# Practical: 1

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



1. **Resister: -**

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 (Ω). 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.



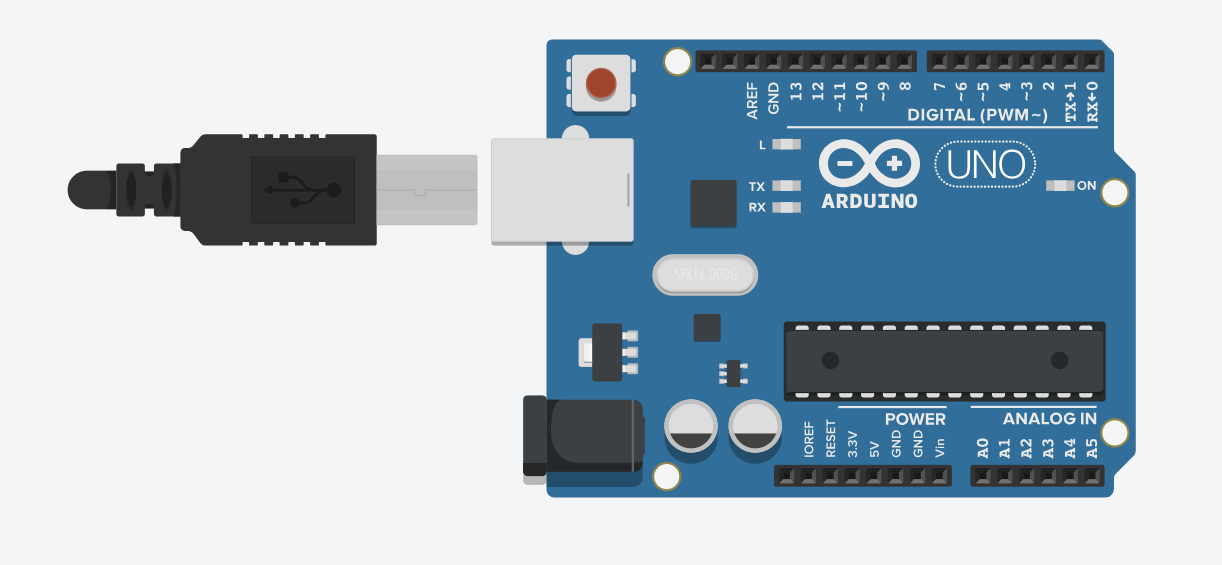
1. **LED: -**

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.



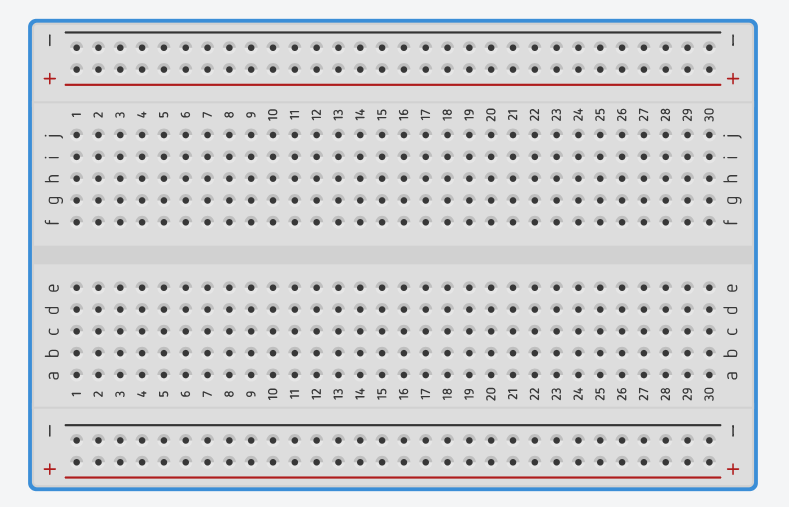
1. **Arduino Uno R3: -**

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.



1. **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.



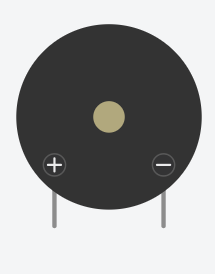
1. **LED RGB: -**

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.



1. **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.



1. **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**

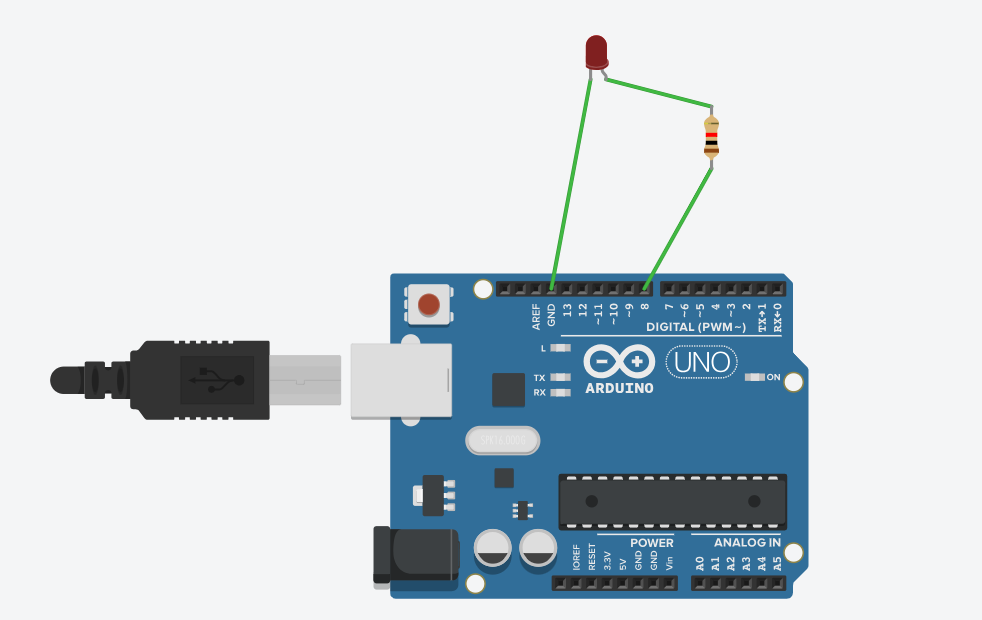
**Aim :** WAP to blink Arduino onboard LED & to interface external LED with Arduino and write a program to turn ON LED for 1 second after every 2 seconds.

**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);

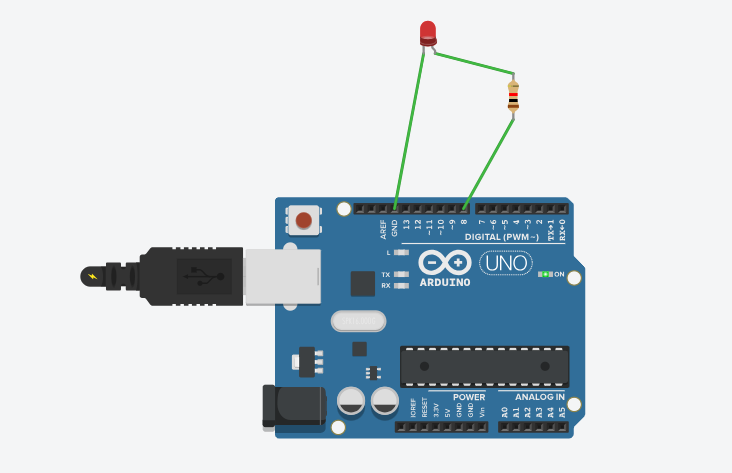
delay(5000); // Wait for 1000 millisecond(s)

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**

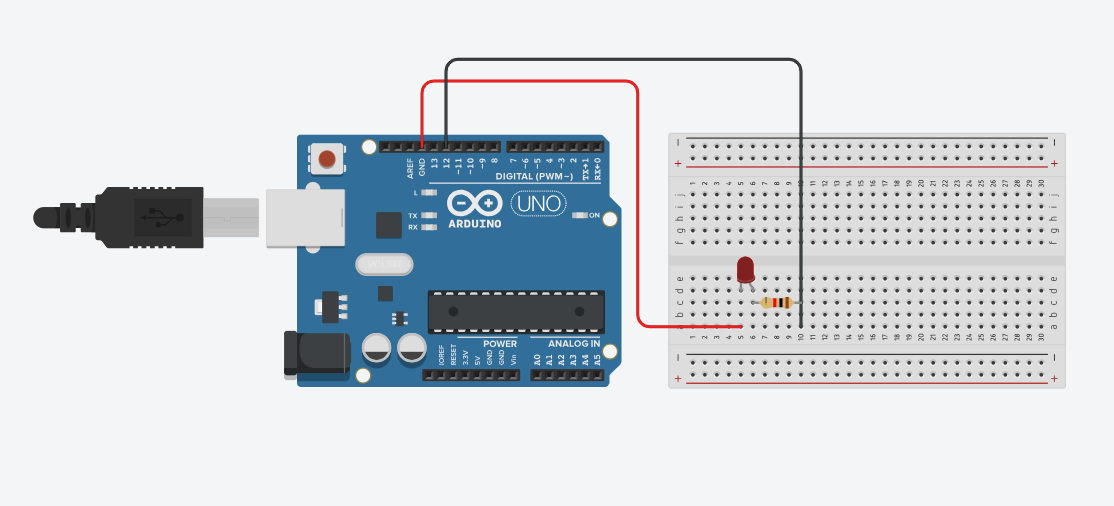
**Aim :** WAP to blink Arduino onboard LED & to interface external LED with Breadboard & Arduino and write a program to turn ON LED for 1 second after every 2 seconds.

**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);

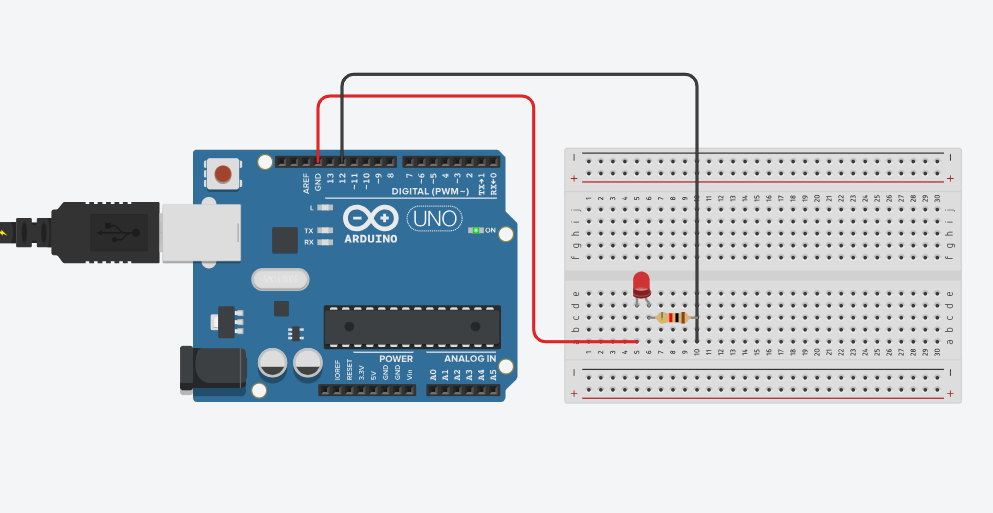
delay(1000); // Wait for 1000 millisecond(s)

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**

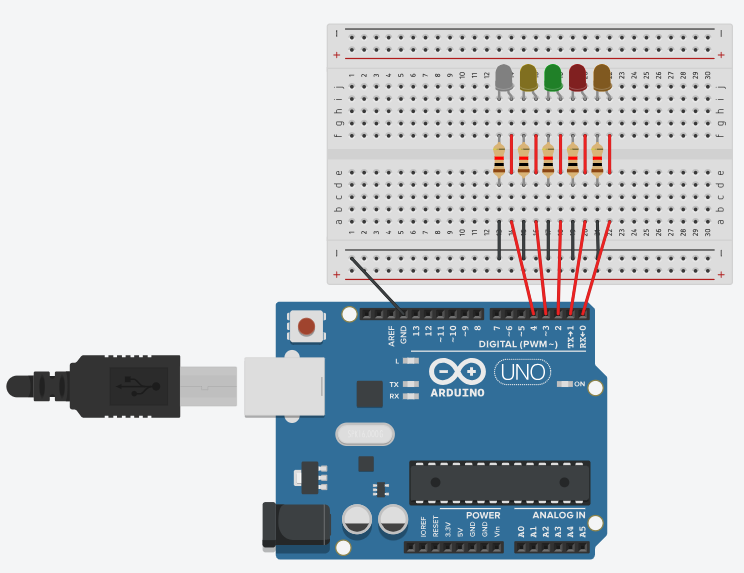
**Aim :** WAP to blink LEDs with Arduino & to interface external 5 LEDs with Breadboard & Arduino and write a program to turn ON/OFF LED.

**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 5LEDs – 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);

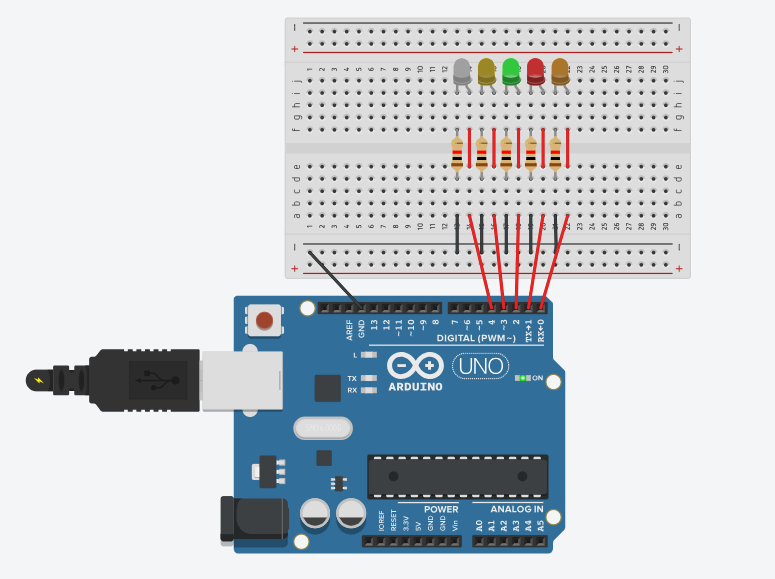
digitalWrite(led5, HIGH);

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**

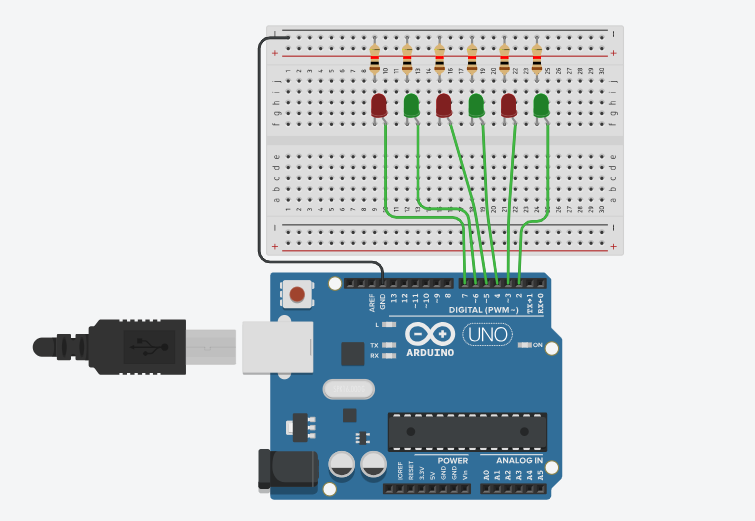
**Aim :** WAP to interface external 6 LED s with Breadboard & Arduino and write a program to blink 6 LEDs , one at a time, in a back and forth formation.

**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);

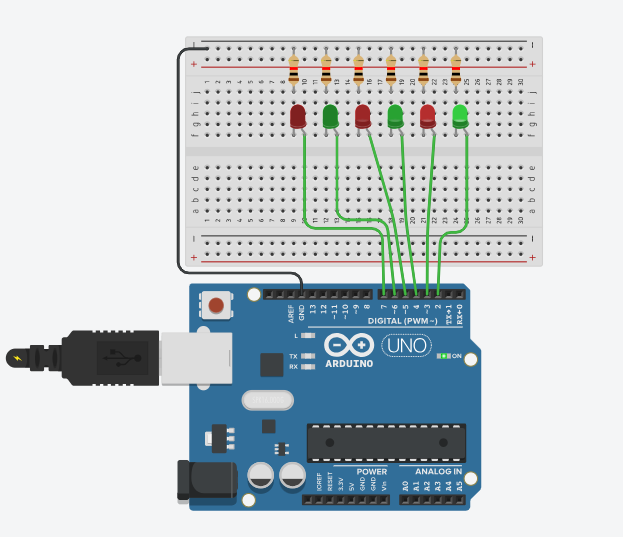
delay(timer);

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**

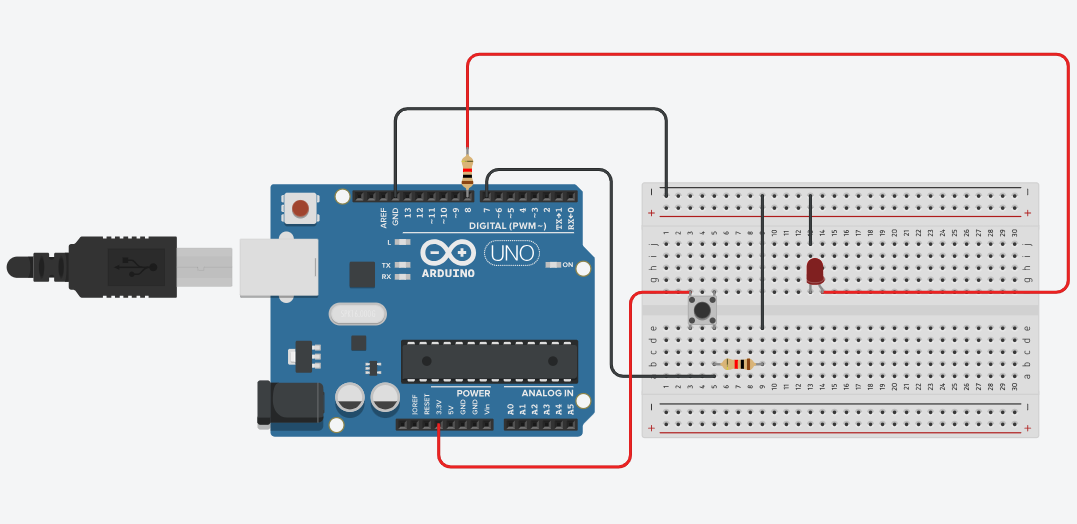
**Aim:** WAP to interface push Button with Arduino and write a program to ON LED when push button is pressed.

**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);

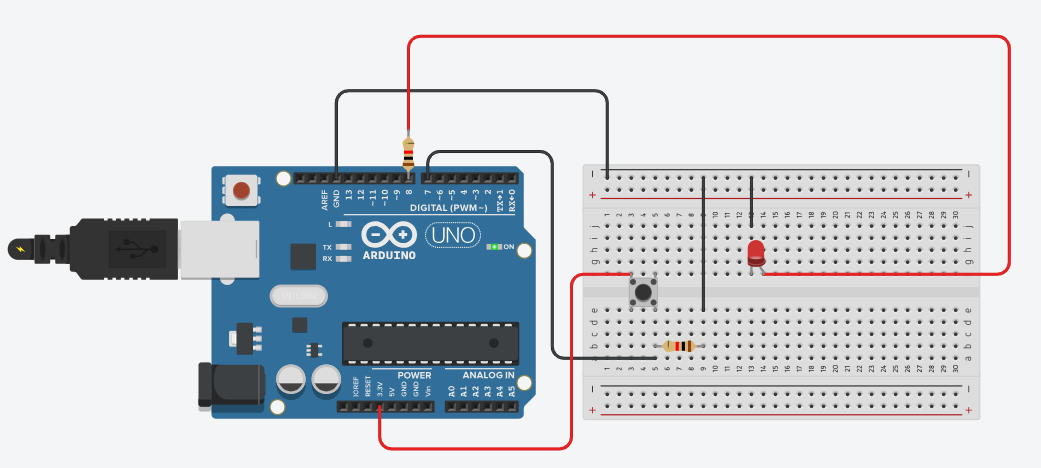
buttonState=0;

}

}

}

**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: 7**

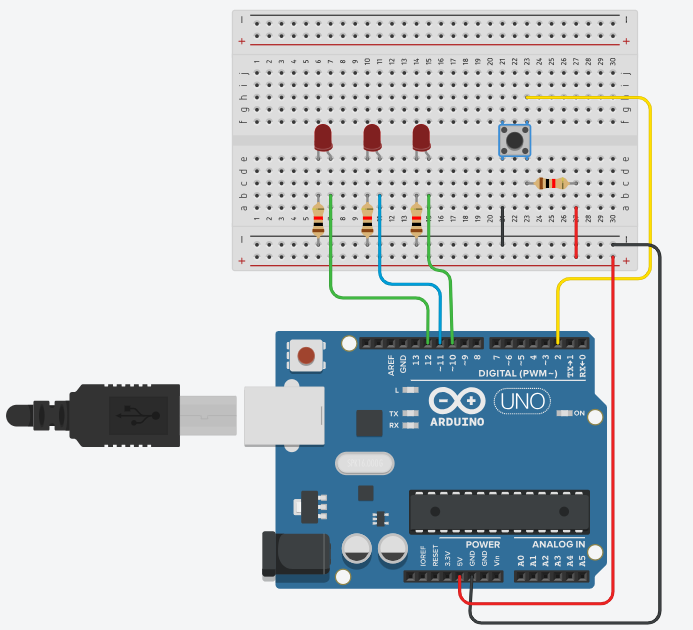
**Aim :** WAP to interface Push Button with Arduino and write a program to turn ON LEDs when push button is pressed and gets OFF after second automatically like a door bell.

**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);

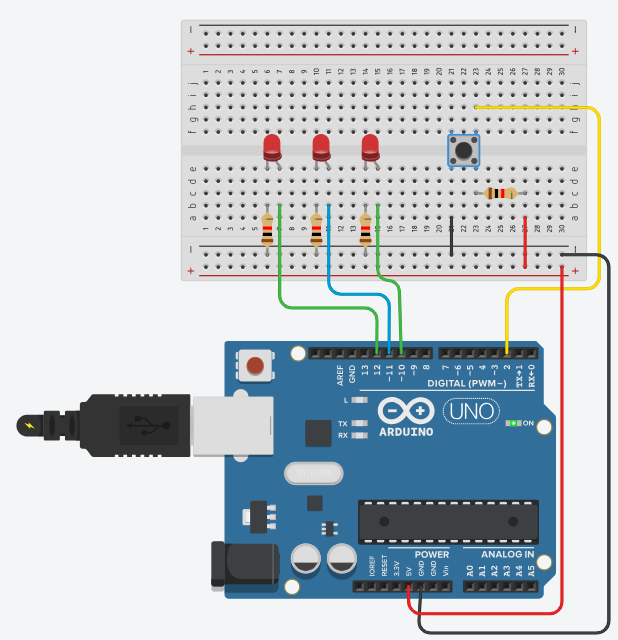
digitalWrite(ledPin3, ledOn);

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: 8**

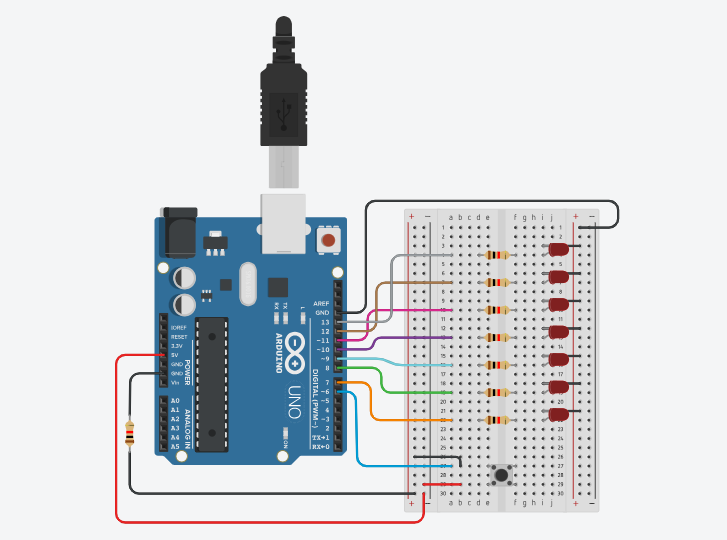
**Aim :** Write a program to interface push button with Arduino and write a program to turn on LEDS when push button is pressed in 7 different patterns.

**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 pattem

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);

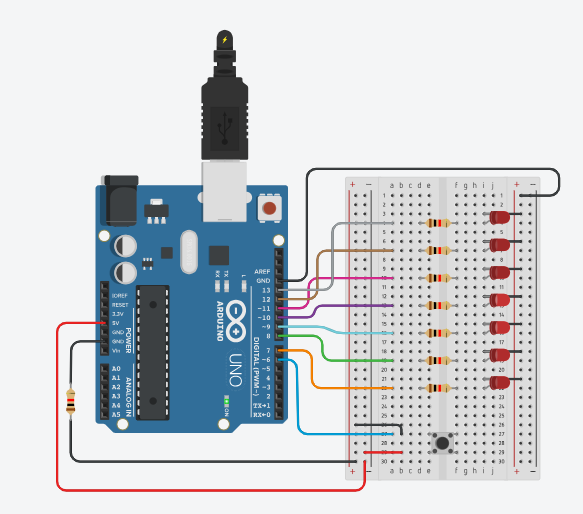
digitalWrite(L7,0); //1,7

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: 9**

**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

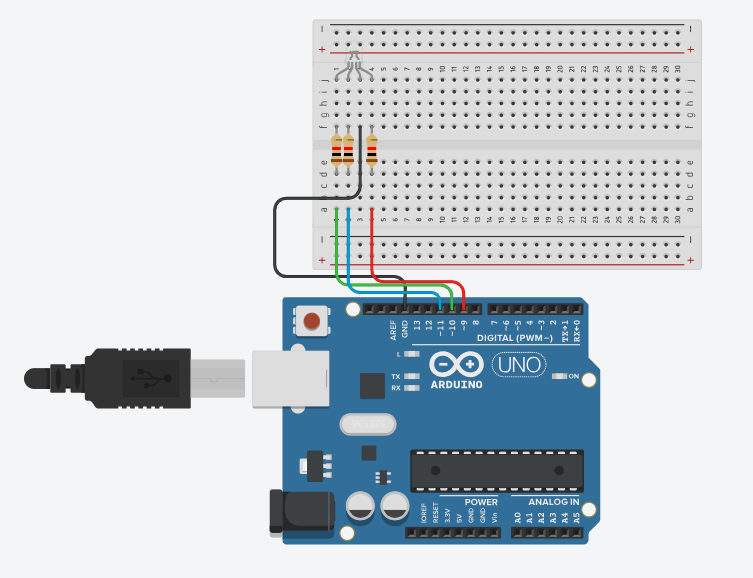
To learn BreadBoard basics

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

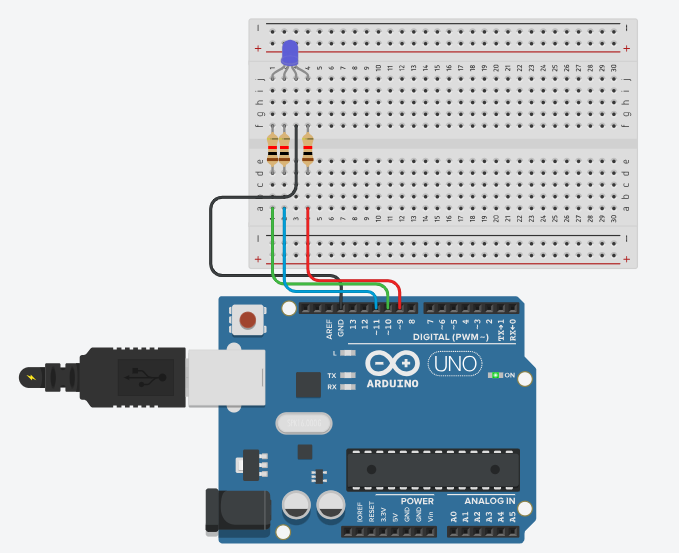
delay(1000);

digitalWrite(blue\_light\_pin, LOW);

//delay(1000);

}

**Output:**



**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){

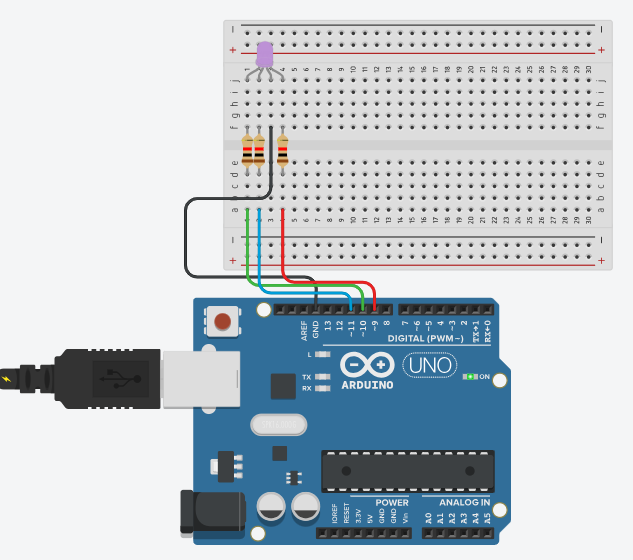
analogWrite(red\_light\_pin, red\_light\_value);

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 :10**

**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

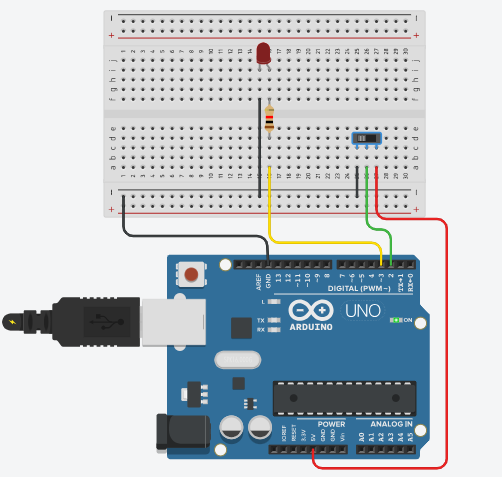
To learn Breadboard basics

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

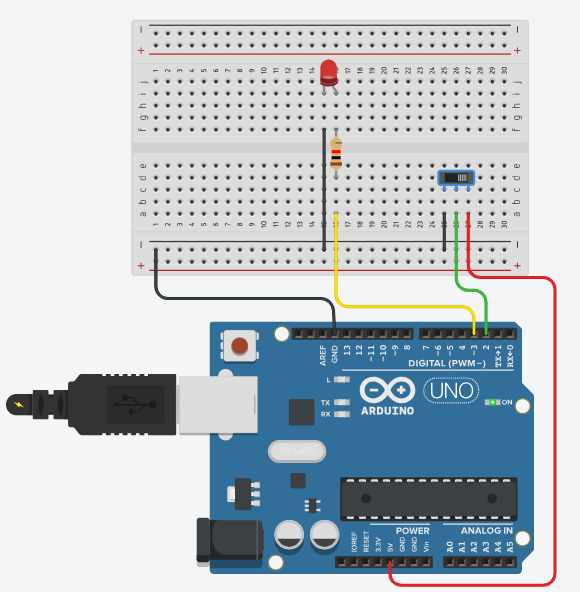
{

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 :11**

**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

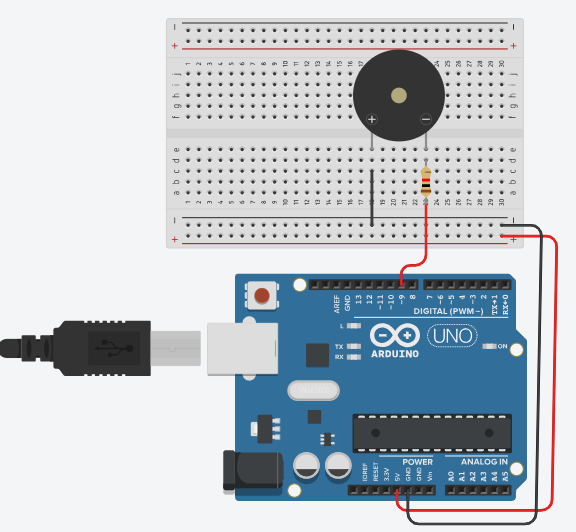
To learn BreadBoard basics

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);

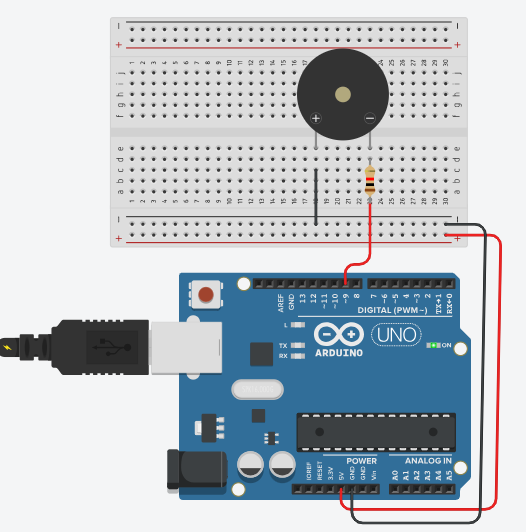
delay(1000); // Wait for 1000 millisecond(s)

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 :12**

**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

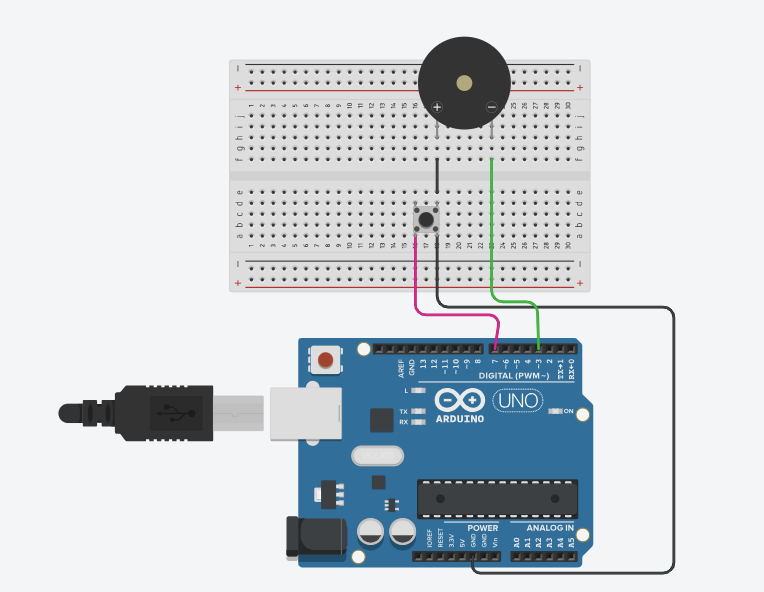
To learn Piezo Speaker

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){

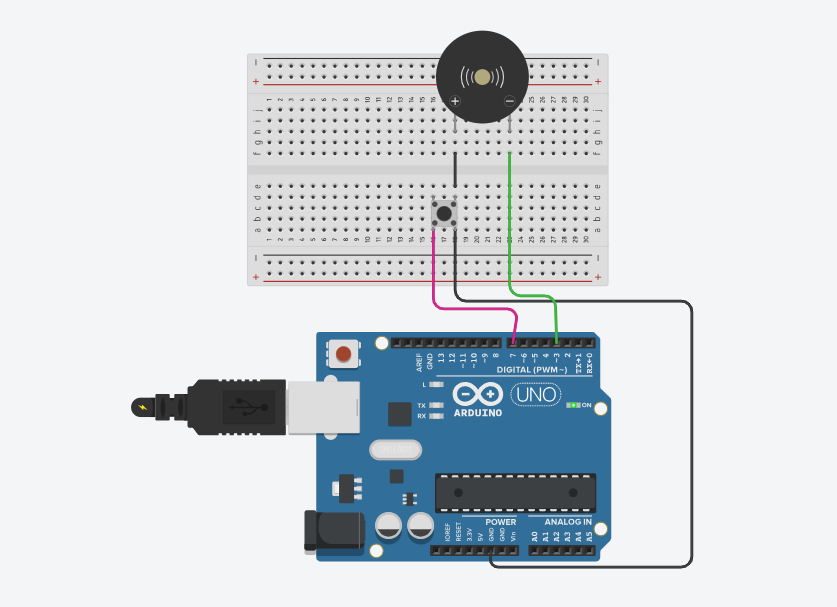
Serial.println("The button is being unpressed");

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 :13**

**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

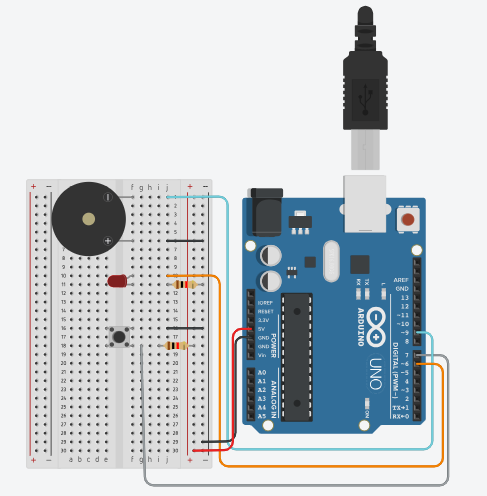
To learn LED

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

{

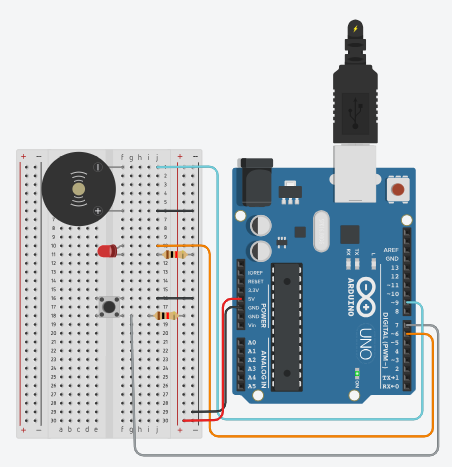
digitalWrite(ledPin,LOW);

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 :14**

**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

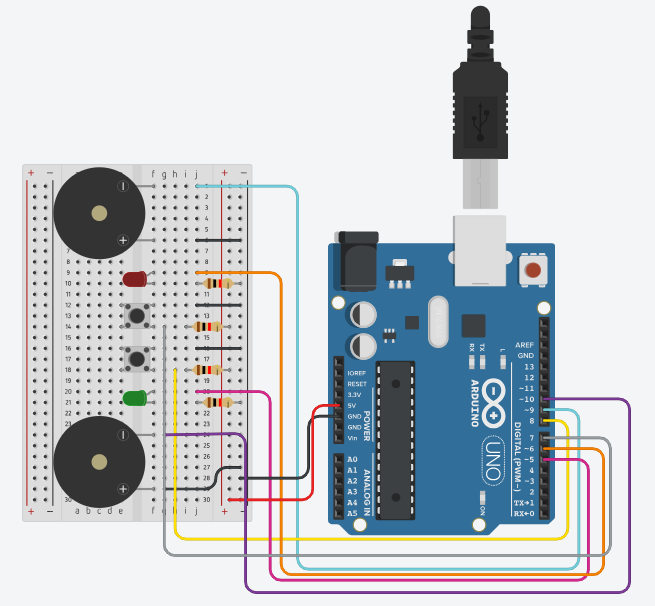
To learn LED

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

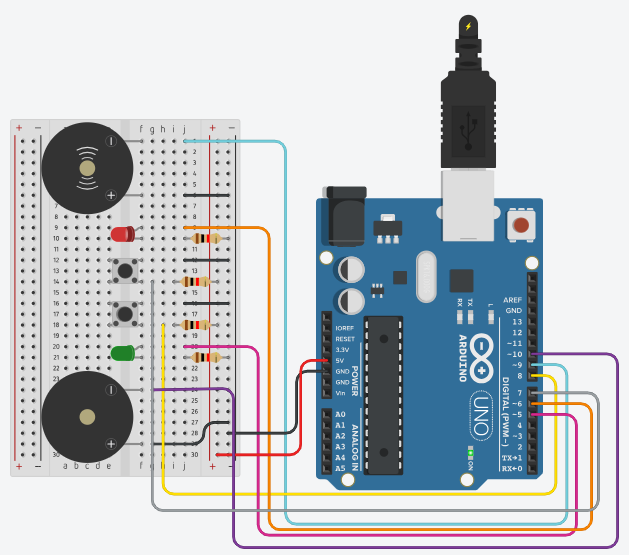
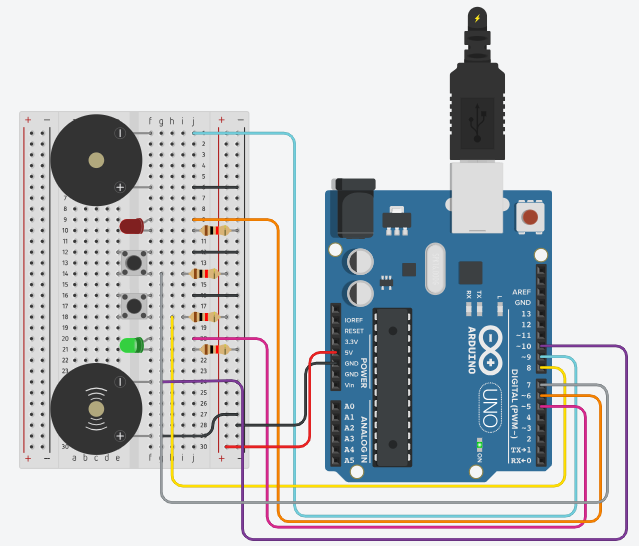
{

digitalWrite(ledPin2, LOW);

noTone(buzzerPin2);

}}}

**Output:**

**** ****

**Conclusion:**

Thus learnt about basic components of IOT like Arduino UNO (Blink Arduino on board LED), Breadboard, LEDS and Interfacing Piezo 2 Push buttons with Arduino to turn ON 2 LEDs and 2 Speakers with different Sounds with connection of resistors breadboard, jumper wires and Arduino.

**Practical :15**

**Aim:** Write a program to interface Peizo 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

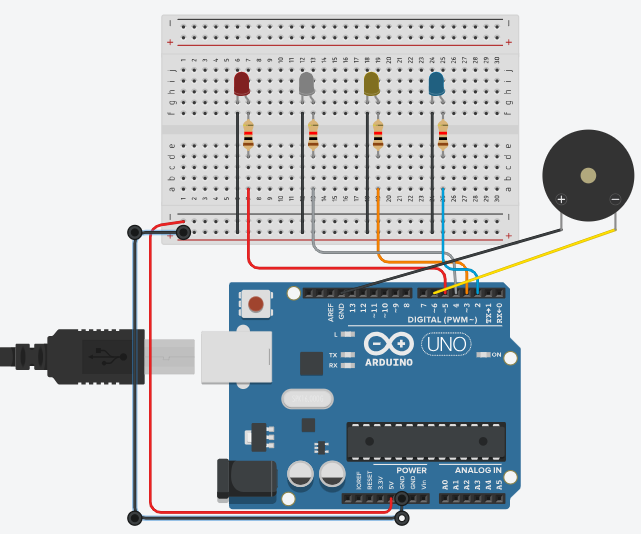
To learn LEDs

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

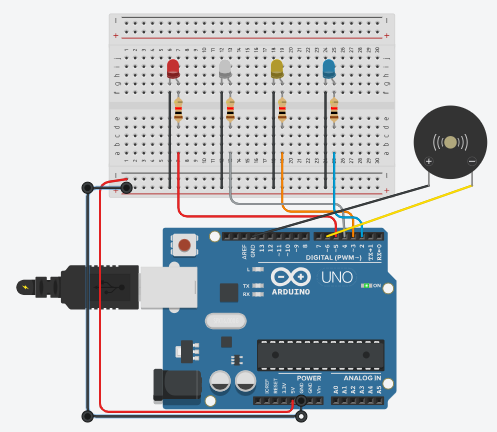
digitalWrite(yellowLED2, HIGH); //Yellow led on

delay(50); //wait for 1/25 of a second

digitalWrite(yellowLED2, LOW); //Yellow led off

}

**Output:**



**Conclusion:**

Thus learnt about basic components of IOT like Arduino UNO (Blink Arduino on board LED), Breadboard, LEDS and Interfacing FAST LED stream that goes back and forward while interfacing with buzzer connection of resistors breadboard, jumper wires and Arduino.

**Practical :16**

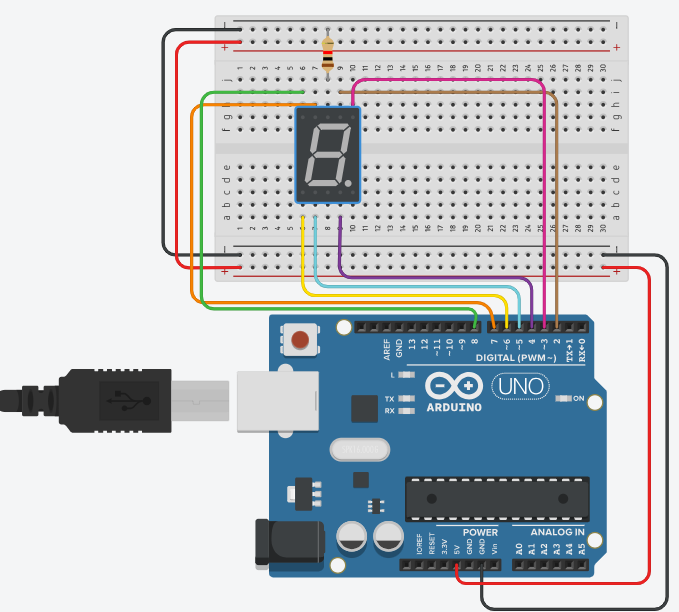
**Aim:** To interface a 7 segment display with Arduino & WAP to turn on display for displaying values from 0 to 9 and A to F.

**Objectives:**

* To learn Arduino UNO basics
* To learn Breadboard basics
* To learn about 7 segment display
* Write a program to interface a 7 segment display with Arduino & turning ON Segment display for displaying values from 0 to 9 and A to F.

**Components:** 7 Segment Display, Resistor, Arduino Uno r3, Breadboard, wires.

**Circuit:**



**Program Code:**

// make an array to save Sev Seg pin configuration of numbers

int num\_array[10][7] = { { 1,1,1,1,1,1,0 }, // 0

{ 0,1,1,0,0,0,0 }, // 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

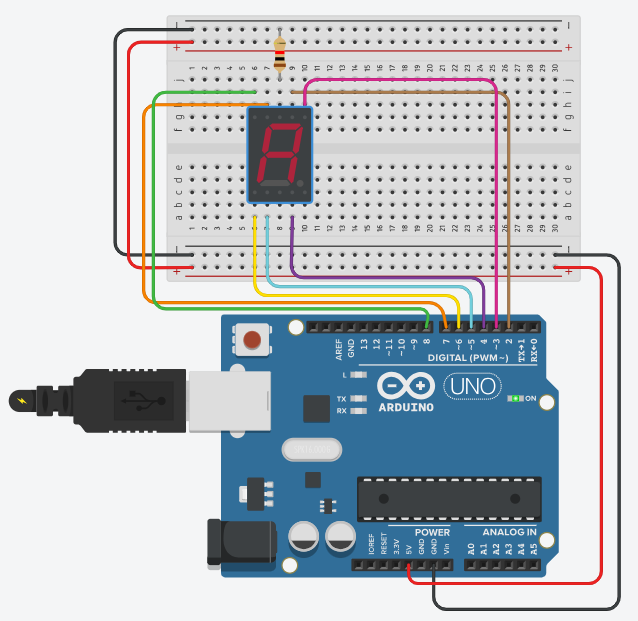
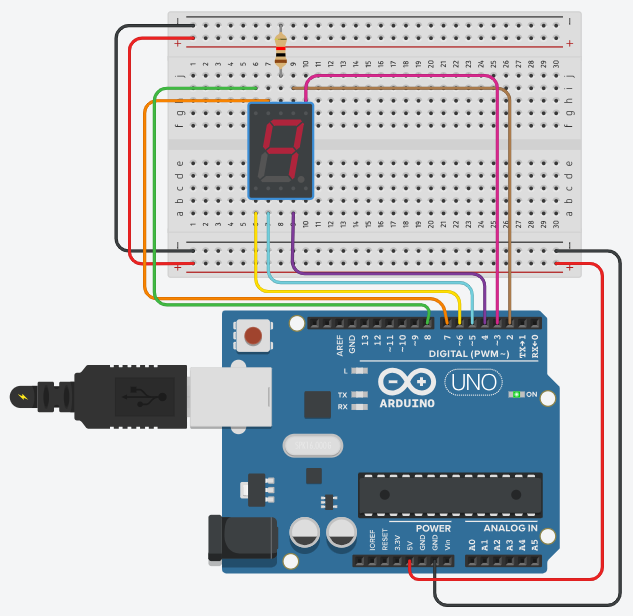
void WriteToSSD(int \* segmentArray )

{ for (int i=0; i < 7; i++)

digitalWrite(segAtoGpinouts[i], segmentArray[i]);

}

**Output:**

**Conclusion:**

Thus learnt about basic components of IOT like Arduino UNO (Blink Arduino on board LED), Breadboard, 7 segment display and Interfacing to turn ON 7 segment Display for displaying values from 0 to 9 & A to F with connection of resistors breadboard, jumper wires and Arduino.

**Practical No. 17**

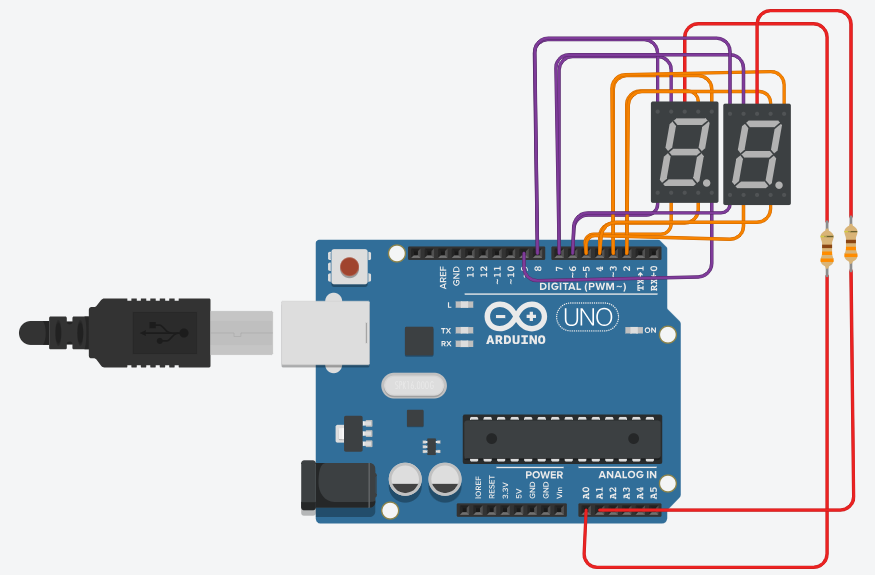
**Aim:** To interface a 7 segment display with Arduino & WAP to turn on display for displaying values from 11 to 51 .

**Objectives:**

* To learn Arduino UNO basics
* To learn Breadboard basics
* To learn about 7 segment display
* Write a program to interface a 7 segment display with Arduino & turning ON Segment display for displaying values from 11 to 51 .

**Components:** 2 (7 Segment Display), Resistor, Arduino Uno r3, Breadboard, wires.

**Circuit :**

****

**Code:**

// Segment values for numbers 0-9

byte segValue[10][7] = {

{0,0,0,0,0,0,1}, //0

{1,0,0,1,1,1,1}, //1

{0,0,1,0,0,1,0}, //2

{0,0,0,0,1,1,0}, //3

{1,0,0,1,1,0,0}, //4

{0,1,0,0,1,0,0}, //5

{0,1,0,0,0,0,0}, //6

{0,0,0,1,1,1,1}, //7

{0,0,0,0,0,0,0}, //8

{0,0,0,0,1,0,0} //9

};

// Define segment pins (A, B, C, D, E, F, G, DP)

byte segPin[8] = {2, 3, 4, 5, 6, 7, 8, 9};

// Define digit control pins

byte digitPin[2] = {A0, A1};

void setup() {

for(int i = 0; i < 8; i++) {

pinMode(segPin[i], OUTPUT);}

pinMode(digitPin[0], OUTPUT);

pinMode(digitPin[1], OUTPUT);

digitalWrite(digitPin[0], LOW);

digitalWrite(digitPin[1], LOW);

}

void loop() {

// Left display changes (1,2,3,4,5), Right display remains 1

int changingNumbers[] = {1, 2, 3, 4, 5};

for (int i = 0; i < 5; i++) {

unsigned long startTime = millis(); // Start time tracking

while (millis() - startTime < 1000) { // Display each number for 1 second

display\_N(changingNumbers[i], 1); // Left digit changes, right stays 1

} }}

void display\_N(int left, int right) {

for(int i = 0; i < 50; i++) { // Dynamic Refresh

segOutput(1, right); // Rightmost digit (always 1)

delay(5);

segOutput(0, left); // Leftmost digit (changes)

delay(5);

}}

void segClear() {

for(int i = 0; i < 8; i++) {

digitalWrite(segPin[i], HIGH);

}}

void segOutput(int d, int Number) {

segClear();

digitalWrite(digitPin[d], HIGH);

for(int i = 0; i < 7; i++) {

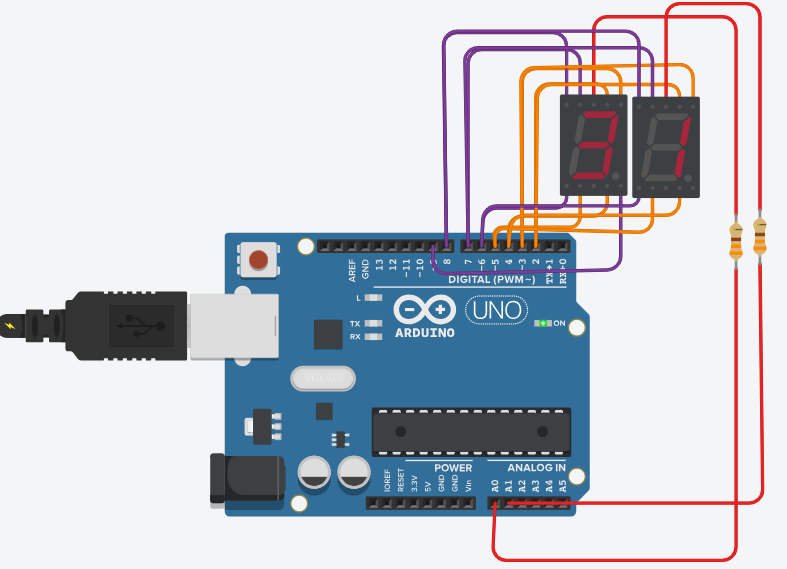
digitalWrite(segPin[i], segValue[Number][i]); }

delayMicroseconds(1000);

digitalWrite(digitPin[d], LOW);

}

**Output:**



**Conclusion:**

Thus learnt about basic components of IOT like Arduino UNO (Blink Arduino on board LED), Breadboard, 7 segment display and Interfacing to turn ON 7 segment Display for displaying values from 11 to 51 with connection of resistors breadboard, jumper wires and Arduino.

**Practical :18**

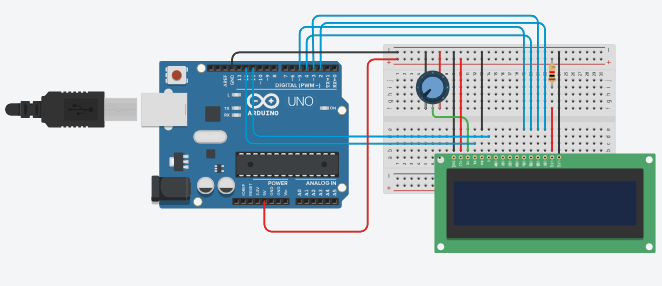
**Aim:** To Interface 16x2 Character LCD Module with Arduinio & WAP to display running with LCD character on parallel interface LCD controller chip.

**Objectives:**

* To learn Arduino UNO basics.
* To Learn Breadboard basics.
* To Learn aqbout 16x2 LCD.
* WAP to Display running with Character LCDs based on parallel interface LCD Controller chip.

**Components:** Arduino UNO R3,Breadboard,1 16x2 LCD,1 Resistor,1 Potentiometer

**Circuit:**



**Program Code:**

#include<LiquidCrystal.h>

LiquidCrystal lcd(12,11,5,4,3,2);

void setup()

{

lcd.begin(16,2);

lcd.clear();

}

void loop()

{

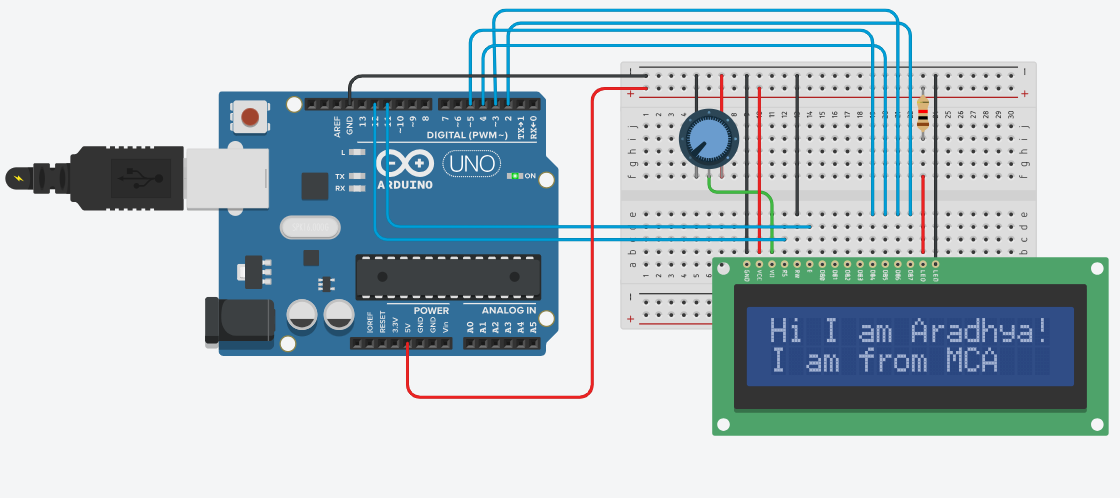
lcd.print("Hi I am Aradhya!");

lcd.setCursor(0,1);

lcd.print("I am from MCA ");

}

**Output:**



**Conclusion:**

Thus, learnt about 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 display characters with LCD controller chip with connections of resistors, breadboard, Jumper wires & Arduino.

**Practical No. 19**

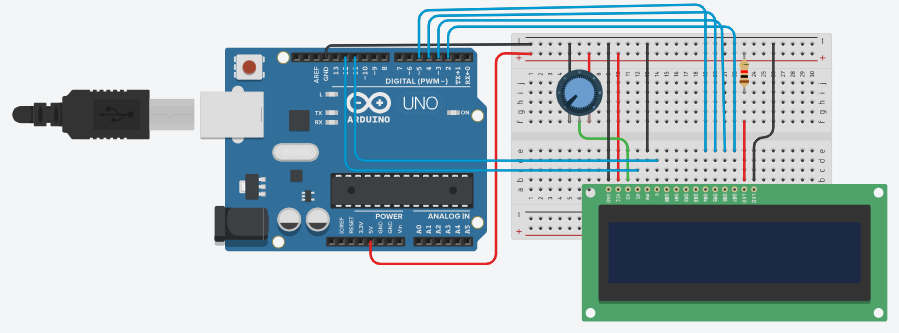
**Aim:** To interface 16x2 Character LCD Module with Arduino & WAP to display running with LCD to custom character generator based on parallel interface LCD Controller chip.

**Objectives:**

* To learn Arduino UNO basics
* To learn Breadboard basics
* To learn about 16x2 LCD
* Write a program to display running with Character LCDs based on parallel interface LCD Controller chip.

**Components:** LCD 16x2, Potentiometere, Resistor, Arduino Uno r3, Breadboard, wires.

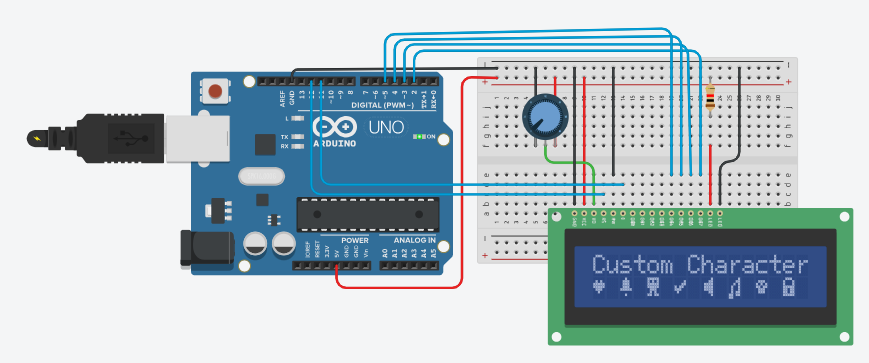
**Circuit Diagram:**



**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  
};  
  
byte Bell[8] = {  
0b00100,  
0b01110,  
0b01110,  
0b01110,  
0b11111,  
0b00000,  
0b00100,  
0b00000  
};  
  
  
byte Alien[8] = {  
0b11111,  
0b10101,  
0b11111,  
0b11111,  
0b01110,  
0b01010,  
0b11011,  
0b00000  
};  
  
byte Check[8] = {  
0b00000,  
0b00001,  
0b00011,  
0b10110,  
0b11100,  
0b01000,  
0b00000,  
0b00000  
};  
  
byte Speaker[8] = {  
0b00001,  
0b00011,  
0b01111,  
0b01111,  
0b01111,  
0b00011,  
0b00001,  
0b00000  
};  
  
  
byte Sound[8] = {  
0b00001,  
0b00011,  
0b00101,  
0b01001,  
0b01001,  
0b01011,  
0b11011,  
0b11000  
};  
  
  
byte Skull[8] = {  
0b00000,  
0b01110,  
0b10101,  
0b11011,  
0b01110,  
0b01110,  
0b00000,  
0b00000  
};  
  
byte Lock[8] = {  
0b01110,  
0b10001,  
0b10001,  
0b11111,  
0b11011,  
0b11011,  
0b11111,  
0b00000  
};  
  
void setup()  
{  
    // initialize LCD and set up the number of columns and rows:  
    lcd.begin(16, 2);  
  
    // create a new character  
    lcd.createChar(0, Heart);  
    // create a new character  
    lcd.createChar(1, Bell);  
    // create a new character  
    lcd.createChar(2, Alien);  
    // create a new character  
    lcd.createChar(3, Check);  
    // create a new character  
    lcd.createChar(4, Speaker);  
    // create a new character  
    lcd.createChar(5, Sound);  
    // create a new character  
    lcd.createChar(6, Skull);  
    // create a new character  
    lcd.createChar(7, Lock);  
  
    // Clears the LCD screen  
    lcd.clear();  
  
    // Print a message to the lcd.  
    lcd.print("Custom Character");  
}  
  
// 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 on board LED), Breadboard, 7 segment display and Interfacing 16 x2 LCD Display for displaying running with LCD to custom character generator with LCD Controller chip with connection of resistors breadboard, jumper wires and Arduino.