```

```

    digitalWrite(L4,0);
    digitalWrite(L5,1);
    digitalWrite(L6,0);
    digitalWrite(L7,0); //5
    delay(de);
    digitalWrite(L1,0);
    digitalWrite(L2,0);
    digitalWrite(L3,0);
    digitalWrite(L4,0);
    digitalWrite(L5,0);
    digitalWrite(L6,1);
    digitalWrite(L7,0); //6
    delay(de);
    digitalWrite(L1,0);
    digitalWrite(L2,0);
    digitalWrite(L3,0);
    digitalWrite(L4,0);
    digitalWrite(L5,0);
    digitalWrite(L6,0);
    digitalWrite(L7,1); //7
    delay(de);
  }
  if(p==2)
  {
    digitalWrite(L1,0);
    digitalWrite(L2,0);
    digitalWrite(L3,0);
    digitalWrite(L4,0);
    digitalWrite(L5,0);
    digitalWrite(L6,0);
    digitalWrite(L7,1); //7
    delay(de);
    digitalWrite(L1,0);
    digitalWrite(L2,0);
    digitalWrite(L3,0);
    digitalWrite(L4,0);
    digitalWrite(L5,0);
    digitalWrite(L6,1);
    digitalWrite(L7,0); //6
    delay(de);
    digitalWrite(L1,0);
    digitalWrite(L2,0);
    digitalWrite(L3,0);
    digitalWrite(L4,0);
    digitalWrite(L5,1);
    digitalWrite(L6,0);
    digitalWrite(L7,0); //5
    delay(de);
    digitalWrite(L1,0);
    digitalWrite(L2,0);
    digitalWrite(L3,0);
    digitalWrite(L4,1);
    digitalWrite(L5,0);
    digitalWrite(L6,0);
    digitalWrite(L7,0); //4
    delay(de);
    digitalWrite(L1,0);
    digitalWrite(L2,0);
    digitalWrite(L3,1);
    digitalWrite(L4,0);
    digitalWrite(L5,0);
    digitalWrite(L6,0);

```

```

digitalWrite(L7,0); //3
delay(de);
digitalWrite(L1,0);
digitalWrite(L2,1);
digitalWrite(L3,0);
digitalWrite(L4,0);
digitalWrite(L5,0);
digitalWrite(L6,0);
digitalWrite(L7,0); //2
delay(de);
digitalWrite(L1,1);
digitalWrite(L2,0);
digitalWrite(L3,0);
digitalWrite(L4,0);
digitalWrite(L5,0);
digitalWrite(L6,0);
digitalWrite(L7,0); //1
delay(de);
}
if(p==3)
{
digitalWrite(L1,1);
digitalWrite(L2,0);
digitalWrite(L3,0);
digitalWrite(L4,0);
digitalWrite(L5,0);
digitalWrite(L6,0);
digitalWrite(L7,0); //1
delay(de);

digitalWrite(L1,0);
digitalWrite(L2,1);
digitalWrite(L3,0);
digitalWrite(L4,0);
digitalWrite(L5,0);
digitalWrite(L6,0);
digitalWrite(L7,0); //2
delay(de);
digitalWrite(L1,0);
digitalWrite(L2,0);
digitalWrite(L3,1);
digitalWrite(L4,0);
digitalWrite(L5,0);
digitalWrite(L6,0);
digitalWrite(L7,0); //3
delay(de);
digitalWrite(L1,0);
digitalWrite(L2,0);
digitalWrite(L3,0);
digitalWrite(L4,1);
digitalWrite(L5,0);
digitalWrite(L6,0);
digitalWrite(L7,0); //4
delay(de);
digitalWrite(L1,0);
digitalWrite(L2,0);
digitalWrite(L3,0);
digitalWrite(L4,0);
digitalWrite(L5,1);
digitalWrite(L6,0);
digitalWrite(L7,0); //5
delay(de);

```

```

digitalWrite(L1,0);
digitalWrite(L2,0);
digitalWrite(L3,0);
digitalWrite(L4,0);
digitalWrite(L5,0);
digitalWrite(L6,1);
digitalWrite(L7,0); //6
delay(de);
digitalWrite(L1,0);
digitalWrite(L2,0);
digitalWrite(L3,0);
digitalWrite(L4,0);
digitalWrite(L5,0);
digitalWrite(L6,0);
digitalWrite(L7,1); //7
delay(de);
digitalWrite(L1,0);
digitalWrite(L2,0);
digitalWrite(L3,0);
digitalWrite(L4,0);
digitalWrite(L5,0);
digitalWrite(L6,1);
digitalWrite(L7,0); //6
delay(de);
digitalWrite(L1,0);
digitalWrite(L2,0);
digitalWrite(L3,0);
digitalWrite(L4,0);
digitalWrite(L5,1);
digitalWrite(L6,0);
digitalWrite(L7,0); //5
delay(de);
digitalWrite(L1,0);
digitalWrite(L2,0);
digitalWrite(L3,0);
digitalWrite(L4,1);
digitalWrite(L5,0);
digitalWrite(L6,0);
digitalWrite(L7,0); //4
delay(de);
digitalWrite(L1,0);
digitalWrite(L2,0);
digitalWrite(L3,1);
digitalWrite(L4,0);
digitalWrite(L5,0);
digitalWrite(L6,0);
digitalWrite(L7,0); //3
delay(de);
digitalWrite(L1,0);
digitalWrite(L2,1);
digitalWrite(L3,0);
digitalWrite(L4,0);
digitalWrite(L5,0);
digitalWrite(L6,0);
digitalWrite(L7,0); //2
delay(de);
}
if(p==4)
{
digitalWrite(L1,1);
digitalWrite(L2,0);
digitalWrite(L3,0);

```

```

digitalWrite(L4,0);
digitalWrite(L5,0);
digitalWrite(L6,0);
digitalWrite(L7,1); //1,7
  delay(de);
digitalWrite(L1,0);
digitalWrite(L2,1);
digitalWrite(L3,0);
digitalWrite(L4,0);
digitalWrite(L5,0);
digitalWrite(L6,1);
digitalWrite(L7,0); //2,6
  delay(de);
digitalWrite(L1,0);
digitalWrite(L2,0);
digitalWrite(L3,1);
digitalWrite(L4,0);
digitalWrite(L5,1);
digitalWrite(L6,0);
digitalWrite(L7,0); //3,5
  delay(de);
digitalWrite(L1,0);
digitalWrite(L2,0);
digitalWrite(L3,0);
digitalWrite(L4,1);
digitalWrite(L5,0);
digitalWrite(L6,0);
digitalWrite(L7,0); //4
  delay(de);
}
  if(p==5)
  {
digitalWrite(L1,0);
digitalWrite(L2,0);
digitalWrite(L3,0);
digitalWrite(L4,1);
digitalWrite(L5,0);
digitalWrite(L6,0);
digitalWrite(L7,0); //4
  delay(de);
digitalWrite(L1,0);
digitalWrite(L2,0);
digitalWrite(L3,1);
digitalWrite(L4,0);
digitalWrite(L5,1);
digitalWrite(L6,0);
digitalWrite(L7,0); //3,5
  delay(de);
digitalWrite(L1,0);
digitalWrite(L2,1);
digitalWrite(L3,0);
digitalWrite(L4,0);
digitalWrite(L5,0);
digitalWrite(L6,1);
digitalWrite(L7,0); //2,6

```

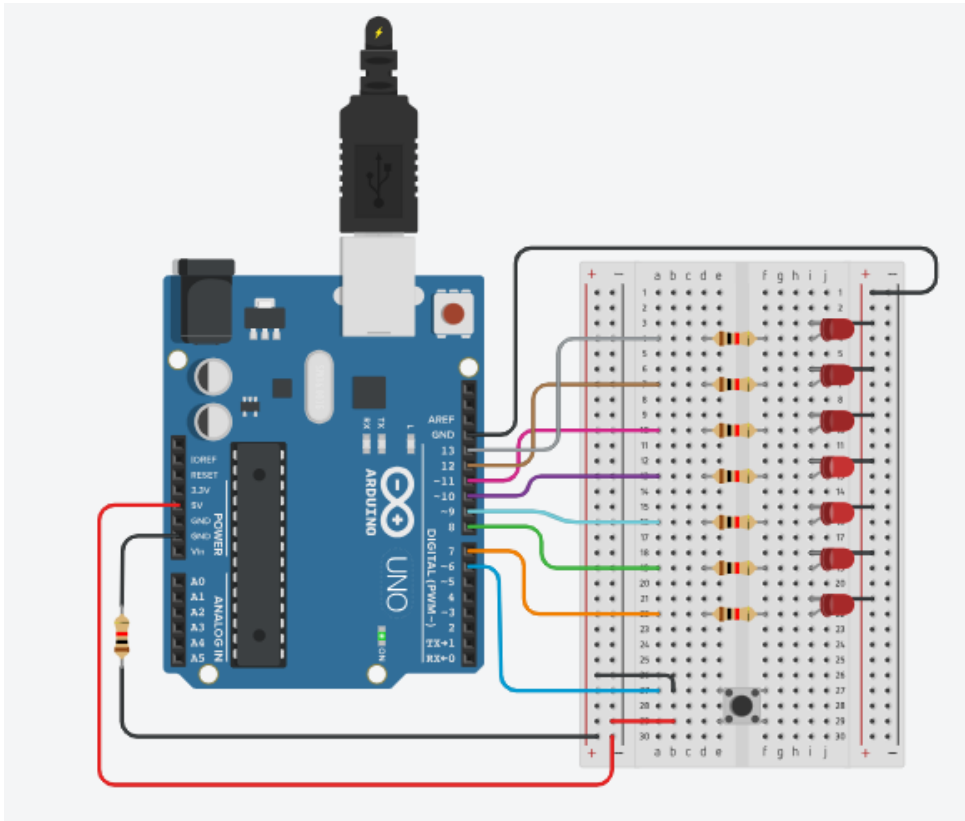
```

  delay(de);
digitalWrite(L1,1);
digitalWrite(L2,0);
digitalWrite(L3,0);
digitalWrite(L4,0);
digitalWrite(L5,0);
digitalWrite(L6,0);
digitalWrite(L7,1); //1,7
  delay(de);
}
if(p==6)
{
digitalWrite(L1,1);
delay(de);
digitalWrite(L2,1);
delay(de);
digitalWrite(L3,1);
delay(de);
digitalWrite(L4,1);
delay(de);
digitalWrite(L5,1);
delay(de);
digitalWrite(L6,1);
delay(de);
digitalWrite(L7,1); //1,7
delay(de);
digitalWrite(L7,0); //1,7
delay(de);
digitalWrite(L6,0);
delay(de);
digitalWrite(L5,0);
delay(de);
digitalWrite(L4,0);
delay(de);
digitalWrite(L3,0);
delay(de);
digitalWrite(L2,0);
delay(de);
digitalWrite(L1,0);
delay(de);
}
if(p==7)
{

digitalWrite(L1,0);
digitalWrite(L2,0);
digitalWrite(L3,0);
digitalWrite(L4,0);
digitalWrite(L5,0);
digitalWrite(L6,0);
digitalWrite(L7,0); //1,7
p=0;
}
}

```



**Output:****Conclusion:**

Thus learnt about basic components of IOT like Arduino UNO (Blink Arduino onboard LED), Breadboard, LEDs and Interfacing push button with Arduino to turn on 7 LEDs when push button is press in 7 different patterns with connection of resistors breadboard, jumper wires and Arduino.

**Practical: 11**

**Aim:** Write a program to interface RGB LED with Arduino and write a program to use RGB LED with Arduino to obtain different colors.

**Objectives:**

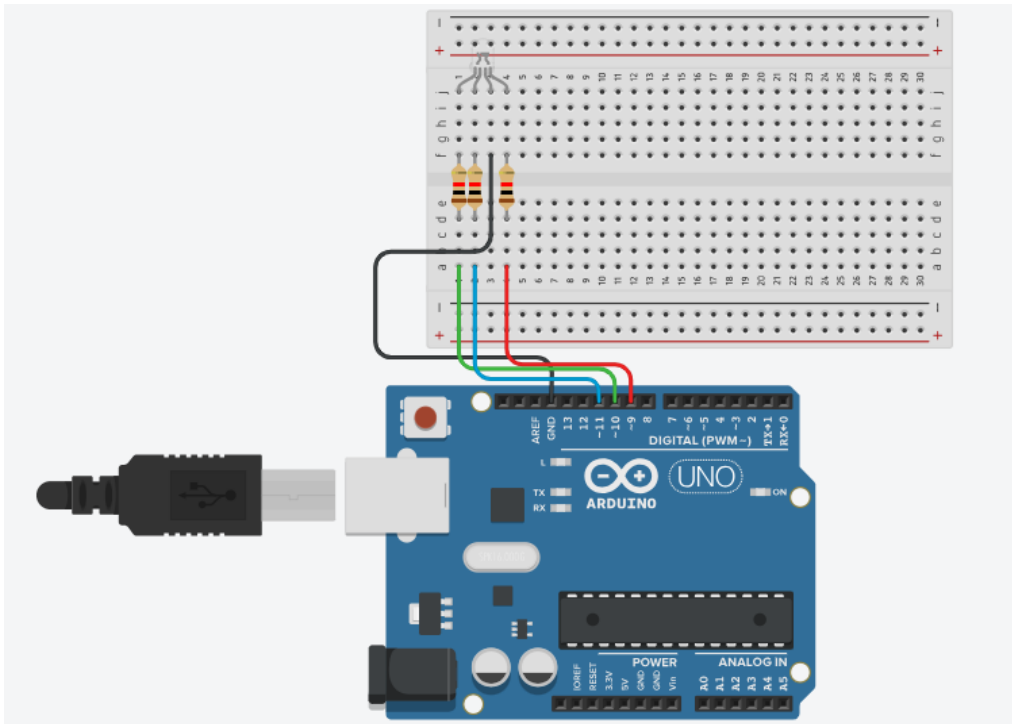
To learn Arduino UNO basics

To learn BreadBoard basics

To learn RGB LED

write a program to use RGB LED with Arduino to obtain different colors.

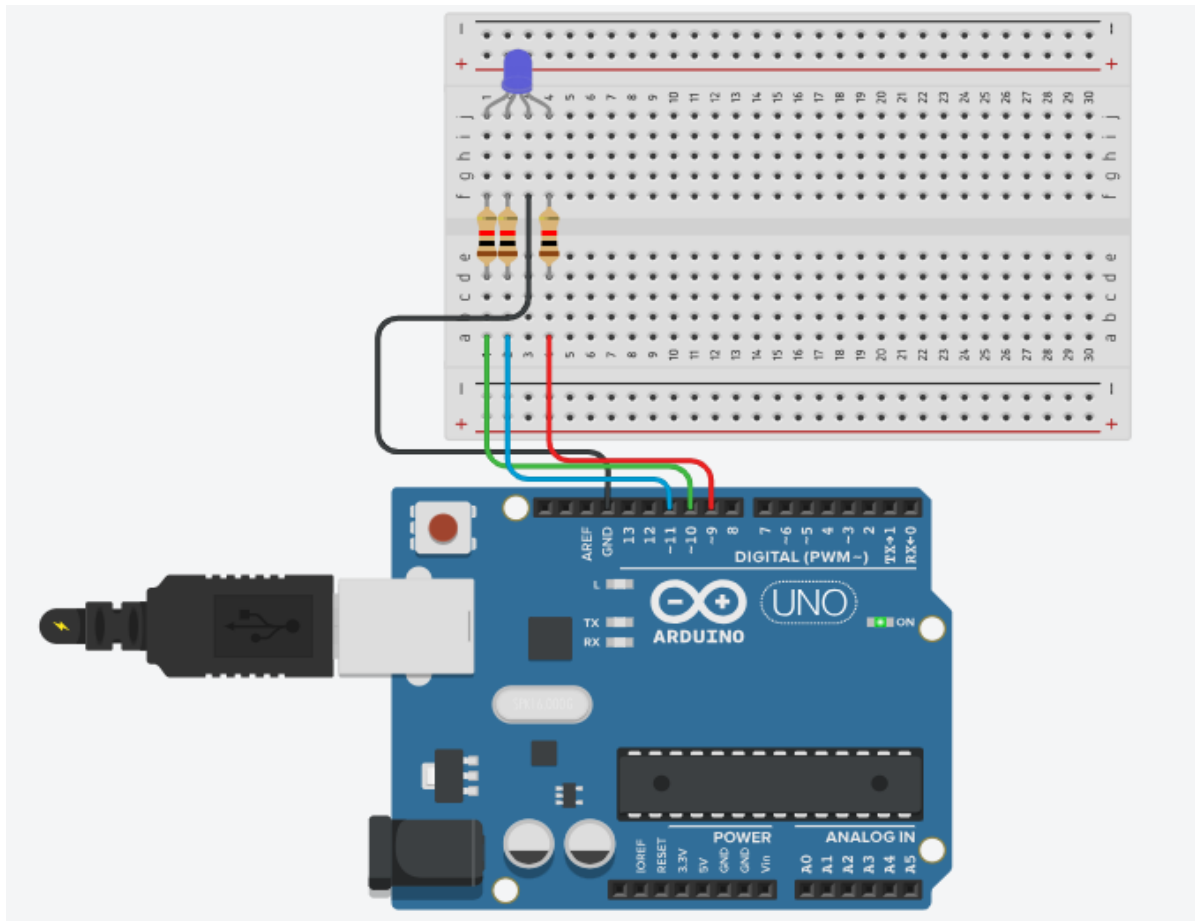
**Components:** 1 RGB, 3 Resistors, Arduino Uno r3, wires

**Circuit:****9.1 Program Code:**

```
// C++ code
int red_light_pin= 9;
int green_light_pin = 10;
int blue_light_pin = 11;
void setup()
{
  pinMode(red_light_pin, OUTPUT);
  pinMode(green_light_pin, OUTPUT);
  pinMode(blue_light_pin, OUTPUT);
}
void loop()
{
  digitalWrite(red_light_pin, HIGH); // Red
```

```
delay(1000);
digitalWrite(red_light_pin, LOW);
//delay(1000);
digitalWrite(green_light_pin, HIGH); // Green
delay(1000);
digitalWrite(green_light_pin, LOW);
delay(1000); //While putting this line
as comment....
digitalWrite(blue_light_pin, HIGH); // Blue
delay(1000);
digitalWrite(blue_light_pin, LOW);
//delay(1000);
}
```

**Output:**

**Conclusion:**

Thus learnt about basic components of IOT like Arduino UNO (Blink Arduino on board LED) , Breadboard, LEDS and Interfacing RGB LED with Arduino to turn on multiple colors as RED, GREEN AND BLUE with connection of resistors breadboard, jumper wires and Arduino.

## Practical: 12

**Aim:** Write a program to interface RGB LED with Arduino and write a program to use RGB LED with Arduino to obtain different colors.

**Objectives:**

To learn Arduino UNO basics

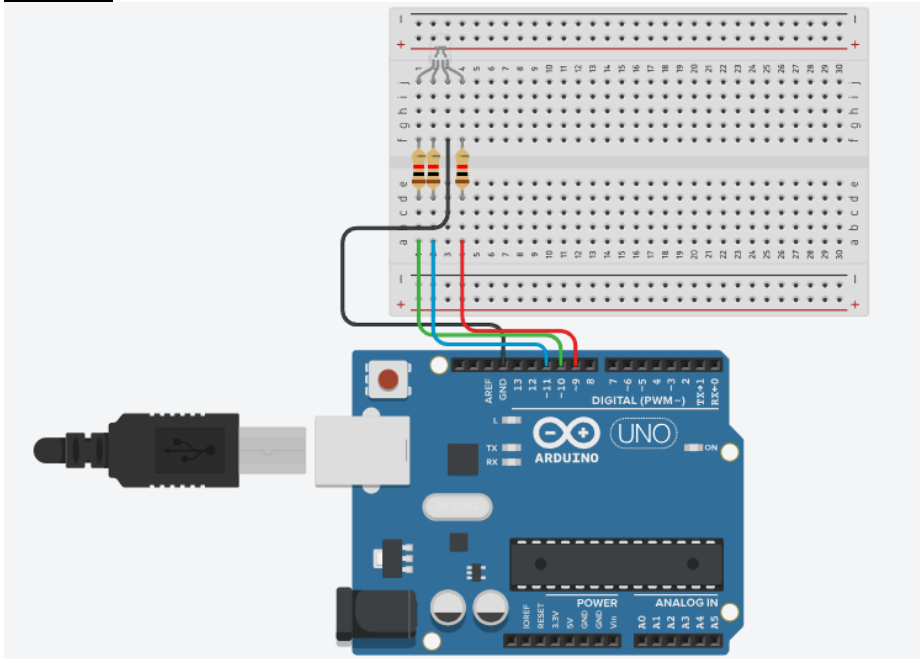
To learn BreadBoard basics

To learn RGB LED

write a program to use RGB LED with Arduino to obtain different colors.

**Components:** 1 RGB, 3 Resistors, Arduino Uno r3, wires

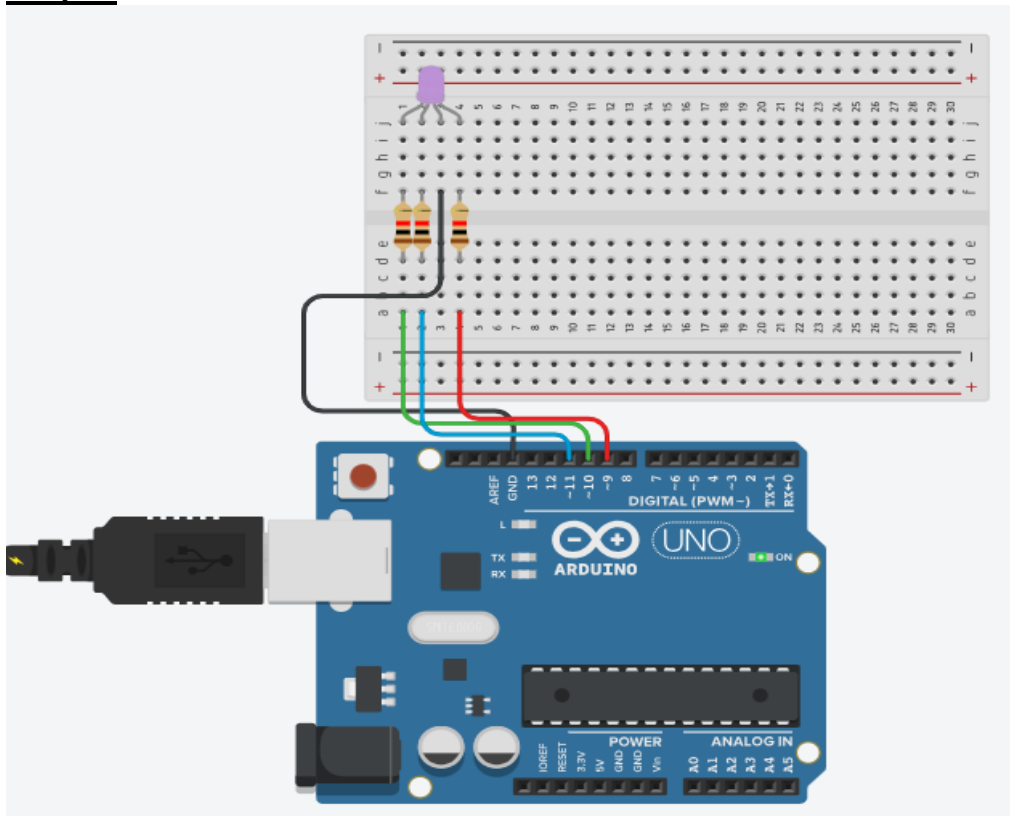
**Circuit:**



### 9.2 ---- Program Code:

```
int red_light_pin= 9;
int green_light_pin = 10;
int blue_light_pin = 11;
void setup()
{pinMode(red_light_pin, OUTPUT);
pinMode(green_light_pin, OUTPUT);
pinMode(blue_light_pin, OUTPUT);
}
void loop(){ RGB_color(0,0,255);
delay(1000);
RGB_color(153,0,153);
delay(1000);
RGB_color(128,0,255);
delay(1000);
RGB_color(34,139,34);
delay(1000);
```

```
RGB_color(0,255,255);
delay(1000);
RGB_color(138,43,226);
delay(1000);
RGB_color(139,69,19);
delay(1000);
RGB_color(0,204,102);
delay(1000);
RGB_color(255,20,147);
delay(1000);
}
void RGB_color(int red_light_value,int
green_light_value, int blue_light_value){
analogWrite(red_light_pin, red_light_value);
analogWrite(green_light_pin, green_light_value);
analogWrite(blue_light_pin, blue_light_value);
}
```

**Output:****Conclusion:**

Thus learnt about basic components of IOT like Arduino UNO (Blink Arduino on board LED) , Breadboard, LEDS and Interfacing RGB LED with Arduino to turn on multiple colors as RED, GREEN AND BLUE with connection of resistors breadboard, jumper wires and Arduino.

**Practical :13**

**Aim:** Write a program to interface ON/OFF LED with Arduino and write a program to use the Slide Button with Arduino to obtain LED ON & OFF.

**Objectives:**

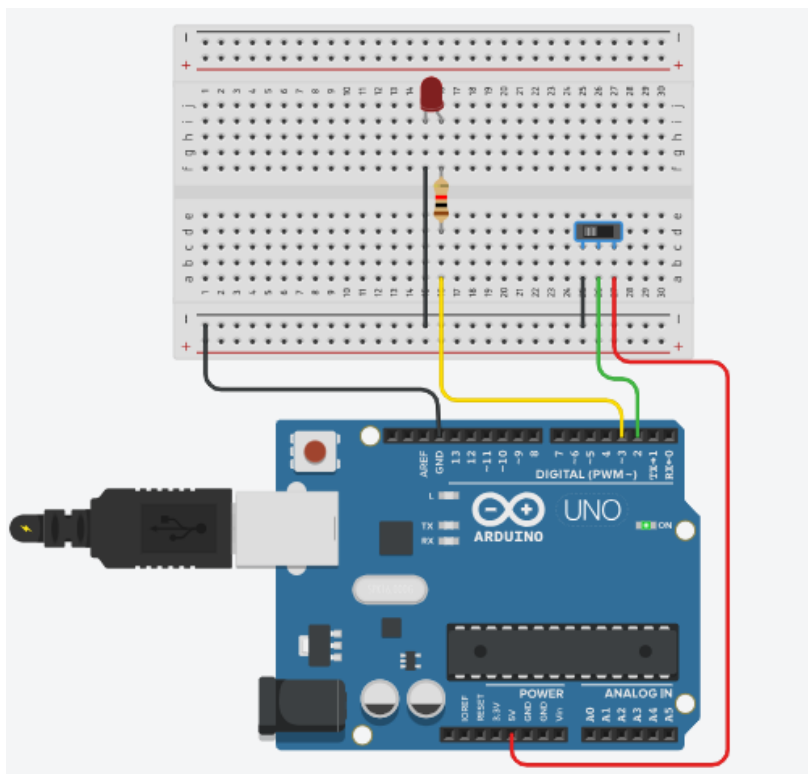
To learn Arduino UNO basics

To learn Breadboard basics

To learn RGB LED

write a program to use the Slide Button with Arduino to obtain LED ON & OFF.

**Components:** 1 LED, 1 Resistor, 1 breadboard, Arduino Uno r3, 1 slide button, wires.

**Circuit:****Program Code:**

// C++ code

int BUTTON=2;

int LED =3;

int BUTTONstate=0;

void setup()

{

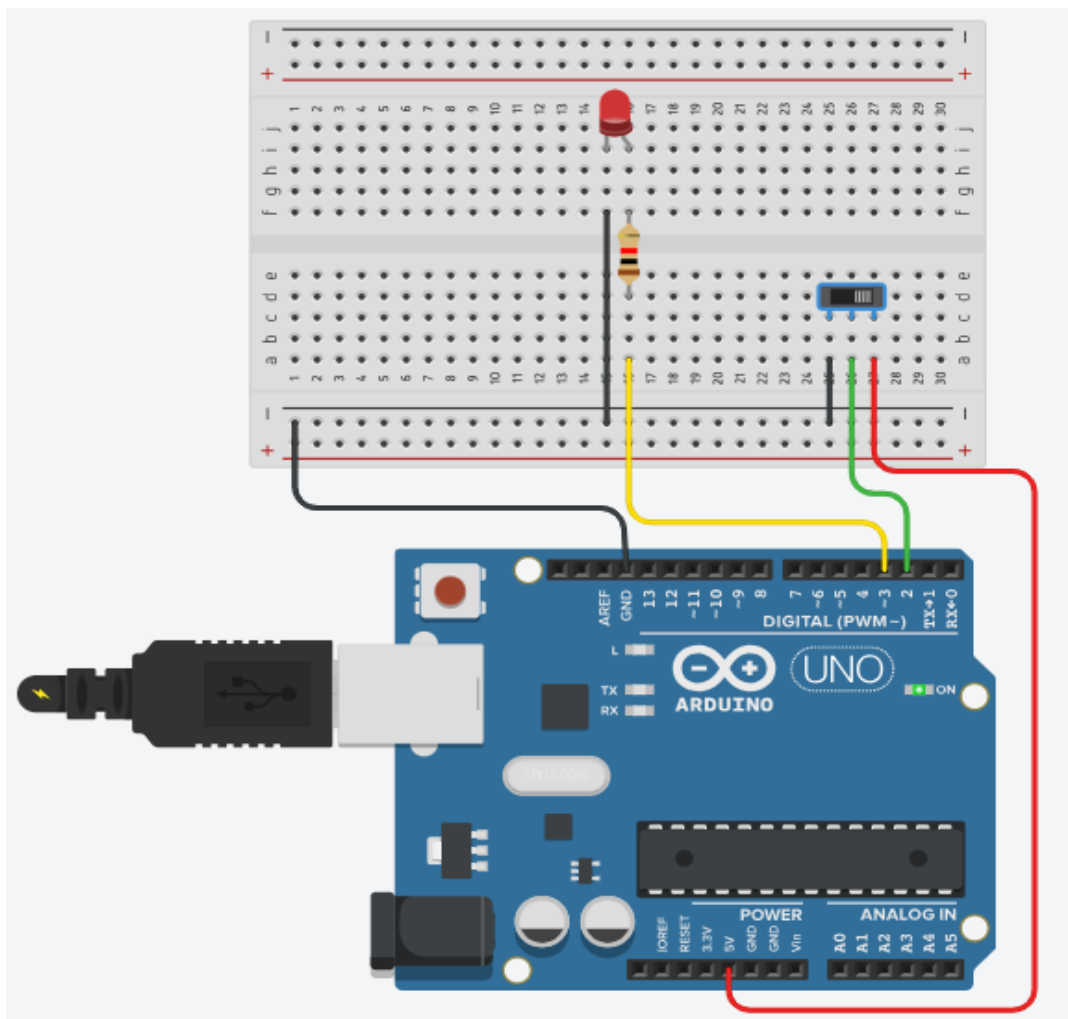
pinMode(BUTTON, INPUT);

pinMode(LED, OUTPUT);

}

void loop()

```
{
  BUTTONstate=digitalRead(BUTTON);
  if(BUTTONstate==HIGH)
  {
    digitalWrite(LED, HIGH);
  }
  else
  {
    digitalWrite(LED, LOW);
  }
}
```

**Output:****Conclusion:**

Thus learnt about basic components of IOT like Arduino UNO (Blink Arduino on board LED), Breadboard, LEDS and Interfacing LED with Arduino to turn on with Slide button ON and OFF with connection of resistors breadboard, jumper wires and Arduino.

**Practical :14**

**Aim:** Write a program to interface Piezo Speaker with Arduino and Write a program to use the Piezo Speaker with Arduino to obtain Continuous buzzer sound.

**Objectives:**

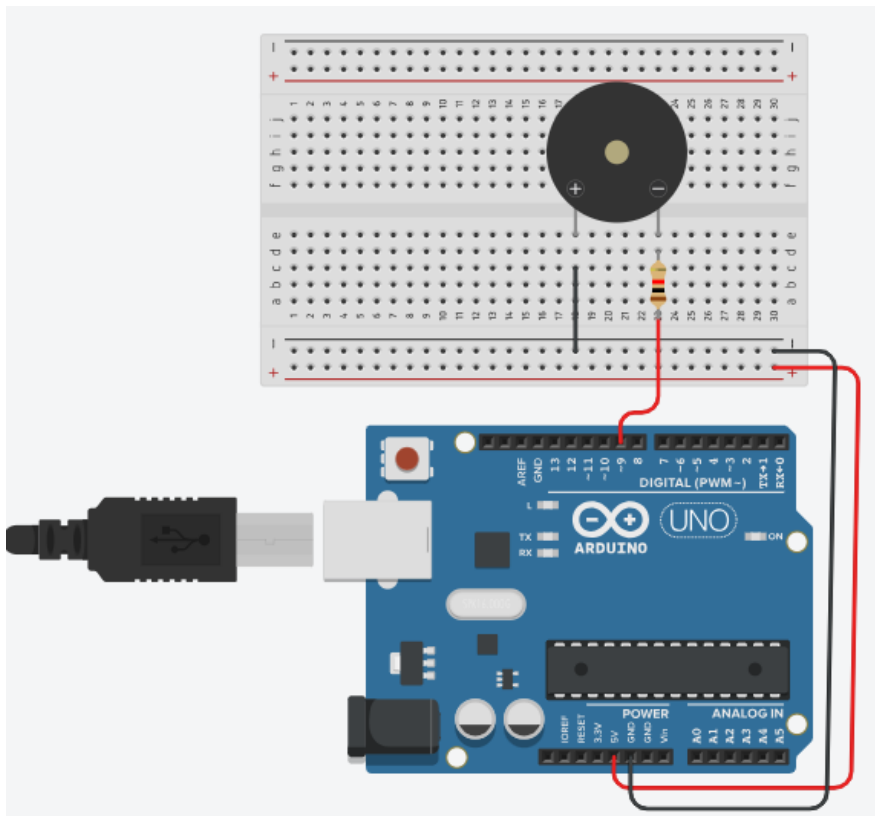
To learn Arduino UNO basics

To learn BreadBoard basics

To learn Piezo Speaker

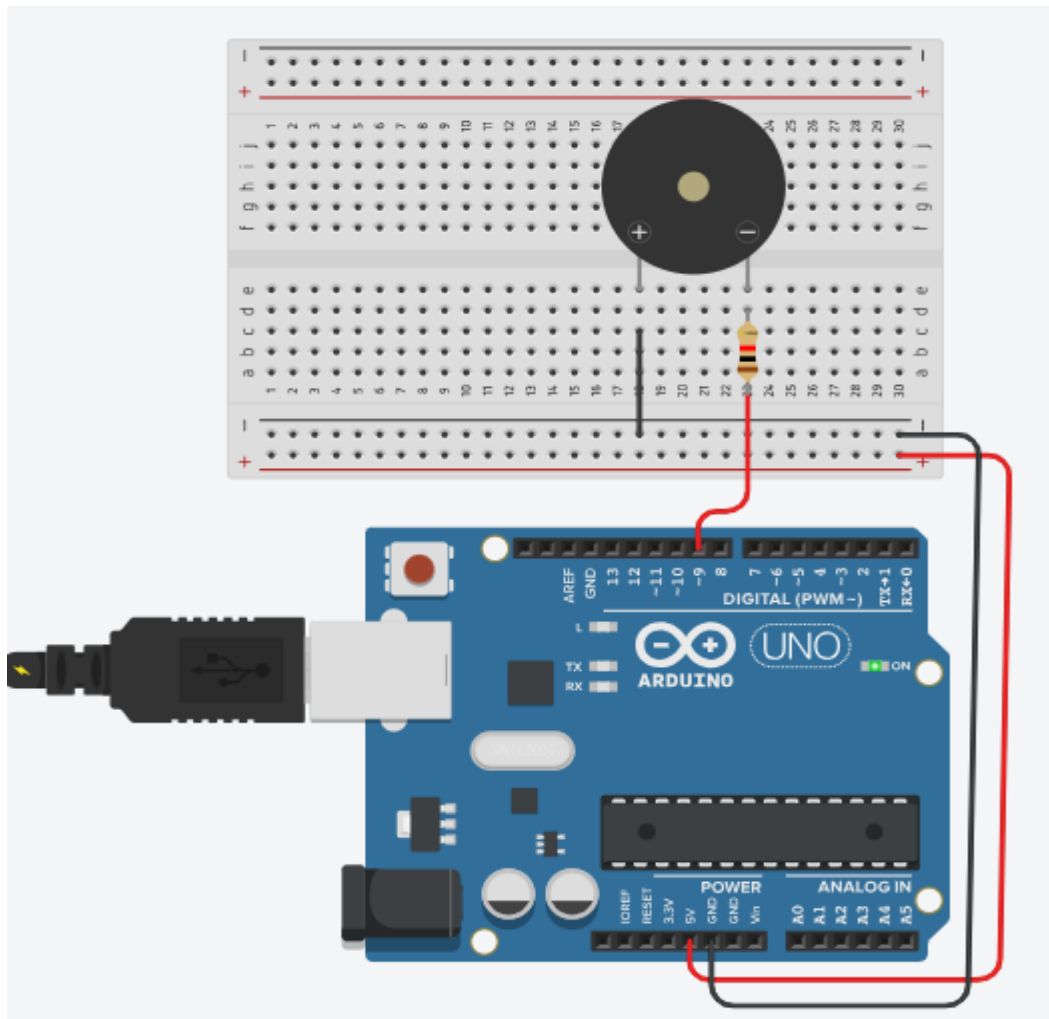
Write a program to use the Piezo Speaker with Arduino to obtain Continuous buzzer sound.

**Components:** 1 Piezo Speaker, 1 Breadboard, 1 Resistor, Arduino Uno r3, wires.

**Circuit:****Program Code:**

```
// C++ code
const int buzzer = 9;
void setup()
{
  pinMode(buzzer, OUTPUT);
}
void loop()
{
  tone(buzzer, 1000);
  delay(1000); // Wait for 1000 millisecond(s)
  noTone(buzzer);
  delay(1000); // Wait for 1000 millisecond(s)
}
```



**Output:****Conclusion:**

Thus learnt about basic components of IOT like Arduino UNO (Blink Arduino on board LED), Breadboard, LEDs and Interfacing Piezo Speaker with Arduino for continuous Buzz Sound with connection of resistors breadboard, jumper wires and Arduino.

**Practical :15**

**Aim:** Write a program to interface Piezo Speaker with Arduino and Write a program to use the Piezo Speaker with Arduino to obtain on Switch button ON/OFF .

**Objectives:**

To learn Arduino UNO basics

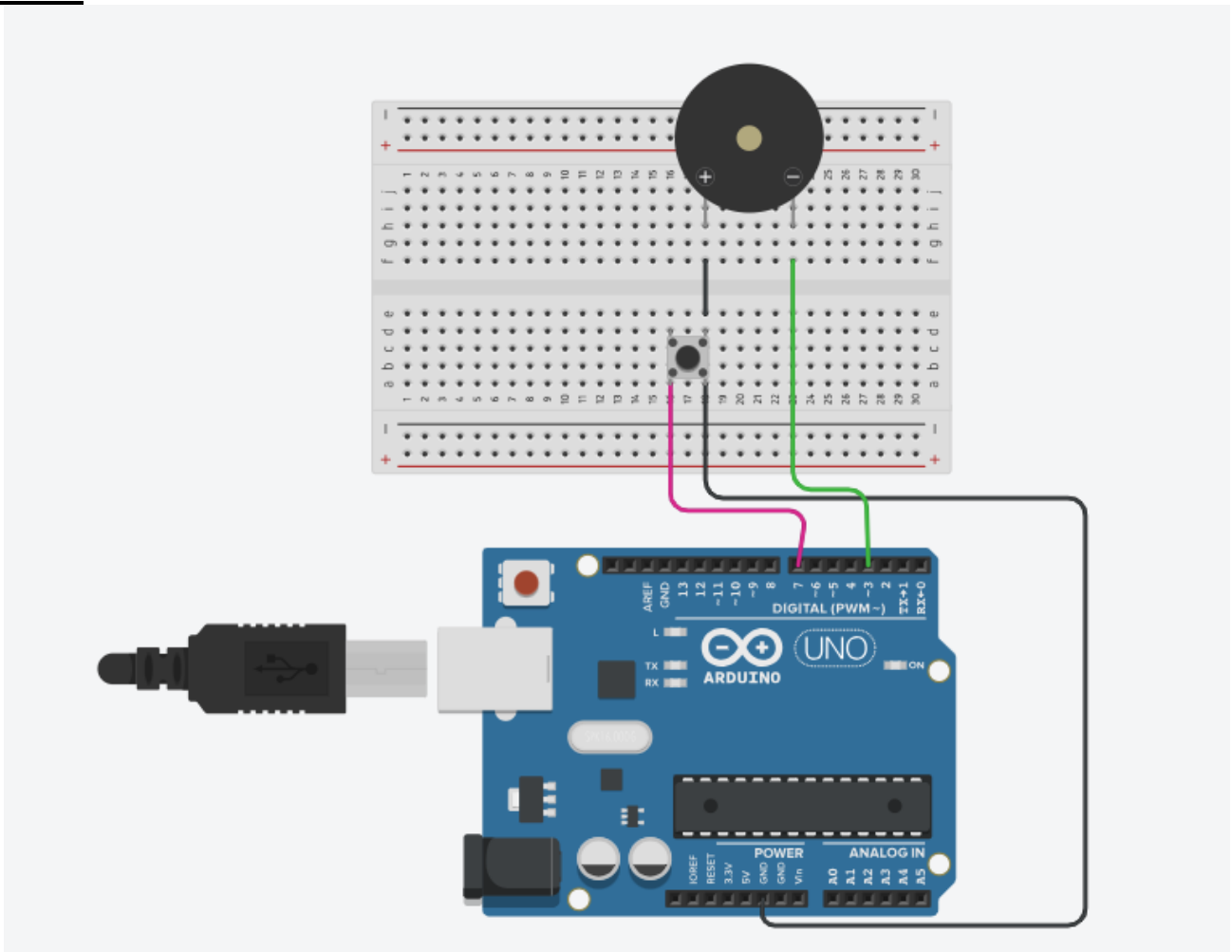
To learn BreadBoard basics

To learn Piezo Speaker

To learn Push Button Switch

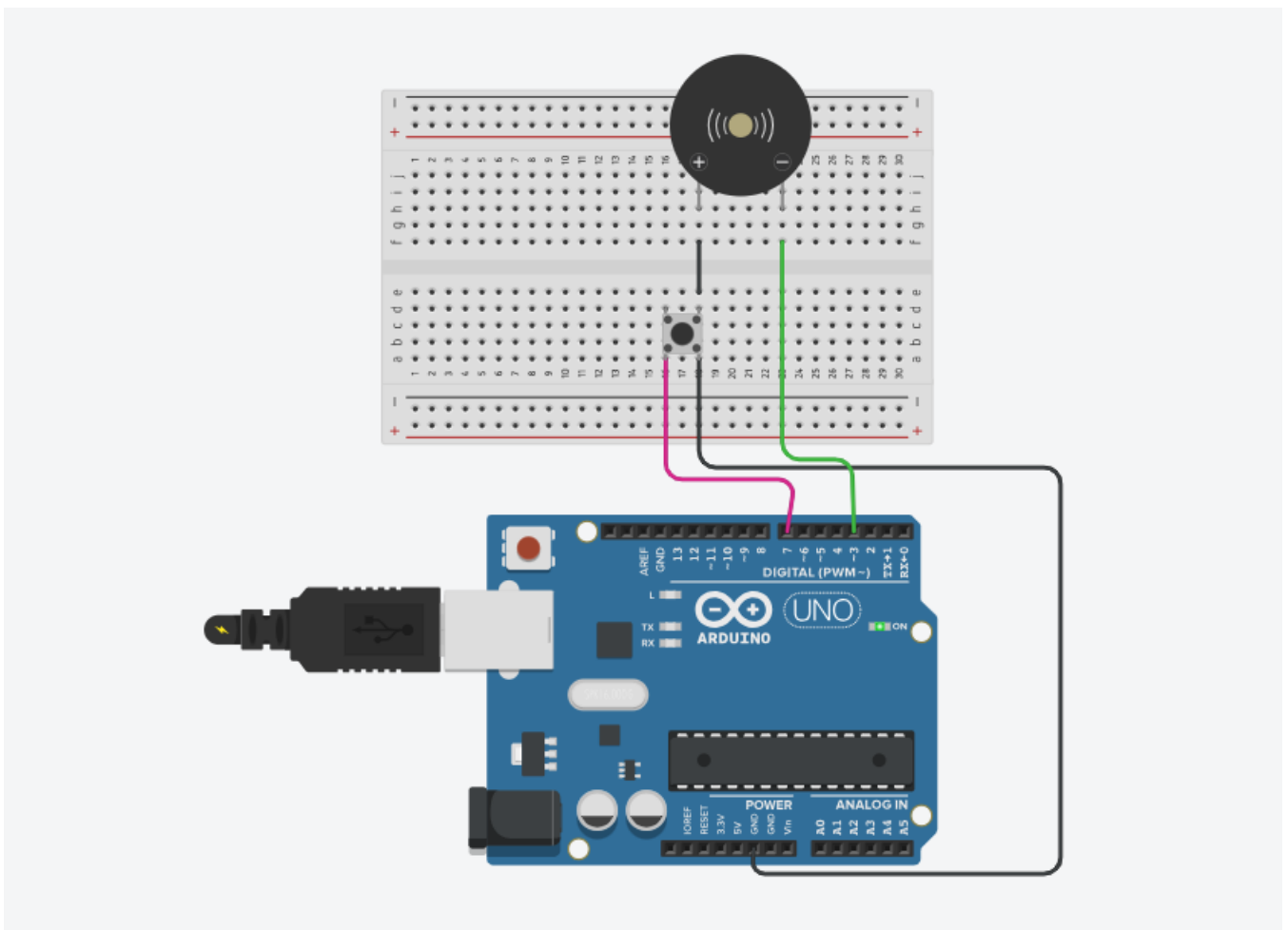
Write a program to use the Piezo Speaker with Arduino to obtain on Switch buttons ON/OFF.

**Components:** 1 Piezo Speaker, 1 Push button, 1 breadboard, Arduino Uno r3, wires.

**Circuit:****Program Code:**

```
// C++ code
const int BUTTON_PIN=7;
const int BUZZER_PIN=3;
void setup()
{
  Serial.begin(9600);
  pinMode(BUTTON_PIN, INPUT_PULLUP);
  pinMode(BUZZER_PIN, OUTPUT);
}
```

```
}  
void loop()  
{  
  int buttonState = digitalRead(BUTTON_PIN);  
  if(buttonState == LOW){  
    Serial.println("The button is being pressed");  
    digitalWrite(BUZZER_PIN, HIGH);  
  }  
  else  
    if(buttonState == HIGH){  
      Serial.println("The button is being unpressed");  
      digitalWrite(BUZZER_PIN, LOW);  
    }  
}
```

**Output:****Conclusion:**

Thus learnt about basic components of IOT like Arduino UNO (Blink Arduino on board LED), Breadboard, LEDs and Interfacing Piezo Speaker with Arduino for Buzz Sound on Switch button ON/OFF with connection of resistors breadboard, jumper wires and Arduino.

**Practical :16**

**Aim:** Write a program to interface Piezo Speaker with Arduino and Write a program to use the Piezo Speaker LED ON/OFF with Arduino to obtain on Switch buttons ON/OFF

**Objectives:**

To learn Arduino UNO basics

To learn BreadBoard basics

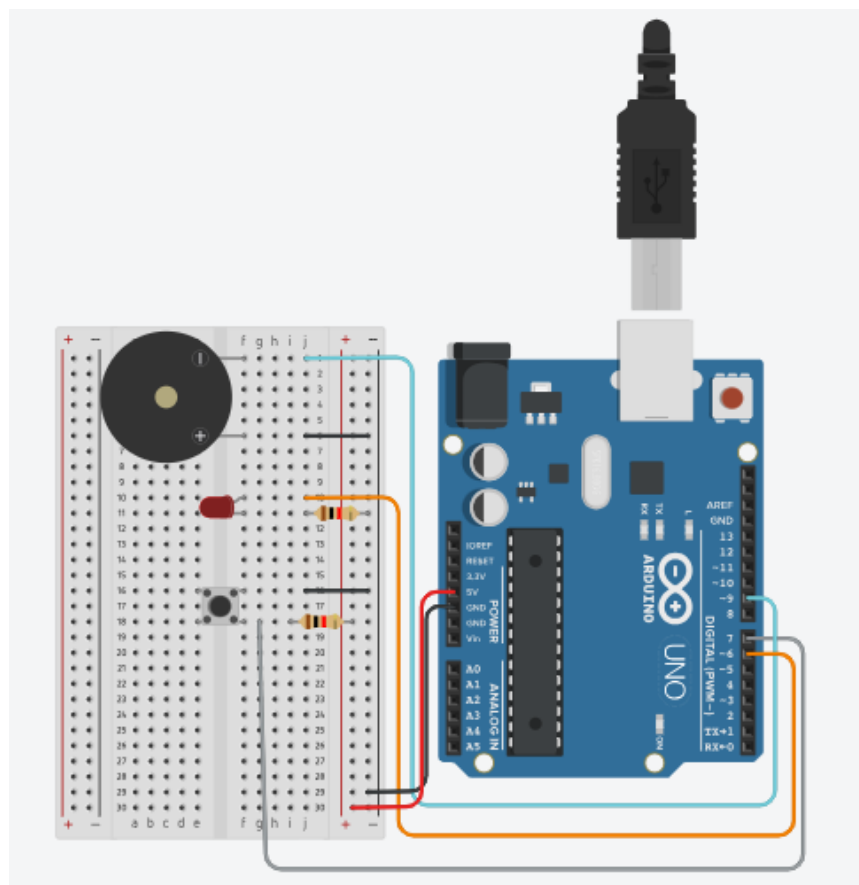
To learn Piezo Speaker

To learn LED

To learn Push Button Switch

Write a program to use the Piezo Speaker with Arduino to obtain on Switch buttons ON/OFF.

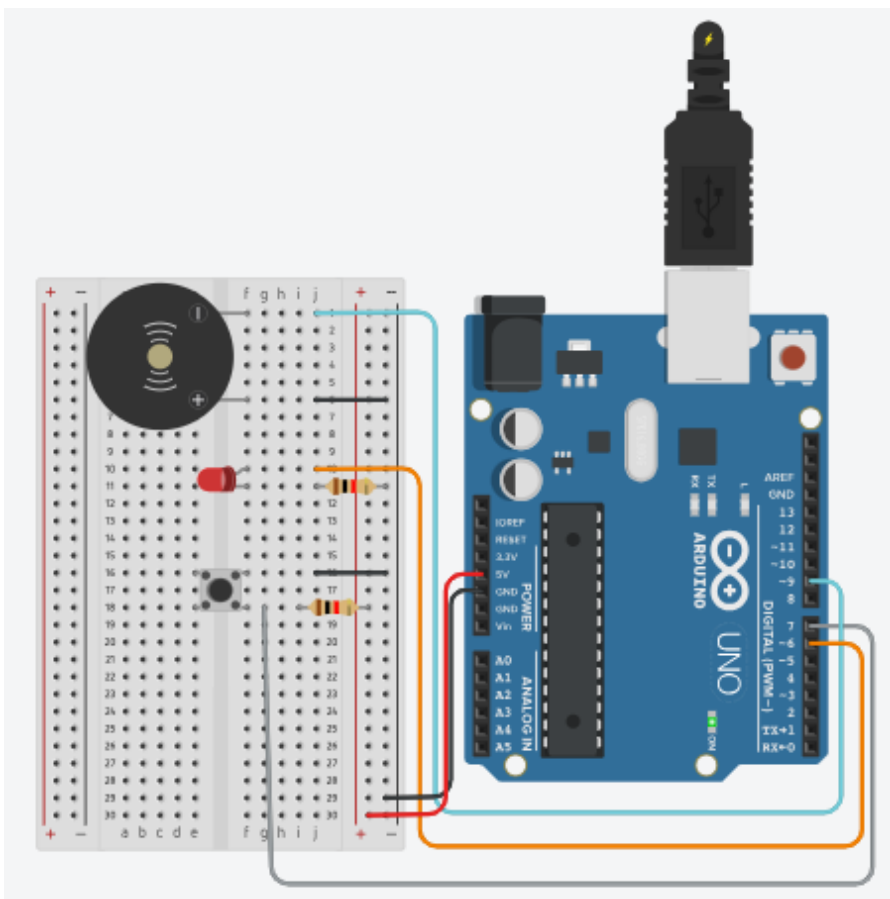
**Components:** 1 LED, 1 Breadboard, 1 Piezo Speaker, 1 Push Button, 2 resistors, Arduino Uno r3, wires.

**Circuit:****Program Code:**

```
// C++ code
//
int buzzerPin = 9;
int buttonPin = 7;
int ledPin = 6;
const int toneFreq = 523;
void setup()
{
  pinMode(buzzerPin, OUTPUT);
  pinMode(ledPin, OUTPUT);
}
```

```
pinMode(buttonPin, INPUT);  
}  
  
void loop()  
{  
  int buttonState = digitalRead(buttonPin);  
  if(buttonState==LOW)  
  {  
    digitalWrite(ledPin, HIGH);  
    tone(buzzerPin, toneFreq);  
  }  
  else  
  {  
    digitalWrite(ledPin,LOW);  
    noTone(buzzerPin);  
  }  
}
```

### Output:



### Conclusion:

Thus learnt about basic components of IOT like Arduino UNO (Blink Arduino on board LED), Breadboard, LEDs and interfacing Piezo Speaker Sound & LED ON/OFF with Arduino to press the Switch Button ON /OFF with connection of resistors breadboard, jumper wires and Arduino.

**Practical :17**

**Aim:** Write a program to interface Piezo Speaker with Arduino and Write a program to use the Piezo Speaker LED ON/OFF with Arduino to obtain on Switch buttons ON/OFF

**Objectives:**

To learn Arduino UNO basics

To learn BreadBoard basics

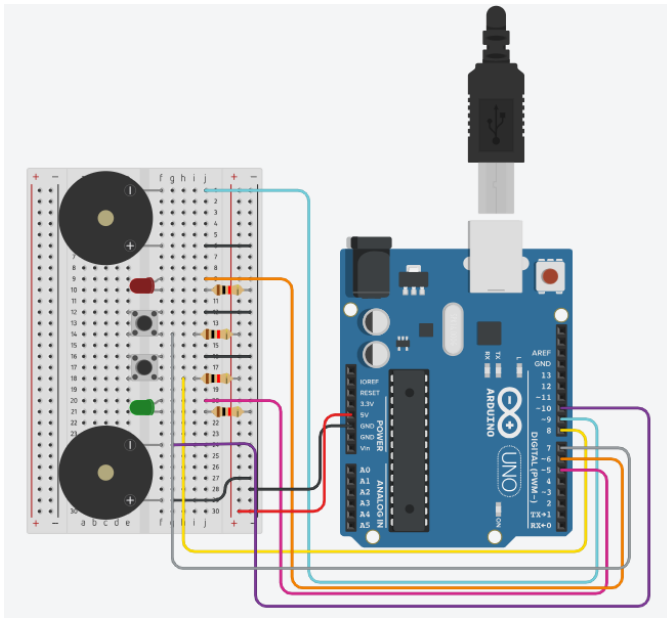
To learn Piezo Speaker

To learn LED

To learn Push Button Switch

Write a program to use the Piezo Speaker with Arduino to obtain on Switch buttons ON/OFF.

**Components:** 2 LED, 1 Breadboard, 2 Piezo Speaker, 2 Push Button, 4 resistors, Arduino Uno r3, wires.

**Circuit:****Program Code:**

```
int buzzerPin1=9;
int buzzerPin2=10;
int buttonPin1 = 7;
int buttonPin2 = 8;
int ledPin1 = 6;
int ledPin2 = 5;
const int toneFreq1 = 523;
const int toneFreq2 = 349;
void setup()
{
  pinMode(buzzerPin1, OUTPUT);
  pinMode(buzzerPin2, OUTPUT);
  pinMode(ledPin1, OUTPUT);
```

```
  pinMode(ledPin2, OUTPUT);
  pinMode(buttonPin1, INPUT);
  pinMode(buttonPin2, INPUT);
}
void loop()
{
  int buttonState = digitalRead(buttonPin1);
  if (buttonState==LOW)
  {
    digitalWrite(ledPin1, HIGH);
    tone(buzzerPin1, toneFreq1);
  }
  else
```

```

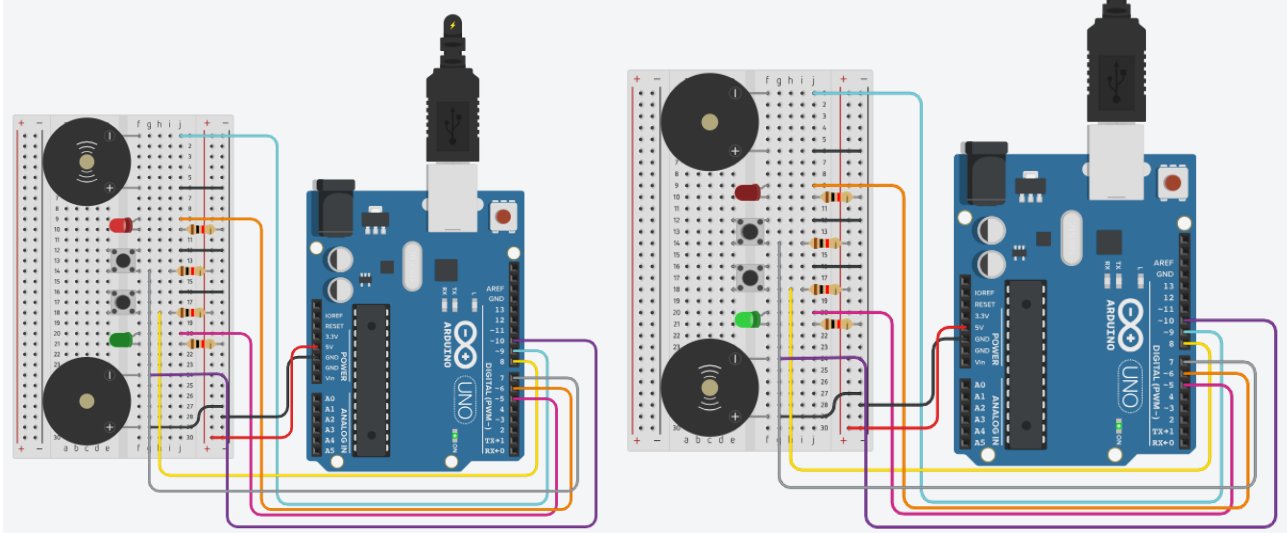
{
digitalWrite(ledPin1, LOW);
noTone(buzzerPin1);
}
{
int buttonState = digitalRead(buttonPin2);
if (buttonState==LOW)
{
digitalWrite(ledPin2, HIGH);

```

```

tone(buzzerPin2, toneFreq2);
}
else
{
digitalWrite(ledPin2, LOW);
noTone(buzzerPin2);
}
}
}

```

**Output:****Conclusion:**

Successfully interfaced a Piezo Speaker and LED with Arduino using a switch. Learned to control output devices (sound and light) based on button input, using components like Arduino UNO, breadboard, resistors, and jumper wires.

**Practical :18**

**Aim:** Write a program to interface Piezo Speaker and Multicolored LED with Arduino and write a program to create fast LED Stream that goes back and forward while interfacing with buzzer.

**Objectives:**

To learn Arduino UNO basics

To learn BreadBoard basics

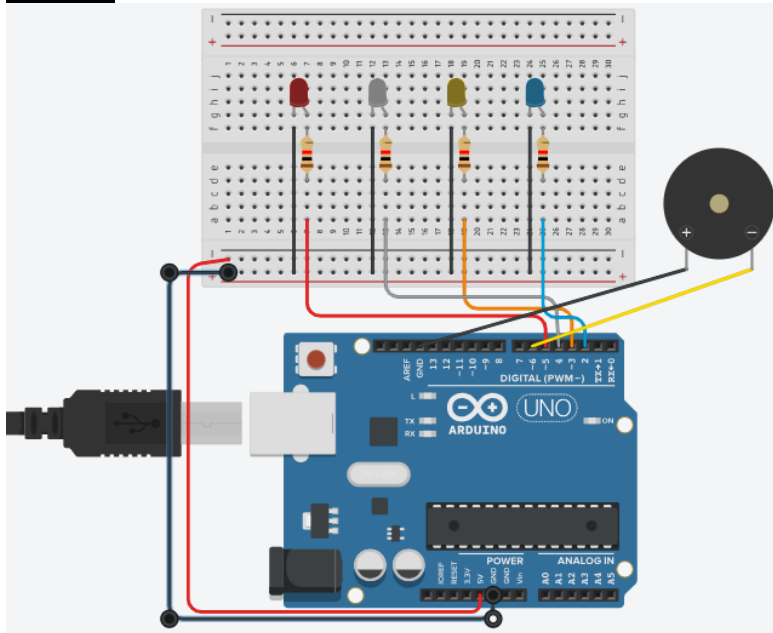
To learn Piezo Speaker

To learn LEDs

To learn Push Button Switch

write a program to create fast LED Stream that goes back and forward while interfacing with buzzer.

**Components:** 4 LEDs, 1 Breadboard, 1 Piezo Speaker, 4 resistors, Arduino Uno r3, wires.

**Circuit:****Program Code:**

```
//Digital Pins/Variables
int blueLED1 = 2; //Blue LED to Pin 2
int yellowLED2 = 3; //Yellow LED to Pin 3
int greenLED3 = 4; //Green LED to Pin 4
int redLED4 = 5; //Red LED to pin 5
int buzzer = 6; //Buzzer to pin 6
const int toneFreq = 261;
void setup() { //Setup Code
  pinMode(blueLED1, OUTPUT); //Blue LED as output
  pinMode(yellowLED2, OUTPUT); //Yellow LED as output
  pinMode(greenLED3, OUTPUT); //Green LED as output
  pinMode(redLED4, OUTPUT); //Red LED as output
  pinMode(buzzer, OUTPUT); //Buzzer as output
  digitalWrite(buzzer, HIGH); //Turn Buzzer on
```

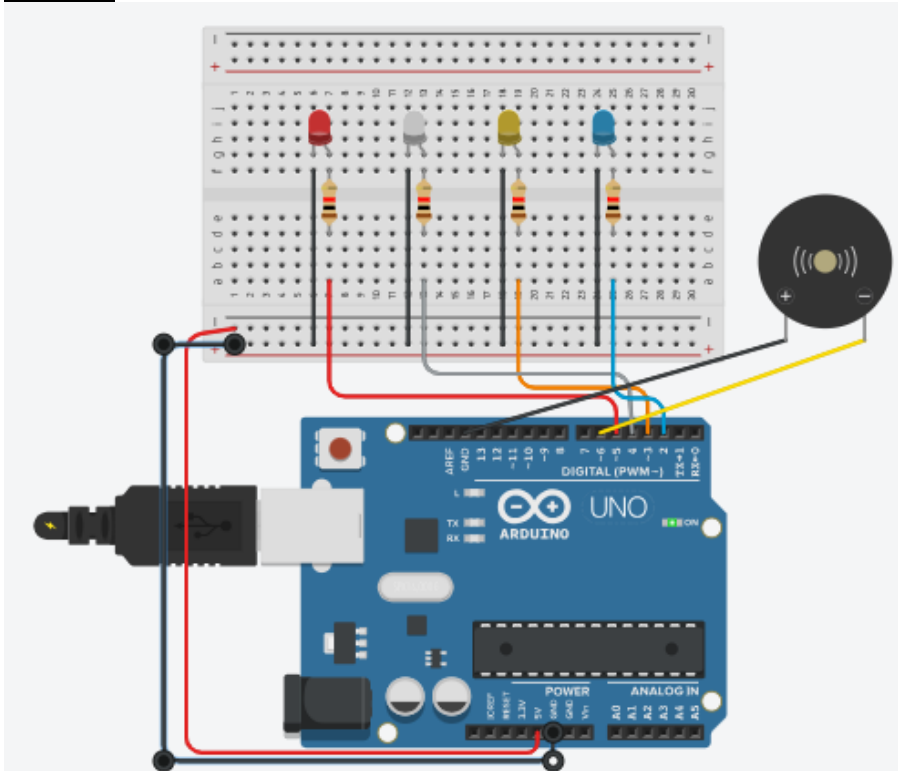
```
tone(buzzer, toneFreq);
}
void loop() { //Loop code
  digitalWrite(blueLED1, HIGH); //Blue led on
  delay(50); //wait for 1/25 of a second
  digitalWrite(blueLED1, LOW); //Blue led off
  digitalWrite(yellowLED2, HIGH); //Yellow led on
  delay(50); //wait for 1/25 of a second
  digitalWrite(yellowLED2, LOW); //Yellow led off
  digitalWrite(greenLED3, HIGH); //Green led on
  delay(50); //wait for 1/25 of a second
  digitalWrite(greenLED3, LOW); //Green led off
  digitalWrite(redLED4, HIGH); //Red led on
  delay(50); //wait for 1/25 of a second
```



```
digitalWrite(redLED4, LOW); //Red led off
digitalWrite(greenLED3, HIGH); //Green led
on
delay(50); //wait for 1/25 of a second
digitalWrite(greenLED3, LOW); //Green led
off
```

```
digitalWrite(yellowLED2, HIGH); //Yellow led
on
delay(50); //wait for 1/25 of a second
digitalWrite(yellowLED2, LOW); //Yellow led
off
}
```

### Output:



### Conclusion:

Successfully created a fast back-and-forth LED stream with buzzer integration using Arduino UNO. Learned the basics of Arduino, Breadboard, Piezo Speaker, LEDs, and switch interfacing. The setup provided practical understanding of synchronized audio-visual output in embedded systems.

**Practical :19**

**Aim:** To interface a 7 Segment Display with Arduino & WAP to turn ON Display for displaying values from 0 to 9 & A to F.

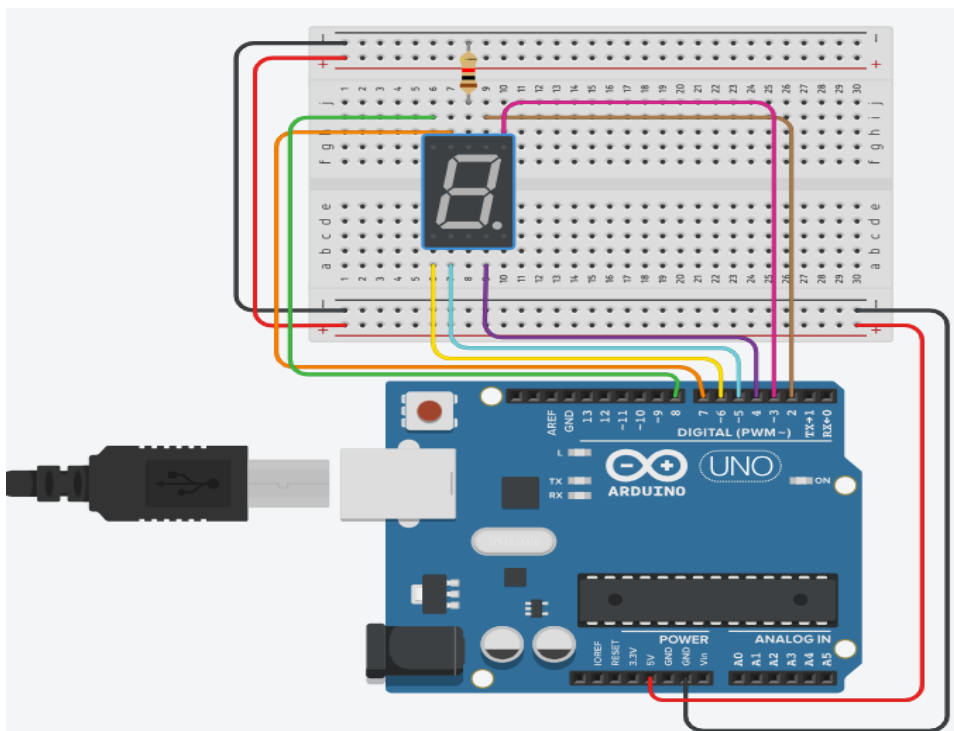
**Objectives:** To learn Arduino UNO basics

To Learn Breadboard basics

To Learn about 7 Segment Display.

WAP to interface a 7 Segment Display with Arduino & turning ON Segment display for displaying values from 0 to 9 & A to F.

**Components:** 7 segment display, 1 Breadboard, 1 resistor, Arduino Uno r3, wires.

**Circuit:****Program Code:**

```
// C++ code
//
// make an array to save Sev Seg pin
configuration of numbers
int num_array[10][7] = { { 1,1,1,1,1,0 }, // 0
                          { 0,1,1,0,0,0 }, // 1
                          { 1,1,0,1,1,0,1 }, // 2
                          { 1,1,1,1,0,0,1 }, // 3
                          { 0,1,1,0,0,1,1 }, // 4
                          { 1,0,1,1,0,1,1 }, // 5
                          { 1,0,1,1,1,1,1 }, // 6
                          { 1,1,1,0,0,0,0 }, // 7
                          { 1,1,1,1,1,1,1 }, // 8
                          { 1,1,1,0,0,1,1 } }; // 9
int letter_array[6][7] = { { 1,1,1,0,1,1,1 }, // A
                           { 0,0,1,1,1,1,1 }, // b
                           { 0,0,0,1,1,0,1 }, // c
                           { 0,1,1,1,1,0,1 }, // d
                           { 1,0,0,1,1,1,1 }, // E
                           { 1,0,0,0,1,1,1 } }; // F

int segAtoGpinouts[7] = {2,3,4,5,6,7,8};
//function header
void WriteToSSD(int *);
void setup()
{ // set pin modes
  for (int i=0;i<7;i++)
    { pinMode(segAtoGpinouts[i],OUTPUT);
    }
}
void loop()
{ //counter loop
```

```

for (int letter_counter = 0; letter_counter < 6;
letter_counter++)
{delay(1000);
  WriteToSSD(letter_array[letter_counter]);}
delay(3000);
for (int num_counter = 0; num_counter <= 9;
num_counter++)
{delay(1000);
  WriteToSSD(num_array[num_counter]);}

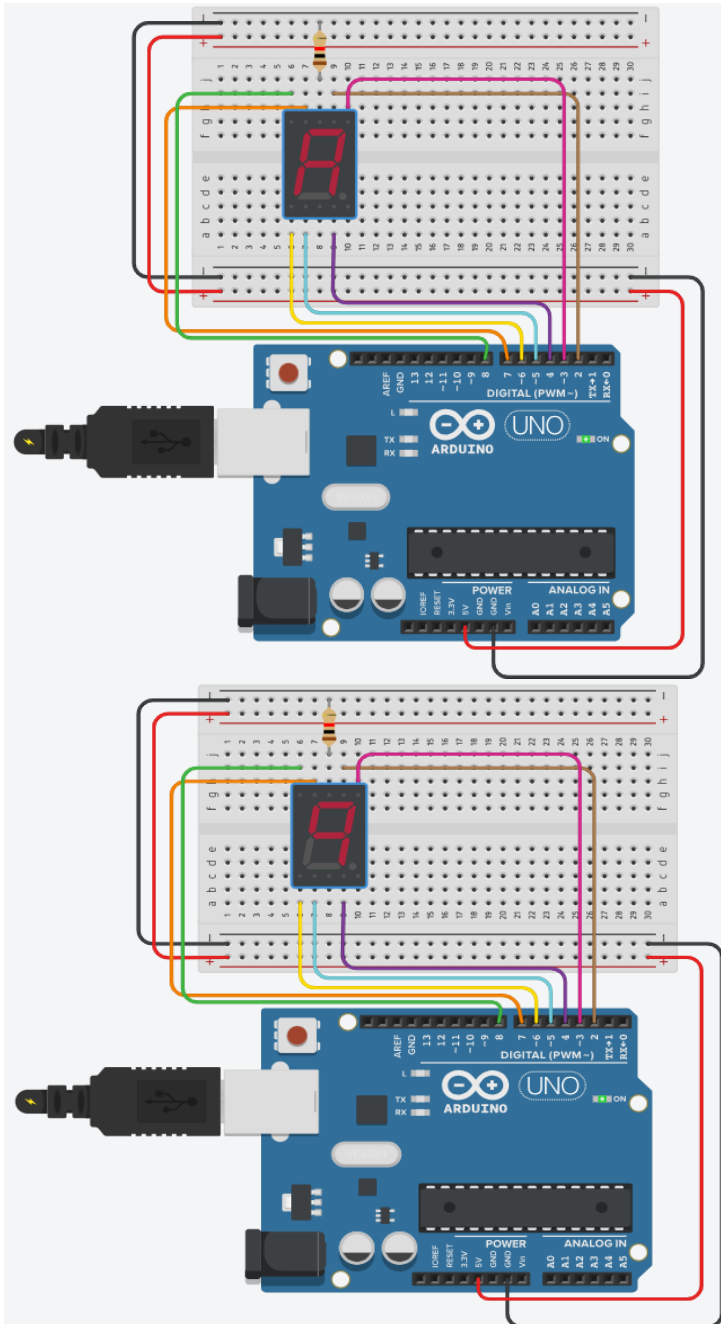
```

```

delay(3000);}
// this functions writes values to the 7seg pins
void WriteToSSD(int * segmentArray )
{for (int i=0; i < 7; i++)
  digitalWrite(segAtoGpinouts[i],
segmentArray[i]);}

```

### Output:



### Conclusion:

Thus, learnt about basic components of lot like Arduino UNO (blink Arduino onboard LED), Breadboard 7 segment display and interfacing to turn ON 7 segment Display for displaying values from 0 to 9 & A to F with connections of resistors, breadboard, Jumper wires & Arduino.

**Practical No:- 20**

**Aim:** To interface 16×2 Character LCD Module with Arduino & WAP to display running with LCD character on parallel interface LCD controller chip.

**Objectives:**

To learn Arduino UNO basics

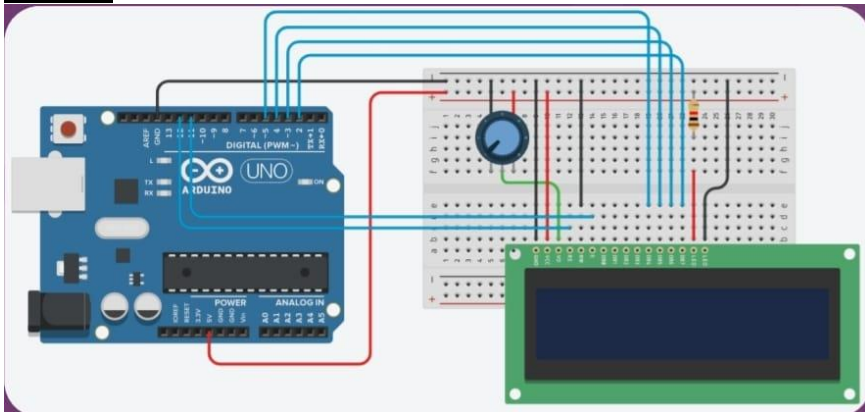
To Learn Breadboard basics

To Learn about 16x2 LCD.

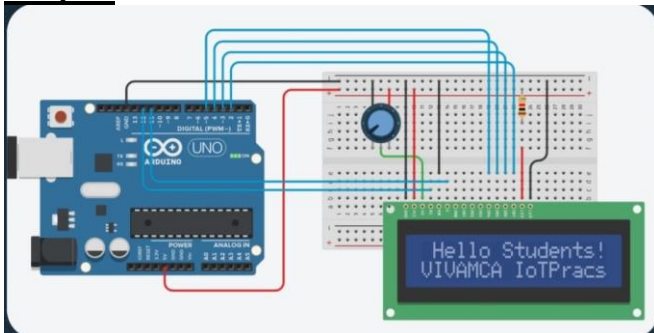
WAP to Display running with Character LCDs based on parallel interface LCD controller chip.

Theory: (concept of Arduino UNO, Breadboard, Resistor, Jumper wires, 16x2 LCD and functions like setup(), loop(), pinMode(), delay(), digitalWrite() etc.).

**Components:** Arduino UNO, 16x2 LCD Display, breadboard, Potentiometer, Jumper Wires, Resistor

**Circuit:****Program Code:**

```
// C++ code
#include <LiquidCrystal.h>
// Creates an LCD object. Parameters: (rs,
enable, d4, d5, d6, d7)
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
void setup()
{ // set up the LCD's number of columns and
rows:
lcd.begin(16, 2);
// Clears the LCD screen
lcd.clear();}
void loop() {
// Print a message to the LCD.
lcd.print(" Hello Students!");
// set the cursor to column 0, line 1
// (note: line 1 is the second row, since
counting begins with 0):
lcd.setCursor(0, 1);
// Print a message to the LCD.
lcd.print("VIVAMCA          IoTPracs");}
```

**Output:****Conclusion:**

Thus, learnt about basic components of lot like Arduino UNO (blink Arduino onboard LED), Breadboard 7 segment display and interfacing 16x2 LCD display for displaying running with LCD to display characters with LCD controller chip with connections of resistors, breadboard, Jumper wires & Arduino.

**Practical No : - 21**

**Aim:** To interface 16×2 Character LCD Module with Arduino & WAP to Display running with LCD to custom character generator based on parallel interface LCD controller chip.

**Objectives:**

To learn Arduino UNO basics

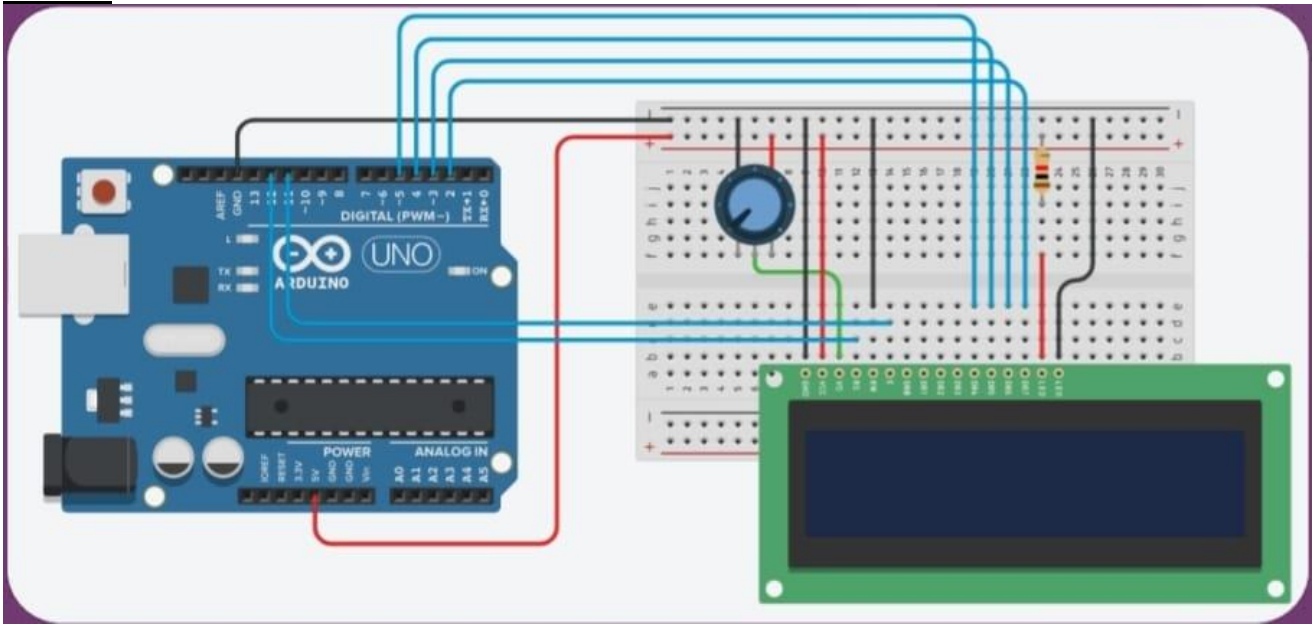
To Learn Breadboard basics

To Learn about 16x2 LCD.

WAP to Display running with Character LCDs based on parallel interface LCD controller chip.

Theory: (concept of Arduino UNO, Breadboard, Resistor, Jumper wires, 16x2 LCD and functions like setup(), loop(), pinMode(), delay(), digitalWrite() etc.).

**Components:** Arduino UNO, 16x2 LCD Display, breadboard, Potentiometer, Jumper Wires, Resistor  
**Circuit:**

**Program Code:**

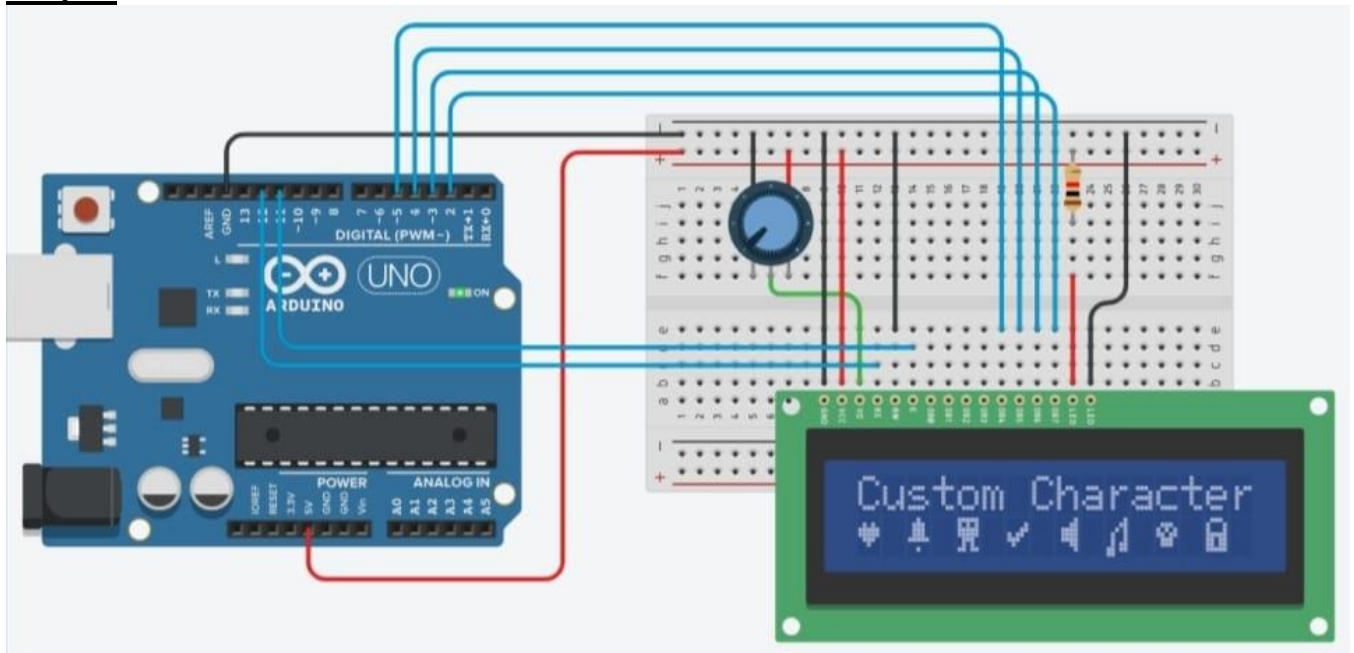
```
// C++ code
// include the library code:
#include <LiquidCrystal.h>
// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
// make some custom characters:
byte Heart[8] = {
  0b00000,
  0b01010,
  0b11111,
  0b11111,
  0b01110,
  0b00100,
  0b00000,
  0b00000 };
// Print All the custom characters
```

```

void loop(){
  lcd.setCursor(0, 1);
  lcd.write(byte(0));
  lcd.setCursor(2, 1);
  lcd.write(byte(1));
  lcd.setCursor(4, 1);
  lcd.write(byte(2));
  lcd.setCursor(6, 1);
  lcd.write(byte(3));
  lcd.setCursor(8, 1);
  lcd.write(byte(4));
  lcd.setCursor(10, 1);
  lcd.write(byte(5));
  lcd.setCursor(12, 1);
  lcd.write(byte(6));
  lcd.setCursor(14, 1);
  lcd.write(byte(7));}

```

### Output:



### Conclusion:

Thus, learnt about basic components of IoT like Arduino UNO (blink Arduino onboard LED), Breadboard 7 segment display and interfacing 16x2 LCD display for displaying running with LCD to custom character generator with LCD controller chip with connections of resistors, breadboard, Jumper wires & Arduino.



**Practical No :- 22**

**Aim:** To interface 16×2 LCD, Push Button, Potentiometer with Arduino and WAP to display message on LCD when push button is pressed.

**Objectives:**

To learn Arduino UNO basics

To Learn Breadboard basics

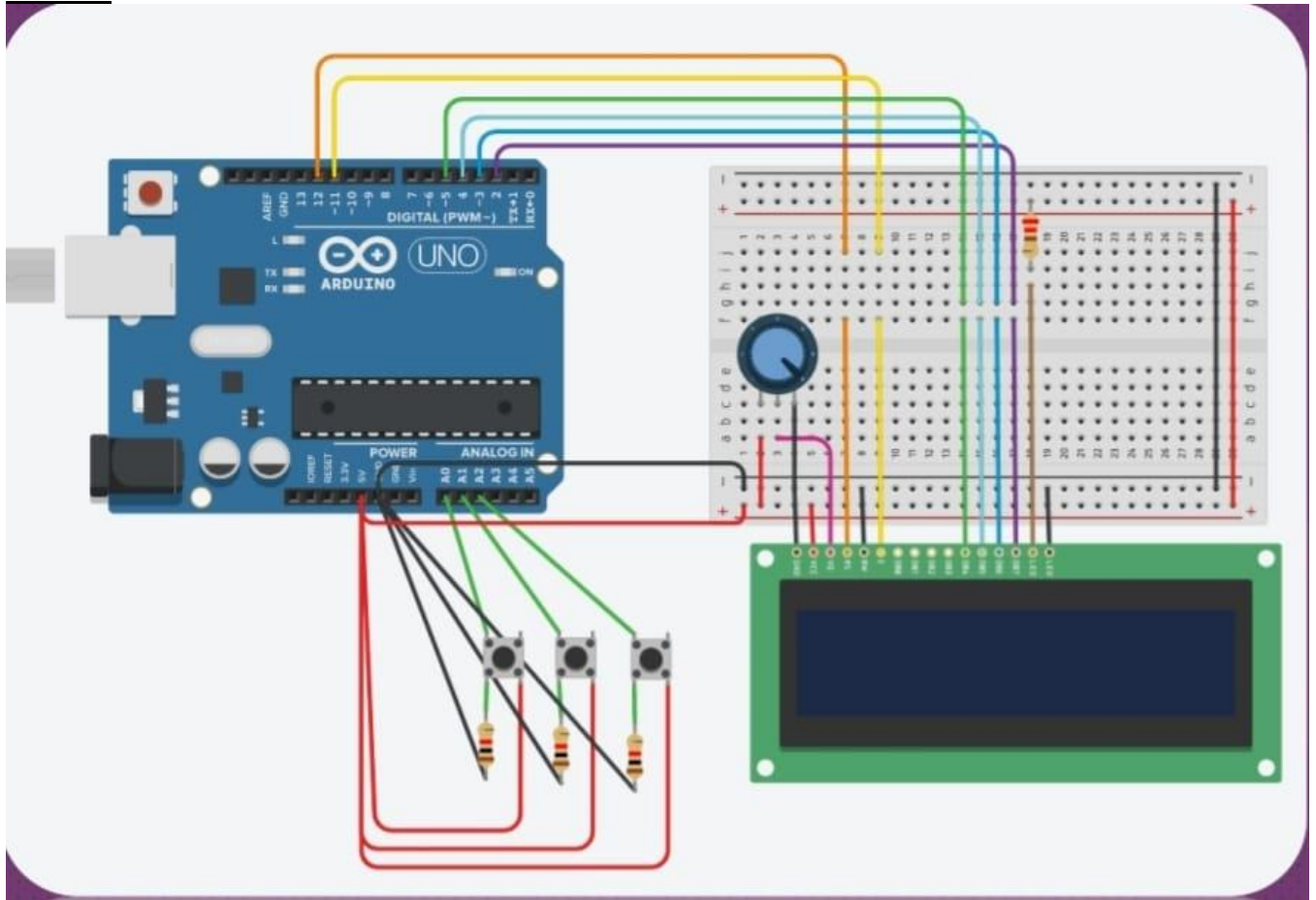
To Learn about 16x2 LCD.

To Learn about Potentiometer

WAP to display message on LCD when push button is pressed.

Theory: (concept of Arduino UNO, Breadboard, Resistor, Jumper wires, 16x2 LCD, Potentiometer and functions like setup(), loop(), pinMode(), delay(), digitalWrite() etc.).

**Components:** Arduino UNO, 16x2 LCD Display, breadboard, Potentiometer, Jumper Wires, Resistor, Push Button.

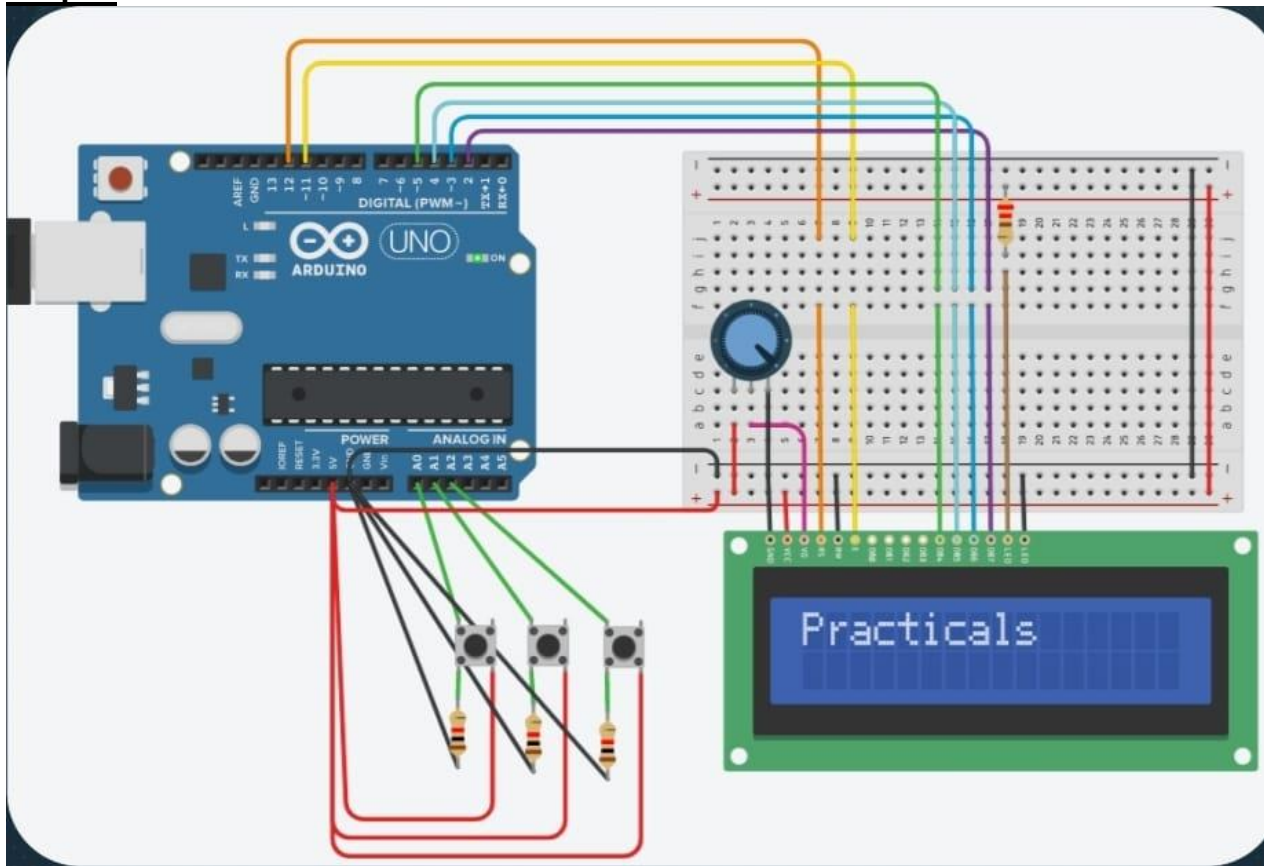
**Circuit:****Program Code:**

```
// include the library code:
#include <LiquidCrystal.h>
// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
void setup(){
  pinMode(Ao, INPUT);
  pinMode(A1, INPUT);
  pinMode(A2, INPUT);
```

```

}
void loop()
{
  // set the cursor to column 0, line 1
  //(note: line 1 is the second row, since counting begins with 0):
  lcd.setCursor(0, 1);
  if (digitalRead(A0) == HIGH)
  { lcd.print("VIT MCA"); }
  if (digitalRead(A1) == HIGH)
  { lcd.print("Sub-IoT"); }
  if (digitalRead(A2) == HIGH)
  { lcd.print("Practicals"); } }

```

**Output:****Conclusion:**

Thus, learnt about basic components of IoT like Arduino UNO (blink Arduino onboard LED), Breadboard, 7 segment display and interfacing 16x2 LCD, Push Button, Potentiometer with Arduino and displaying message on LCD when push button pressed with connections of resistors, breadboard, Jumper wires & Arduino.



**Practical No. 23****Part 1**

**Aim:** To upload data on Thingspeak cloud

**Manually.Steps:**

1. Go to Google and search for Thingspeak.
2. If you are new to Thingspeak, Do sign up and make sure you are on your Channel page.
3. Click on the NEW CHANNEL button (Green color) and create a new channel.
4. Enter a channel name, any description of your choice, and make sure one field is selected or ticked and give that field a name of your choice. Click on save.
5. Now in the private view, make sure you see a graph (empty).
6. Now click on API KEYS tab, scroll down to find API requests section and in that copy the link of Write a Channel Feed and paste it in the Address bar of your browser. And press enter to get a blank screen with a number which indicates the number of data uploaded manually. Following is the example link:

[https://api.thingspeak.com/update?api\\_key=6WEDQNFN3GBKNCQ3&field1=0](https://api.thingspeak.com/update?api_key=6WEDQNFN3GBKNCQ3&field1=0)

7. Suppose you want to change the data to be entered in the graph, just change the
8. =0 to any value of your choice in the link. Above is the link, where we had changed 0 to 40.

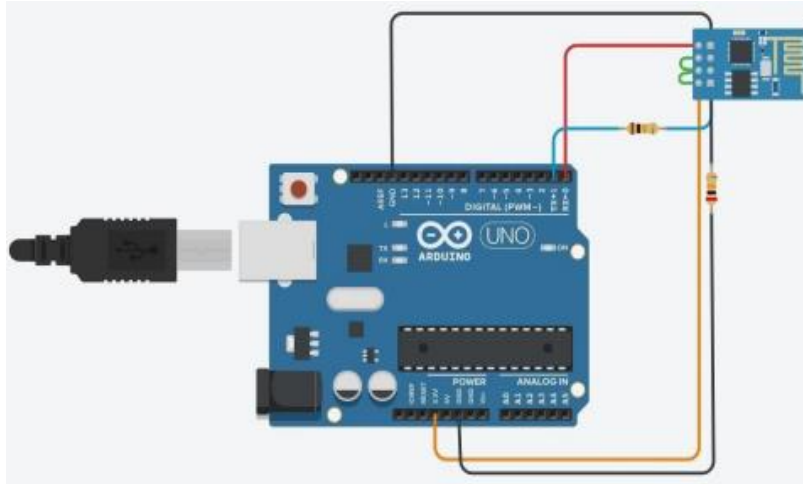
[https://api.thingspeak.com/update?api\\_key=6WEDQNFN3GBKNCQ3&field1=40](https://api.thingspeak.com/update?api_key=6WEDQNFN3GBKNCQ3&field1=40)

8. As a result, to see the visualization (graph), go to the private view and see the graph.

**(Part 2)**

**Aim:** To update readings to Thingspeak from Arduino

**using**

**Tinkercad.Steps:**

1. Click on the NEW CHANNEL button (Green colour) and create a new channel.
2. Enter a channel name, any description of your choice, and make sure one field is selected or ticked and give that field a name of your choice. Click on save.
3. Now in the API Keys tab copy the Write API Key and Paste it in your program \

**PROGRAM:**

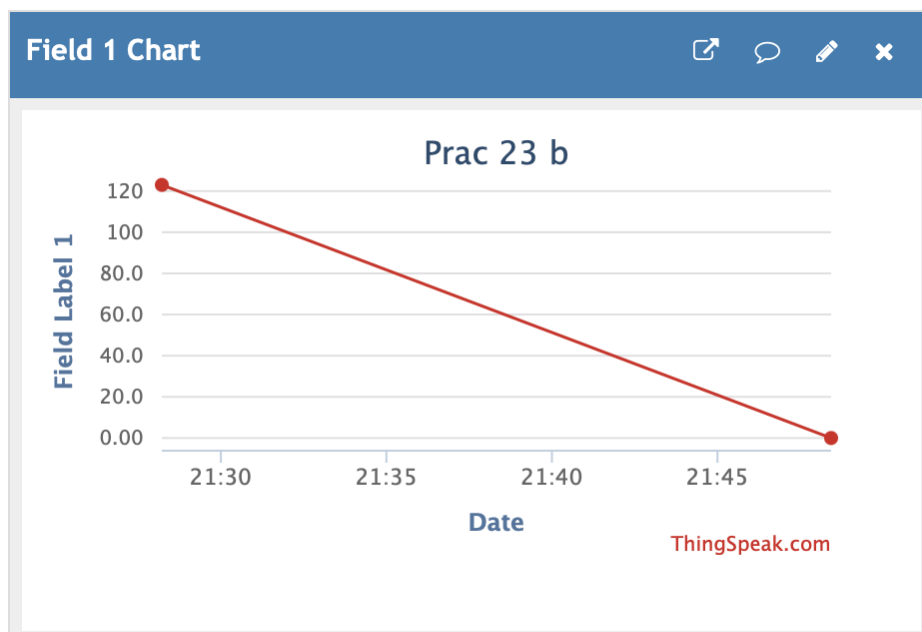
```
void setup() {
  Serial.begin(1152
  00);delay(1000);
  //if you want thingsspeak through
  tinkercad use simulator
  wifi as your
  ssidSerial.println("AT+CWJAP=\"Simulat
  or
  Wifi\", \"\"\\r\\n");
  delay(3000);
}
void loop() {
{
  Serial.println("AT+CI
  PSTART=\"TCP\", \"a
  pi.thingspeak.com\", 80
```

```
\\r\\n"); delay(5000);int
len = 57;//length of
line 15
Serial.print("AT+CIPSEND=");
Serial.println(len);
delay(10);
Serial.print("GET
/update?api_key=ZRGGNASXTIB4M3B
&field1=120
HTTP/1.1\\r\\n"); // Change the field value to
see the
variations in the data
delay(100);
Serial.println("AT+CIPCLOSE=0
\\r\\n");delay(6000);
}
}
```

**Output:****Serial Monitor**

```
AT+CWJAP="Simulator Wifi",""  
AT+CIPSTART="TCP","api.thingspeak.com",80  
AT+CIPSEND=104  
GET /update?api_key=7JWB3X9H404VQGL7&field1=99 HTTP/1.1  
Host: api.thingspeak.com  
Connection: close
```

```
AT+CIPCLOSE
```

**NOTE:**

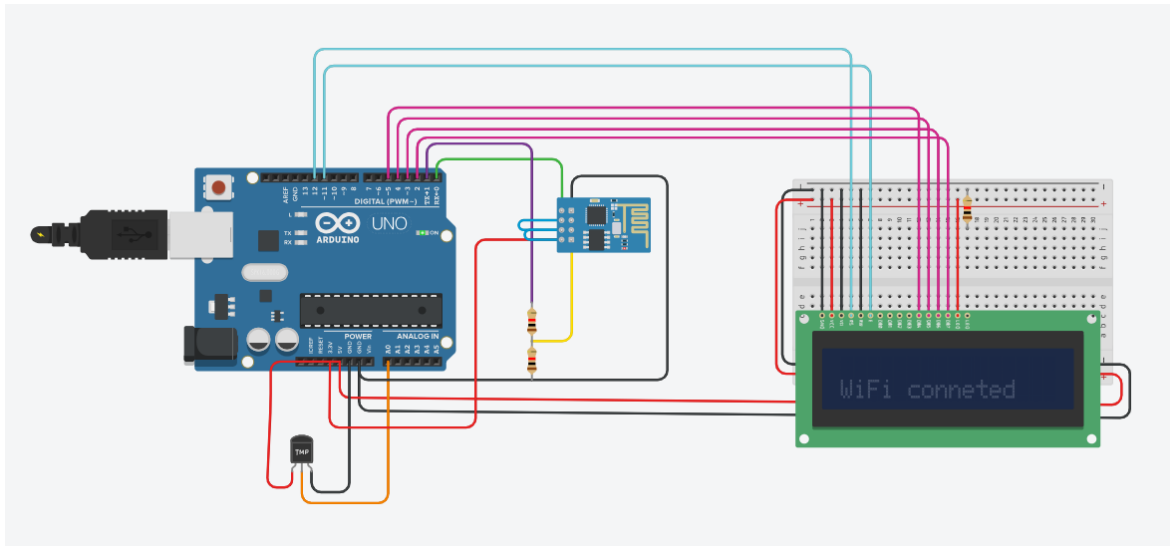
1. TEXT IN **RED** COLOUR IS YOUR WRITE API KEY
2. Make sure you see your Serial Monitor and check if Data is sent to Thingspeak.
3. To check result about data upload, go to thingspeak, click on private view and see the graph.

**(Part 3)**

**Aim:** To interface Temperature sensor and ESP8266 with Arduino and update temperature reading to Thingspeak.

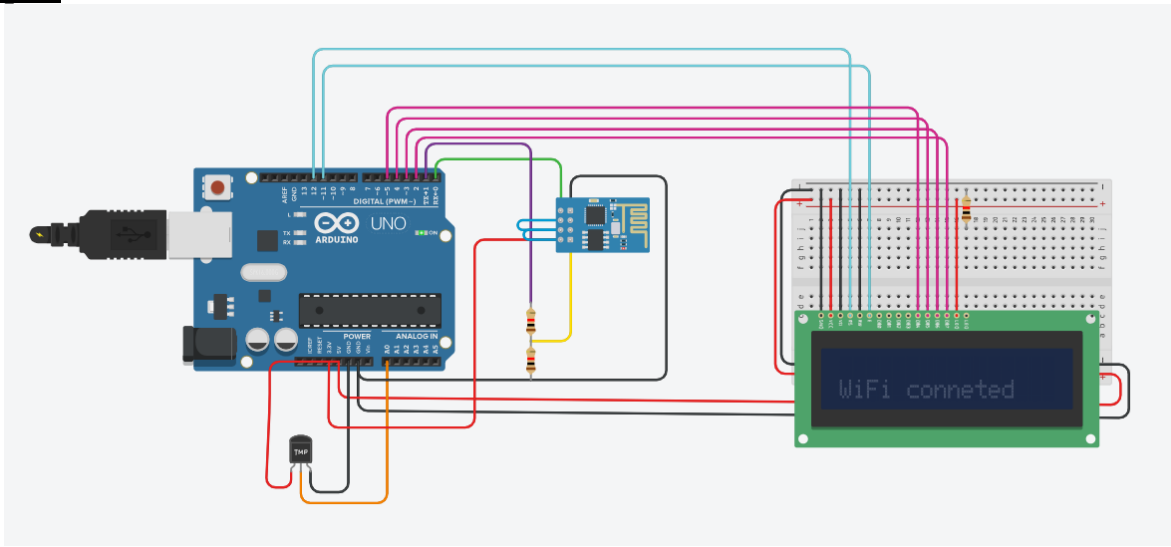
**Steps:**

1. Click on the NEW CHANNEL button (Green colour) and create a new channel.
2. Enter a channel name, any description of your choice, and make sure one field is selected or ticked and give that field a name of your choice. Click on save.
3. Now in the API Keys tab copy the Write API Key and Paste it in your program

**PROGRAM:**

```
void setup() {
  Serial.begin(115200);
  Delay(1000);
  Serial.println("AT+CWLJAP=\"Simulator
  Wifi\", \"\"\\r\\n"); delay(3000);
}
void loop() {
{
  int sensorValue = analogRead(A0);
  float volt = (sensorValue/1020.0) * 4.9;
  //Volts floattempC = (volt -0.5) * 100;
  //Celcius Serial.println(tempC);
  Serial.println("AT+CI
  PSTART=\"TCP\", \"a
  pi.thingspeak.com\",80
```

```
\\r\\n"); delay(5000);int
len = 65;
Serial.print("AT+CIPS
END=");
Serial.println(l
en);delay(10);
Serial.print("GET
/update?api_key=EDLBQ1UJ9ZLNXD
57&field1=" + String(tempC) + "
HTTP/1.1\\r\\n");
delay(100);
Serial.println("AT+CIPCLOSE=0
\\r\\n");delay(6000);
}
}
```

**Output:****NOTE:**

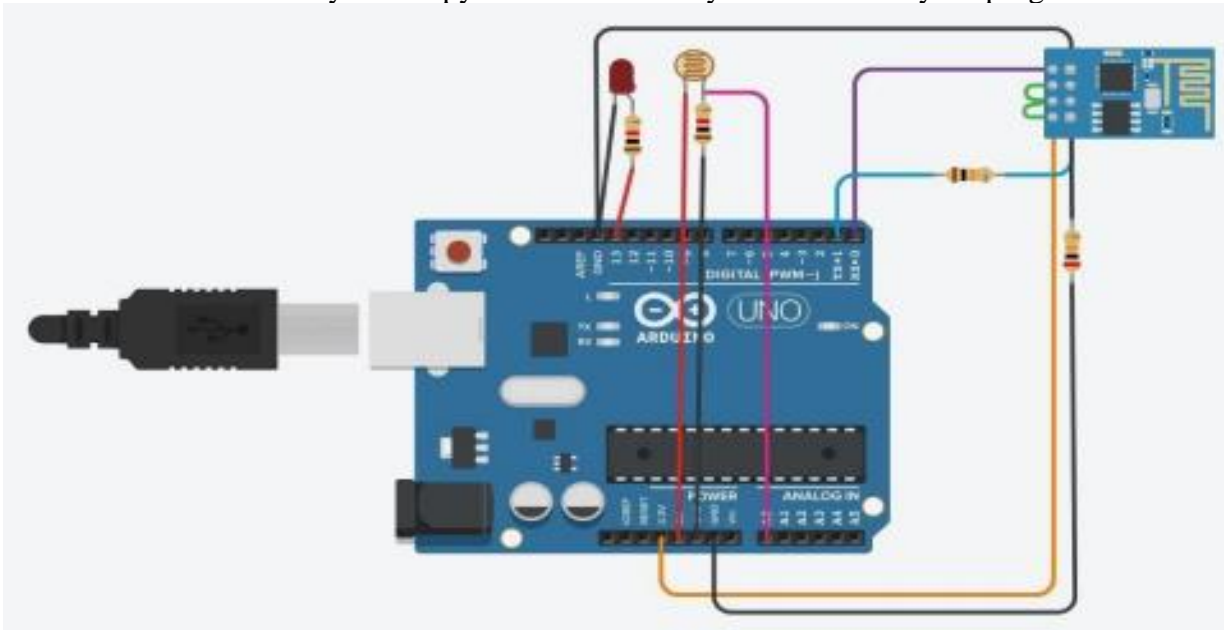
1. TEXT IN **RED** COLOUR IS YOUR WRITE API KEY
2. Make sure you see your Serial Monitor and check if Data is sent to Thingspeak. 3. To check result about data upload, go to thingspeak, click on private view and see the graph.

## Practical No. 24

**Aim:** To interface LDR sensor, LED and ESP8266 with Arduino and update light intensity values to Thingspeak and tweet "LIGHT ON" message on tweeter when light intensity value is less than 300.

### Steps:

1. Click on the NEW CHANNEL button (Green colour) and create a new channel.
2. Enter a channel name, any description of your choice, and make sure one field is selected or ticked and give that field a name of your choice. Click on save.
3. Now in the API Keys tab copy the Write API Key and Paste it in your program



### PROGRAM:

```
int ldr=A0;//Set A0(Analog Input)
for LDR.int value=0;
void setup()
{ Serial.begin(115200 );
  pinMode(13,OUTPUT);
  delay(1000);
  Serial.println("AT+CWJAP=\"Simulator
  Wifi\",\"\"\\r\\n");delay(3000);}
void loop()
{
  { value=analogRead(ldr);
    Serial.println("LDR value
    is :");Serial.println(value);
    if(value<300)
    { digitalWrite(13,HIGH); }
    else
    { digitalWrite(13,LOW);//Turns the LED OFF in Light. }
```

```

Serial.println("AT+CIPSTART=\"TCP\", \"thingspeak
.com\",80");delay(5000);
int len = 65;
Serial.print("AT+CIPSEN
D=");Serial.println(len);
delay(10);
Serial.print("GET /update?api_key=6WEDQNFN3GBKNCQ3&field1="+
String(value) + "HTTP/1.1\r\n");
delay(100);
Serial.println("AT+CIPCLOSE=0
\r\n");delay(6000); } }

```

**NOTE:**

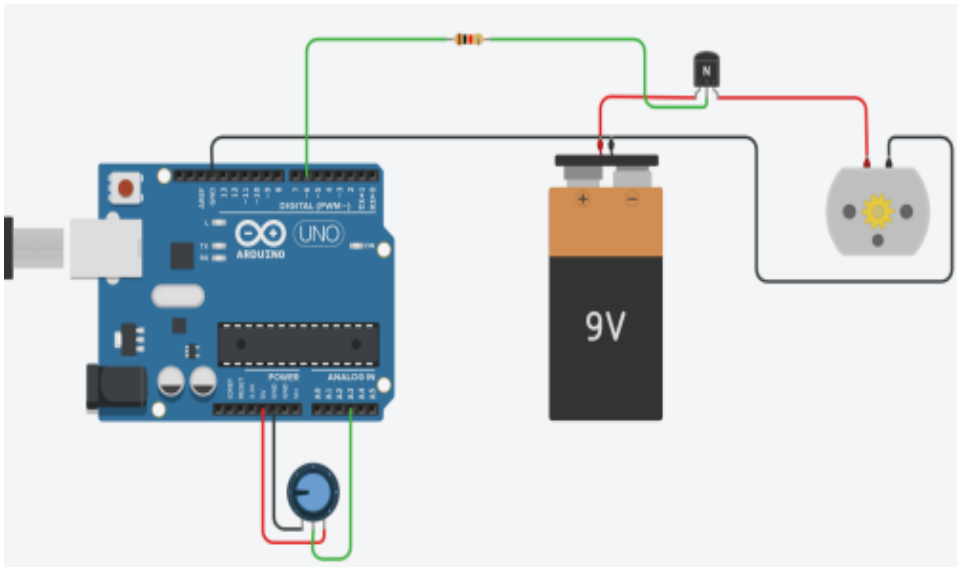
1. TEXT IN RED COLOUR IS YOUR WRITE API KEY
2. Make sure you see your Serial Monitor and check if Data is sent to Thingspeak. 3. To check result about data upload, go to thingspeak, click on private view and see the graph.
4. Once you finish doing the above steps go back to Thingspeak and next to the CHANNELS tab ,click on the APPS tab and select React option.
5. Click on the NEW REACT button (Green colour) and give a React name. Here it is LDR Reactand do following settings:

The screenshot shows the 'React' configuration page in Thingspeak. The 'React Name' is 'LDR React'. The 'Condition Type' is 'Numeric'. The 'Test Frequency' is 'On Data Injection'. The 'Condition' is 'If channel' with 'Packag Channel [1440000]'. The 'Field' is '2 (Light Intensity)'. The condition is 'is less than' with a value of '300'. The 'Action' is 'ThingTweet'. The 'then tweet' message is 'LIGHT ON'. The 'using Twitter account' is 'PANJALINUDHOLGA1'. The 'Options' section has the first option selected: 'Run action only the first time the condition is met'. A green 'Save React' button is at the bottom.

6. In the Action tab in the above figure, select ThingTweet option and then it will ask to link with your twitter account
7. Link and Choose your Twitter Account.
8. Save the React.
9. Make sure your intensity of LDR is less than 300 in the TinkerCad LDR slider and then check your Twitter account for the Tweet of "LIGHT ON" message.

**Practical No. 25**

**Aim:** To interface servo motor / DC motor with Arduino and WAP to sweep a servo back and forth through its full range of motion/to control a DC motor.

**Code:**

```
const int poten = A3;
int var;
void setup()
{
  Serial.begin(9600);
  pinMode(6, OUTPUT);
}
void loop()
{
  var = analogRead(poten);
  analogWrite(6, var);
  Serial.println(var);
}
```

**Output:**