

**Name: Mohammed Ahmed Ali**

**USN:1BM18CS135**

## IOT LAB CONSOLIDATED REPORT

Program No. – 01

Program Title – LED Blinking

---

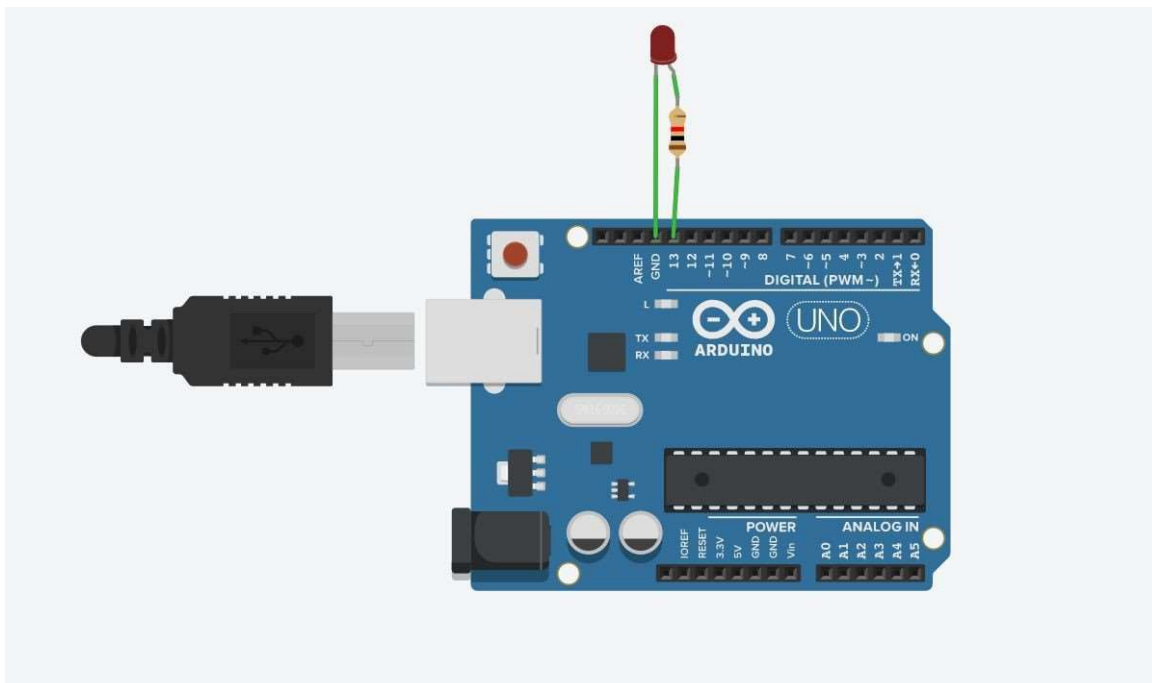
### AIM

Turn the LED on for a second, then off for a second, repeatedly.

### HARDWARES REQUIRED

- Arduino Board
- LEDs

### CIRCUIT DIAGRAM



## CODE

```
void setup()

{

  pinMode(13, OUTPUT);

}

void loop()

{

  digitalWrite(13, HIGH);

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

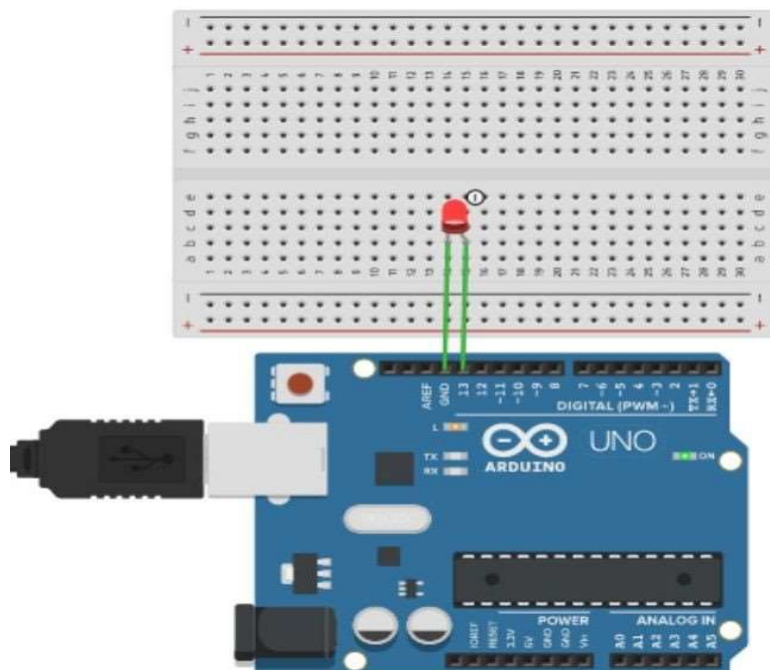
  digitalWrite(13, LOW);

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

}
```

## OUTPUT

The LED was found to be blinking at an interval of 1000 ms.



Program No. – 02

Program Title – Traffic Controller

---

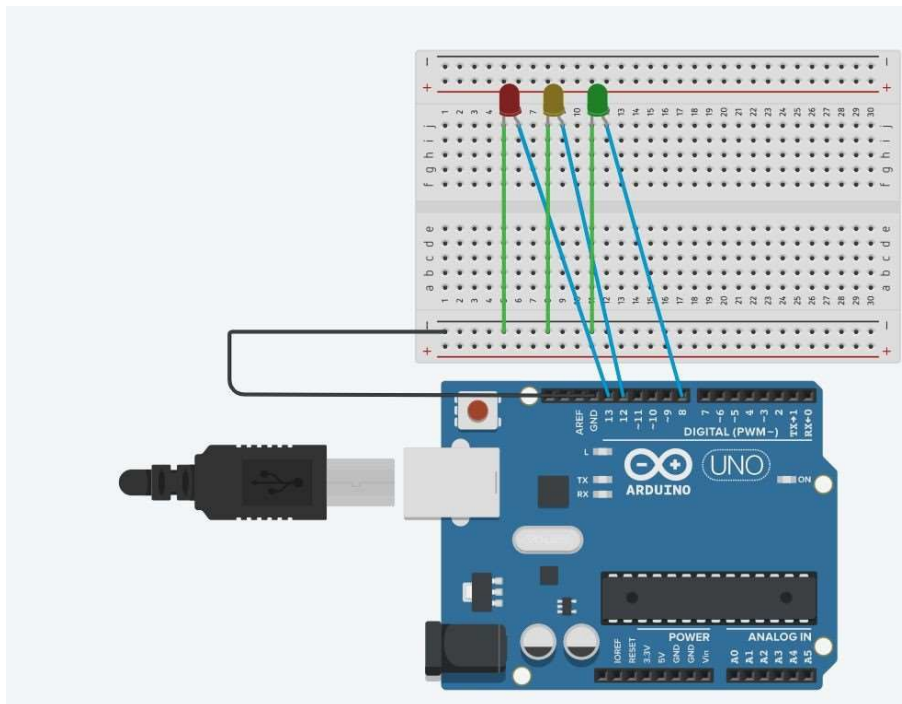
AIM

Traffic Signal Simulator.

HARDWARES REQUIRED

- Arduino Board
- LEDs
- Breadboard

CIRCUIT DIAGRAM



## CODE

```
void setup()
{
  pinMode(13, OUTPUT);
  pinMode(12,OUTPUT);
  pinMode(8,OUTPUT);
}

void red()
{
  digitalWrite(13, HIGH);
  digitalWrite(12,LOW)
```

```
digitalWrite(8,LOW);
```

```
}
```

```
void yellow()
```

```
{
```

```
    digitalWrite(13, LOW);
```

```
    digitalWrite(12,HIGH);
```

```
    digitalWrite(8,LOW);
```

```
}
```

```
void green()
```

```
{
```

```
    digitalWrite(13, LOW);
```

```
    digitalWrite(12,LOW);
```

```
    digitalWrite(8,HIGH);
```

```
}
```

```
void loop()
```

```
{
```

```
    red(); delay(3000);
```

```
    yellow();
```

```
    delay(1500);
```

```
    green();
```

```
    delay(3000);
```

```
yellow();
```

```
delay(1500);
```

```
}
```

## OUTPUT

All the three LEDs blink one after the other at an interval of 1000ms.

Program No. – 03

Program Title – LED fading without potentiometer

---

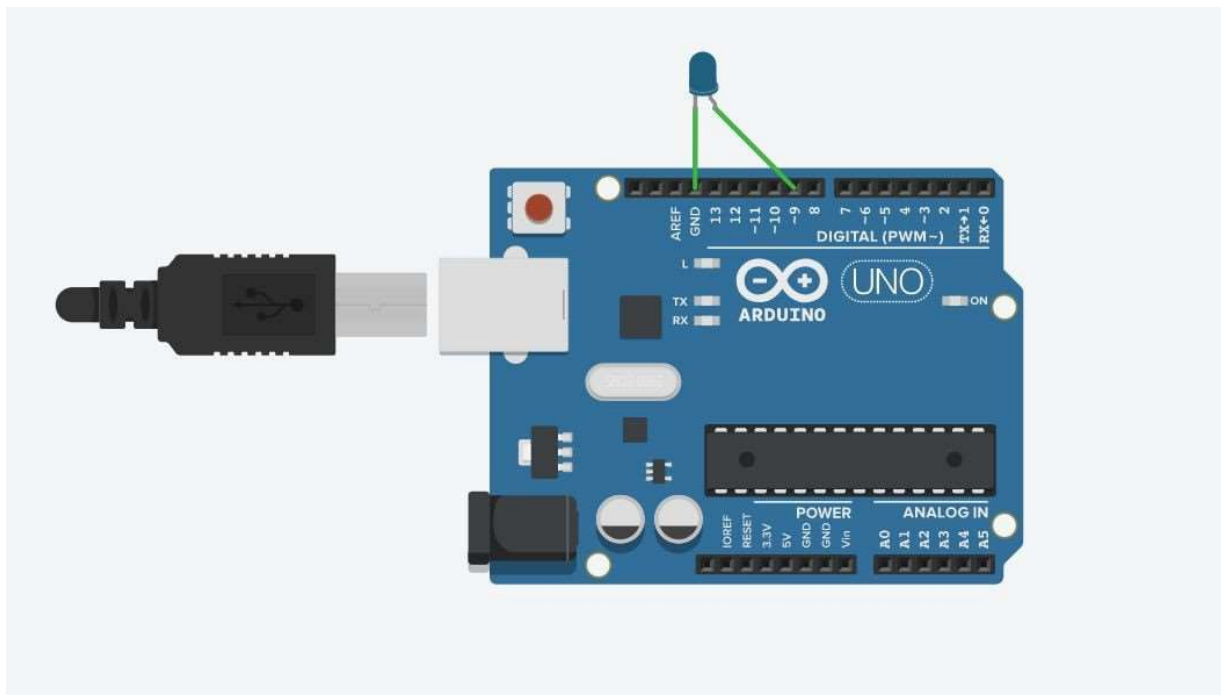
## AIM

Demonstrate to show LED fading.

## HARDWARES REQUIRED

- Arduino Board
- LED bulb

## CIRCUIT DIAGRAM



## CODE

```
void setup()

{

  pinMode(2, OUTPUT);

}


void loop()

{

  for(int fade =0;fade <=255; fade+=5)

  {

    analogWrite(9,fade); delay(30);

  }

  for(int fade = 255; fade>=0;fade-=5)

  {

    analogWrite(9, fade);

    delay(30);

  }
```



}

## OUTPUT

Fading of LED.

Program No. – 04

Program Title – LED fading with potentiometer

---

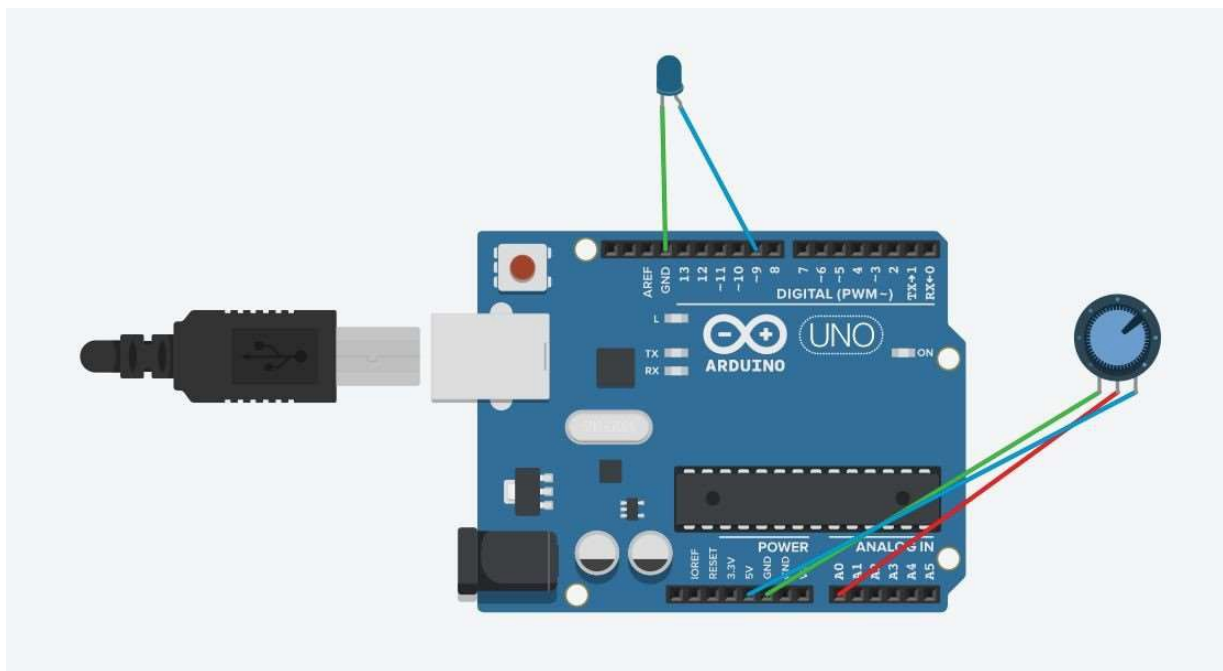
## AIM

Demonstrate to show LED fading (analog output).

## HARDWARES REQUIRED

- Arduino Board
- LED bulb
- Potentiometer

## CIRCUIT DIAGRAM



## CODE

```
int LED_PIN = 9;

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); Serial.print("Analog:

  ");

  Serial.print(analogValue);

  Serial.print(", Brightness : ");

  Serial.println(brightness);
```

$$\}$$

## Fading of LED with potentiometer.



Program No. – 05

Program Title – ON/OFF LED using Push Button

---

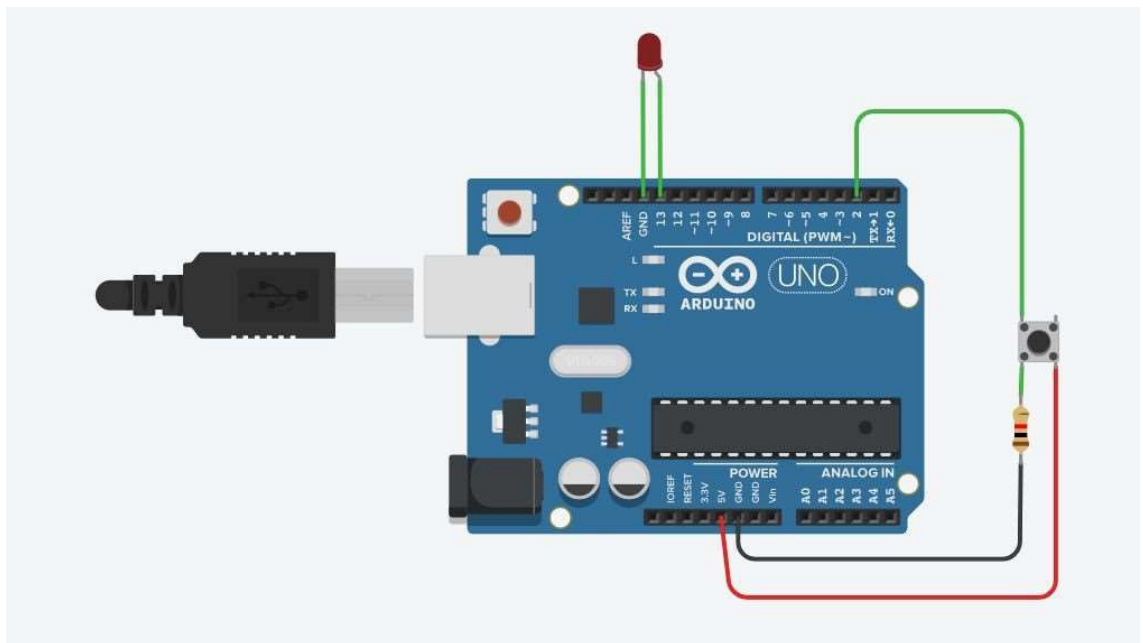
## AIM

Demonstrate to show ON/OFF of a LED using push button(Digital Output).

## HARDWARES REQUIRED

- Arduino Board
- LED bulb
- Push Button
- Resistor

## CIRCUIT DIAGRAM



## CODE

```
int buttonstate=0; void  
setup()  
{  
  pinMode(13, OUTPUT);  
  pinMode(2, OUTPUT);  
  
}  
  
void loop()  
{  
  buttonstate=digitalRead(2);  
  if(buttonstate == HIGH)  
  {digitalWrite(13,HIGH);}  
  else  
  {digitalWrite(13,LOW);}  
}
```

## OUTPUT

ON/OFF of a LED using push button(Digital Output).

Program No. – 06 Program

Title – LDR

---

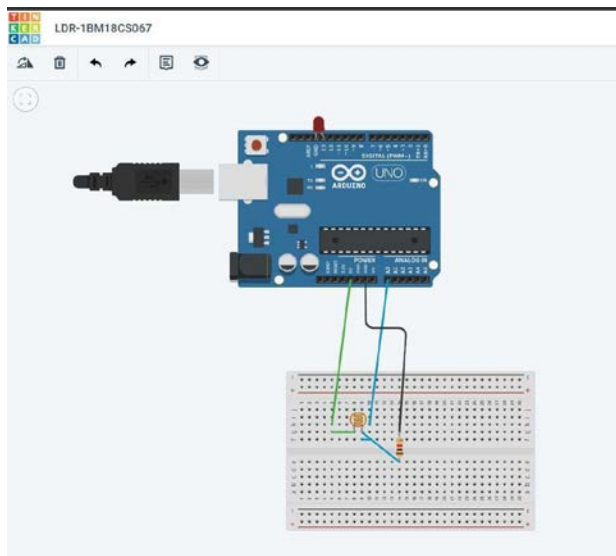
AIM

Demonstrate to show on/off of a LED using LDR night light simulation.

HARDWARES REQUIRED

- Arduino Board
- PhotoResistor
- Resistor
- LED
- Breadboard Small

CIRCUIT DIAGRAM



## CODE

```
const int ledPin = 13;
```

```
const int ldrPin = A0;
```

```
void setup()
```

```
{
```

```
    Serial.begin(9600); pinMode(ledPin,
```

```
    OUTPUT); pinMode(ldrPin, INPUT);
```

```
}
```

```
void loop()
```

```
{
```

```
    int ldrStatus = analogRead(ldrPin);
```

```
    Serial.println(ldrStatus); if(ldrStatus
```

```
    <=10)
```

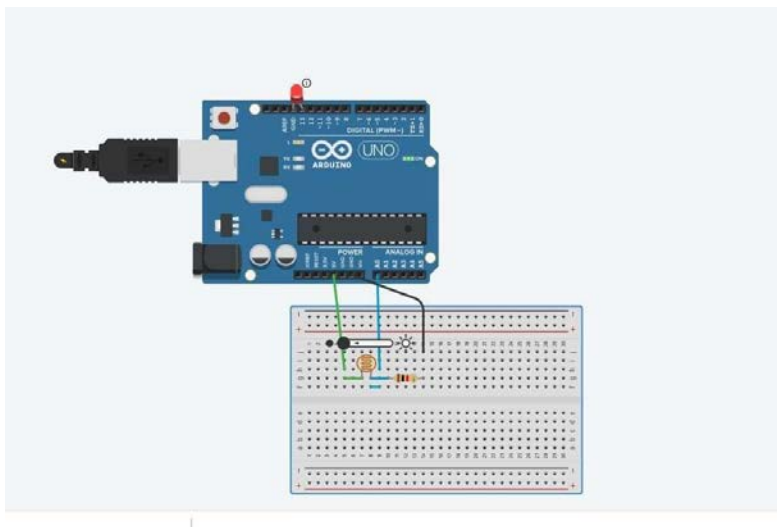
```
    {
```



```
        digitalWrite(ledPin, HIGH); Serial.println("LDR  
is DARK, LED is ON");  
    }  
    else  
    {  
        digitalWrite(ledPin, LOW);  
        Serial.println(" -----");  
    }  
}
```

## OUTPUT

Design a system to show on/off of a LED using LDR night light simulation.



Program No. – 07 Program

Title – PIR

---

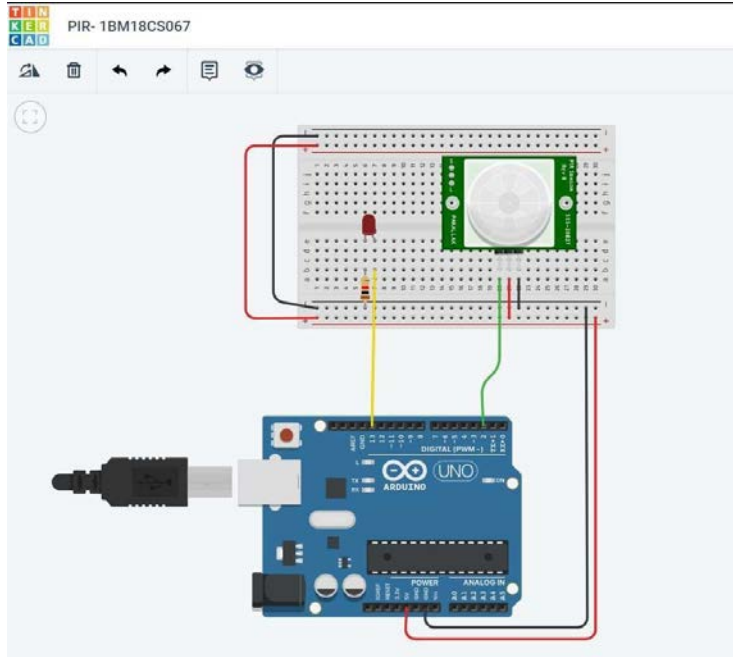
AIM

Demonstrate to show working of PIR sensor.

HARDWARES REQUIRED

- Arduino Board
- PIR sensor
- Resistor
- LED pin
- Breadboard Small

CIRCUIT DIAGRAM



## CODE

```
int sensorState = 0;

void setup()

{

  pinMode(2, INPUT);

  pinMode(13, OUTPUT);

  Serial.begin(9600);

}

void loop()

{

  sensorState = digitalRead(2); if

(sensorState == HIGH) {

digitalWrite(13, HIGH);

  Serial.println("Sensor activated!");

} else {

digitalWrite(13, LOW);

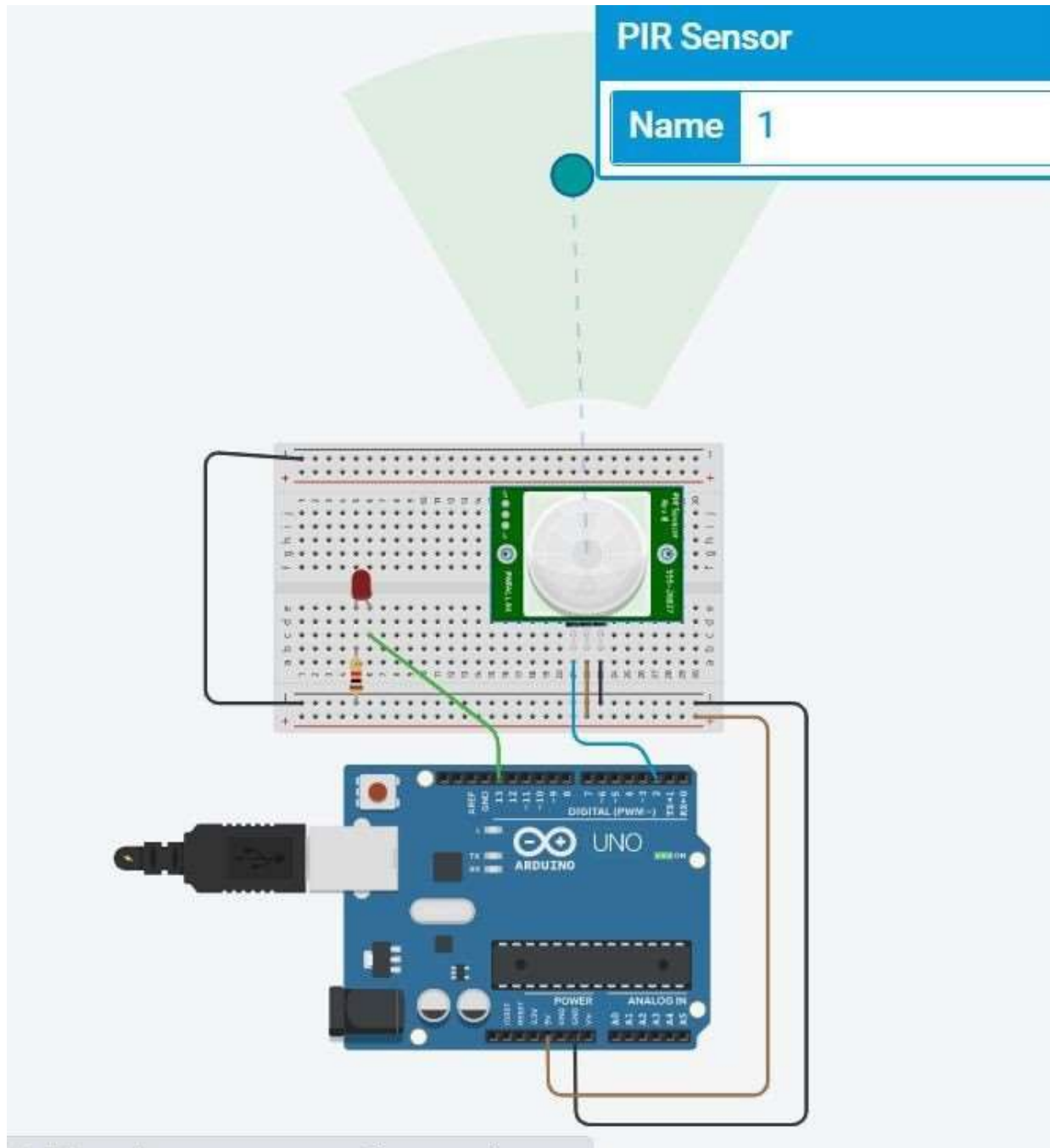
  Serial.println("Sensor deactivated!");

}
```

```
delay(5); }
```

## OUTPUT

Designed a system to show working of PIR Sensor.



Program No. – 08

Program Title – Distance Measurement using ultrasonic sensor

---

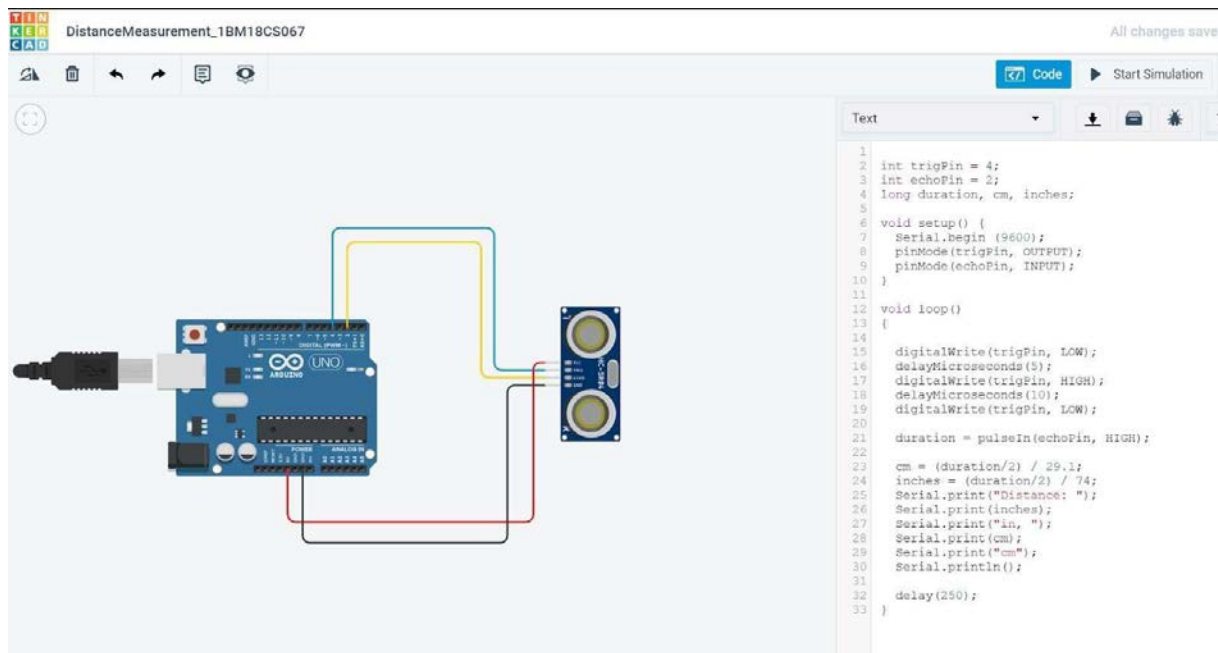
## AIM

Design a system to measure the distance between objects.

## HARDWARES REQUIRED

- Arduino Board
- Ultrasonic sensor HC-SR04

## CIRCUIT DIAGRAM



## CODE

```
int trigPin = 4;

int echoPin = 2;

long duration, cm, inches;

void setup() { Serial.begin
(9600);

pinMode(trigPin, OUTPUT); pinMode(echoPin,
INPUT);

}

void loop()

{
```

```

digitalWrite(trigPin, LOW);

delayMicroseconds(5);

digitalWrite(trigPin, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin, LOW);


duration = pulseIn(echoPin, HIGH); cm
= (duration/2) / 29.1;

inches = (duration/2) / 74;

Serial.print("Distance: ");

Serial.print(inches);

Serial.print("inch, ");

Serial.print(cm);

Serial.print("cm");

Serial.println(); delay(250);

}

```

## OUTPUT

Design a system to measure the distance between objects using ultrasonic device.

Simulator time: 00:00:07

Code

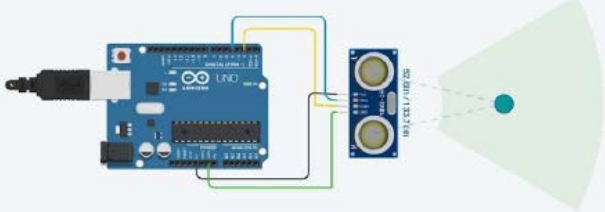
Stop Simulation

Export

Share

Ultrasonic Distance Sensor

Name 1



Text

```
1 int trigPin = 4;
2 int echoPin = 2;
3 long duration, cm, inches;
4
5 void setup() {
6   Serial.begin (9600);
7   pinMode(trigPin, OUTPUT);
8   pinMode(echoPin, INPUT);
9 }
10
11 void loop()
12 {
13
14   digitalWrite(trigPin, LOW);
15   delayMicroseconds(5);
16   digitalWrite(trigPin, HIGH);
17 }
```

Serial Monitor

Distance: 52Inches, 133cm  
Distance: 52Inches, 133cm  
Distance: 52Inches, 133cm  
Distance: 52Inches, 133cm  
Distance: 52Inches, 133cm  
Distance: 52Inches, 133cm  
Distance: 52Inches, 133cm  
Distance: 52Inches, 133cm

1 (Arduino Uno R3)



Program No. – 09

Program Title – Fire Alarm using flame Sensor

---

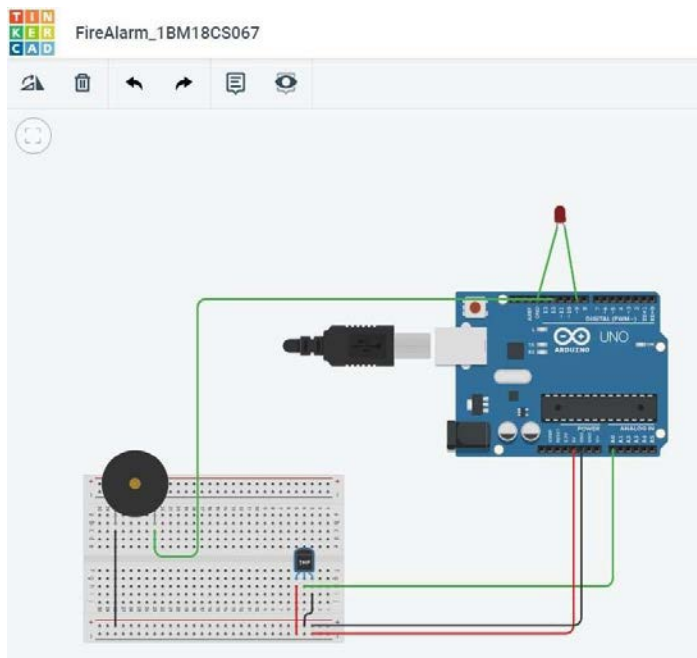
## AIM

Design an alert system using a flame sensor.

## HARDWARES REQUIRED

- Arduino Board
- Piezo
- Temperature Sensor
- Breadboard small

## CIRCUIT DIAGRAM



## CODE

```
const int temperaturePin = 0; int  
buzzer = 12;  
  
void setup()  
{  
  Serial.begin (9600);  
  pinMode(buzzer, OUTPUT);  
  pinMode(9, OUTPUT);  
}  
  
void loop()  
{  
  float voltage, degreesC;  
  voltage = getVoltage(temperaturePin);  
  degreesC = (voltage-0.5)*100.0;  
  
  if(degreesC < 37)
```

```
{  
  Serial.print(degreesC);  
  Serial.println(" SAFE!");  
}
```

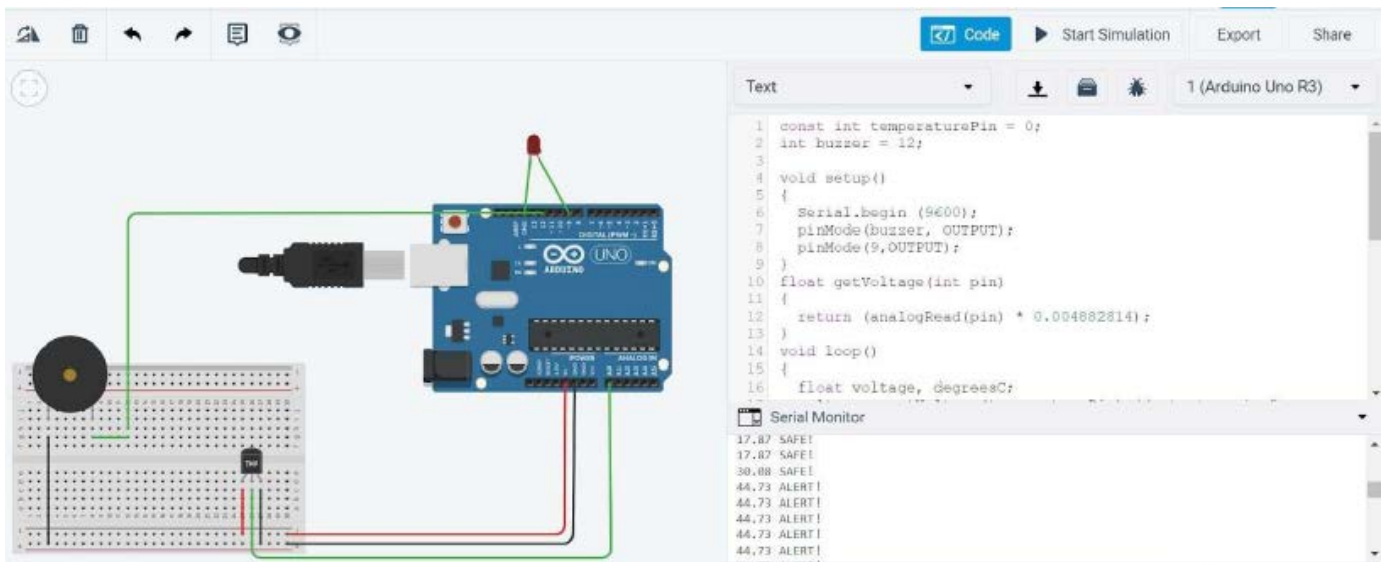
```
if(degreesC > 37)  
{  
  Serial.print(degreesC);  
  Serial.println("FIRE !!!");  
  digitalWrite(9, HIGH);  
  digitalWrite(buzzer, LOW);  
  tone(12, 10000,100);  
  delay(100);  
}
```

```
}
```

```
float getVoltage(int pin)  
{  
  return (analogRead(pin) * 0.004882814);  
}
```

## OUTPUT

Designed an alert system using flame sensor.



Program No. – 10

Program Title – Gas Sensor

---

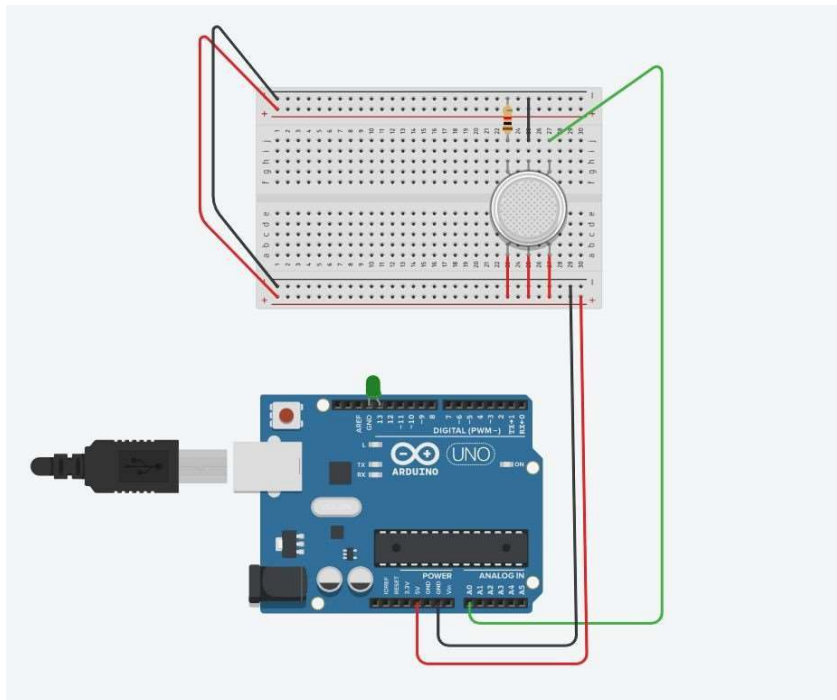
## AIM

To design a smart gas leakage indicator system.

## HARDWARES REQUIRED

- Arduino Board
- Gas sensor
- Resistor
- LED
- Breadboard Small

## CIRCUIT DIAGRAM



## CODE

```
int LED = 13; const  
  
int gas = 0; int  
Gaspin = A0;  
  
void setup()  
{  
    Serial.begin(9600);  
}  
  
void loop()  
{  
    float sensorValue = analogRead(Gaspin);  
    if(sensorValue >= 300)  
    {  
        digitalWrite(LED, HIGH);  
        Serial.print(sensorValue);  
        Serial.println(" *** SMOKE DETECTED ***");  
        delay(sensorValue);  
    }  
}
```

$$\}$$

else

$$\{$$

```
digitalWrite(LED, LOW);
```

```
Serial.println("Serial Value : ");
```

```
Serial.println(sensorValue);
```

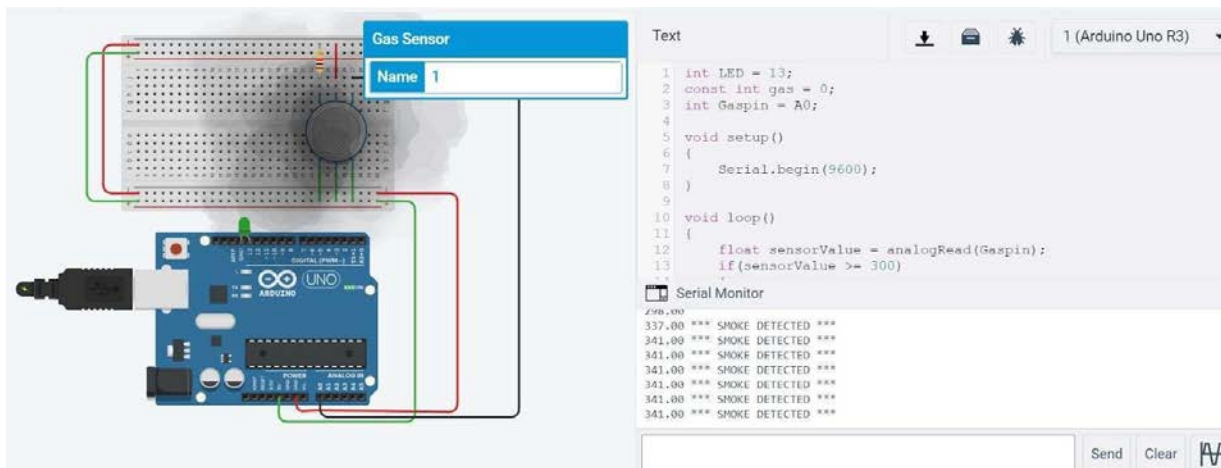
$$\}$$

```
delay(1000);
```

$$\}$$

## OUTPUT

Designed a smart gas leakage indicator system.



Program No. – 11

Program Title – Vibration motor and LDR

---

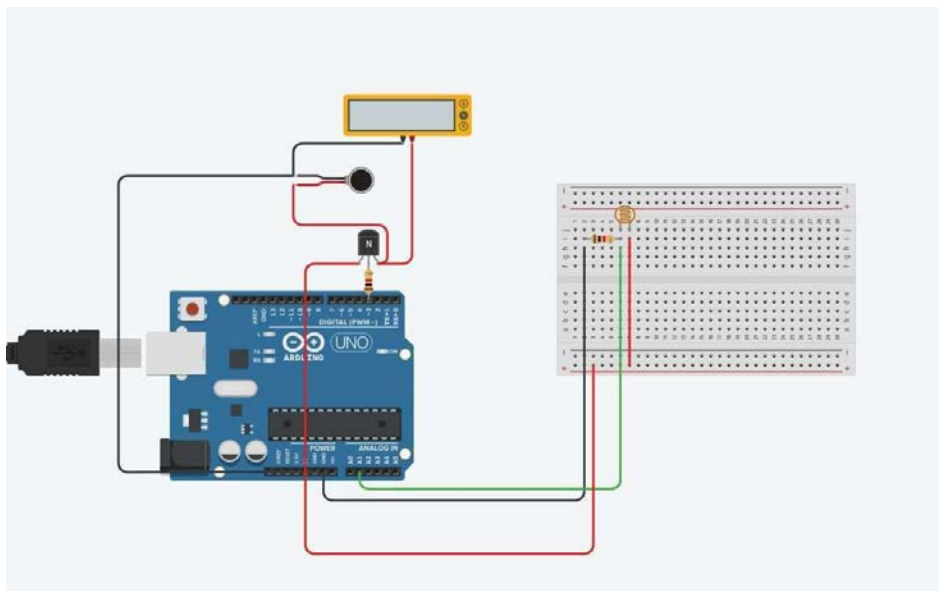
## AIM

To design an automated day indicator system.

## HARDWARES REQUIRED

- Arduino Board
- NPN Transistor
- Resistor
- Vibration motor
- Multimeter
- Photoresistor

## CIRCUIT DIAGRAM





## CODE

```
int motorPin = 3;

int sensorPin = A1; int

threshold = 400;


void setup()

{

    pinMode(motorPin, OUTPUT);

    Serial.begin(9600);

}


void loop()

{

    int sensorValue = analogRead(sensorPin);

    Serial.println(sensorValue);

    if(sensorValue > threshold)

    {

        digitalWrite(motorPin, HIGH);
```

```

    }

    else

    {

        digitalWrite(motorPin, LOW);

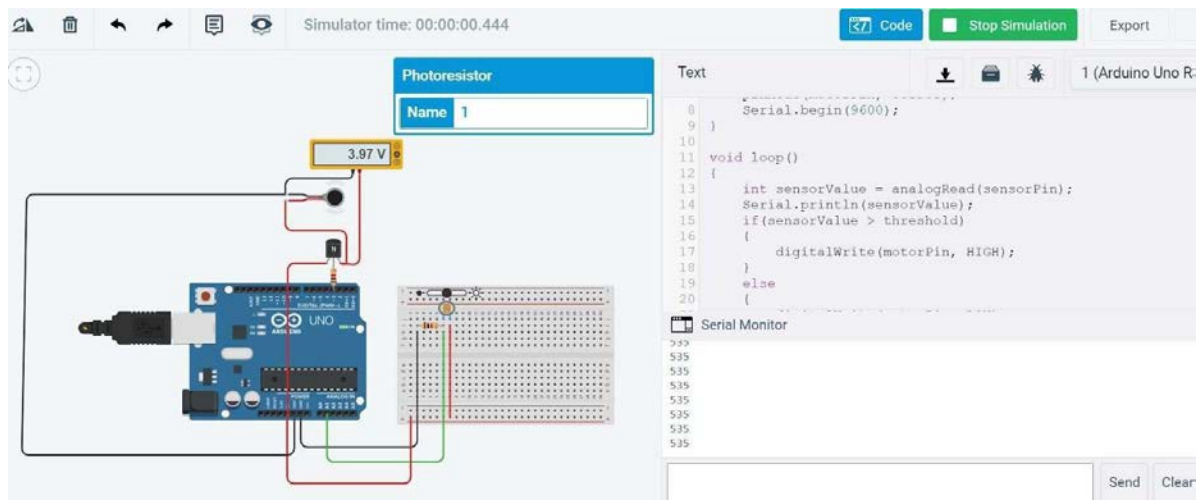
    }

}

```

## OUTPUT

Designed an automated day indicator system.



Program No. – 12

Program Title – Tilt Sensor

---

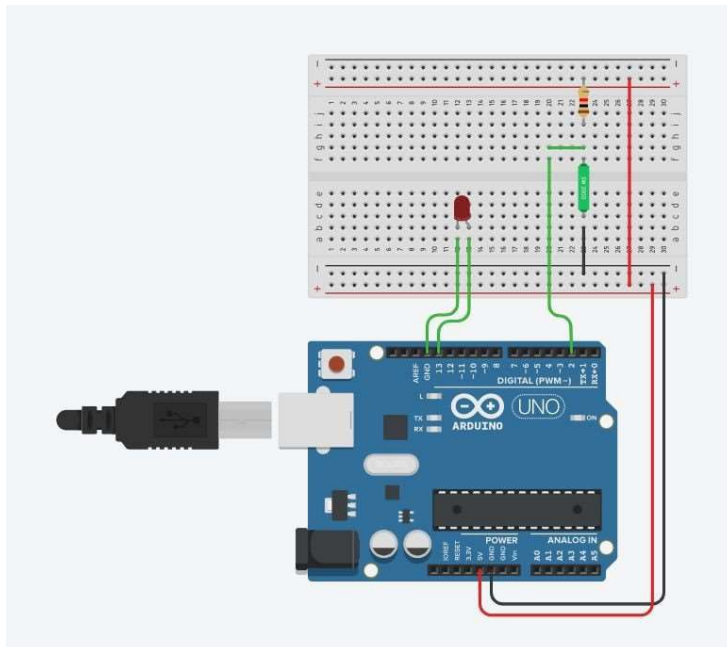
## AIM

Design a Smart Package handling system (Tilt sensor and LED)

## HARDWARES REQUIRED

- Arduino Board
- Breadboard Small
- LEDs
- Tilt Sensor
- Resistor

## CIRCUIT DIAGRAM



## CODE

```
int tilt = 2;

int led = 13;

void setup()

{

    pinMode(tilt, INPUT);

    pinMode(led, OUTPUT);

}


void loop()

{

    int reading;

    reading = digitalRead(tilt);

    if(reading) digitalWrite(led,

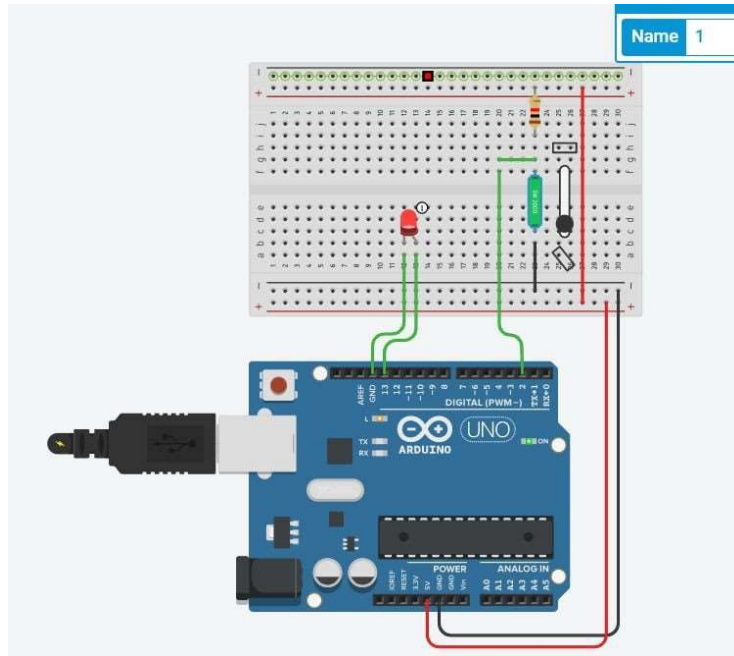
LOW); else

        digitalWrite(led, HIGH);

}
```

## OUTPUT

Designed a Smart Package handling system using Tilt Sensor and LED.



Program No. – 13

Program Title – IR based SERVO Motor controller

---

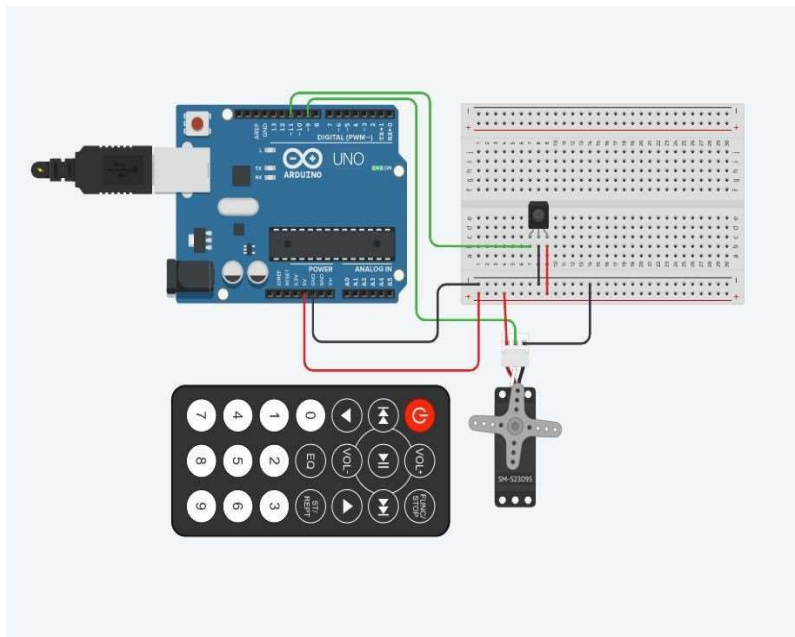
## AIM

Design IR based SERVO Motor controller. (Clockwise and CounterClockwise rotation of shaft).

## HARDWARES REQUIRED

- Arduino Board,
- Breadboard Small,
- IR Sensor,
- IR Remote,
- Micro Servo

## CIRCUIT DIAGRAM



## CODE

```
#include <Servo.h>
```

```
#include <IRremote.h>
```

```
int RECV_PIN = 11;
```

```
IRrecv irrecv(RECV_PIN);
```

```
decode_results results;
```

```
Servo myservo;
```

```
void setup(){ Serial.begin(9600);
```

```
  irrecv.enableIRIn();
```

```
}
```

```
void loop(){
```

```
  if (irrecv.decode(&results))
```

```
{
```

```
  switch (results.value)
```

```
{
```

```
  case 0xFD00FF:
```

```
    myservo.attach(9);
```

```
    Serial.println("Start");
```

```
    break;
```

```
  case 0xFD609F:
```

```
    myservo.write(360);
```

```
    Serial.println("Clockwise");
```

```
    break;
```

```
  case 0xFD20DF:
```

```
    myservo.write(-360); Serial.println("Counter
```

```
    Clockwise");
```

```
    break; default:
```



```
    Serial.print("Unrecognized code received: 0x");  
  
    Serial.println(results.value, HEX);  
  
    break;  
  
}  
  
irrecv.resume();  
  
}  
}
```

## OUTPUT

Designed a Smart Package handling system using Tilt Sensor and LED.



Program No. – 14

Program Title – RGB Led and LCD

---

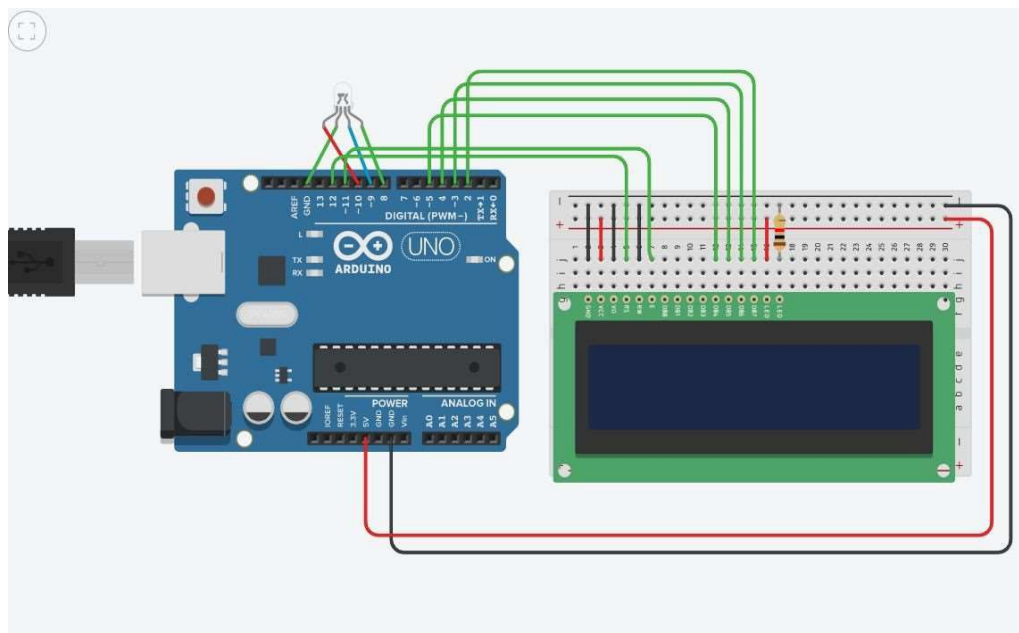
## AIM

Design a display system to print the RED,BLUE and Green colors (RGB Led and LCD).

## HARDWARES REQUIRED

- Arduino Board
- Breadboard Small
- LCD 16x2
- RGB LED
- Resistor

## CIRCUIT DIAGRAM



## CODE

```
#include <LiquidCrystal.h> LiquidCrystal
```

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

```
int red=10;
```

```
int green=8;
```

```
int blue=9;
```

```
void setup()
```

```
{
```

```
pinMode(10, OUTPUT);  
pinMode(9, OUTPUT);  
pinMode(8, OUTPUT);  
}  
  
void loop()  
{  
  lcd.setCursor(0,0); lcd.print("RGB  
  Color Print!"); delay(1000);  
  lcd.clear();  
  RGB_color(255,0,0);//red  
  lcd.print("RED");  
  delay(1000);  
  lcd.clear();  
  
  RGB_color(0,255,0);//Green  
  lcd.print("GREEN"); delay(1000);  
  lcd.clear();  
  
  RGB_color(0,0,255);//Blue  
  lcd.print("BLUE"); delay(1000);  
  lcd.clear();
```

```

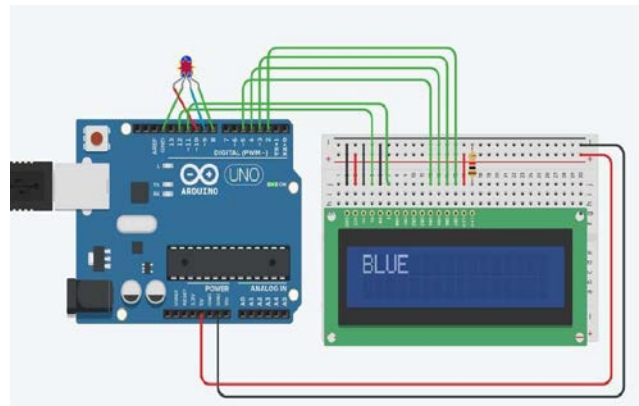
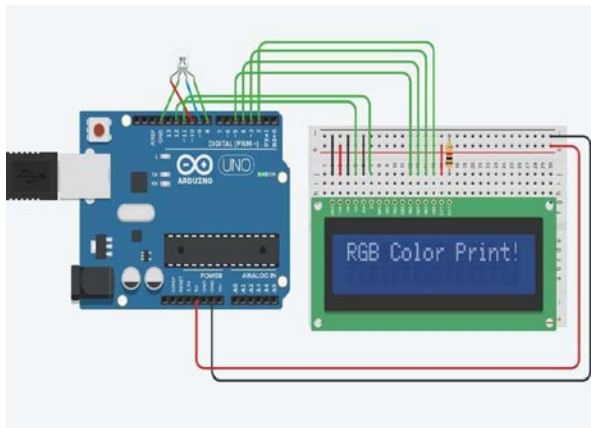
    RGB_color(0,0,0); //White
    lcd.print("WHITE"); delay(1000);
    lcd.clear();
}

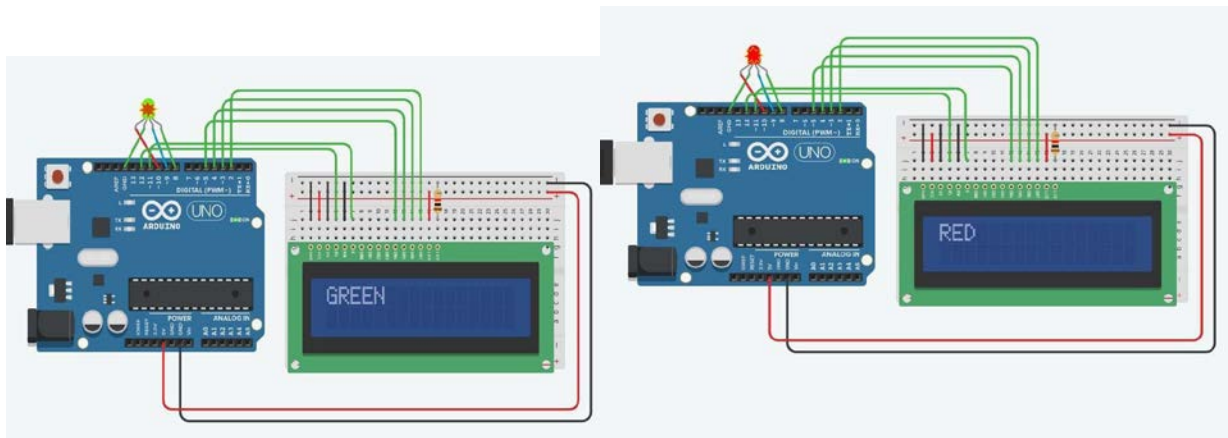
void RGB_color(int red_value, int green_value, int blue_value)
{
    analogWrite(red,red_value); analogWrite(green,green_value);
    analogWrite(blue,blue_value);
}

```

## OUTPUT

Designed a display system to print the RED,BLUE and Green colors (RGB Led and LCD).





Program No. – 16

Program Title – Smart irrigation system

---

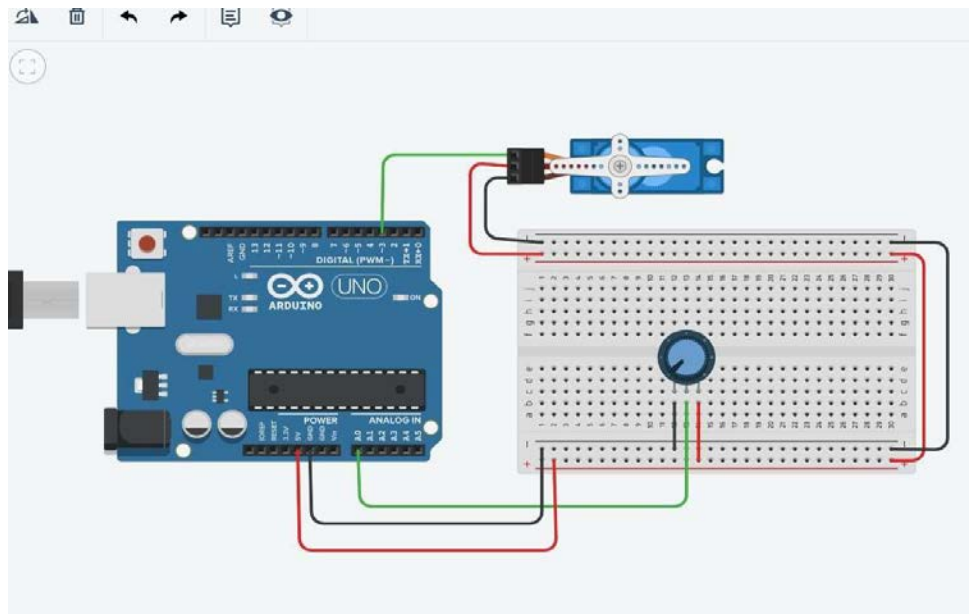
## AIM

Design a smart irrigation system (Potentiometer, Servo motor shaft).

## HARDWARES REQUIRED

- Arduino Board
- Breadboard Small
- Potentiometer
- Servo motor shaft

## CIRCUIT DIAGRAM



## CODE

```
#include <Servo.h>

Servo myservo; // create servo object to control a servo
// twelve servo objects can be created on most boards

int pos = 0;    // variable to store the servo position

int sensorPin = A0;    // select the input pin for the potentiometer

int sensorValue = 0; // variable to store the value coming from the sensor void
setup() {
  myservo.attach(3); // attaches the servo on pin 9 to the servo object
  Serial.begin(9600);
```



```

}

void loop() {

  // read the value from the sensor: sensorValue =
  analogRead(sensorPin); Serial.println
  (sensorValue); if(sensorValue>500)
  {
    for (pos = 0; pos <= 180; pos += 1) { // goes from 0 degrees to 180 degrees
      // in steps of 1 degree

      myservo.write(pos);           // tell servo to go to position in variable 'pos'
      delay(15);                   // waits 15ms for the servo to reach the position
    }
    for (pos = 180; pos >= 0; pos -= 1) { // goes from 180 degrees to 0degrees
      myservo.write(pos);           // tell servo to go to position in variable 'pos'
      delay(15);                   // waits 15ms for the servo to reach the position
    }
  }
  delay (1000);
}

```

## OUTPUT

Designed a smart irrigation system (Potentiometer, Servo motor shaft).

