**Practical No. 1**

**Aim:** Introduction to Basic IoT Components.

**Objectives:**

1. To learn Arduino UNO basics
2. Write a program to blink Arduino onboard LED and to interface external LED with Arduino.

**Theory:**

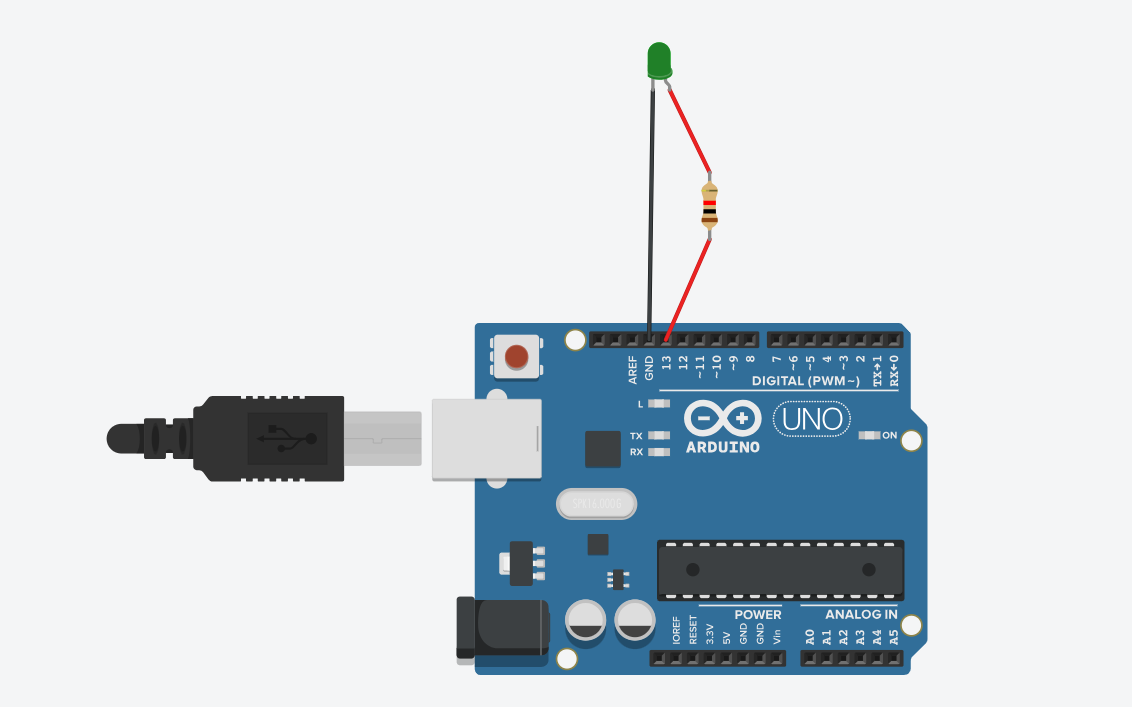
Arduino UNO R3 is an open-source electronics platform based on easy-to-use hardware and software. The setup() function is called when a sketch starts.

A loop() is a programming function that iterates a statement or condition based on specified boundaries.

The PinMode() function is used to configure a specific pin to behave either as an input or an output. The

delay() function allows you to pause the execution of your Arduino program for a specified period. The digitalWrite() function is used to write a HIGH or a LOW value to a digital pin.

**Circuit Diagram:**

****

**Program:**

void setup()

{

pinMode(LED\_BUILTIN, OUTPUT);

}

void loop()

{

digitalWrite(LED\_BUILTIN, HIGH);

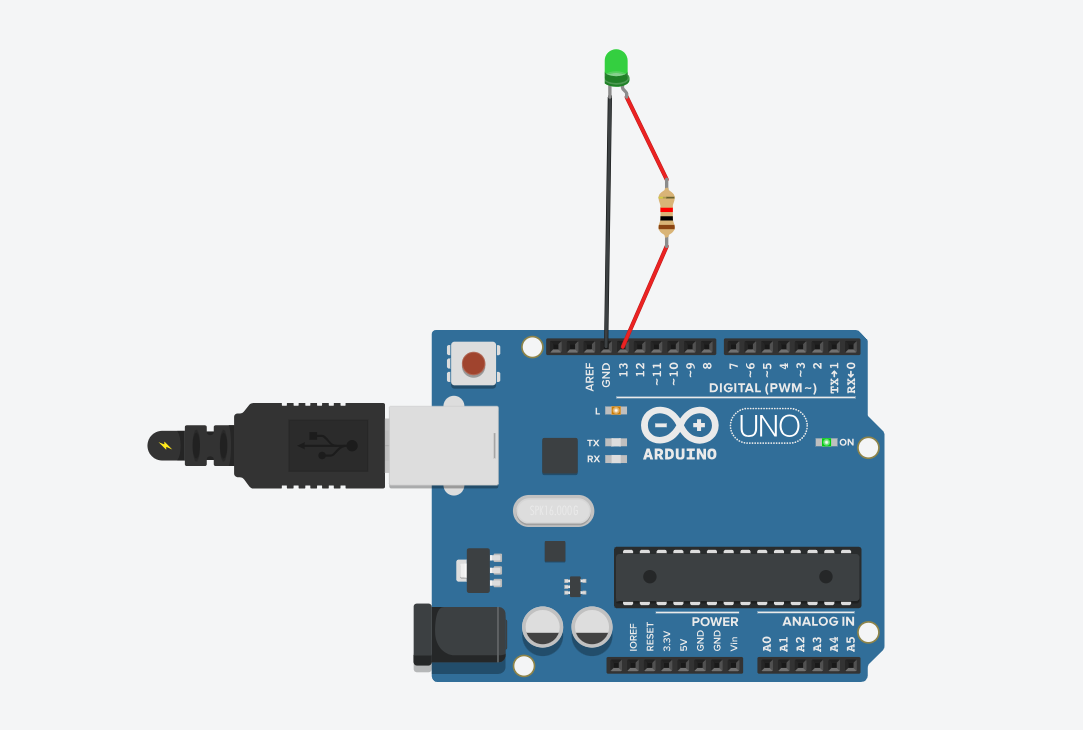
delay(2000);

digitalWrite(LED\_BUILTIN, LOW);

delay(2000);

}

**Output:**



**Conclusion:** Thus, learnt about basic components of IoT like Arduino UNO, Breadboard, resisters, LED’s and interfacing LED with Arduino.

**Practical No. 2**

**Aim:** To interface 5 LED’s with Arduino and write a program to blink 6 LEDs, one at a time, in a back-and-forth formation.

**Objectives:**

1. To interface 6 LEDs with Arduino
2. Write a Program to blink 6 LEDs, one at a time, in a back-and-forth formation.

**Theory:**

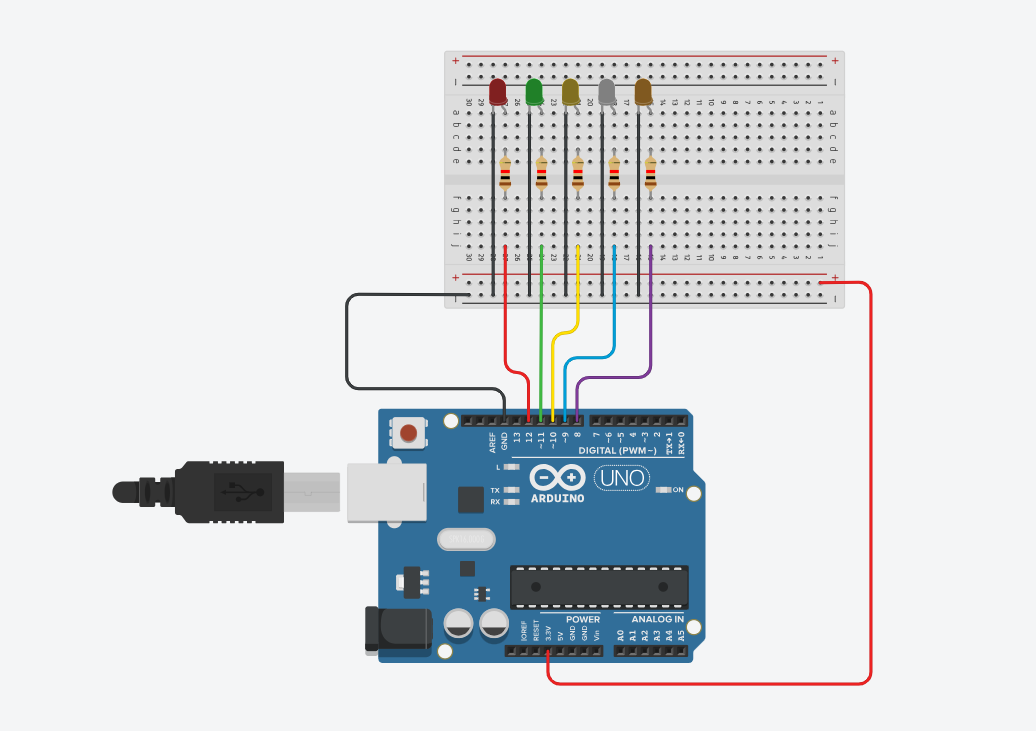
Arduino UNO R3 is an open-source electronics platform based on easy-to-use hardware and software. The setup() function is called when a sketch starts.

A loop() is a programming function that iterates a statement or condition based on specified boundaries.

The PinMode() function is used to configure a specific pin to behave either as an input or an output. The

delay() function allows you to pause the execution of your Arduino program for a specified period. The digitalWrite() function is used to write a HIGH or a LOW value to a digital pin.

**Circuit Diagram:**

****

**Program:**

// C++ code

//

void setup()

{

pinMode(12, OUTPUT);

pinMode(11, OUTPUT);

pinMode(10, OUTPUT);

pinMode(9, OUTPUT);

pinMode(8, OUTPUT);

}

void loop()

{

digitalWrite(12, HIGH);

delay(1000);

digitalWrite(12, LOW);

digitalWrite(11, HIGH);

delay(1000);

digitalWrite(11, LOW);

digitalWrite(10, HIGH);

delay(1000);

digitalWrite(10, LOW);

digitalWrite(9, HIGH);

delay(1000);

digitalWrite(9, LOW);

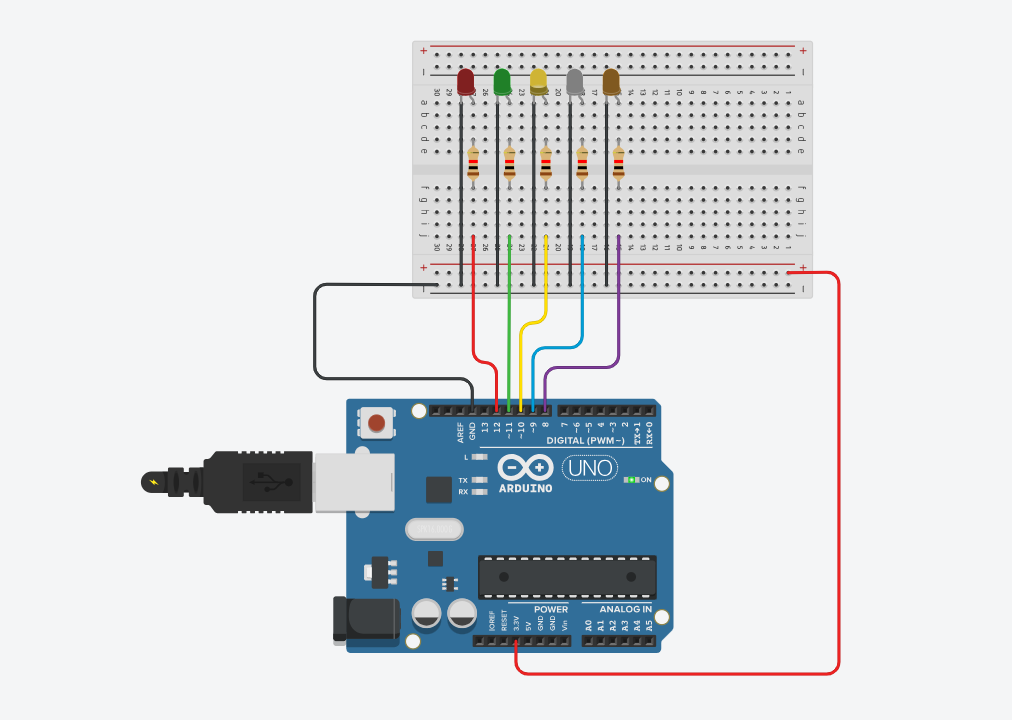
digitalWrite(8, HIGH);

delay(1000);

digitalWrite(8, LOW);

}

**Output:**



**Conclusion:** Thus, learnt about basic components of IoT like Arduino UNO, Breadboard, resisters, LED’s and interfacing LED with Arduino.

**Practical No. 3**

**Aim:** To interface Push button with Arduino and write a program to turn ON LED when push button is pressed.

**Objectives:**

1. To interface the Push button with Arduino
2. Write a Program to turn ON the LED wthe hen push button is pressed.

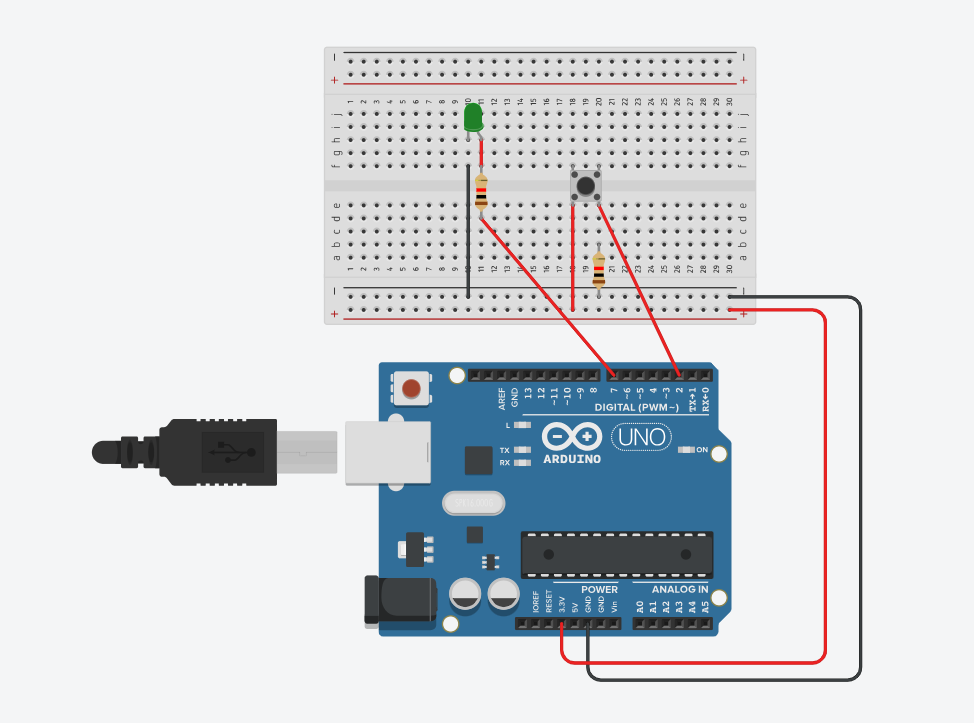
**Theory:**

Arduino UNO R3 is an open-source electronics platform based on easy-to-use hardware and software.

A Breadboard is a solderless construction base used for developing an electronic circuit and wiring for projects with microcontroller boards like Arduino.

A Resistor is a passive components, which means that they don't generate any electricity at all, but rather reduce voltage and current by dissipating power in the form of heat. The unit of resistance is ohms (Ω).

**Circuit Diagram:**

****

**Program:**

// C++ code

//

void setup()

{

pinMode(7, OUTPUT);

pinMode(2, INPUT);

}

void loop()

{

int bn = digitalRead(2);

if(bn==HIGH)

{

digitalWrite(7, HIGH);

}

else

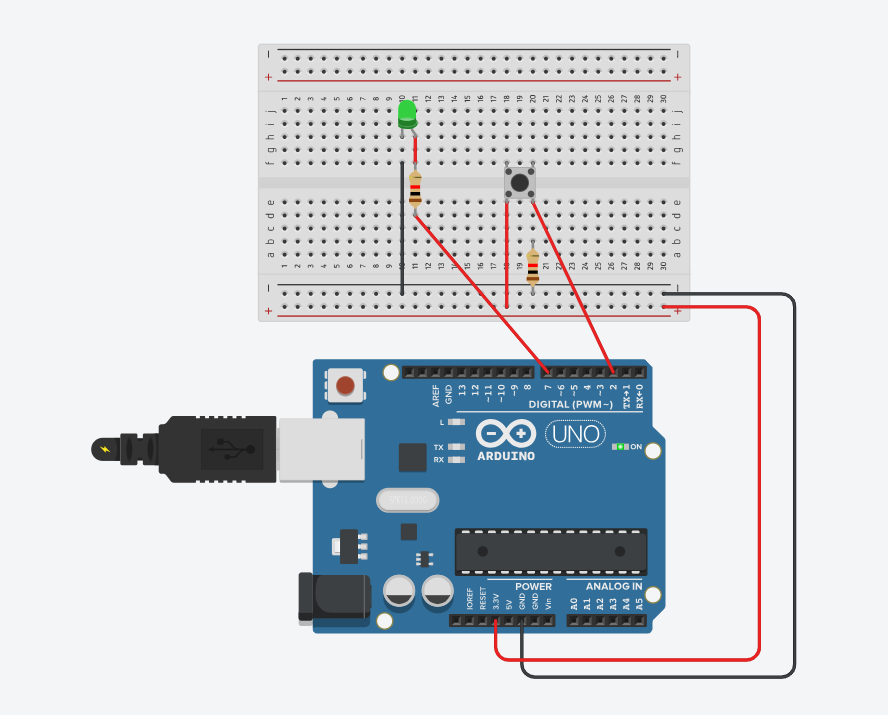
{

digitalWrite(7, LOW);

}

}

**Output:**



**Conclusion:** Thus, learned about basic components of IoT like Arduino UNO, Breadboard, resistors, LEDs, push buttons and controlling LED glow using push buttons.

**Practical No. 4**

**Aim:** To interface the Push button, Speaker/buzzer with Arduino, and write a program to turn ON LED and generate a note or tone when the push button is pressed.

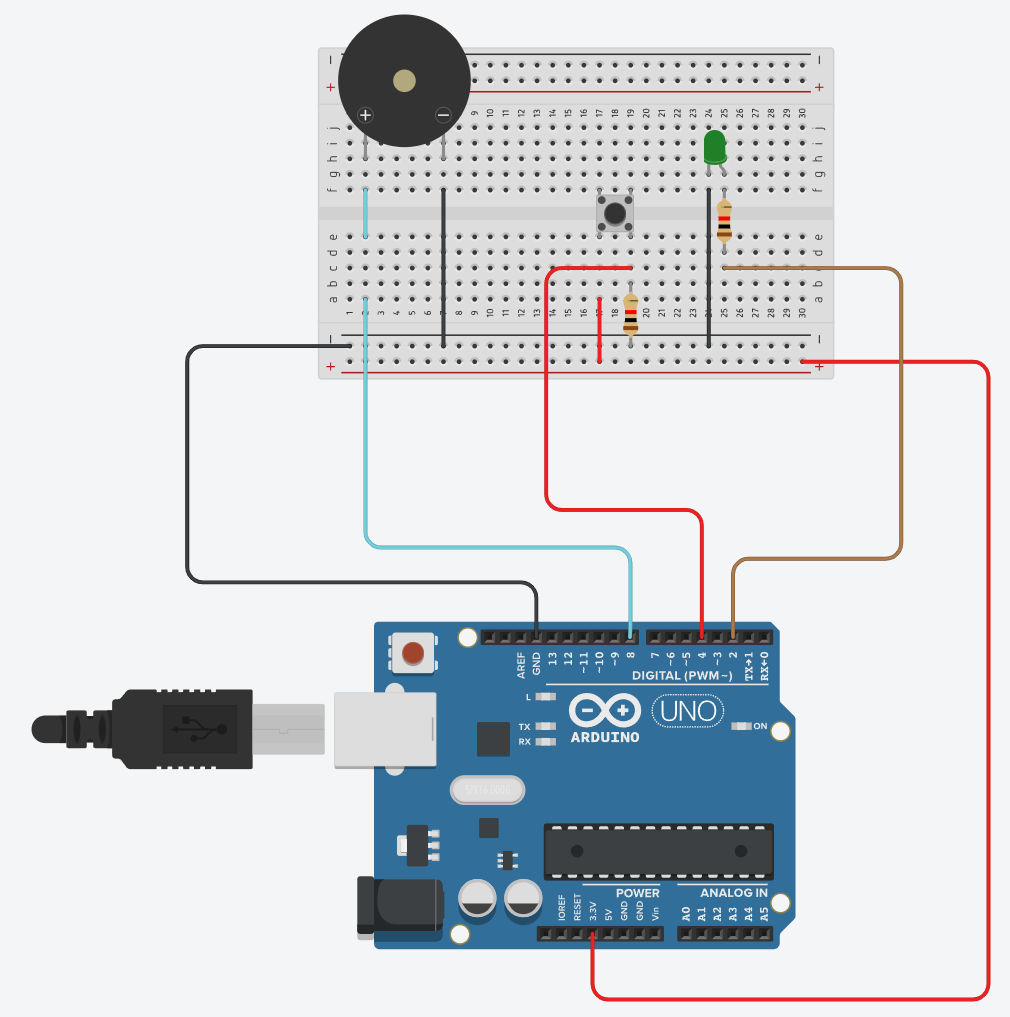
**Objectives:**

1. To interface the Push button, Speaker/buzzer with Arduino.
2. Write a Program to turn ON the LED and generate a note or tone when the push button is pressed.

**Theory:**

Buzzer:An Arduino Buzzer is basically a beeper. The Arduino buzzer is a device that produces sound when an electric current is passed through it.

**Circuit Diagram:**



**Program:**

int LED = 2;

int BUTTON = 4;

int BUZZER = 8;

void setup()

{

pinMode(BUTTON, INPUT);

pinMode(BUTTON, INPUT);

pinMode(BUZZER, OUTPUT);

}

void loop()

{

int bs = digitalRead(BUTTON);

if (bs == HIGH)

{

digitalWrite(BUZZER,HIGH);

digitalWrite(LED,HIGH);

}

else

{

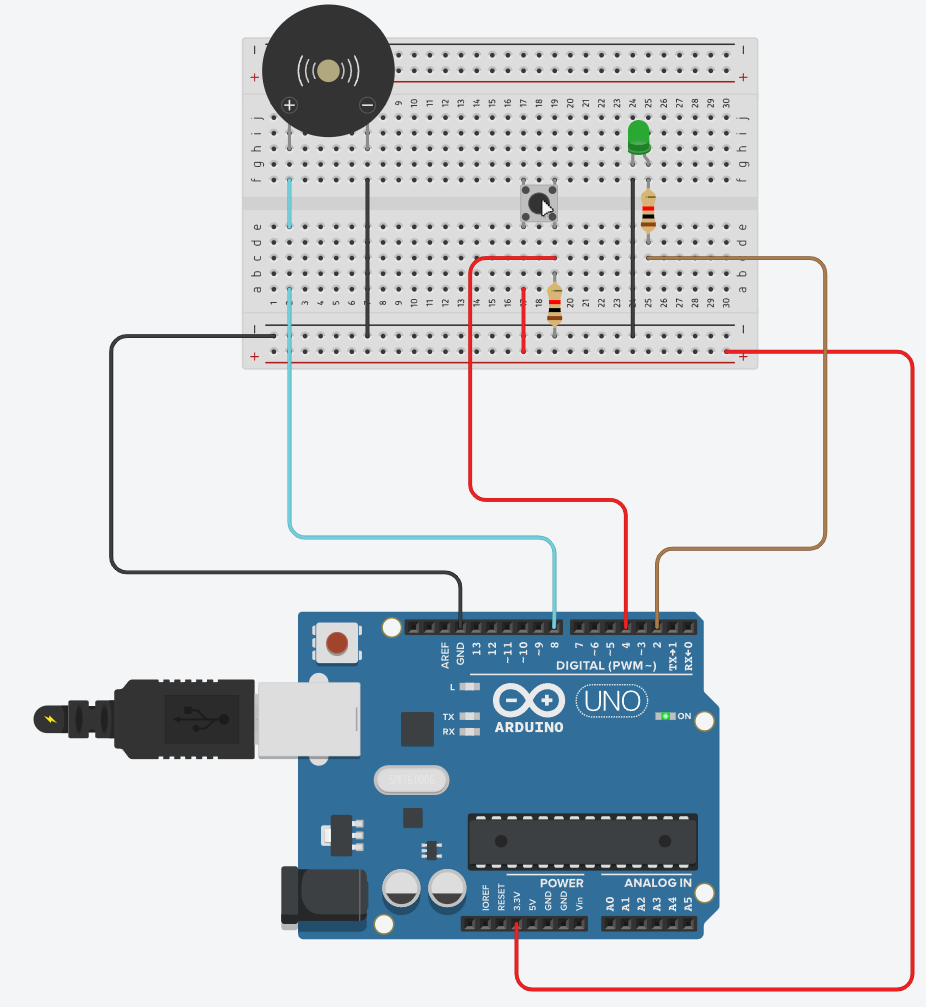
digitalWrite(BUZZER,LOW);

digitalWrite(LED,LOW);

}

}

**Output:**



**Conclusion:** Thus, learnt about basic components of IoT like Arduino UNO, Breadboard, resisters, LED’s and interfacing LED with Arduino.

**Practical No. 5**

**Aim:** To interface 2 Push buttons, a Speaker with Arduino and write a program to turn ON LED and generate 2 different notes on two button keyboard.

**Objectives:**

1. To interface 2 Push buttons, a Speaker with Arduino
2. Write a Program To to turn ON LED and generate 2 different notes on two button keyboards.

**Theory:**

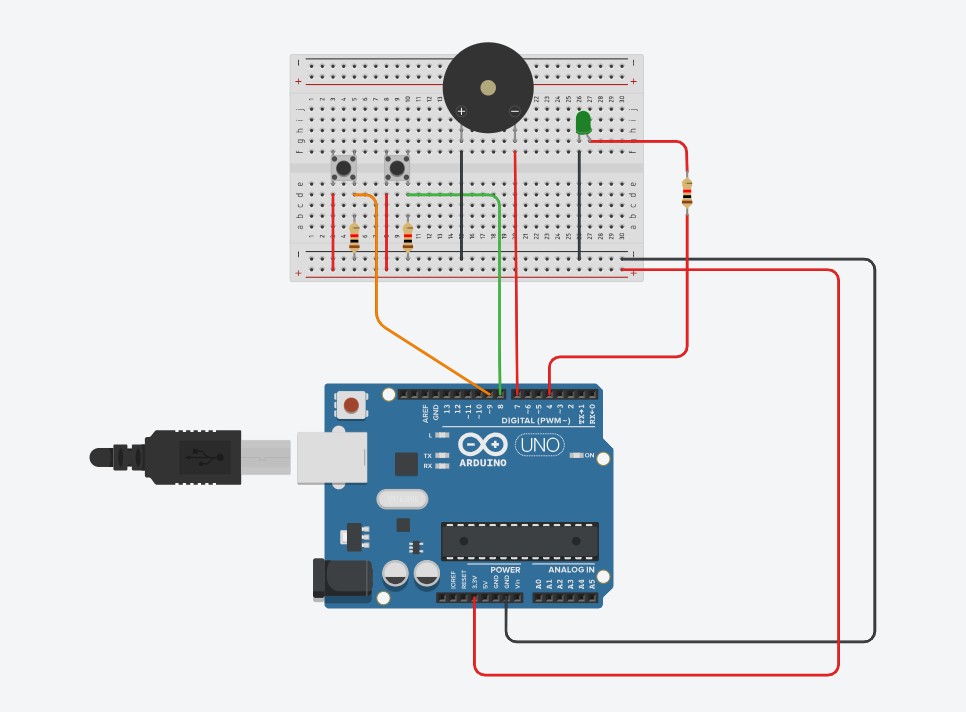
Arduino UNO R3 is an open-source electronics platform based on easy-to-use hardware and software. The setup() function is called when a sketch starts.

A loop() is a programming function that iterates a statement or condition based on specified boundaries.

The PinMode() function is used to configure a specific pin to behave either as an input or an output. The

delay() function allows you to pause the execution of your Arduino program for a specified period. The digitalWrite() function is used to write a HIGH or a LOW value to a digital pin.

**Circuit Diagram:**

****

**Program:**

// C++ code

//

int btnStatus;

int btnStatus2;

#define NOTE\_D8 4699

#define NOTE\_DS8 4978

#define NOTE\_A4 440

#define NOTE\_AS4 466

void setup()

{

pinMode(7, OUTPUT);

pinMode(4, OUTPUT);

pinMode(8, INPUT);

pinMode(9, INPUT);

}

void loop()

{

btnStatus = digitalRead(8);

btnStatus2 = digitalRead(9);

if(btnStatus == HIGH)

{

tone(7, NOTE\_A4, 100);

digitalWrite(4, HIGH);

digitalWrite(7, HIGH);

}

else if(btnStatus2 == HIGH)

{

tone(7, NOTE\_AS4, 100);

digitalWrite(4, HIGH);

digitalWrite(7, HIGH);

}

else

{

noTone(7);

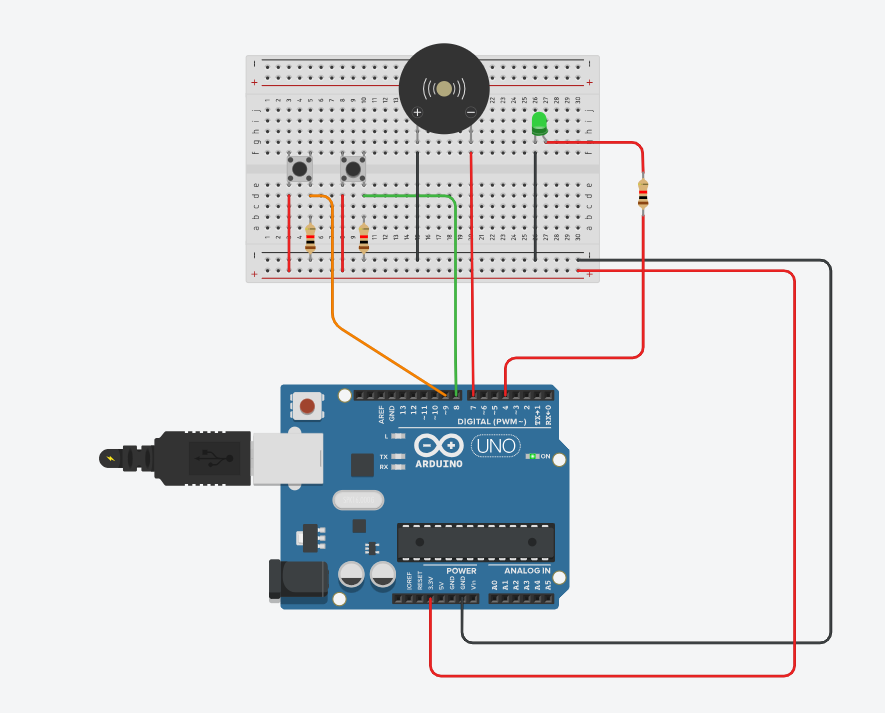
digitalWrite(4, LOW);

digitalWrite(7, LOW);

}

}

**Output:**



**Conclusion:** Thus, learned about basic components of IoT like Arduino UNO, Breadboard, resistors, Piezo buzzer, LED, push buttons, and controlling LED glow and Piezo buzzer using two push buttons simultaneously with different tones.

**Practical No. 6**

**Aim:** To interface Seven Segment Display (SSD) with Arduino and write a program to blink SSD.

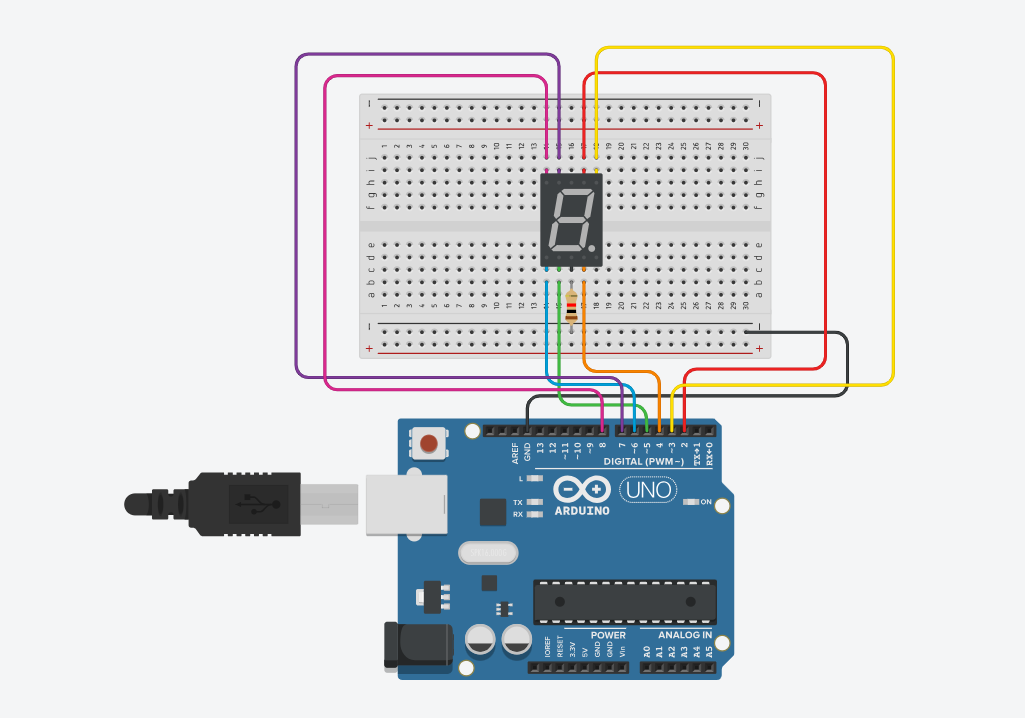
**Objectives:**

1. To interface Seven Segment Display (SSD) with Arduino
2. Write a Program to blink SSD.

**Theory:**

SSD (Seven Segment Display) are made up of 8 LED segments. They are used to display Numbers (0-9) and certain Alphabets (like C, A, H, P, etc.). 7 of these LED segments are in the shape of a line, whereas 1 segment is circular.

**Circuit Diagram:**

****

**Program:**

// C++ code

//

int a = 2;

int b = 3;

int c = 4;

int d = 5;

int e = 6;

int f = 7;

int g = 8;

void setup()

{

pinMode(a, OUTPUT);

pinMode(b, OUTPUT);

pinMode(c, OUTPUT);

pinMode(d, OUTPUT);

pinMode(e, OUTPUT);

pinMode(f, OUTPUT);

pinMode(g, OUTPUT);

}

void loop()

{

digitalWrite(a, HIGH);

digitalWrite(b, HIGH);

digitalWrite(c, HIGH);

digitalWrite(d, HIGH);

digitalWrite(e, HIGH);

digitalWrite(f, HIGH);

digitalWrite(g, LOW);

delay(1000);

digitalWrite(a, LOW);

digitalWrite(b, LOW);

digitalWrite(c, LOW);

digitalWrite(d, LOW);

digitalWrite(e, LOW);

digitalWrite(f, LOW);

digitalWrite(g, LOW);

delay(1000);

}

**Output:**



**Conclusion:** Thus, learned about basic components of IoT like Arduino UNO, Breadboard, resistor, SSD, and interfacing SSD with Arduino through the short blinking duration.

**Practical No. 7**

**Aim:** To interface Seven Segment Display (SSD) with Arduino and write a program to print numbers from 1 to 9 on SSD. Arduino.

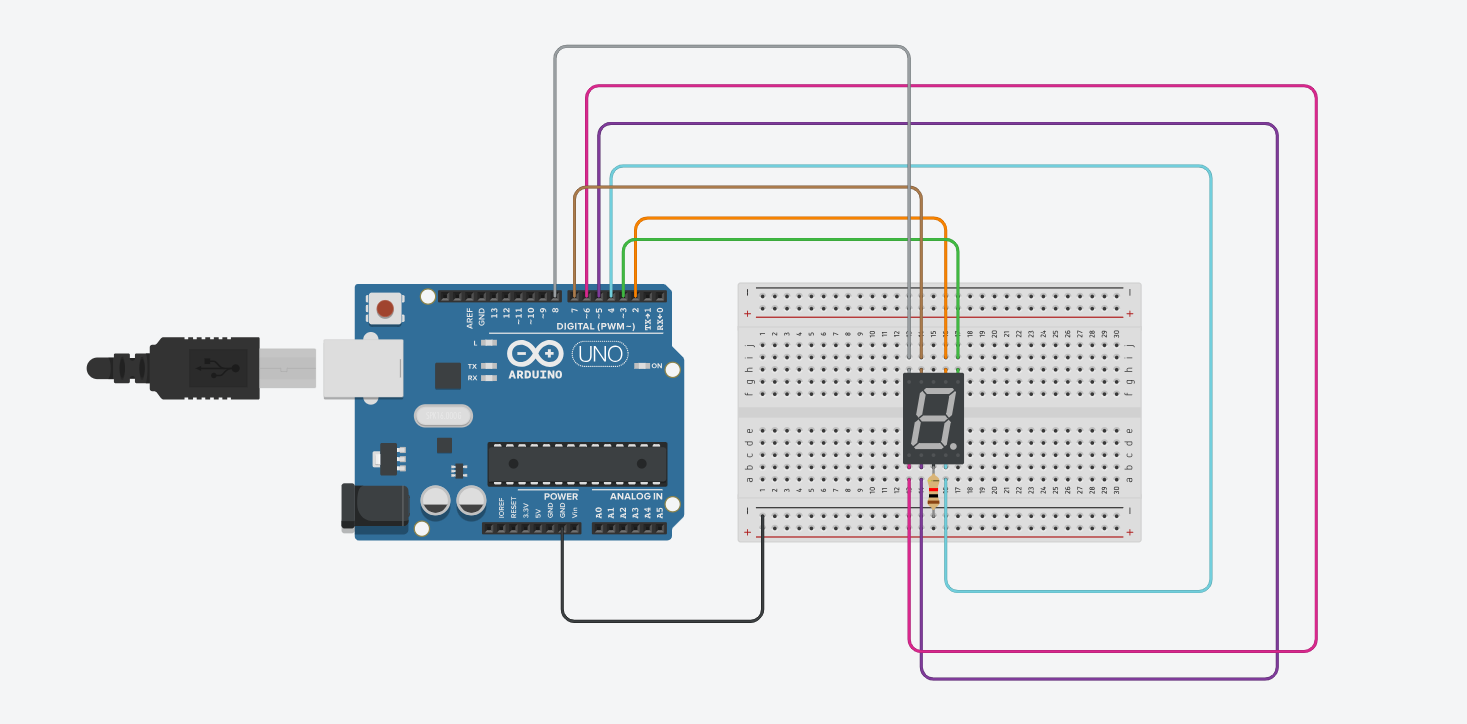
**Objectives:**

1. To interface Seven Segment Display (SSD) with Arduino
2. Write a Program to print numbers from 1 to 9 on SSD.

**Theory:**

SSD (Seven Segment Display) are made up of 8 LED segments. They are used to display Numbers (0-9) and certain Alphabets (like C, A, H, P, etc.). 7 of these LED segments are in the shape of a line, whereas 1 segment is circular.

**Circuit Diagram:**

****

**Program:**

// C++ code

//

int a = 2;

int b = 3;

int c = 4;

int d = 5;

int e = 6;

int f = 7;

int g = 8;

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

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

};

//function header

void Num\_Write(int);

void setup()

{

pinMode(a, OUTPUT);

pinMode(b, OUTPUT);

pinMode(c, OUTPUT);

pinMode(d, OUTPUT);

pinMode(e, OUTPUT);

pinMode(f, OUTPUT);

pinMode(g, OUTPUT);

}

void loop()

{

//counter loop

for(int counter = 0; counter < 10; ++counter) //1

{

delay(1000);

Num\_Write(counter);

}

delay(1000);

}

// this function writes values to the seven segment pins

void Num\_Write(int number) //0

{

int pin = 2;

for(int j = 0; j < 7; j++)

{

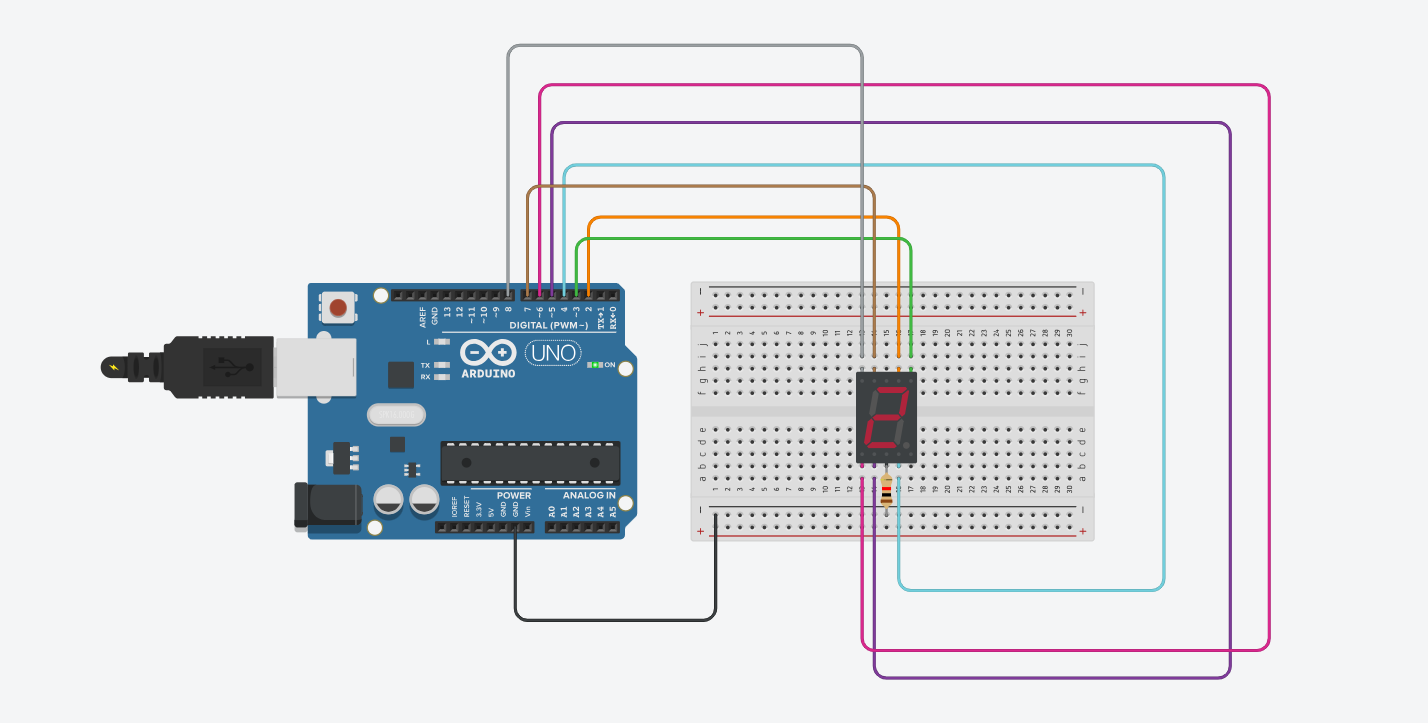
digitalWrite(pin, num\_array[number][j]); //0 0

pin++;

}

}

**Output:**



**Conclusion:** Thus, learned about basic components of IoT like Arduino UNO, Breadboard, resistor, SSD, and interfacing SSD with Arduino displaying numbers from 1 to 9.

**Practical No. 8**

**Aim:** To interface LCD, push button, and potentiometer with Arduino and write a program to display a message on LCD when the push button is pressed.

**Objectives:**

1. To interface LCD, push button, and potentiometer with Arduino
2. Write a Program to display a message on LCD when the push button is pressed.

**Theory:**

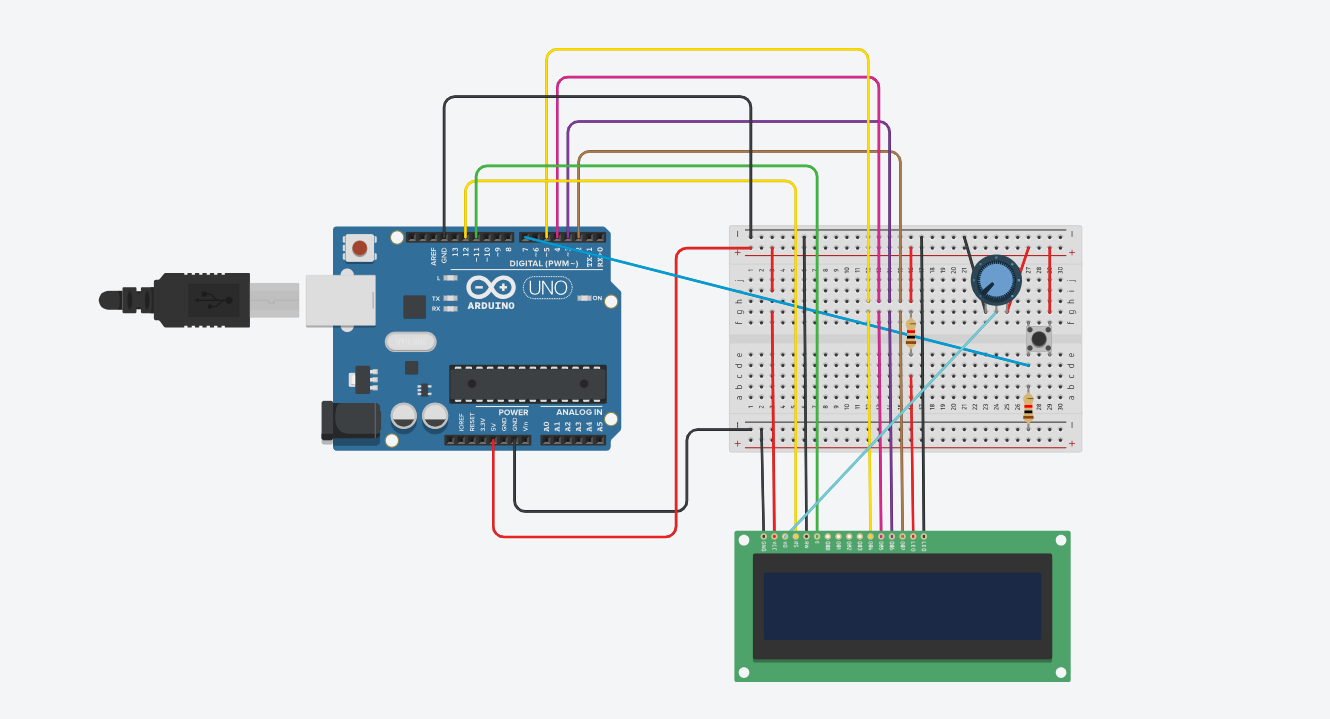
**Arduino UNO R3** is an open-source electronics platform based on easy-to-use hardware and software.

A **breadboard** is a solderless construction base used for developing an electronic circuit and wiring for projects with microcontroller boards like Arduino.

The **LCD (Liquid Crystal Display)** is a type of display that uses the liquid crystals for its operation. Here, we will accept the serial input from the computer and upload the sketch to the Arduino. The characters will be displayed on the LCD.

A **potentiometer** is a simple mechanical device that provides a varying amount of resistance when its shaft is turned. By passing voltage through a potentiometer and into an analog input on your board, it is possible to measure the amount of resistance produced by a potentiometer (or *pot* for short) as an analog value.

**Circuit Diagram:**

****

**Program:**

// C++ code

//

#include <LiquidCrystal.h>

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

int e=7; //button pin

// Practical 8

void setup()

{

pinMode(e, INPUT);

lcd.begin(16,2);

}

void loop()

{

if(digitalRead(e)==HIGH)

{

lcd.clear();

lcd.setCursor(1,0);

lcd.print("Button Pressed");

delay(200);

}

else

{

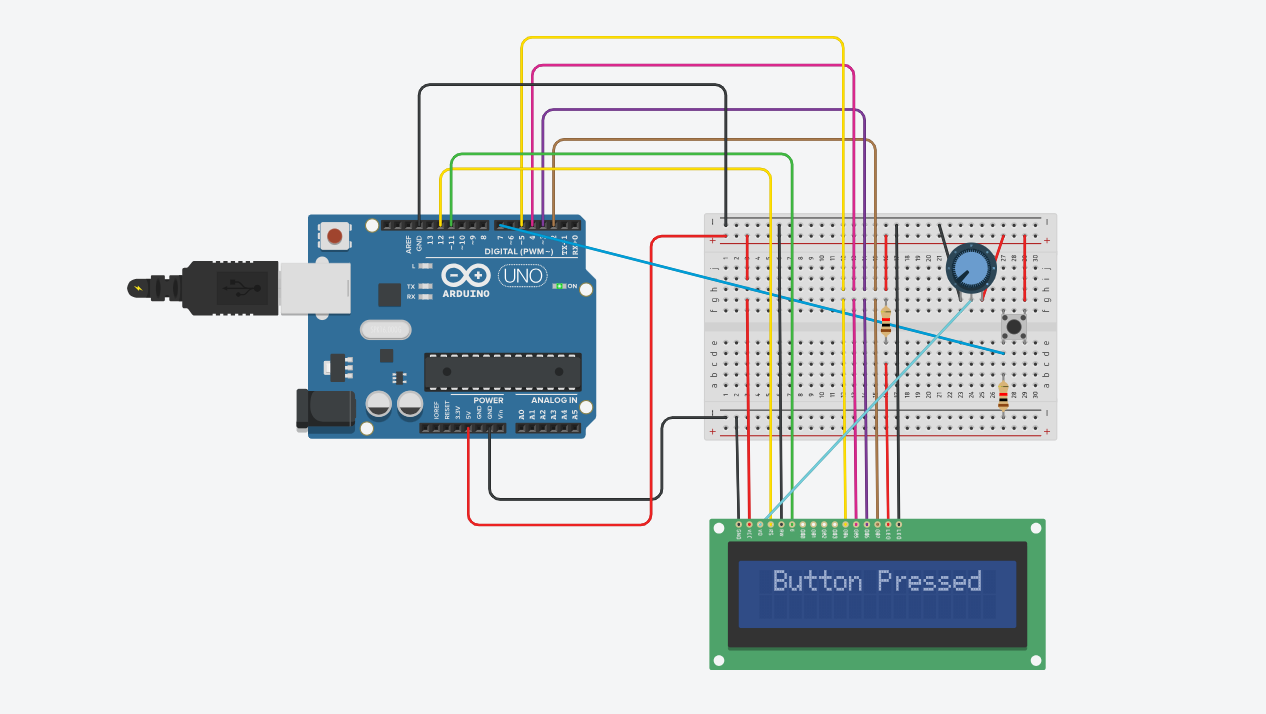
lcd.setCursor(1,0);

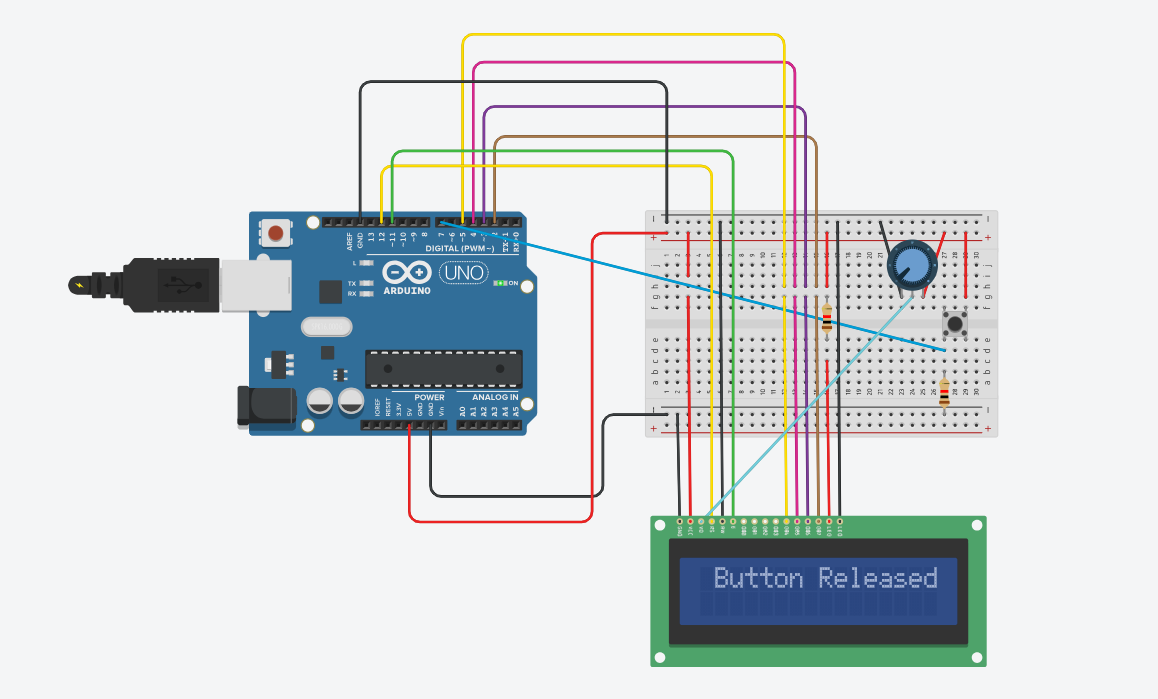
lcd.print("Button Released");

}

}

**Output:**





**Conclusion:** Thus, learned how to interface LCD, push button, and potentiometer with Arduino and write a program to display messages on LCD when push button is pressed.

**Practical No. 9**

**Aim:** To interface LCD, push button, potentiometer with Arduino and write a program to display the no. of times (count) the push button is pressed on LCD.

**Objectives:**

1. To interface LCD, push button, potentiometer with Arduino
2. Write a Program to display the no. of times (count) the push button is pressed on LCD.

**Theory:**

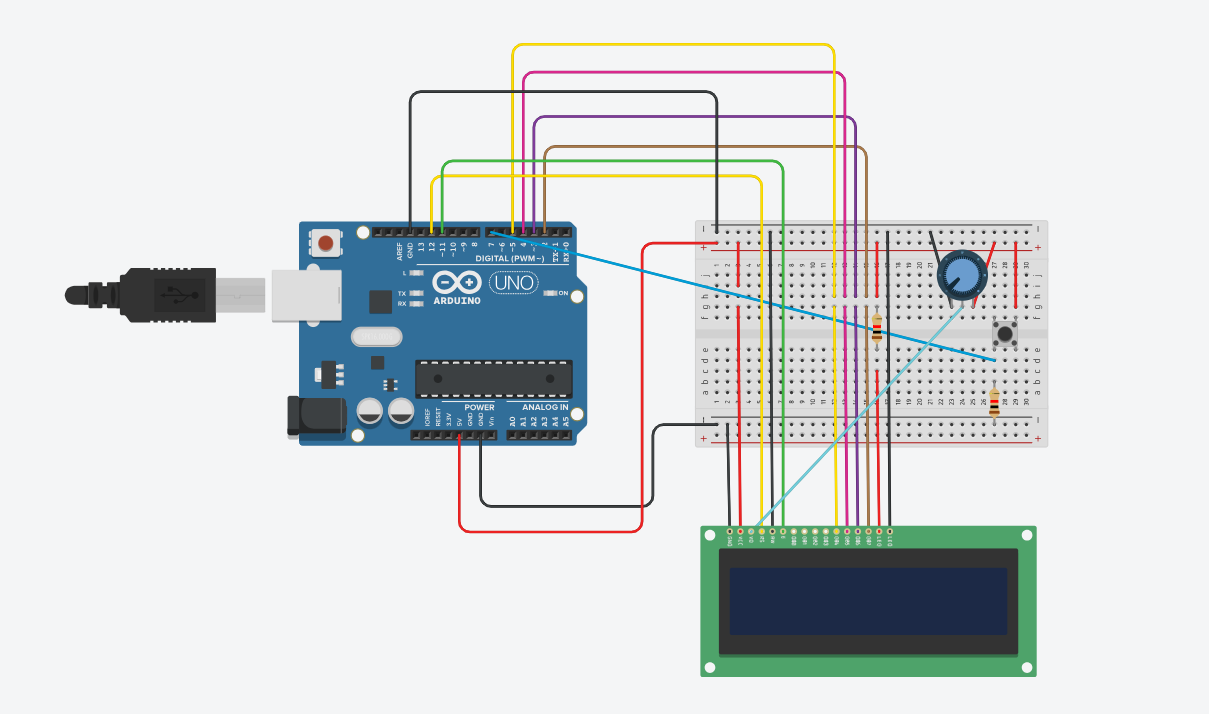
**Arduino UNO R3** is an open-source electronics platform based on easy-to-use hardware and software.

A **breadboard** is a solderless construction base used for developing an electronic circuit and wiring for projects with microcontroller boards like Arduino.

The **LCD (Liquid Crystal Display)** is a type of display that uses the liquid crystals for its operation. Here, we will accept the serial input from the computer and upload the sketch to the Arduino. The characters will be displayed on the LCD.

A **potentiometer** is a simple mechanical device that provides a varying amount of resistance when its shaft is turned. By passing voltage through a potentiometer and into an analog input on your board, it is possible to measure the amount of resistance produced by a potentiometer (or *pot* for short) as an analog value.

**Circuit Diagram:**

****

**Program:**

// C++ code

//

#include <LiquidCrystal.h>

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

int e=7; //button pin

// Practical 9 - Push Button Count with LCD

int count=0;

void setup()

{

pinMode(e,INPUT);

lcd.begin(16,2);

}

void loop()

{

int button = digitalRead(e);

lcd.setCursor(3,1);

lcd.print(count);

if(button==HIGH)

{

count++;

lcd.setCursor(3,1);

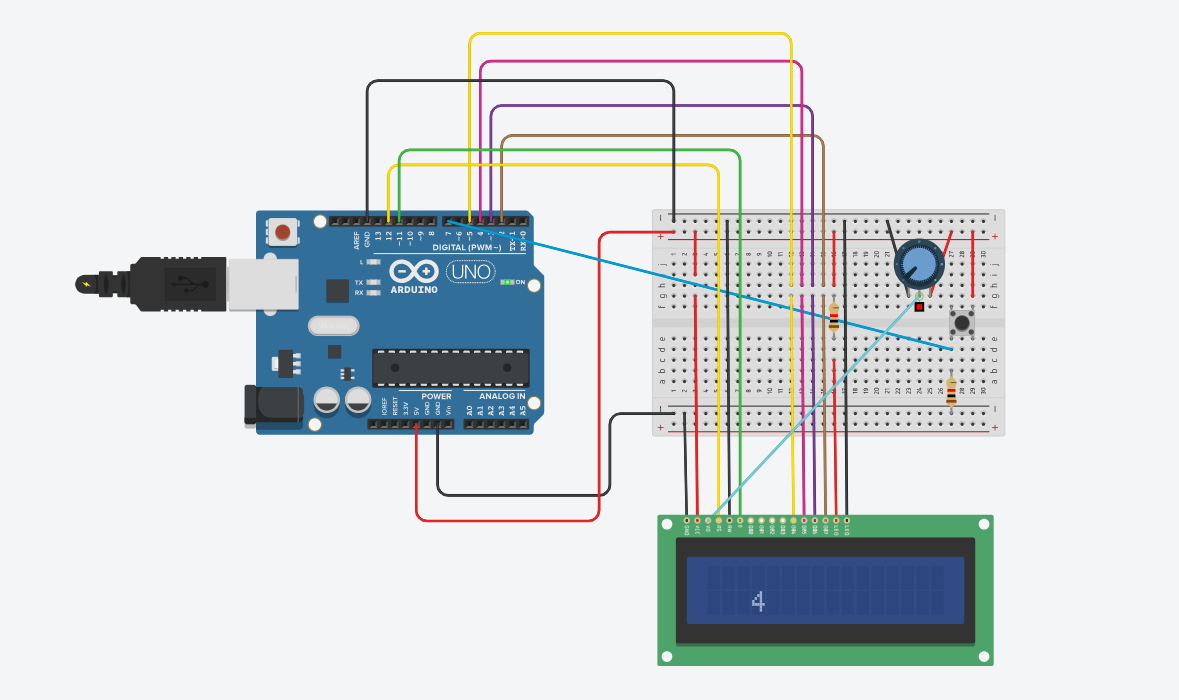
lcd.print(count);

delay(200);

}

}

**Output:**



**Conclusion:** Thus, learned how to interface LCD, push button, and potentiometer with Arduino and write a program to display the no. of times (count) the push button is pressed on the LCD.

**Practical No. 10**

**Aim:** To interface LEDs, and potentiometer with Arduino and write a program to turn on or off more of the LEDs by turning the potentiometer knob.

**Objectives:**

1. To interface LEDs, and potentiometer with Arduino
2. Write a Program to turn on or off more of the LEDs by turning the potentiometer knob.

**Theory:**

**Arduino UNO R3** is an open-source electronics platform based on easy-to-use hardware and software.

A **breadboard** is a solderless construction base used for developing an electronic circuit and wiring for projects with microcontroller boards like Arduino.

A **potentiometer** is a simple mechanical device that provides varying resistance when its shaft is turned. By passing a voltage through a potentiometer and into an analog input on your board, it is possible to measure the amount of resistance produced by a potentiometer (or *pot* for short) as an analog value.

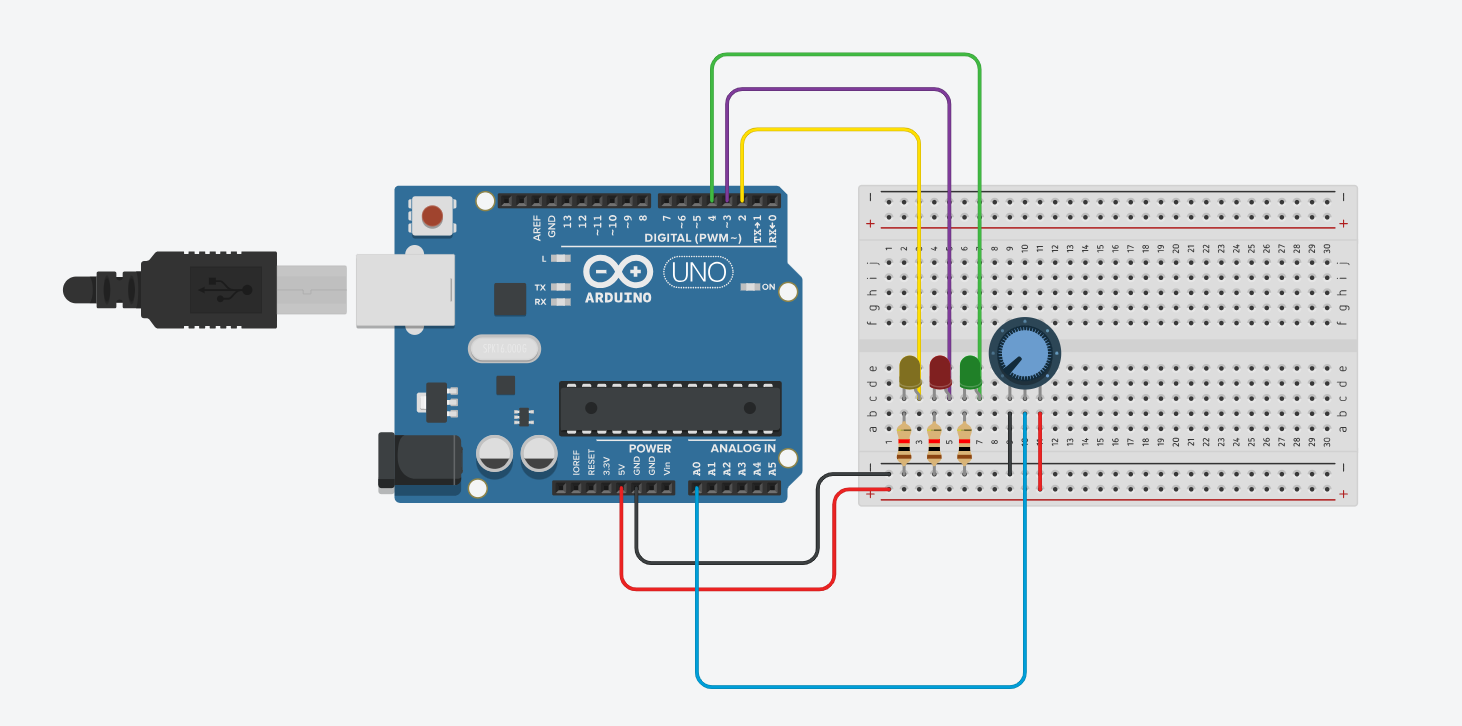
A **resistor is** a passive component, which means that they don't generate any electricity at all, but rather reduces voltage and current by dissipating power in the form of heat. The unit of resistance is ohms (Ω).

The **setup()** function is called when a sketch starts.

A **loop** is a programming function that iterates a statement or condition based on specified boundaries

The **pinMode()** function is used to configure a specific pin to behave either as an input or an output.

**Circuit Diagram:**

****

**Program:**

// C++ code

// Practical 10 Potentiometer Controls 3 LEDs

int input = 0;

int potpin = A0;

void setup()

{

pinMode(2, OUTPUT);

pinMode(3, OUTPUT);

pinMode(4, OUTPUT);

pinMode(A0, INPUT);

}

/\*

void loop()

{

input = analogRead(potpin);

if(input < 341) // 682 1024

{

digitalWrite(2,HIGH);

digitalWrite(3, LOW);

digitalWrite(4, LOW);

}

else if (input < 682)

{

digitalWrite(2, LOW);

digitalWrite(3, HIGH);

digitalWrite(4, LOW);

}

else if (input < 1024)

{

digitalWrite(2, LOW);

digitalWrite(3, LOW);

digitalWrite(4, HIGH);

}

}

\*/

void loop()

{

input = analogRead(potpin);

if(input <= 0)

{

digitalWrite(2, LOW);

digitalWrite(3, LOW);

digitalWrite(4, LOW);

}

else if(input < 341) // 682 1024

{

digitalWrite(2, HIGH);

digitalWrite(3, LOW);

digitalWrite(4, LOW);

}

else if (input < 682)

{

digitalWrite(2, HIGH);

digitalWrite(3, HIGH);

digitalWrite(4, LOW);

}

else if (input < 1024)

{

digitalWrite(2, HIGH);

digitalWrite(3, HIGH);

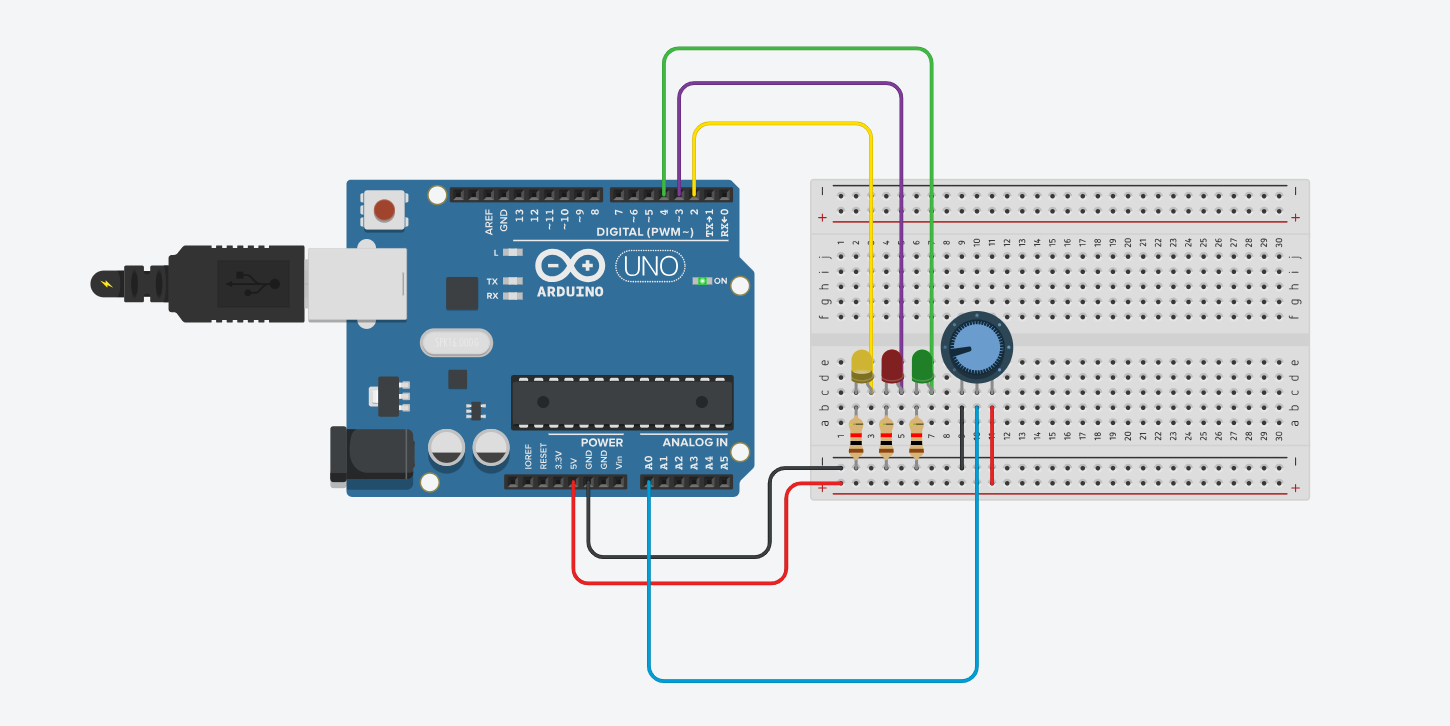
digitalWrite(4, HIGH);

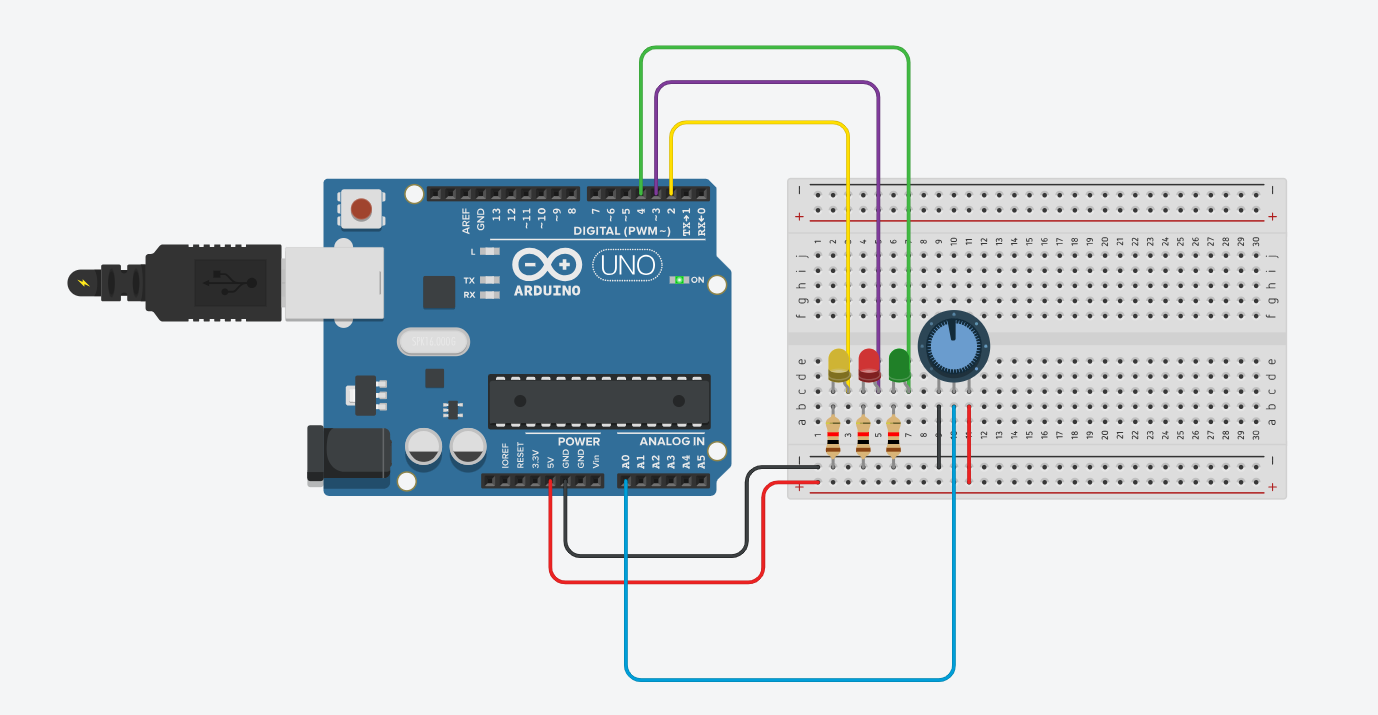
}

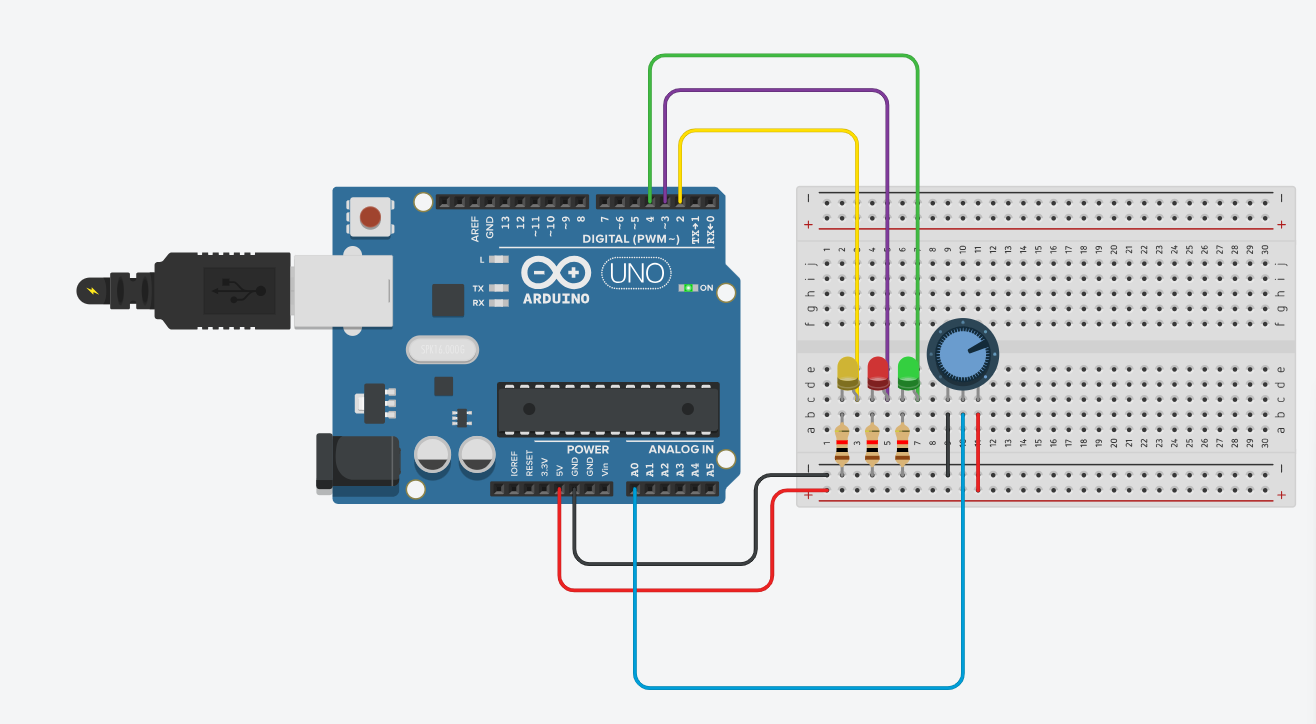
}

// one is on, 2 is on, 3 is on same in reverse

**Output:**







**Conclusion:** Thus, learned how to interface LEDs, and potentiometers with Arduino and write a program to turn on or off more of the LEDs by turning the potentiometer knob.

**Practical No. 11**

**Aim:** To interface LED, Photo resistor (LDR) with Arduino and write a program to increase and decrease the brightness of the LED based on the amount of light present.

**Objectives:**

1. To interface LED, Photo resistor (LDR) with Arduino.
2. Write a Program to increase and decrease the brightness of the LED based on the amount of light present.

**Theory:**

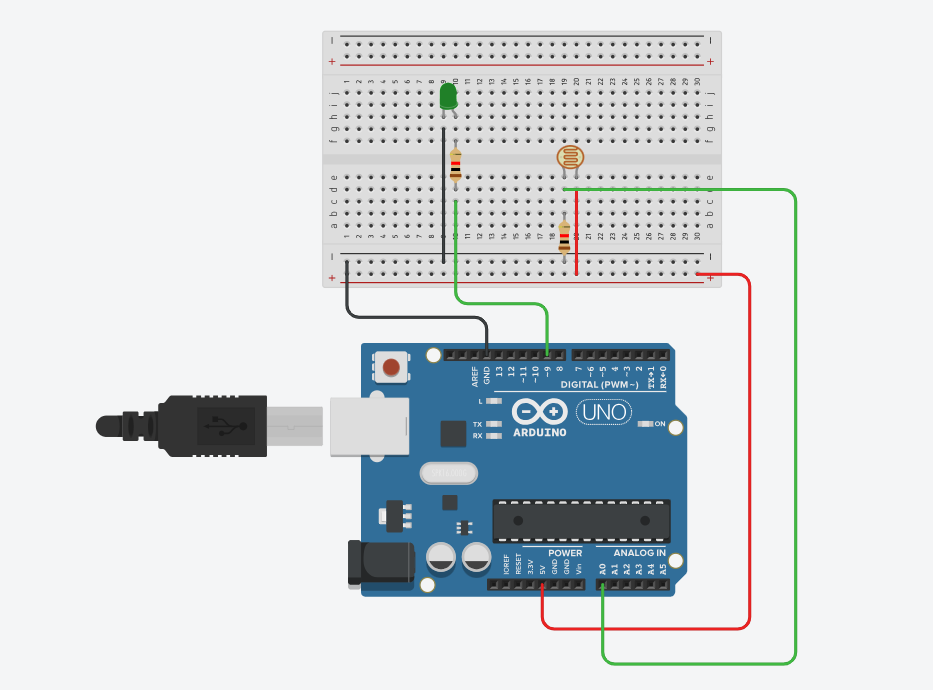
Arduino UNO R3 is an open-source electronics platform based on easy-to-use hardware and software. The setup() function is called when a sketch starts.

A loop() is a programming function that iterates a statement or condition based on specified boundaries.

The PinMode() function is used to configure a specific pin to behave either as an input or an output. The

delay() function allows you to pause the execution of your Arduino program for a specified period. The digitalWrite() function is used to write a HIGH or a LOW value to a digital pin.

**Circuit Diagram:**

****

**Program:**

// C++ code

// Practical 11 - LDR Photoregister

int p=0;

void setup()

{

pinMode(A0, INPUT);

pinMode(9, OUTPUT);

}

void loop()

{

p=analogRead(A0);

if (p<=500)

{

digitalWrite(9, HIGH);

}

else

{

digitalWrite(9, LOW);

}

delay(100);

}

**Output:**





**Conclusion:** Thus, learned how to interface LED, Photo resistor (LDR) with Arduino and write a program to increase and decrease the brightness of the LED based on the amount of light present.

**Practical No. 12**

**Aim:** To interface LEDs with Arduino and write a program to show the fading effect on leds.

**Objectives:**

1. To interface LED with Arduino.
2. Write a Program to show the fading effect on LEDs.

**Theory:**

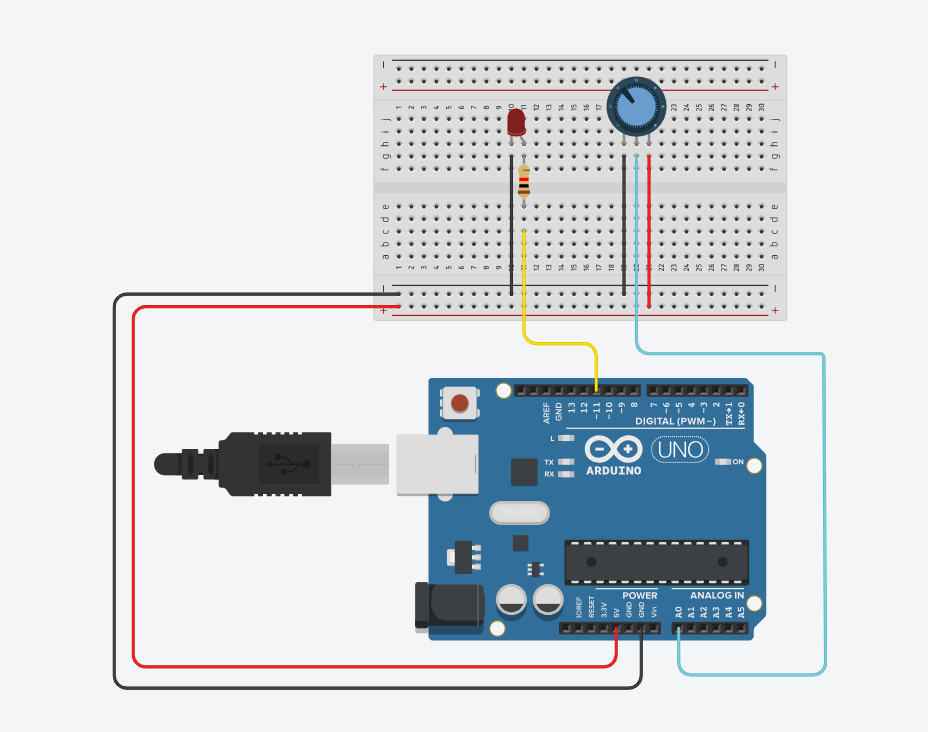
Arduino UNO R3 is an open-source electronics platform based on easy-to-use hardware and software. The setup() function is called when a sketch starts.

A loop() is a programming function that iterates a statement or condition based on specified boundaries.

The PinMode() function is used to configure a specific pin to behave either as an input or an output. The

delay() function allows you to pause the execution of your Arduino program for a specified period. The digitalWrite() function is used to write a HIGH or a LOW value to a digital pin.

**Circuit Diagram:**

****

**Program:**

// C++ code

//

int LED\_PIN = 11;

void setup()

{

Serial.begin(9600);

pinMode(LED\_PIN, OUTPUT);

}

void loop()

{

int analogValue = analogRead(A0);

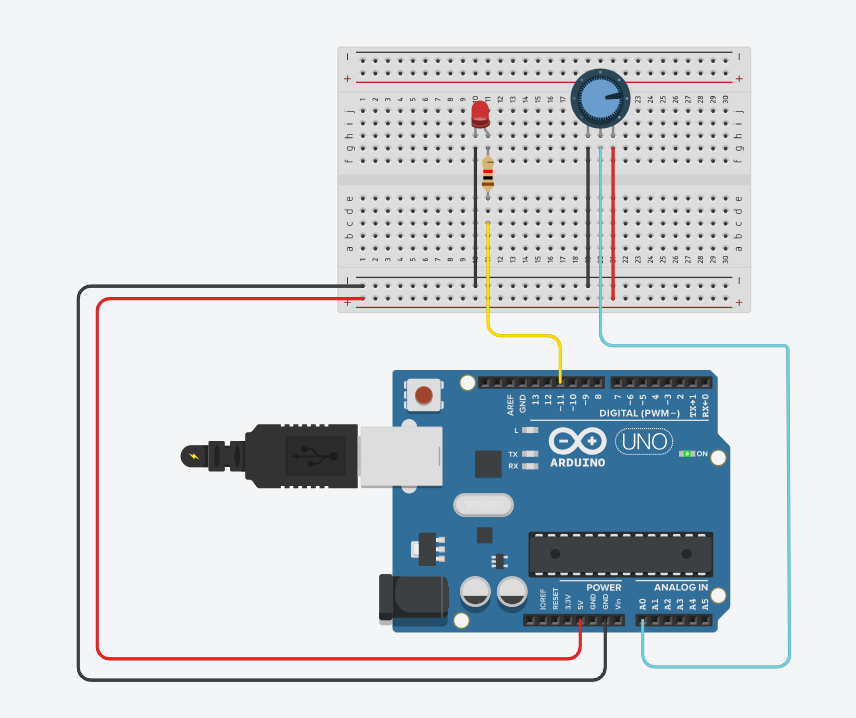
int brightness = map(analogValue, 0, 1023, 0, 255);

analogWrite(LED\_PIN, brightness);

delay(100);

}

**Output:**



**Conclusion:** Thus, learned how to interface LEDs with Arduino and write a program to show the fading effect on LEDs.

**Practical No. 13**

**Aim:** To interface DHT11/temperature sensor with Arduino and write a program to display temperature and humidity data on a serial monitor.

**Objectives:**

1. To interface temperature sensor with Arduino.
2. Write a Program to show the fading effect on LEDs.

**Theory:**

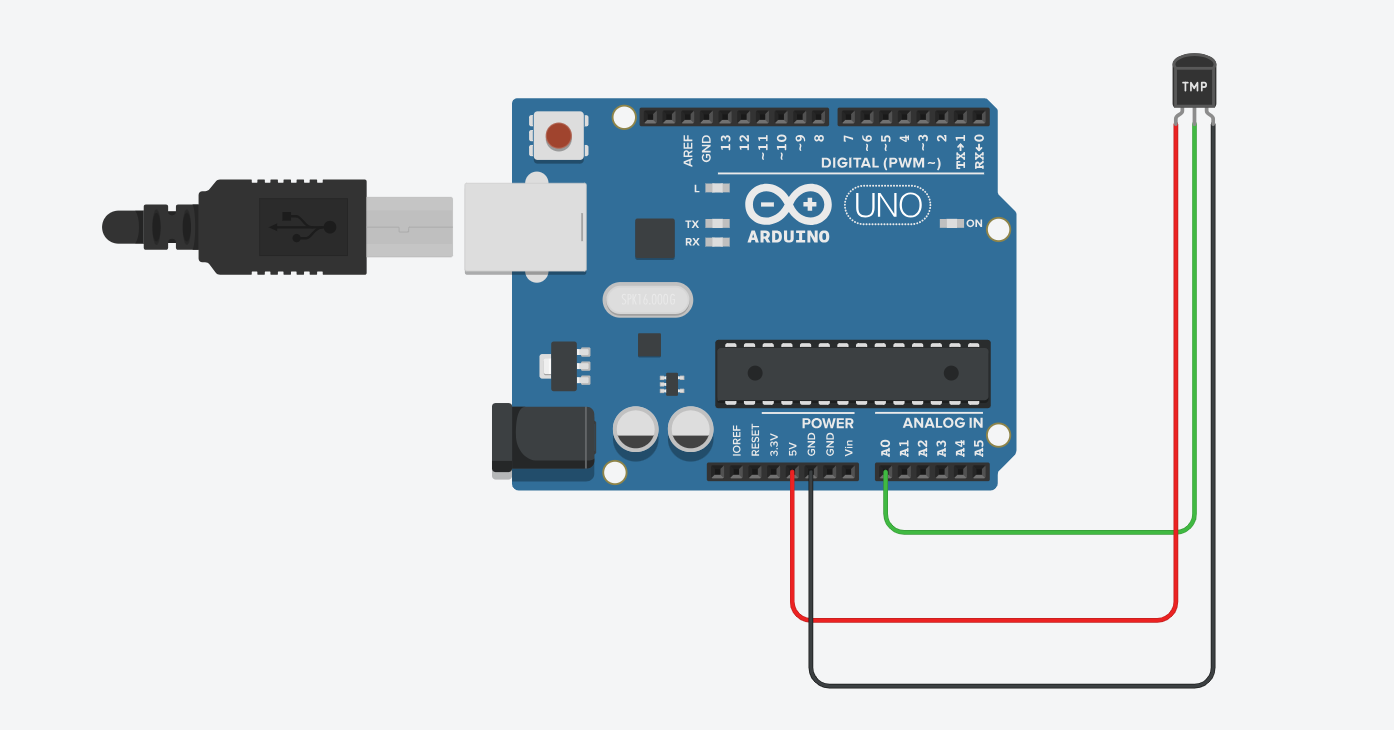
Arduino UNO R3 is an open-source electronics platform based on easy-to-use hardware and software. The setup() function is called when a sketch starts.

A loop() is a programming function that iterates a statement or condition based on specified boundaries.

The PinMode() function is used to configure a specific pin to behave either as an input or an output. The

delay() function allows you to pause the execution of your Arduino program for a specified period. The digitalWrite() function is used to write a HIGH or a LOW value to a digital pin.

**Circuit Diagram:**

****

**Program:**

// C++ code

//

void setup()

{

Serial.begin(9600);

}

void loop()

{

int var = analogRead(A0);

float voltage = var \* (5.0 / 1024);

float temperature = (voltage - 0.5) \* 100;

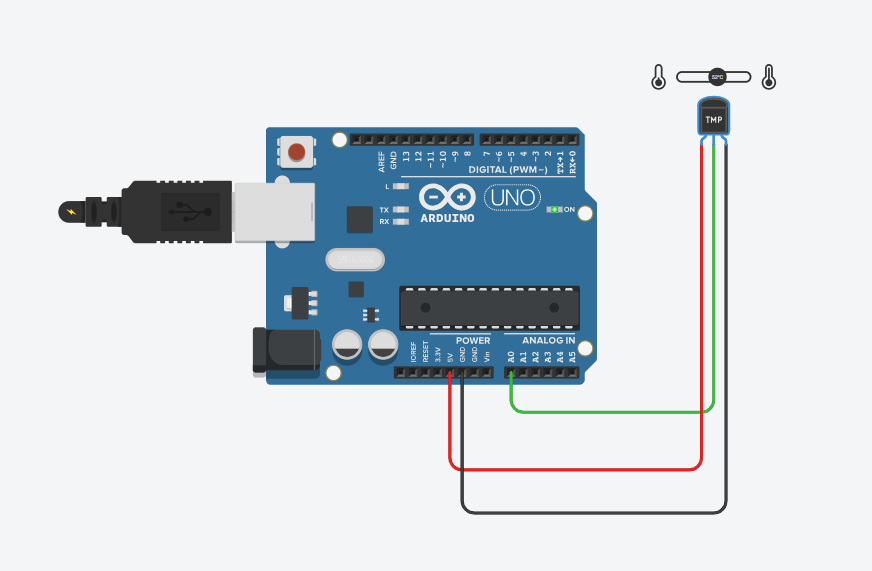
//Serial.println(temperature);

float Fahrienheit = (temperature \* 9.0 / 5.0) + 32.0;

Serial.println(Fahrienheit);

}

**Output:**





**Conclusion:** Thus, learned how to interface DHT11 or temperature sensor with Arduino and write a program to display temperature and humidity data on a serial monitor.

**Practical No. 14**

**[A] PIR Sensor**

**Aim:** To interface PIR/ Ultrasonic sensor with Arduino and write a program to turn on and off LED depending on motion detection/sound detection.

**Objectives:**

1. To interface PIR/ Ultrasonic sensor with Arduino.
2. Write a Program to turn on and off LED depending on motion detection/sound detection.

**Theory:**

Arduino UNO R3 is an open-source electronics platform based on easy-to-use hardware and software. The setup() function is called when a sketch starts.

A loop() is a programming function that iterates a statement or condition based on specified boundaries.

The PinMode() function is used to configure a specific pin to behave either as an input or an output. The

delay() function allows you to pause the execution of your Arduino program for a specified period. The digitalWrite() function is used to write a HIGH or a LOW value to a digital pin.

**Circuit Diagram:**

****

**Program:**

int LED = 3;

int BUZZER = 4;

int PIR = 2;

int PIRState = 0;

void setup()

{

pinMode(LED, OUTPUT);

pinMode(BUZZER, OUTPUT);

pinMode(PIR, INPUT);

}

void loop()

{

PIRState = digitalRead(PIR);

if(PIRState == HIGH){

digitalWrite(LED,HIGH);

digitalWrite(BUZZER,HIGH);

}

else {

digitalWrite(LED,LOW);

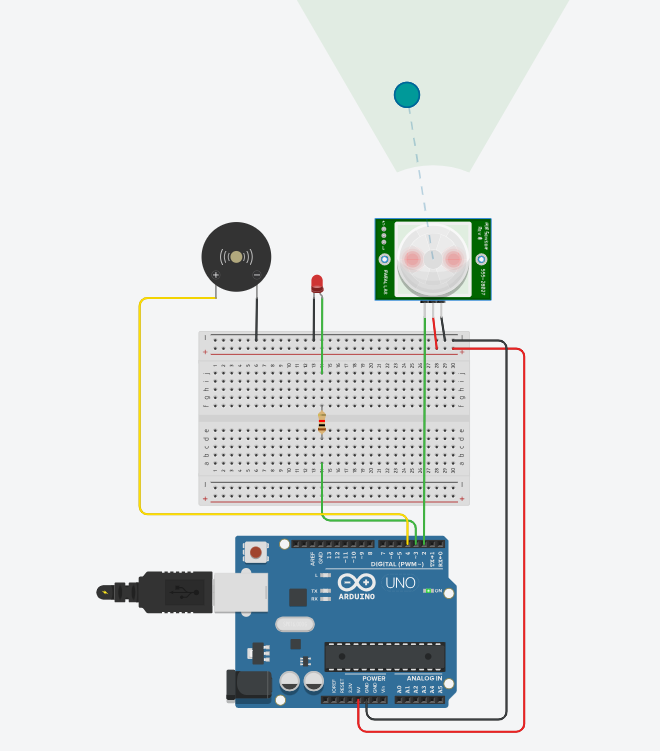
digitalWrite(BUZZER,LOW);

}

delay(10);

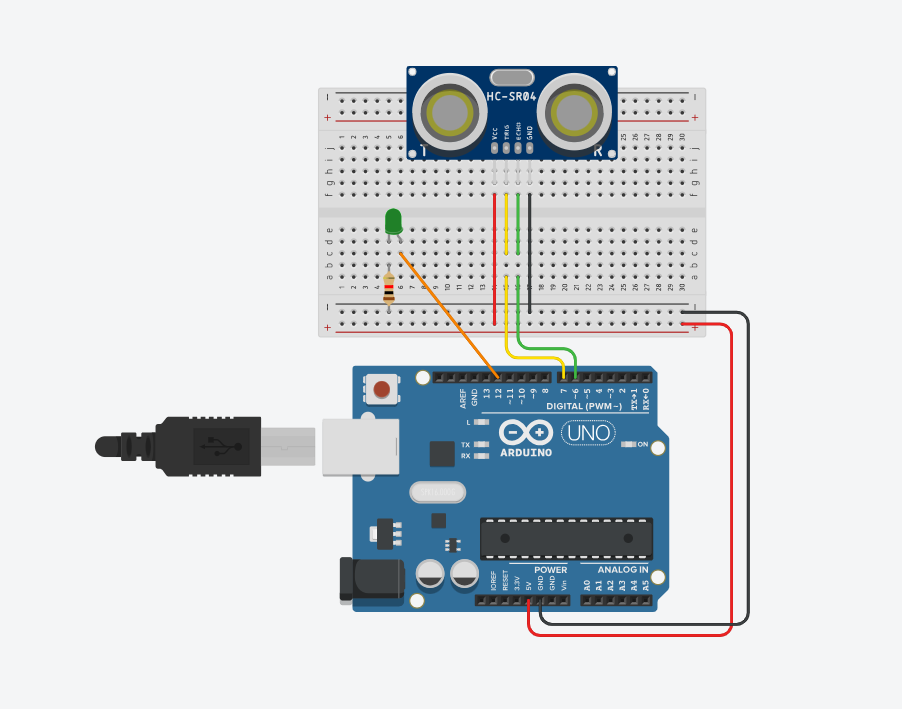
}

**Output:**



**[B] Ultrasonic Sensor**

**Circuit Diagram:**



**Program:**

// C++ code

//

float inches = 0;

float cm = 0;

int led = 12;

const int trigPin = 7;

const int echoPin = 6;

void setup()

{

pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output

pinMode(echoPin, INPUT); // Sets the echoPin as an Input

Serial.begin(9600); // Starts the serial communication

}

void loop()

{

// Clears the trigPin

digitalWrite(trigPin, LOW);

delayMicroseconds(20);

// Sets the trigPin on HIGH state for 10 micro seconds

digitalWrite(trigPin, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin, LOW);

int duration= pulseIn(echoPin, HIGH);

cm = duration\*0.034/2;

inches = (cm / 2.54);

if (cm < 200)

{ digitalWrite(led,HIGH);

}

else {

digitalWrite(led,LOW);

}

// Prints the distance on the Serial Monitor

Serial.print("Duration");

Serial.println(duration);

Serial.print(inches);

Serial.print("in, ");

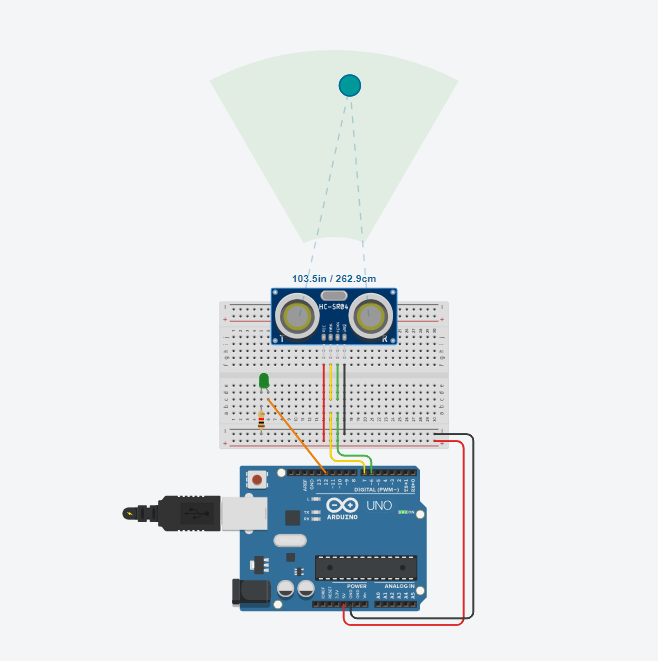
Serial.print(cm);

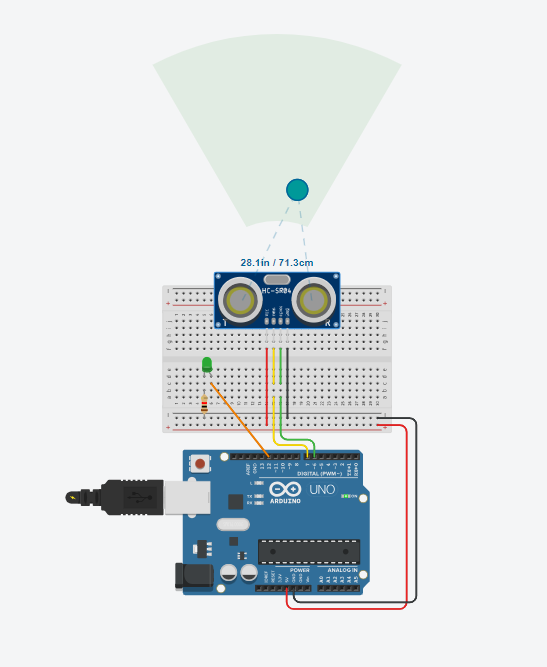
Serial.println("cm");

delay(1000);

}

**Output:**

****

****

**Conclusion:** Thus, learned how to interface PIR/ Ultrasonic sensor with Arduino and write a program to turn on and off LED depending on motion detection/sound detection.

**Practical No. 15**

**Aim:** To interface servo motor with Arduino and write a program to sweep a servo back and forth through its full range of motion.

**Objectives:**

1. To interfaceservo motor with Arduino.
2. Write a Program to s program to sweep a servo back and forth through its full range of motion.

**Theory:**

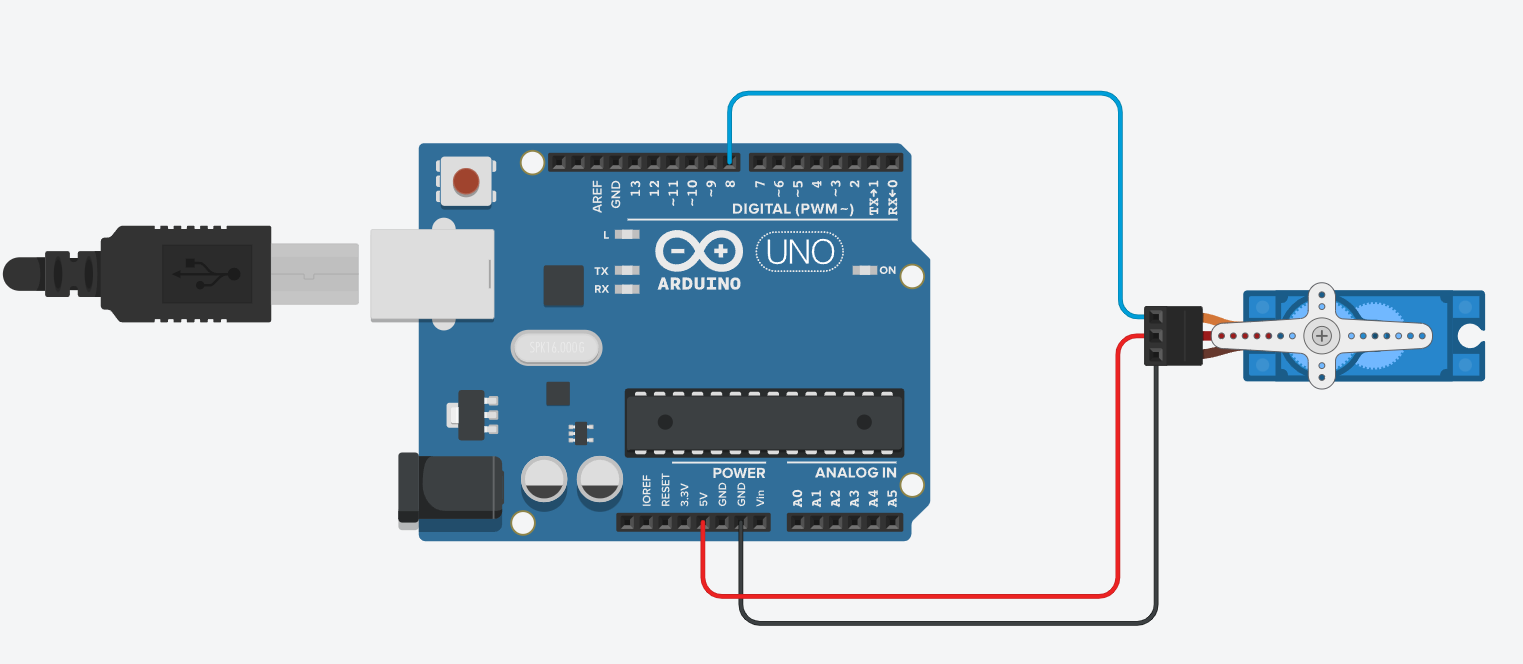
Arduino UNO R3 is an open-source electronics platform based on easy-to-use hardware and software.

A servo motor is a rotary actuator that allows for precise control of angular position. It consists of a motor coupled to a sensor for position feedback. It also requires a servo drive to complete the system. The setup() function is called when a sketch starts.

A loop is a programming function that iterates a statement or condition based on specified boundaries.

The pinMode() function is used to configure a specific pin to behave either as an input or an output.

**Circuit Diagram:**



**Program:**

#include<Servo.h>

Servo myservo;

int angle;

void setup()

{

myservo.attach(8);

}

void loop()

{

for (angle =0; angle<=180; angle++)

{

myservo.write(angle);

delay(40);

}

for (angle =180; angle>=0; angle--)

{

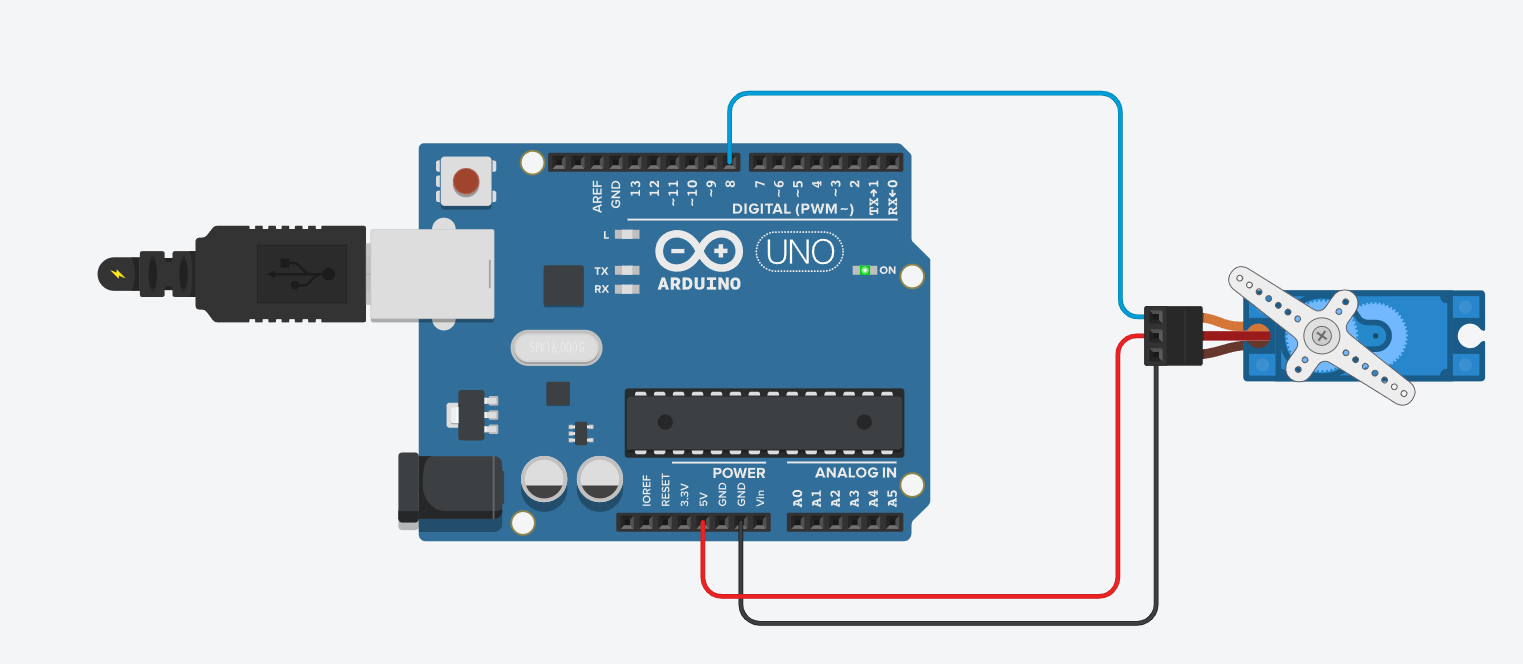
myservo.write(angle);

delay(40);

}

}

**Output:**



**Conclusion:** Thus learned how To interface servo motor with Arduino and write a program to sweep a servo back and forth through its full range of motion.

**Practical No. 16**

**Aim:** To interface LED with Arduino and write a program to send sensor data to the cloud using ThingSpeak/ AWS and receive notification.

**Objectives:**

1. To interface LED with Arduino
2. Write a programto send sensor data to the cloud using ThingSpeak/ AWS and receive notification.

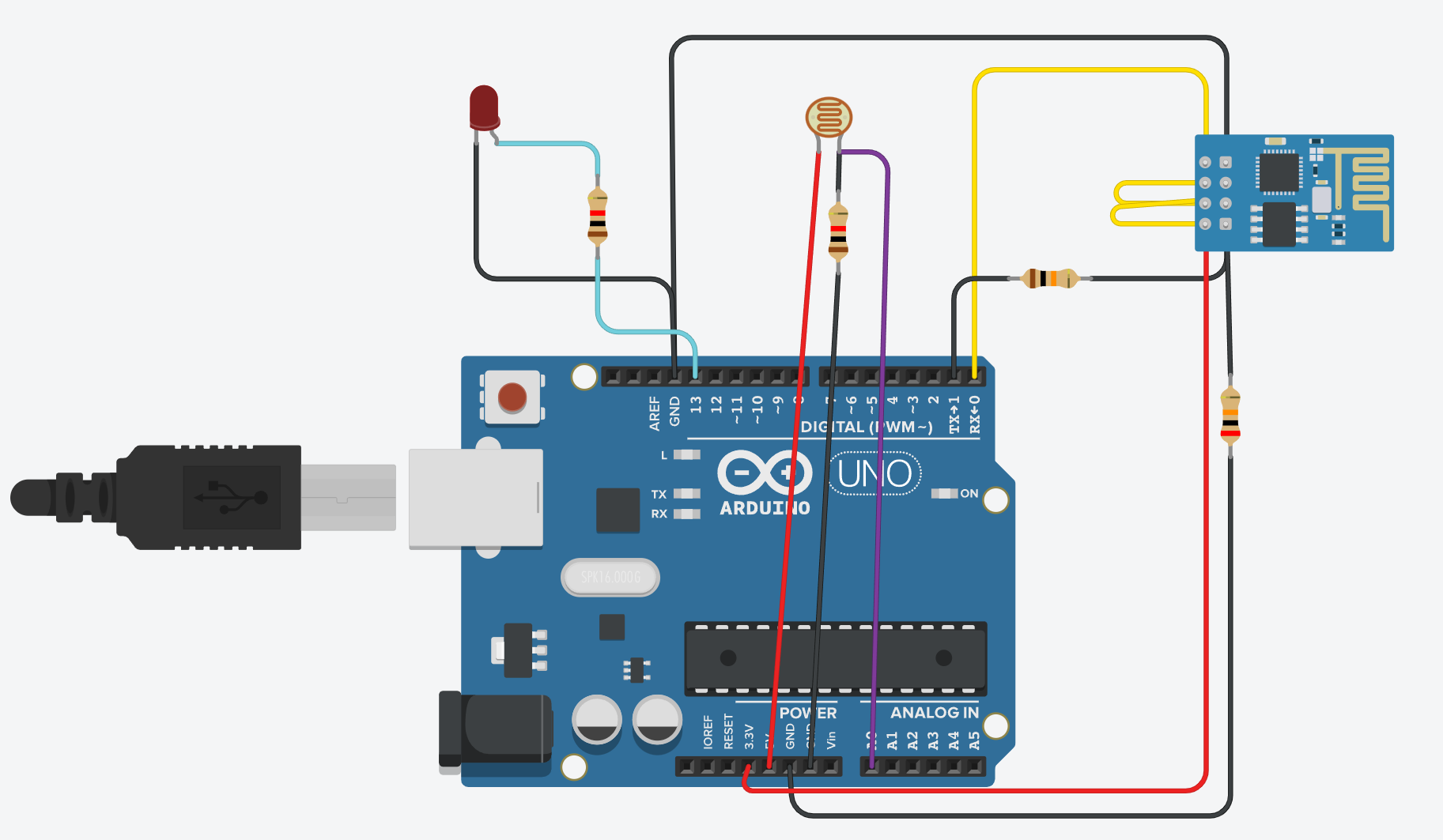
**Theory:**

**ThingSpeak** is an open-source software written in Ruby which allows users to communicate with internet enabled devices. It facilitates data access, retrieval and logging of data by providing an API to both the devices and social network websites.

**Arduino UNO R3** is an open-source electronics platform based on easy-to-use hardware and software.

A **Resistor** is a passive components, which means that they don't generate any electricity at all, but rather reduce voltage and current by dissipating power in the form of heat. The unit of resistance is ohms (Ω).

**Circuit Diagram:**



**Program:**

int ldr=A0;

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);

}

Serial.println("AT+CIPSTART=\"TCP\",\"thingspeak.com\",80");

delay(5000);

int len = 65;

Serial.print("AT+CIPSEND=");

Serial.println(len);

delay(10);

Serial.print("GET /update?api\_key=9040MF08OW6Q5DMX&field1="+ String(value) +" HTTP/1.1\r\n");

delay(100);

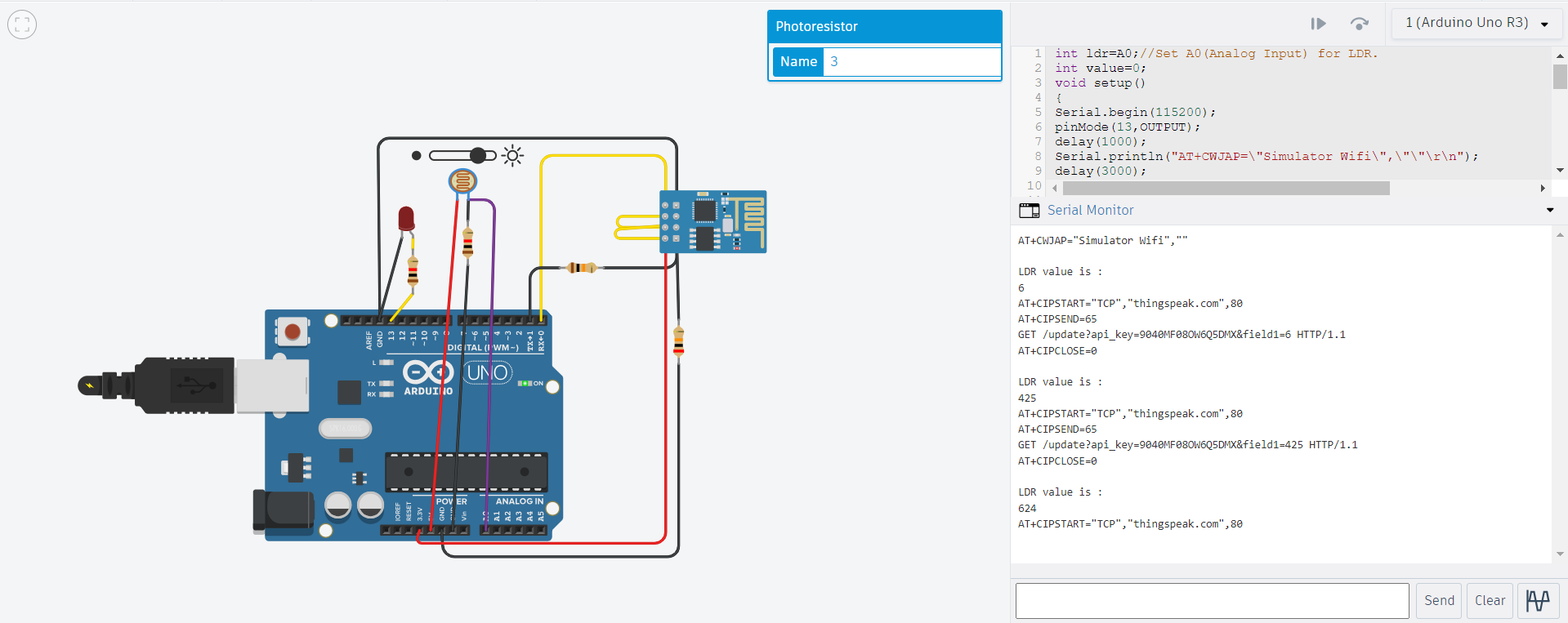
Serial.println("AT+CIPCLOSE=0\r\n");

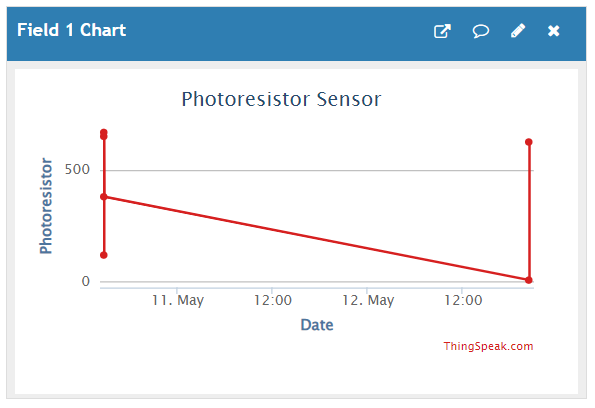
delay(6000);

}

}

**Output:**





**Conclusion:** In this, temperature sensor is used to sense the data and we can send this data to cloud using ThinkSpeak/AWS and receive noitification.

**Practical No. 17**

**Aim:** To interface Temperature sensor with Arduino and write a program to send sensor data to the cloud using ThingSpeak/ AWS and receive notification.

**Objectives:**

1. To interface Temperature sensor with Arduino
2. Write a program to send to send sensor data to the cloud using ThingSpeak/ AWS and receive notification.

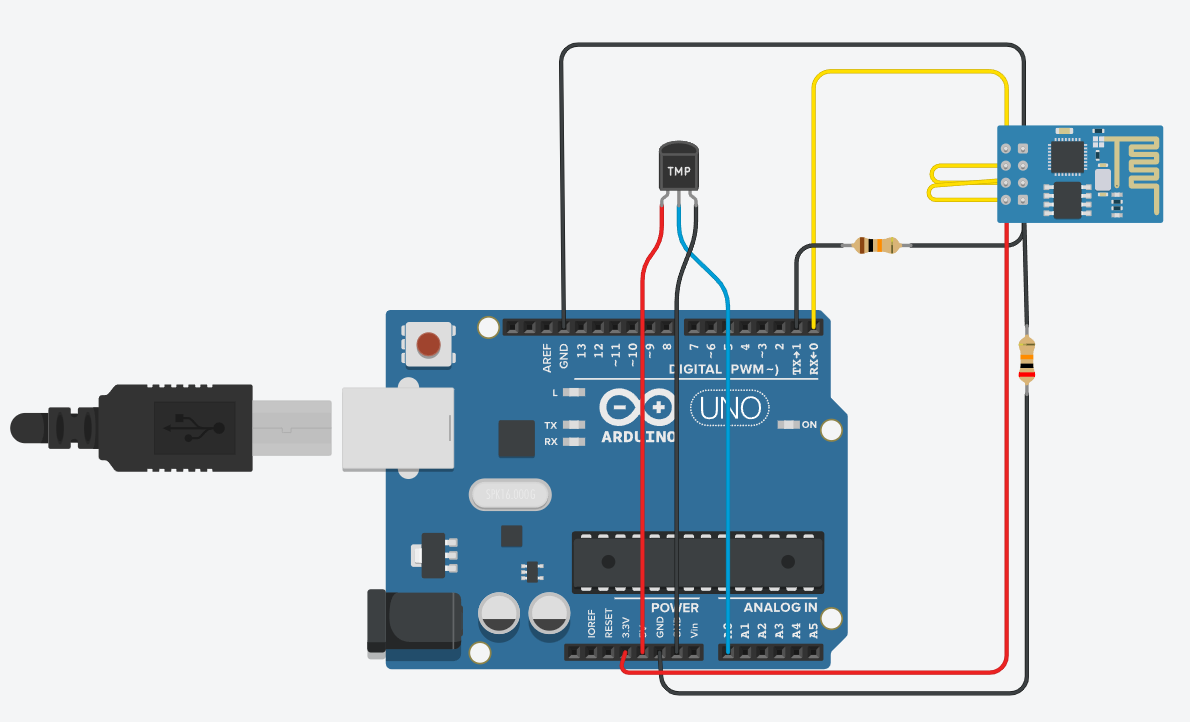
**Theory:**

ThingSpeak is an open-source software written in Ruby which allows users to communicate with internet enabled devices. It facilitates data access, retrieval and logging of data by providing an API to both the devices and social network websites.

Arduino UNO R3 is an open-source electronics platform based on easy-to-use hardware and software.

A Resistor is a passive components, which means that they don't generate any electricity at all, but rather reduce voltage and current by dissipating power in the form of heat. The unit of resistance is ohms (Ω).

**Circuit Diagram:**



**Program:**

void setup() {

Serial.begin(115200);

delay(1000);

Serial.println("AT+CWJAP=\"Simulator Wifi\",\"\"\r\n");

delay(3000);

}

void loop() {

{

int sensorValue = analogRead(A0);

float volt = (sensorValue/1020.0) \* 4.9;

float tempC = (volt -0.5) \* 100;

Serial.println(tempC);

Serial.println("AT+CIPSTART=\"TCP\",\"api.thingspeak.com\",80\r\n");

delay(5000);

int len = 65;

Serial.print("AT+CIPSEND=");

Serial.println(len);

delay(10);

Serial.print("GET /update?api\_key=8GH3DZKJUWRS363&field1=" + String(tempC)+" HTTP/1.1\r\n");

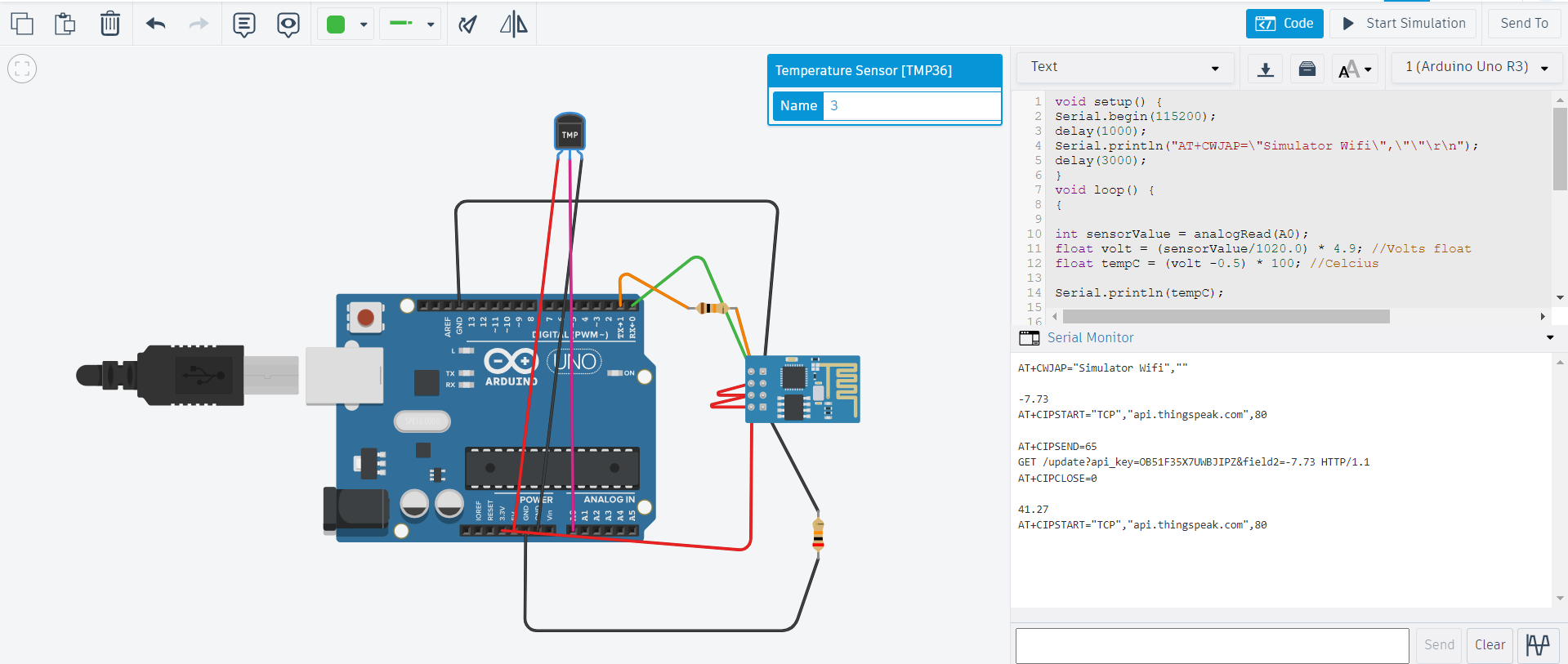
Serial.println("AT+CIPCLOSE=0\r\n");

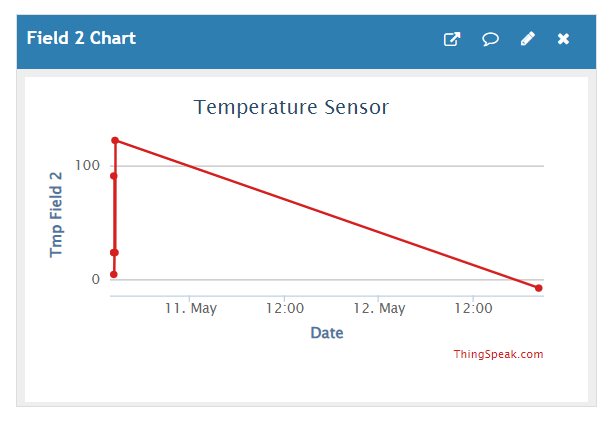
delay(6000);

}

}

**Output:**





**Conclusion:** Thus, learnt about basic components of IoT like Arduino UNO, Breadboard, resisters, LED’s and interfacing LED with Arduino.

**Practical No. 18 (Project)**

**TITLE: Arduino-based LED Chaser Controlled by Sensors**

**Aim:** To create an Arduino-based system that controls a series of LEDs in a chasing pattern, with the speed of the chase determined by a potentiometer and influenced by a PIR sensor.

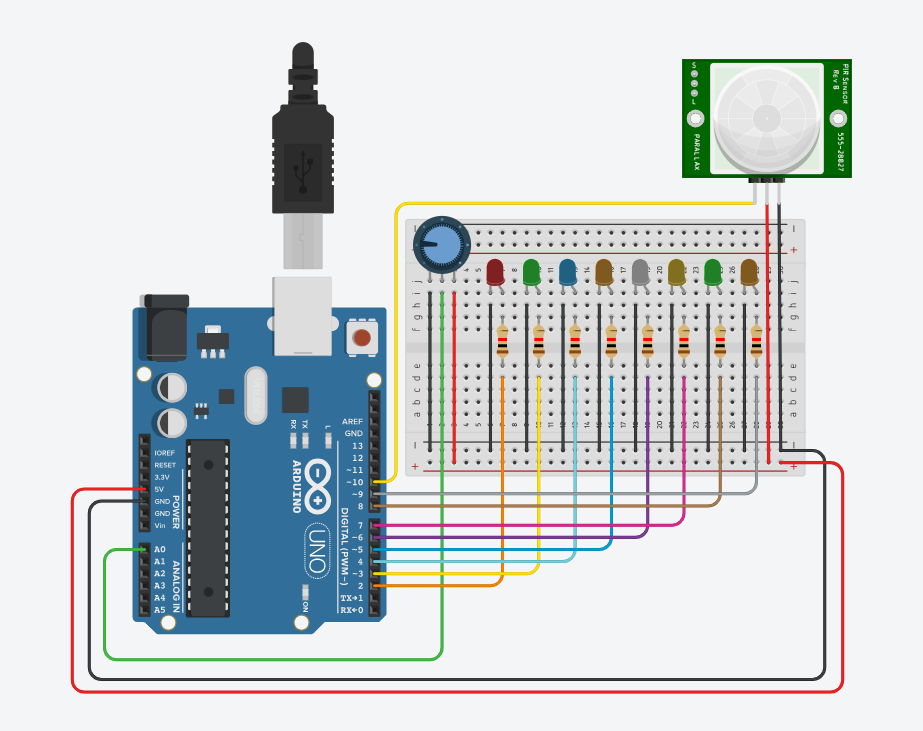
**Objectives:**

The project aims to create an Arduino-driven LED chaser system, utilizing a potentiometer for speed adjustment and a PIR sensor for motion-triggered behavior. Monitoring potentiometer values via the serial monitor enhances debugging and calibration. Objectives include designing dynamic lighting, integrating sensors for interactivity, and optimizing usability through monitoring features. The project demonstrates Arduino's versatility in creating responsive IoT solutions with sensor integration.

**Theory:**

In this project, LEDs are arranged in a chaser pattern, sequentially lighting up. The speed of this sequence is controlled by a potentiometer, providing analog input to the Arduino. The presence of a PIR sensor ensures that the LED chaser behavior is dynamically influenced by detected motion. This project showcases the integration of sensors to control output devices, resulting in an interactive and responsive lighting system.

**Circuit Diagram:**

****

**Program:**

// PROJECT - LEDs CHASER

const int numLEDs = 8;

int ledPins[numLEDs] = {2, 3, 4, 5, 6, 7, 8, 9};

int potentioPin = A0;

int PIRPin = 10;

int delay\_period = 0;

int input\_val;

int PIRState = 0;

void setup() {

Serial.begin(9600);

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

pinMode(ledPins[i], OUTPUT);

}

pinMode(potentioPin, INPUT);

pinMode(PIRPin, INPUT);

}

void loop() {

input\_val = analogRead(potentioPin);

Serial.print("Potentiometer Value: ");

Serial.println(input\_val);

delay\_period = map(input\_val, 0, 1023, 0, 1000);

PIRState = digitalRead(PIRPin);

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

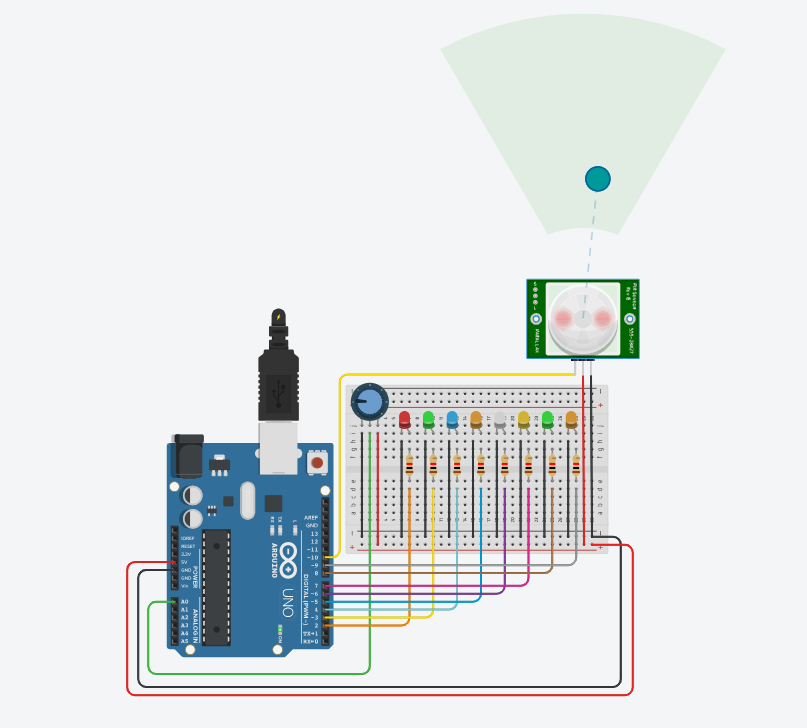
digitalWrite(ledPins[i], (PIRState == HIGH) ? HIGH : LOW);

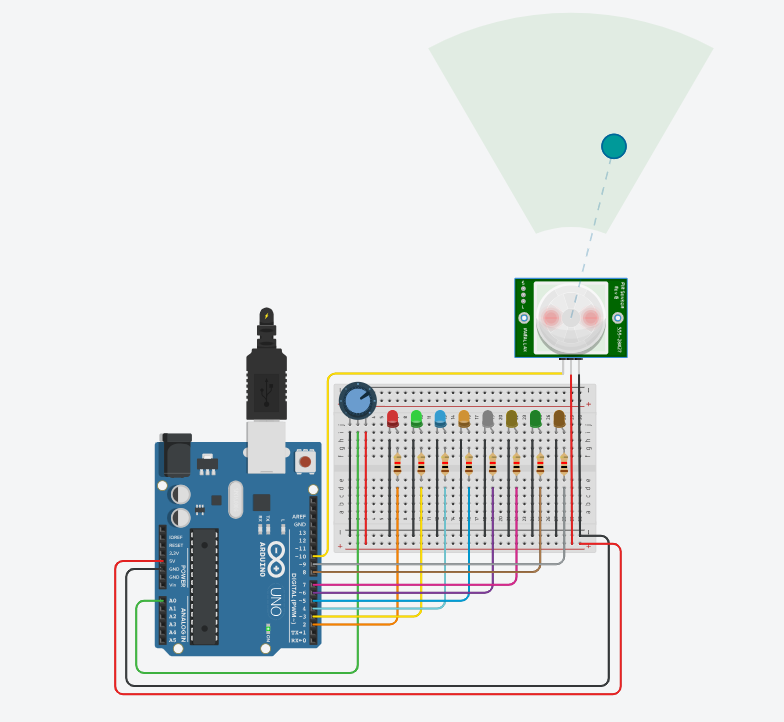
delay(delay\_period);

}

}

**Output:**





**Conclusion:** This project successfully demonstrates the implementation of an Arduino-based LED chaser system controlled by a potentiometer and influenced by a PIR sensor. By integrating a PIR sensor, the LED chaser behavior responds dynamically to detected motion, enhancing its interactivity and usability in various applications. Additionally, the monitoring of potentiometer values via the serial monitor facilitates debugging and calibration, contributing to the project's functionality and versatility.