**MCA SEM – II**

**IOT PRACTICAL PROBLEMS**

|  |  |
| --- | --- |
| **Practical No** | **Problem Statement** |
| 1. | Program to blink Arduino onboard LED and To interface external LED with Arduino and write a program to turn ON LED for 1 sec after every 2 seconds. |
| 2. | Tointerface5 LED’swithArduinoandwritea programto blink 6 LEDs, oneat a time, in a back-and-forth formation. |
| 3. | To interface the Push button with Arduino and write a program to turn the ON LED when the push button is pressed. |
| 4. | To interface Push button, Speaker/buzzer with Arduino and write a program to turn ON the LED and generate a note or tone when push button is pressed. |
| 5. | To interface 2 Push buttons, a Speaker with Arduino and write a program to turn  ON LED and generate a 2 different notes on two button keyboard. |
| 6. | To interface Seven Segment Display (SSD) with Arduino and write a program to blink SSD. |
| 7. | To interface Seven Segment Display (SSD) with Arduino and write a program to print numbers from 1 to 4 on SSD. |
| 8. | To interface LCD, push button, a potentiometer with Arduino and write a program to display a message on LCD whether n push button is pressed. |
| 9. | 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 LCD. |
| 10. | To interface LEDs, potentiometer with Arduino and write a program to turn on or off more of the LEDs by turning the potentiometer knob. |
| 11. | 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. |
| 12. | To interface LEDs with Arduino and write a program to show the fading effect on leds. |
| 13. | To interface the DHT11 sensor with Arduino and write a program to display temperature and humidity data on a serial monitor. |
| 14. | To interface PIR/ Ultrasonic sensor with Arduino and write a program to turn on and off LED depending on motion detection/sound detection. |
| 15. | To interface servo motor/DC motor with Arduino and write a program to sweep a servo back and forth through its full range of motion/ to control a DC motor. |
| 16. | To interface LED with Arduino and write a program to send sensor data to the  cloud using ThingSpeak/ AWS and receive notifications.to |
| 17. | To interface the Temperature sensor with Arduino and write a program to send sensor data to the cloud using ThingSpeak/ AWS and receive notification. |
| 18. | To build a mini project based on interfacing any combination of sensors with Raspberry pi and Internet. |

**Practical No. 1**

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

# Objectives:

* To learn Arduino UNO basics
* Program to blink Arduino onboard LED and To interface external LED with Arduino and write a program to turn ON LED for 1 sec after every 2 seconds.

# Theory:

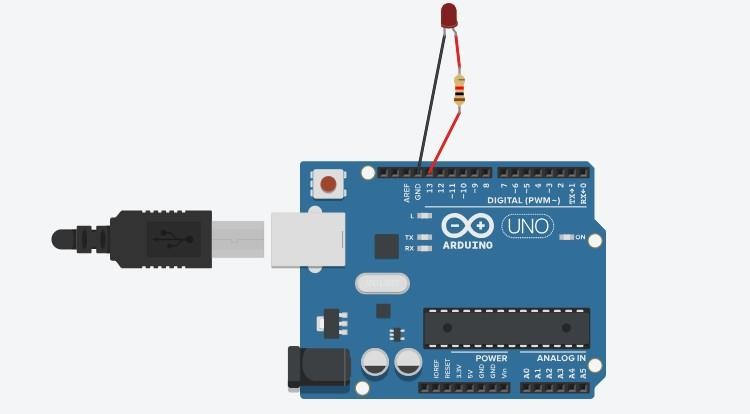
**Arduino UNO R3** is an open-source electronics platform based on easy-to-use hardware and software. The **setup()** function is called when a sketch starts.

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);

}

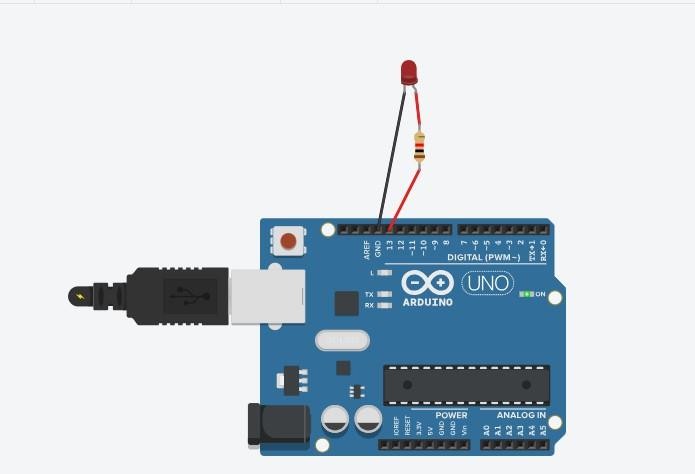
void loop()

{

digitalWrite(LED\_BUILTIN, HIGH); delay(1000); // Wait for 1000 millisecond(s) digitalWrite(LED\_BUILTIN, LOW); delay(1000); // Wait for 1000 millisecond(s)

}

# Output:



**Conclusion:** Thus, learned about basic components of IoT like Arduino UNO, Breadboard, resisters, LED, 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:

⦁ To interface 6 LEDs with Arduino

⦁ Write a Program to blink 6 LEDs, one at a time, in a back-and-forth formation.

# Theory:

We have used one Arduino, breadboard, 6 resistors of resistance 220 ohms, and 6 LEDs of different colors.

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

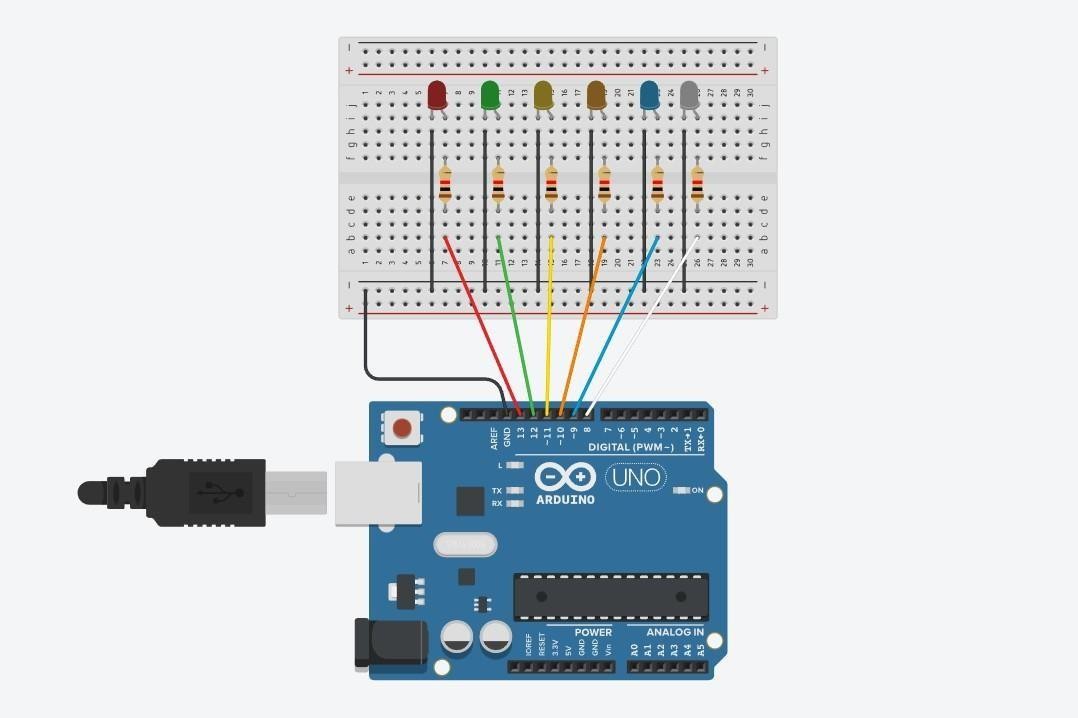
A **loop()** is a programming function that iterates a statement or condition based on specified boundaries.

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:



**Code:**

int timer = 1000; void setup()

{

for (int thisPin =8; thisPin<= 13; thisPin++){ pinMode(thisPin, OUTPUT);

}

}

void pattern1(){ digitalWrite(13, HIGH); delay(timer); digitalWrite(12, HIGH); delay(timer); digitalWrite(11, HIGH); delay(timer); digitalWrite(10, HIGH); delay(timer); digitalWrite(9, HIGH); delay(timer); digitalWrite(8, HIGH); delay(timer); digitalWrite(13, LOW); delay(timer); digitalWrite(12, LOW); delay(timer); digitalWrite(11, LOW); delay(timer); digitalWrite(10, LOW); delay(timer); digitalWrite(9, LOW); delay(timer); digitalWrite(8, LOW);

}

void pattern2(){ digitalWrite(13, HIGH); digitalWrite(11, HIGH); digitalWrite(9, HIGH); delay(timer); digitalWrite(13, LOW); digitalWrite(11, LOW); digitalWrite(9, LOW); delay(timer); digitalWrite(12, HIGH); digitalWrite(10, HIGH); digitalWrite(8, HIGH); delay(timer); digitalWrite(12, LOW); digitalWrite(10, LOW); digitalWrite(8, LOW);

delay(timer);

}

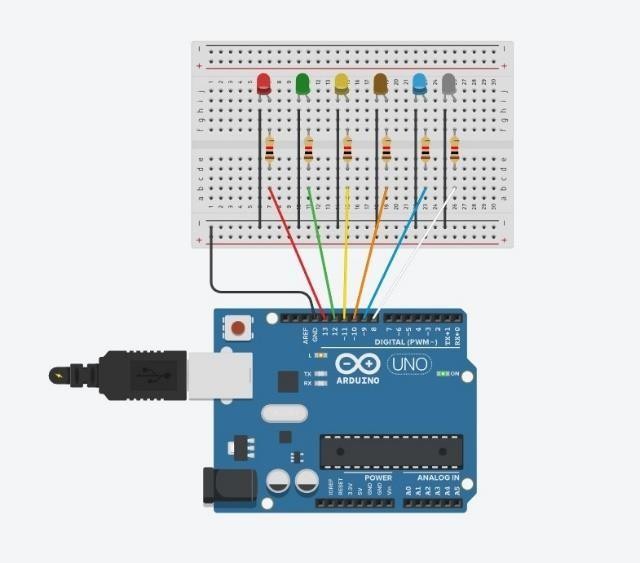
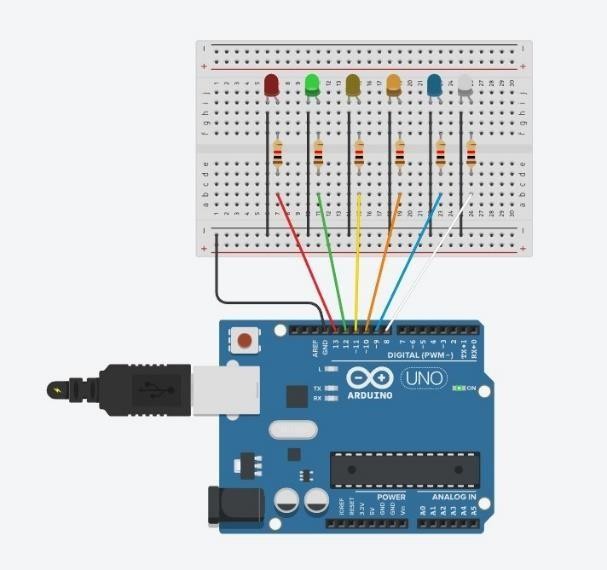
void loop()

{

pattern2();

}

# Output:



**Conclusion:** Thus, learned about basic components of IoT like Arduino UNO, Breadboard, resistors, LE,D and interfacing 6 LELEDsith Arduino with bacand-for howow formation.

# Practical No. 3

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

# Objectives:

⦁ To interface the Push button with Arduino

⦁ Write a Program to turn ON the LED wthe hen push button is pressed.

# Theory:

We have used one Ardunio, breadboard, 2 resistors of resistance 220 oohmsand 10 ohms respectively,and 1 Led.

**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).

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

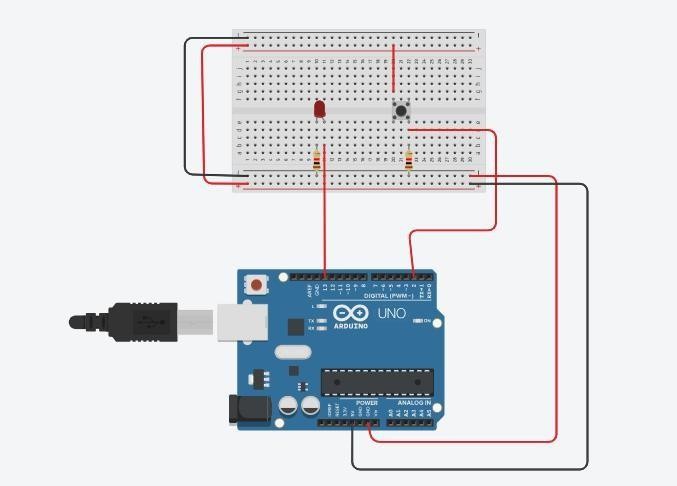
A loop() is a programming function that iterates a statement or condition based on specified boundaries.

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:



**Code :**

void setup()

{

pinMode(7, OUTPUT); pinMode(3, INPUT);

}

void loop()

{

int bs=digitalRead(3); if( bs==HIGH)

{

digitalWrite(7, HIGH);

}

else

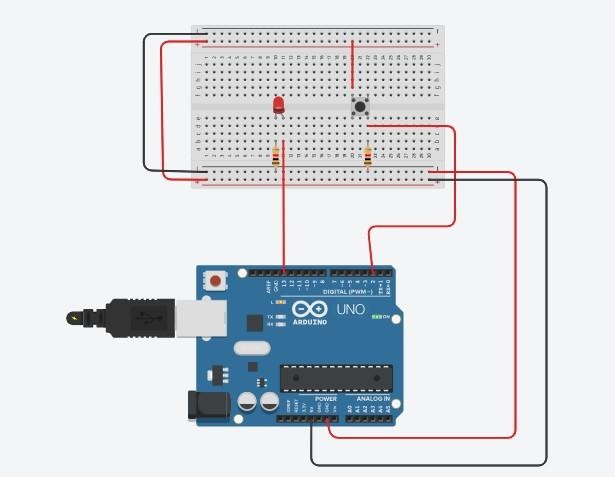
{

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:

⦁ To interface the Push button, Speaker/buzzer with Arduino.

Write a Program to turn ON the LED and generate a note or tone when the push button is pressed.

# Theory:

We have used one Arduino, breadboard, 2 resistors of resistance 220 ohms and 10 ohms respectively, 1 Buzzer, and 1 LED.

**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).

A **Piezo buzzer** is basically a tiny speaker that you can connect directly to an Arduino. The setup() function is called when a sketch starts.

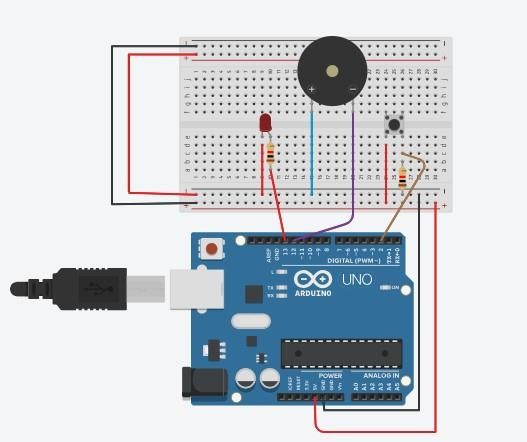
A **loop** is a programming function that iterates a statement or condition based on specified boundaries.

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:



**Code :**

int piezopin = 8; void setup(){

pinMode(piezopin,OUTPUT); pinMode(7, OUTPUT); pinMode(3, INPUT);

}

void loop()

{

int bs=digitalRead(3); if(bs==HIGH){ digitalWrite(piezopin,HIGH);

}

if(bs==HIGH){ digitalWrite(7,HIGH);

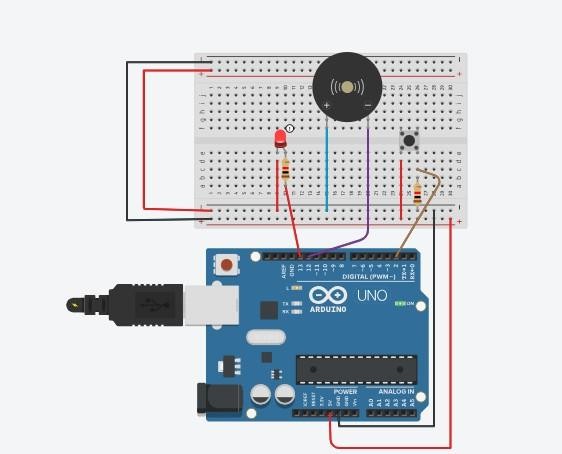
}

else{ digitalWrite(piezopin,LOW); digitalWrite(7,LOW);

}

}

# Output:



**Conclusion:** Thus, learned about basic components of IoT like Arduino UNO, Breadboard, resistors, Piezo buzzer, LED, push button, and controlling LED glow and Piezo buzzer using a push button at the same time.

# 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:

⦁ To interface 2 Push buttons, a Speaker with Arduino

⦁ Write a Program To to turn ON LED and generate 2 different notes on two button keyboards.

# Theory:

We have used one Ardunio , breadboard, 2 resistors of resistance 220 ohm and 10 ohm respectively, 1 Buzzer and 1 LED.

**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).

A **Piezo buzzer** is basically a tiny speaker that you can connect directly to an Arduino. 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:

**Code :**

#define NOTE\_D8 4699

#define NOTE\_DS8 4978 const int buttonpin1=7; const int buttonpin2=4;

const int ledpin = 8; const int speakerpin=13;

int buttonstate1; int buttonstate2;

void setup()

{

pinMode(ledpin,OUTPUT); pinMode(buttonpin1,INPUT); pinMode(buttonpin2,INPUT); pinMode(ledpin,OUTPUT);

}

void loop()

{

buttonstate1 = digitalRead(buttonpin1); buttonstate2 = digitalRead(buttonpin2);

if(buttonstate1==HIGH)

{

digitalWrite(ledpin,HIGH); tone(speakerpin,NOTE\_D8,100);

}

else if(buttonstate2==HIGH)

{

digitalWrite(ledpin,HIGH); tone(speakerpin,NOTE\_DS8,100);

}

else

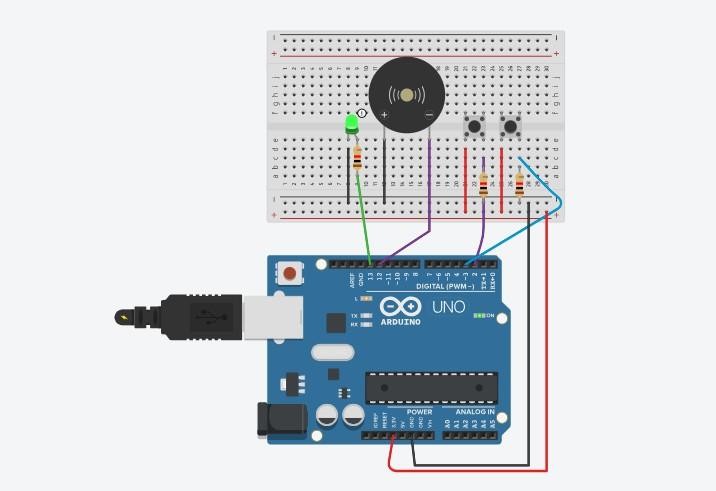
{

noTone(speakerpin); digitalWrite(ledpin,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:

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

# 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.

**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.

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

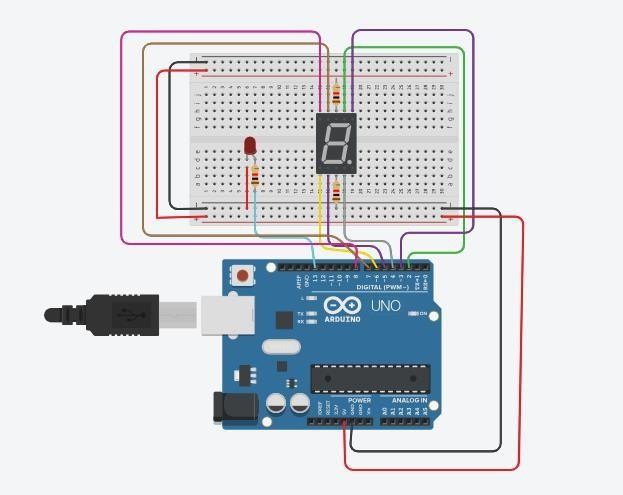
A **loop** is a programming function that iterates a statement or condition based on specified boundaries.

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:



**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()

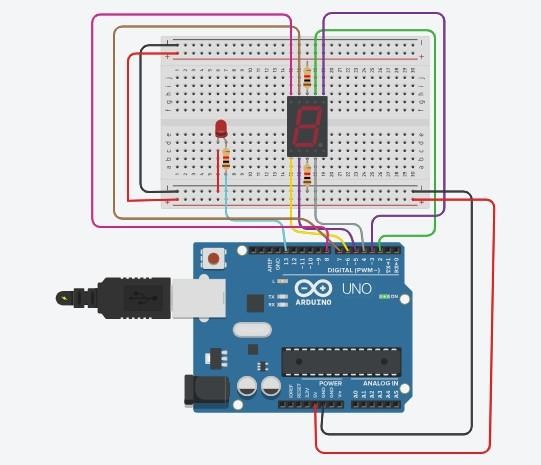
{

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:

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

# 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.

**SSD (Seven Segment Display)** is 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.

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

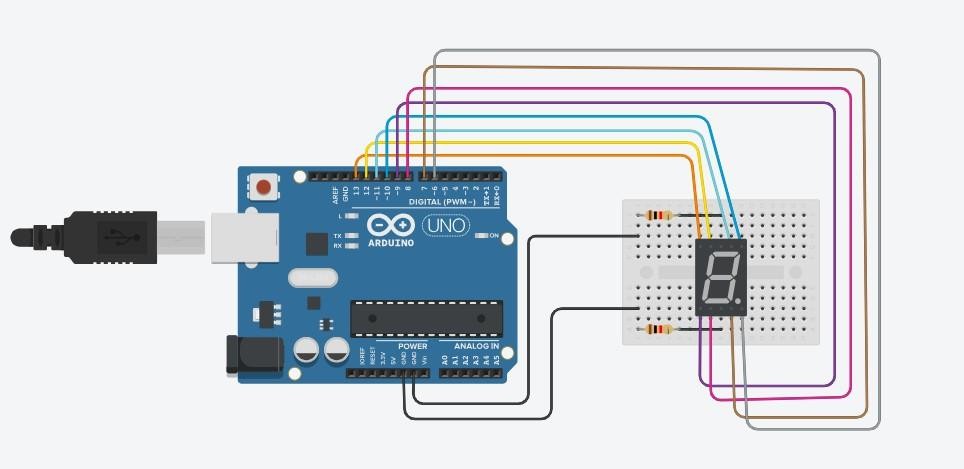
A **loop** is a programming function that iterates a statement or condition based on specified boundaries

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:



**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

{ 1,1,1,0,0,1,1 }}; // 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)

{

delay(1000); Num\_Write(counter);

}

delay(1000);

}

// this functions writes values to the sev seg pins void Num\_Write(int number)

{

int pin= 2;

for (int j=0; j < 7; j++) { digitalWrite(pin, num\_array[number][j]); pin++;

}

}

/\* zero();

one();

two();

three();

four();

five();

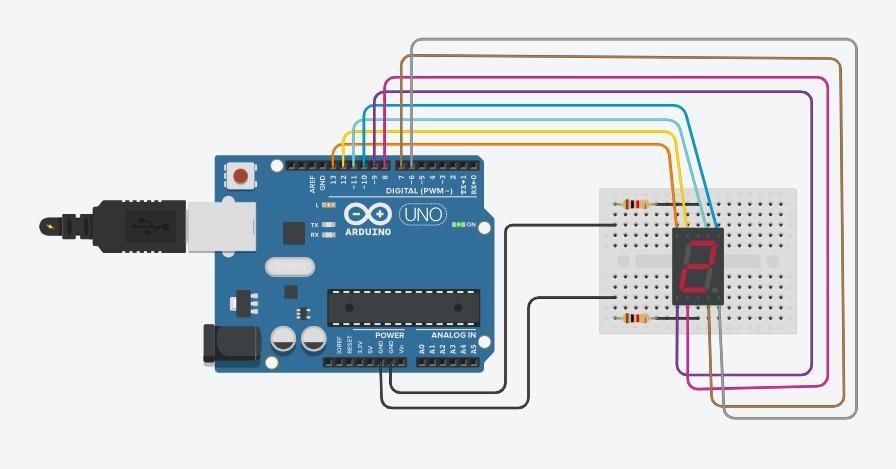
six();

seven();

eight();

nine();\*/

# 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:

* To interface LCD, push button, and potentiometer with Arduino
* 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.](https://www.javatpoint.com/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.

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

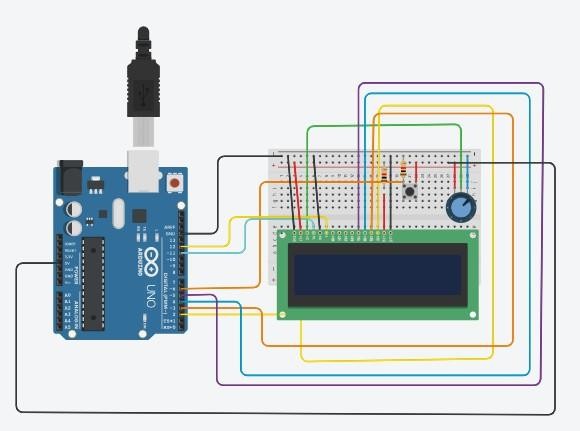
A **loop** is a programming function that iterates a statement or condition based on specified boundaries

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:



**Code:** #include<LiquidCrystal.h> LiquidCrystal lcd(12,11,5,4,3,2); int e=7;

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);

while(digitalRead(e)==HIGH);

}

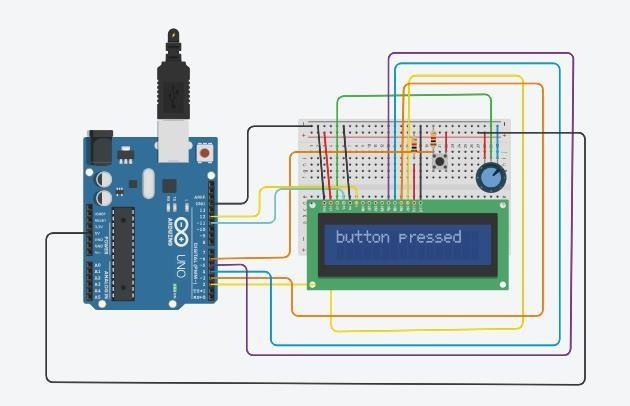
else{ lcd.setCursor(1,0);

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:

* To interface LCD, push button, potentiometer with Arduino
* 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.](https://www.javatpoint.com/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.

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

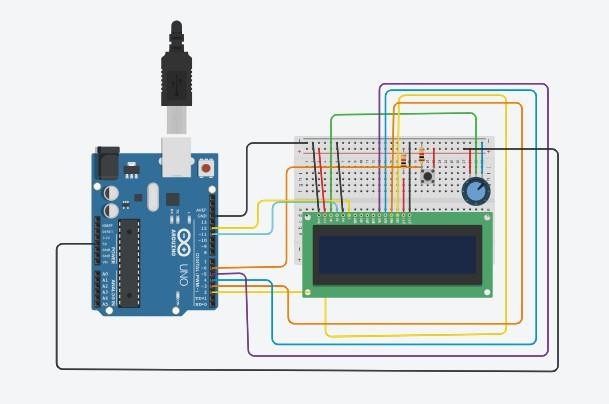
A **loop** is a programming function that iterates a statement or condition based on specified boundaries

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:



**Code:**

// C++ code

// #include<LiquidCrystal.h>

LiquidCrystal lcd(12,11,5,4,3,2); int e=7;

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:

* To interface LEDs, and potentiometer with Arduino
* 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.

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:

**Code:**

int input=0;

int pointpin=A0; void setup()

{

pinMode (2, OUTPUT);

pinMode (3, OUTPUT);

pinMode (4, OUTPUT); pinMode (A0,INPUT);

}

void loop()

{

input = analogRead(pointpin); if(input < 341)

{

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)

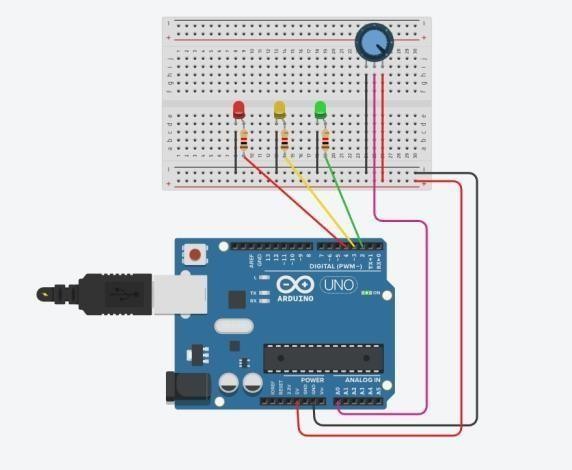
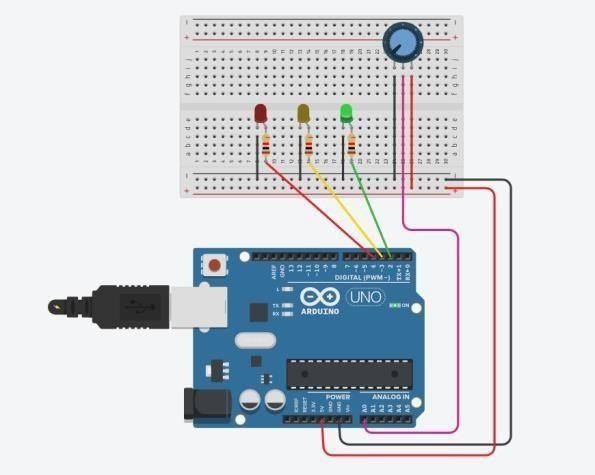
{

digitalWrite(2, HIGH); digitalWrite(3, HIGH); digitalWrite(4, HIGH);

}

}

# 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:

* To interface LED, Photo resistor (LDR) with Arduino.
* 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.

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.

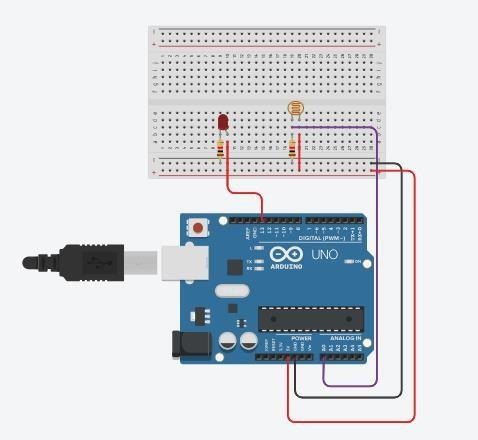
A **loop** is a programming function that iterates a statement or condition based on specified boundaries

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

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:



**Code:**

int p= 0; void setup()

{

pinMode(A0, INPUT); Serial.begin(9600); pinMode(9,OUTPUT);

}

void loop()

{

p=analogRead(A0); Serial.println(p); if(p<= 500){

digitalWrite(9,HIGH);

}

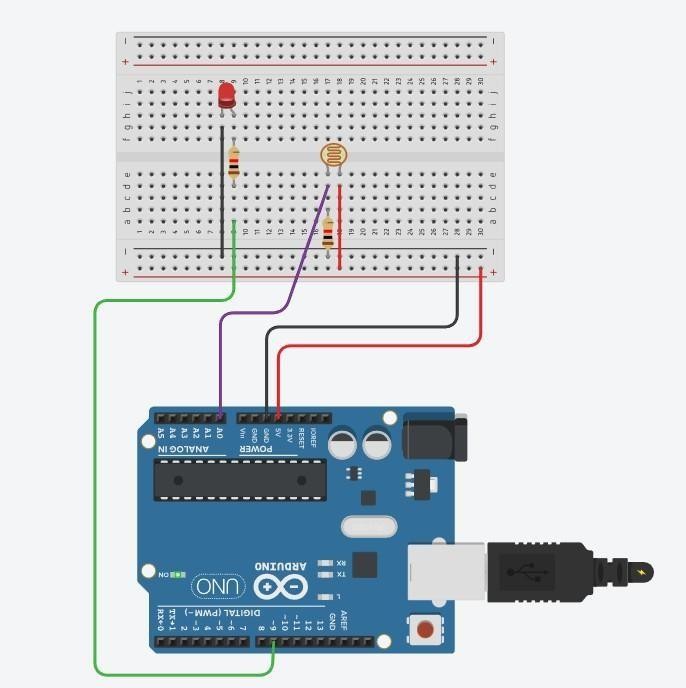
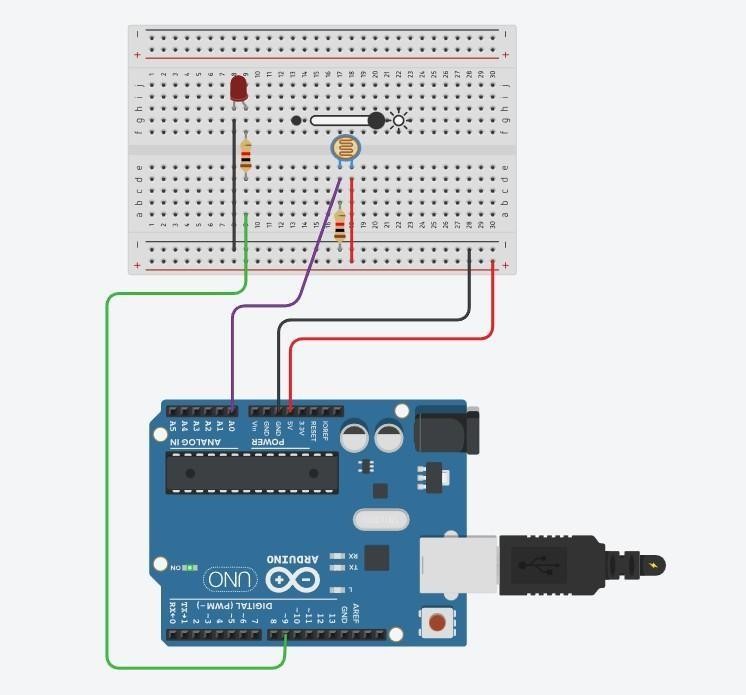
else{ digitalWrite(9,LOW);

}

delay(1000);

}

# 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 interfaceLEDss with Arduino and write a program to show the fading effect on leds.

# Objectives:

* To interface LED with Arduino.
* 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.

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. Passing a voltage through a potentiometer and into an analog input on your board makes it 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.

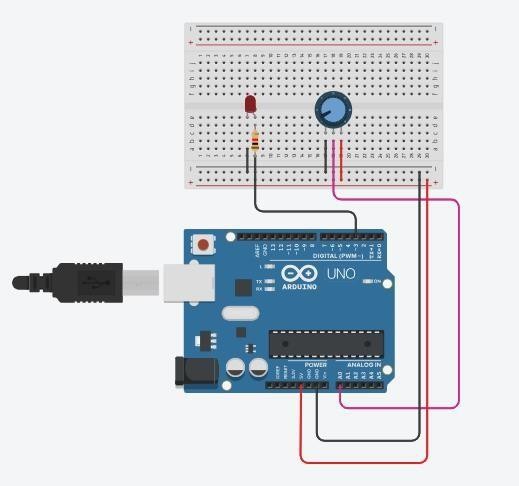
A **loop** is a programming function that iterates a statement or condition based on specified boundaries.

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

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:



**Code:**

int LED\_PIN =11;

void setup()

{

Serial.begin(9600); pinMode(LED\_PIN, OUTPUT);

}

void loop()

{

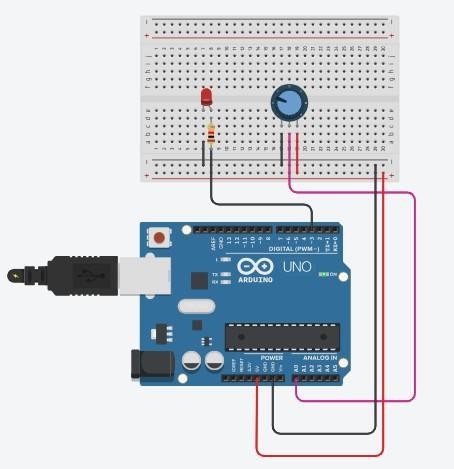
int analogValue = analogRead(A0);

int brightness = map(analogValue,0, 1023, 0, 100); Serial.println(brightness); 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:

* To interface temperature sensor with Arduino.
* 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.

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. Passing a voltage through a potentiometer and into an analog input on your board makes it 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.

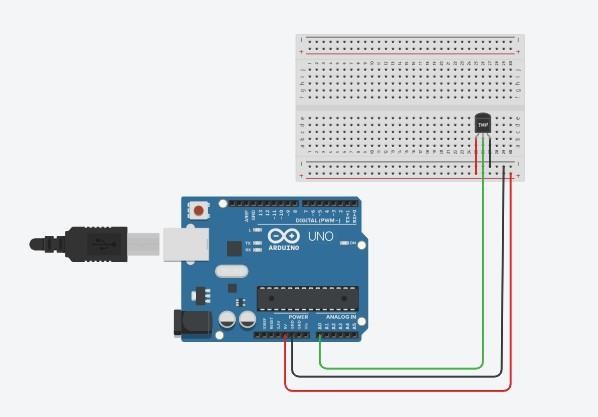
A **loop** is a programming function that iterates a statement or condition based on specified boundaries.

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

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:



**Code:**

void setup()

{

Serial.begin(9600);

}

void loop()

{

int reading= analogRead(A0);

float voltage = reading \* (5.0/1024.0); float temperatureC = (voltage - 0.5) \* 100; Serial.println("Temperatue:"); Serial.print(temperatureC); Serial.print("\xB0");

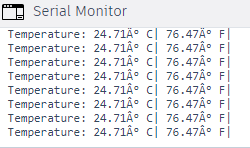
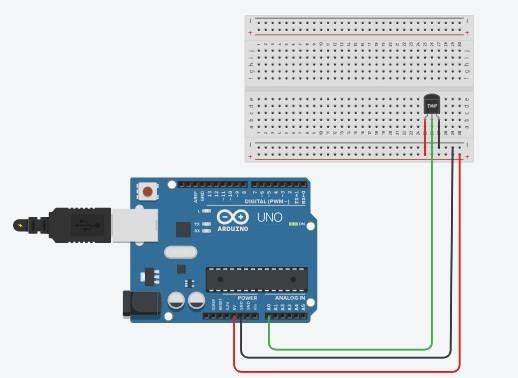
Serial.print("C");

float temperaturef = (temperatureC \* 9.0/ 5.0 + 32.0); Serial.print(" | ");

Serial.print(temperaturef); Serial.print("\xB0"); Serial.print("F"); delay(1000);

}

# 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

**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:

* To interface PIR/ Ultrasonic sensor with Arduino.
* 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.

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

**PIR Sensor** is a passive infrared sensor that does not radiate energy to space. It receives the infrared radiation from the human body to make an alarm.

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

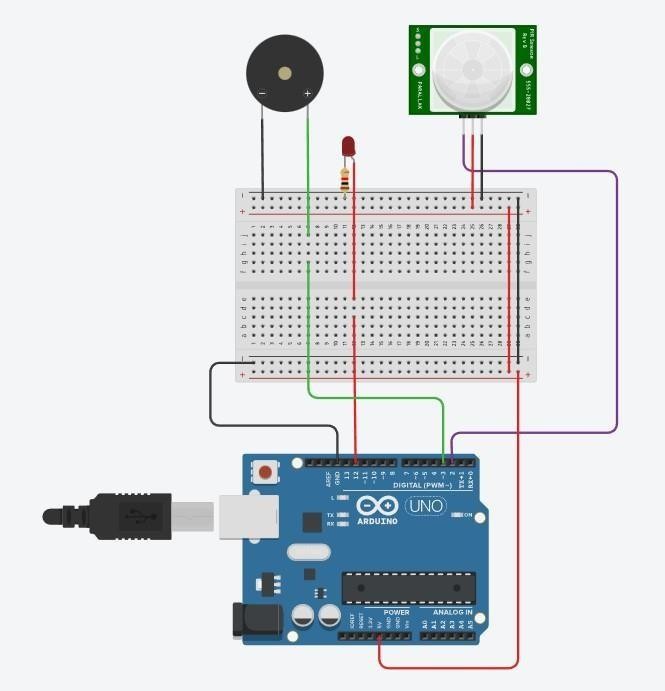
A **loop** is a programming function that iterates a statement or condition based on specified boundaries

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:



**Code:**

int sensorState = 0; void setup()

{

pinMode(2,INPUT); pinMode(3,OUTPUT); pinMode(12, OUTPUT); Serial.begin(9600);

}

void loop()

{

sensorState = digitalRead(2); if(sensorState == HIGH)

{

digitalWrite(12, HIGH); digitalWrite(3,HIGH); Serial.println("Sensor Activated");

}

else

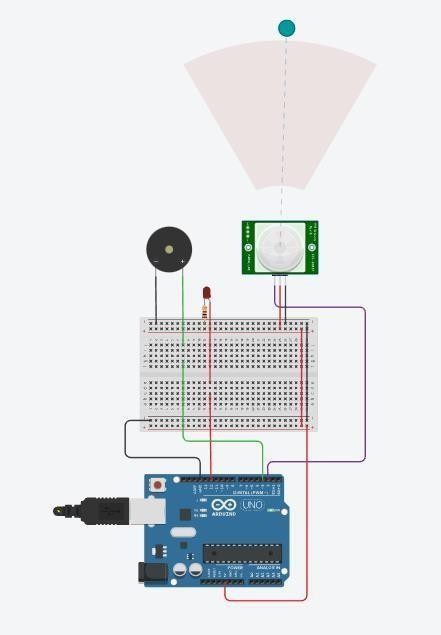
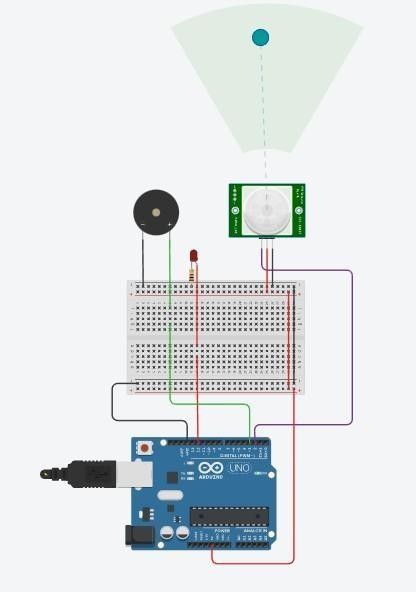
{

digitalWrite(12,LOW ); digitalWrite(3,LOW); Serial.println("Sensor Deactivated");

}

}

# 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.

# Ultrasonic 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:

* + To interface PIR/ Ultrasonic sensor with Arduino.
  + 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.

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

**PIR Sensor** is a passive infrared sensor that does not radiate energy to space. It receives the infrared radiation from the human body to make an alarm.

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

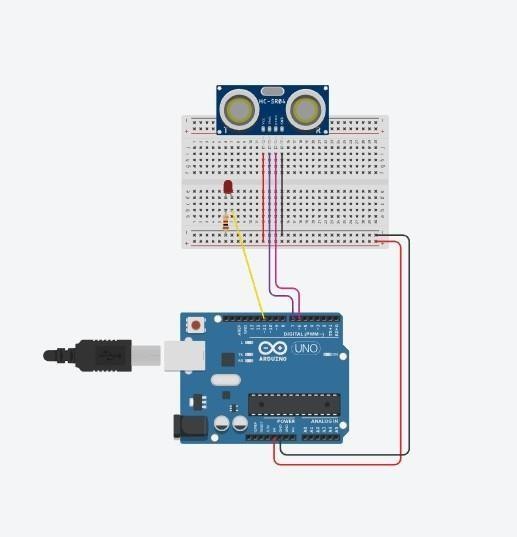
A **loop** is a programming function that iterates a statement or condition based on specified boundaries

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:



**Code:**

float inches = 0; float cm = 0; int led = 11;

const int trigPin = 7; //Output const int echoPin = 6; //Input void setup()

{

pinMode (trigPin, OUTPUT); pinMode(echoPin,INPUT); Serial.begin(9600);

}

void loop()

{

digitalWrite(trigPin, LOW); delayMicroseconds(2); digitalWrite(trigPin, HIGH); delayMicroseconds(10); digitalWrite(trigPin, LOW);

int duration = pulseIn(echoPin, HIGH);

if(cm < 200)

{

digitalWrite(led,HIGH);

}

else

{

digitalWrite(led,LOW);

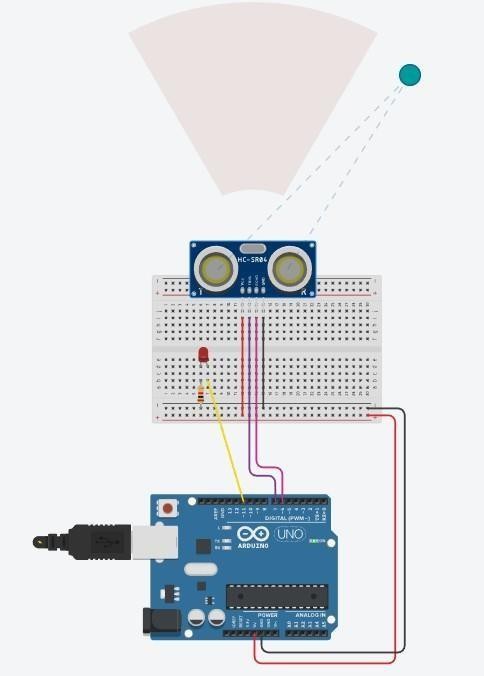
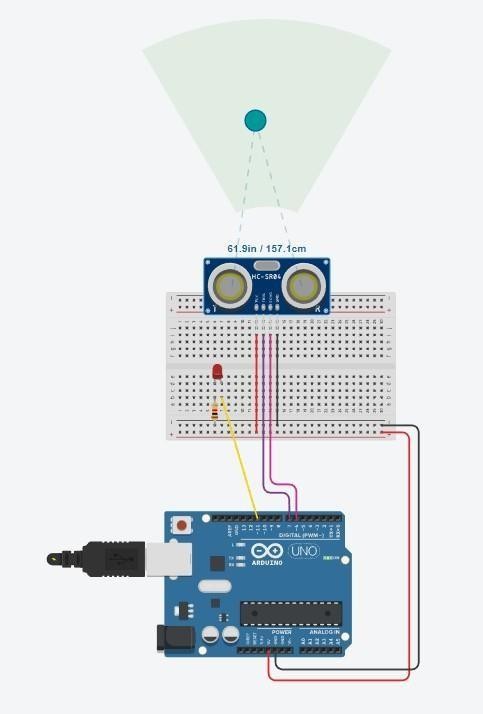
}

Serial.print("Duration"); Serial.println(duration);

Serial.print("in, "); Serial.print(cm); Serial.println("cm"); delay(1000);

}

# Output:



**Conclusion:** Thus, learnt 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:

* To interfaceservo motor with Arduino.
* 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.

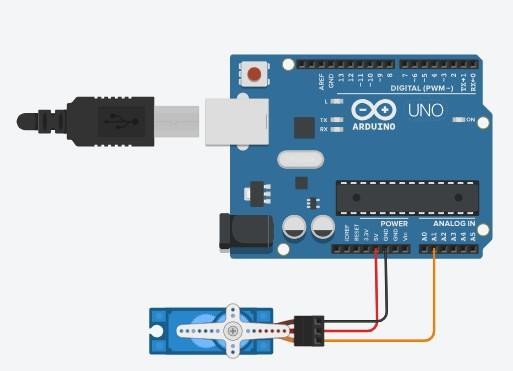
A **loop** is a programming function that iterates a statement or condition based on specified boundaries.

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

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:



**Code:**

//#include<Servo.h> Servo myservo;

#define servoPin 9 int angle = 0; void setup()

{

myservo.attach(servoPin);

}

void loop()

{

myservo.write(90); delay(1000); myservo.write(180); delay(1000); myservo.write(0); delay(1000);

for(angle = 0;angle<=180; angle += 1)

{

myservo.write(angle); delay(40);

}

for(angle = 180; angle>=0; angle -= 1)

{

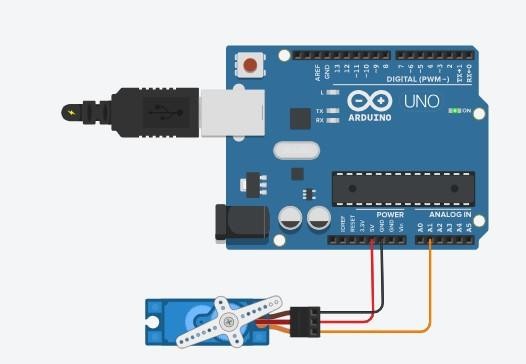
myservo.write(angle); delay(30);

}

delay(1000);

}

# 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.

# B]DC Motor:

**Aim:**To interface DC motor with Arduino and write a program to control a DC motor.

# Objectives:

* To interfaceDC motor with Arduino.
* Write a Program to s program to control a DC motor

# Theory:

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

A **direct current (DC) motor** is a type of electric machine that converts electrical energy into mechanical energy. DC motors take electrical power through direct current, and convert this energy into mechanical rotation.

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

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

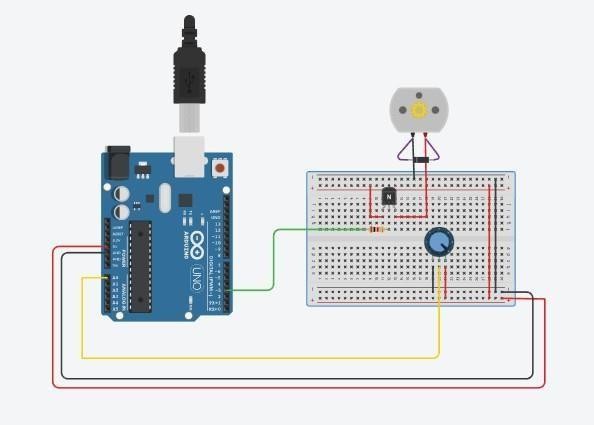
A **loop** is a programming function that iterates a statement or condition based on specified boundaries.

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:



**Code :**

int analogInPin = A0; int sensorValue = 0; int outputValue = 0; int transisterPin = 3; void setup()

{

Serial.begin(9600); pinMode(transisterPin, OUTPUT);

}

void loop()

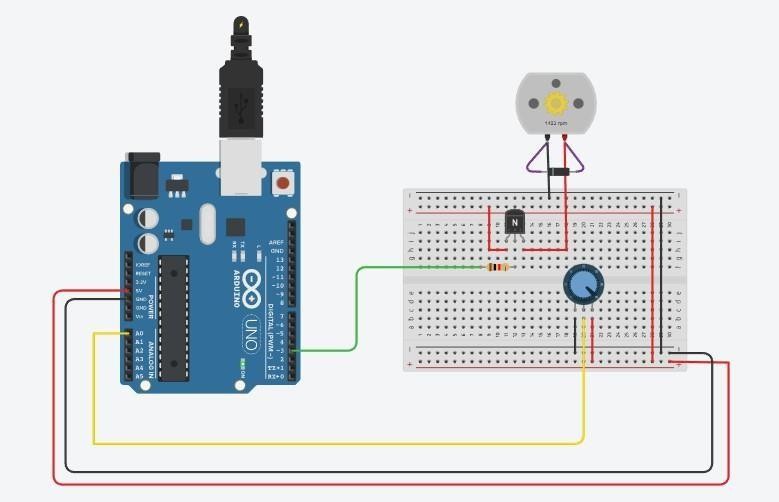
{

sensorValue = analogRead(analogInPin); outputValue = map(sensorValue, 0, 1023, 0, 255); analogWrite(transisterPin, sensorValue); Serial.print(outputValue);

Serial.print("\n"); delay(10);

}

# Output :



**Conclusion:** Thus learnt how to interface DC motor with Arduino and write a program to control a DC motor.

# 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.to

# Objectives:

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

# 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).

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

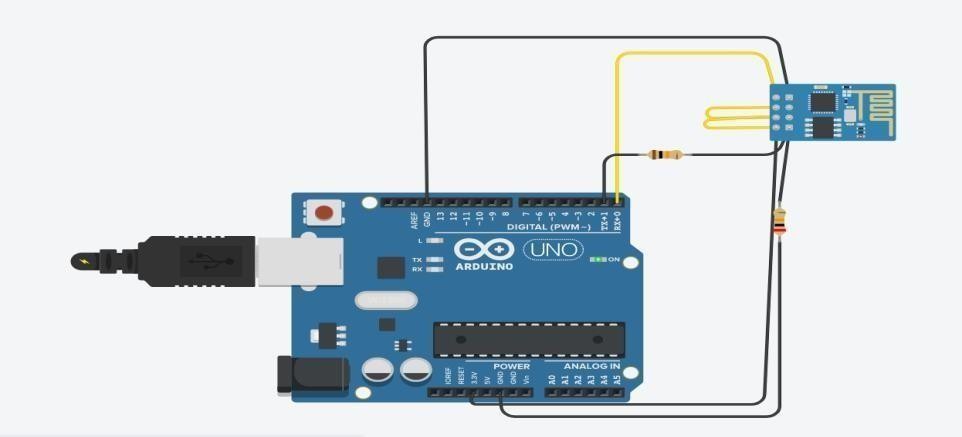
A **loop** is a programming function that iterates a statement or condition based on specified boundaries

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 :



**Code :**

void setup()

Serial.begin(115200); delay(1000);

//Access to the internet using AT+CWJAP="SSID","password"

//if you want thingspeak through Tinkercad use Simulator Wifi as your ssid Serial.println("AT+CWJAP=\"Simulator Wifi\",\"\"\r\n");

delay(3000);

}

void loop()

{

{

//Establish TCP connection, UDP transmission, or SSL connection. Use

AT+CIPSTART=<"type">,<"remote host">,<remote port> Serial.println("AT+CIPSTART=\"TCP\",\"thingspeak.com\",80"); delay(5000);

//Set the data length to be send in the Normal Transmission Mode using AT+CIPSEND=<length>

int len = 57;//length of line 15 Serial.print("AT+CIPSEND="); Serial.println(len);

delay(10);

//Connect to thingspeak server to post data using your API\_WRITE\_KEY Serial.print("GET /update?api\_key=6TPSHYV5TDYD7NLX&field1=120 HTTP/1.1\r\n"); delay(100);

//Close TCP/UDP/SSL Connection using AT+CIPCLOSE=<link ID> Serial.println("AT+CIPCLOSE=0\r\n");

delay(6000);

}

}

# Output:

**Conclusion**: Thus, learnt how to interfaceLED with Arduino and write a program to send sensor data to the cloud using ThingSpeak/ AWS and receive notification.to.

# 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:

* To interface Temperature sensor with Arduino
* 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).

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

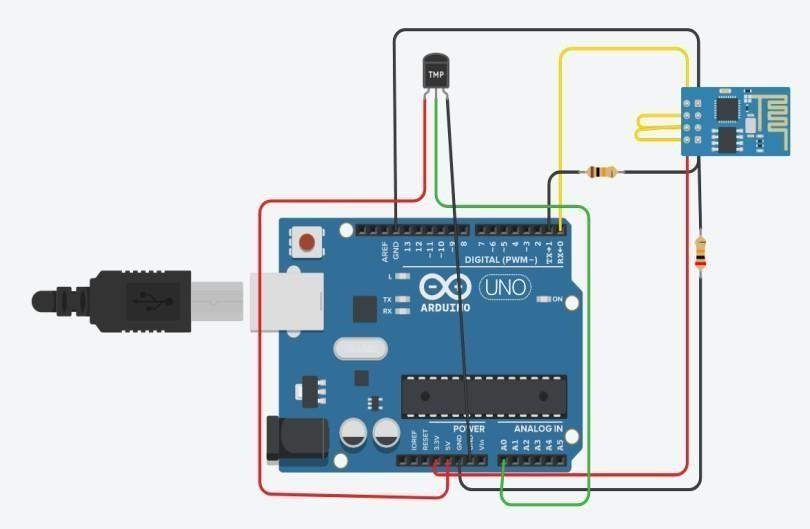
A **loop** is a programming function that iterates a statement or condition based on specified boundaries

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:



**Code :**

void setup()

Serial.begin(115200); delay(1000);

//Access to the internet using AT+CWJAP="SSID","password"

//if you want thingspeak through Tinkercad use Simulator Wifi as your ssid Serial.println("AT+CWJAP=\"Simulator Wifi\",\"\"\r\n");

delay(3000);

}

void loop()

{

int sensorValue = analogRead(A0); float volt = (sensorValue/1020.0) \*

4.9; //Volts float int tempC = (volt - 0.5) \* 100; //Celcius Serial.println(tempC);

Serial.println("AT+CIPSTART=\"TCP\",\"api.thingspeak.com\", 80\r\n"); delay(5000);

int len = 65; Serial.print("AT+CIPS END=");

Serial.println(len); delay(10);

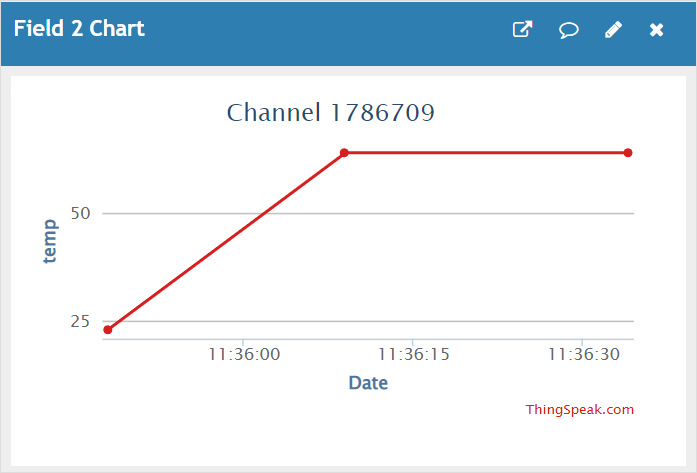
Serial.print("GET /update?api\_key=Q3C799K0VQKULOQM&field2=" + String(tempC)+" HTTP/1.1\r\n");

delay(100); Serial.println("AT+CIP CLOSE=0\r\n");

delay(6000);

}

# Output :



**Conclusion:** Thus, learnt how to interfaceTemperature sensor with Arduino and write a program to send sensor data to the cloud using ThingSpeak/ AWS and receive notification.

# Practical No. 18 (Mini Project) Topic : Arduino Fire Alarm System

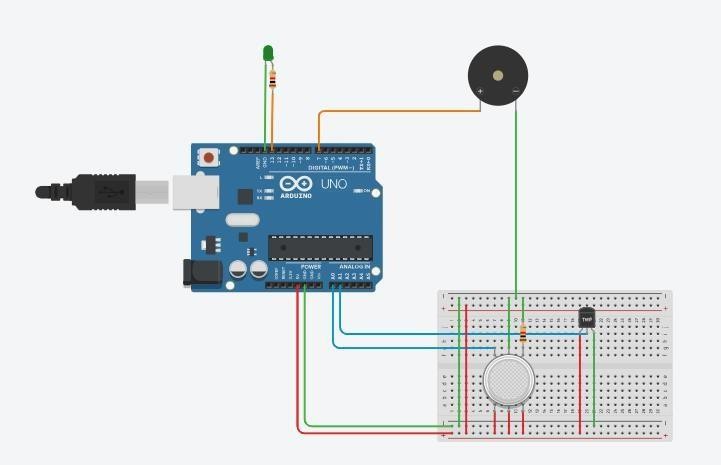
**Aim :**

The Arduino Fire Alarm System is a project aimed at designing and implementing a fire detection and alarm system using an Arduino microcontroller. The system will utilize various sensors to detect the presence of smoke or fire and trigger an alarm to alert occupants and minimize potential damage.

# Objectives:

Objective of the Arduino Fire Alarm System project is to enhance fire safety by creating a reliable and efficient fire detection and alarm system using Arduino technology, contributing to the protection of lives and properties in various settings.

# Circuit Diagram:



**Code:** float temp; float vout;

float vout1; int LED = 13; int gasSensor; int piezo = 7; void setup()

{

pinMode(A0,INP UT);

pinMode(A1, INPUT);

pinMode(LED,OU TPUT);

pinMode(piezo,O UTPUT);

Serial.begin(9600)

;

}

void loop()

{

vout=analogRead( A1);

vout1=(vout/1023)

\*5000;

temp=(vout1- 500)/10;

gasSensor=analog Read(A0);

if (temp>=80)

{

digitalWrite(LED, HIGH);

}

else

{

digitalWrite(LED, LOW);

}

if (gasSensor>=100)

{

digitalWrite(piezo, HIGH);

}

else

{

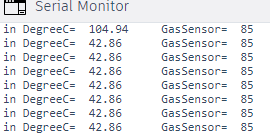
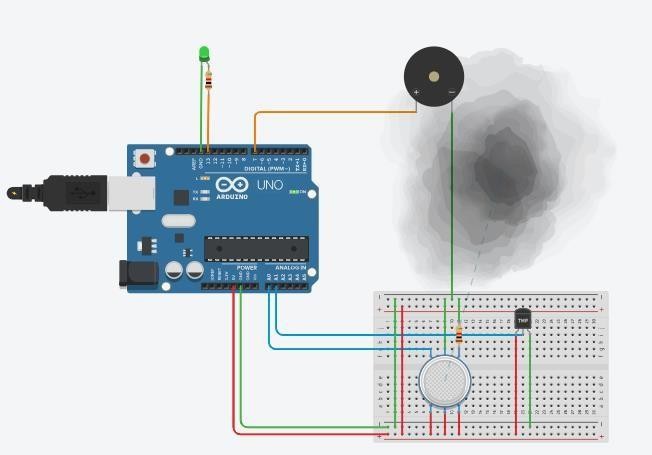
digitalWrite(piezo, LOW);

}

Serial.print("in DegreeC= "); Serial.print(" "); Serial.print(temp); Serial.print("\t"); Serial.print("GasS ensor= "); Serial.print(" "); Serial.print(gasSen sor); Serial.println(); delay(1000);

}

# Output:



**Conclusion :**

Arduino Fire Alarm System project contributes to enhancing fire safety by providing a reliable, efficient, and customizable solution for detecting and alerting occupants to fire hazards. The project serves as a foundation for future advancements and improvements in the field of fire alarm systems, with the potential for integration with other smart technologies for enhanced safety and automation.