

**Name – Sai Sriram Vemparala**

Program No. – 01

Program Title – LED Blinking

---

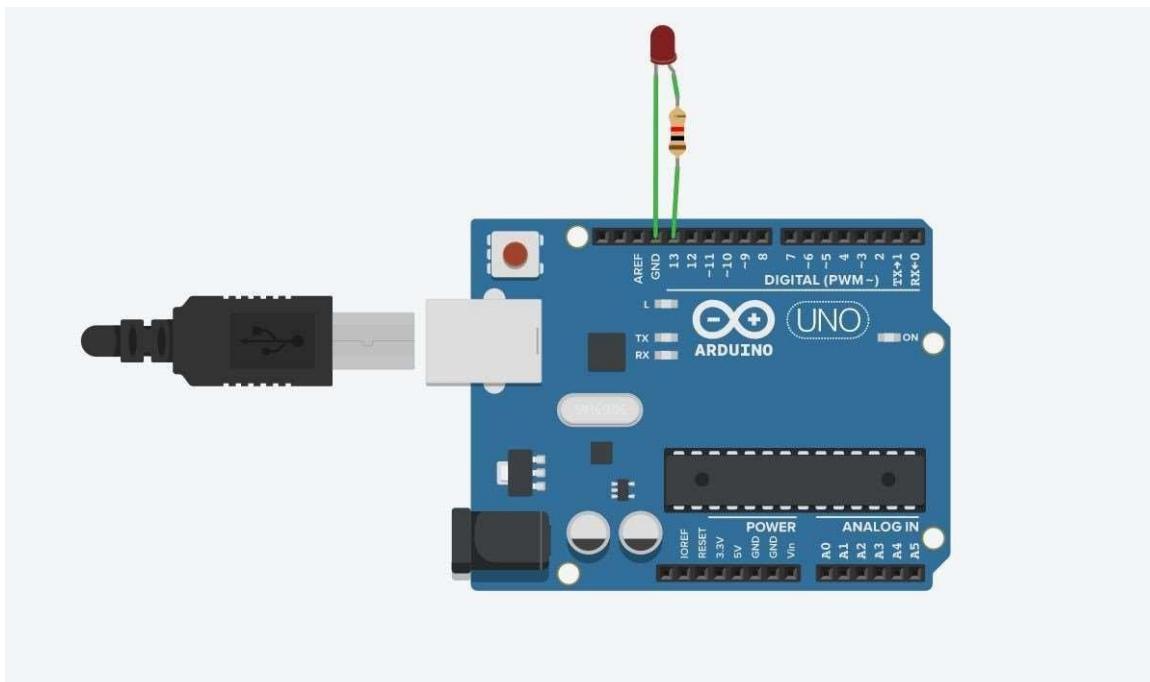
## **AIM**

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

## **HARDWARES REQUIRED**

- Arduino Board
- LEDs

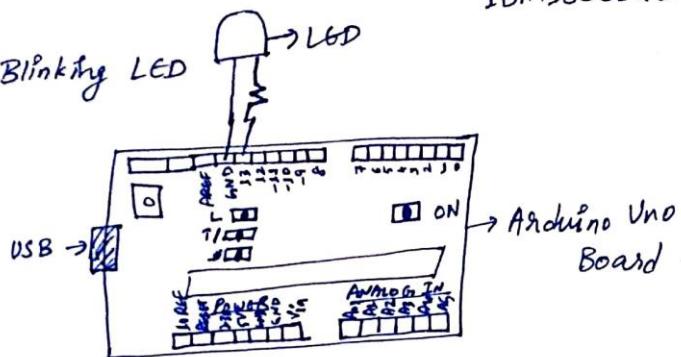
## **CIRCUIT DIAGRAM**



## WRITE-UP

Sai Sriram.V  
1BM18CS140

Program 1:- Blinking LED



CODE:-

```
int ledPin = 13;  
void setup()  
{  
    pinMode(ledPin, OUTPUT);  
}  
  
void loop()  
{  
    digitalWrite(ledPin, HIGH);  
    delay(1000);  
    digitalWrite(ledPin, LOW);  
    delay(1000);  
}
```

①

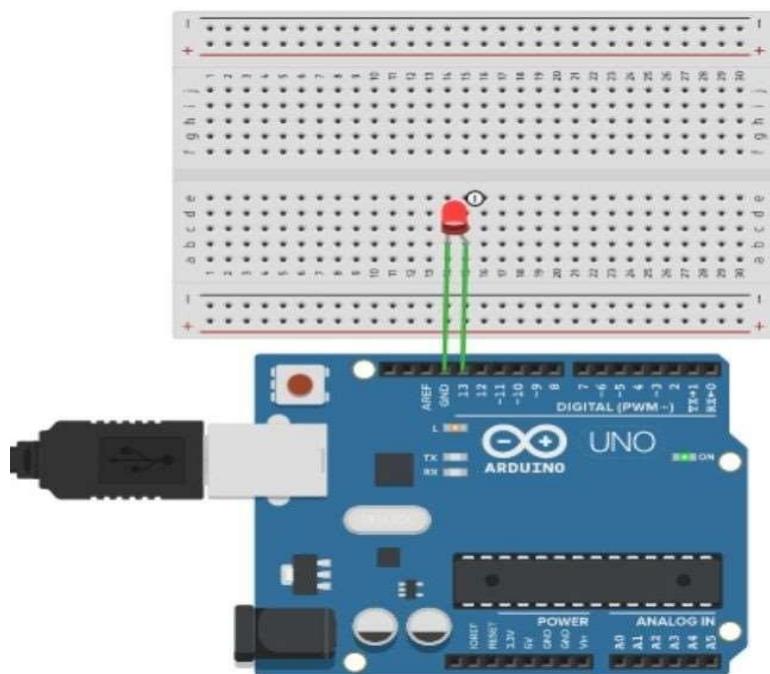
## CODE

```
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.



**Name – Sai Sriram Vemparala**

Program No. – 02

Program Title – Traffic Controller

---

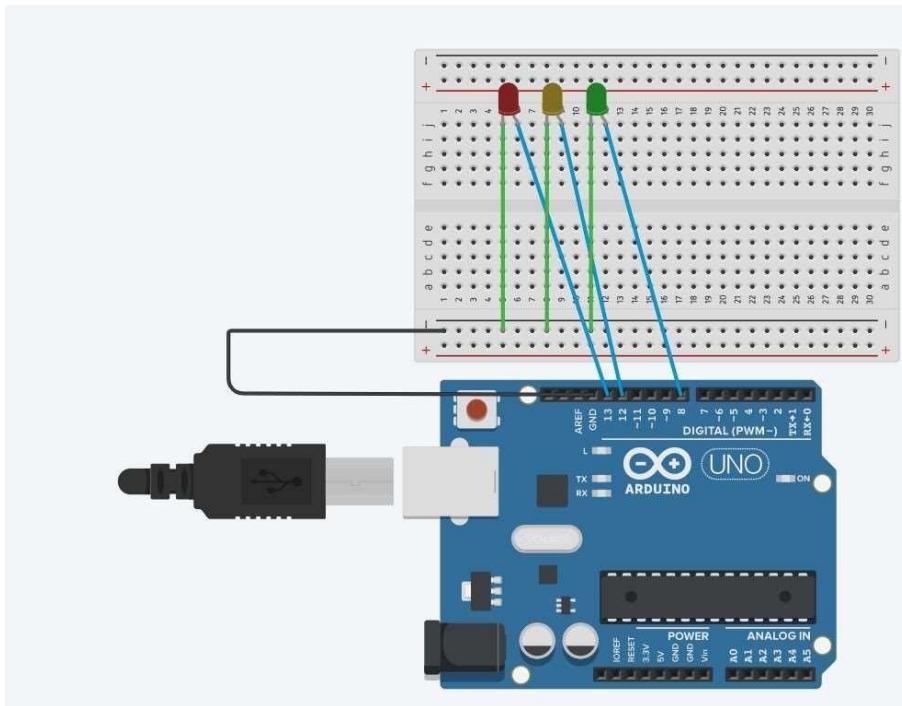
## **AIM**

Traffic Signal Simulator.

## **HARDWARES REQUIRED**

- Arduino Board
- LEDs
- Breadboard

## **CIRCUIT DIAGRAM**

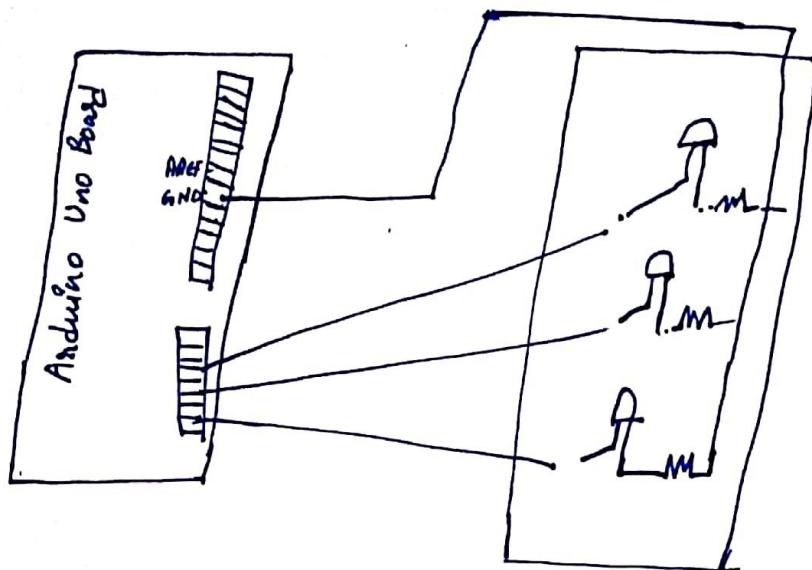


## WRITE-UP

Program 2

Traffic Light

Sai Sriram V  
1BM18CS140



```
void setup() {  
    Serial.begin(9600);  
    pinMode(13, output);  
    pinMode(12, output);  
    pinMode(11, output);  
}  
  
void loop() {  
    digitalWrite(13, HIGH);  
    digitalWrite(12, LOW);  
    digitalWrite(11, LOW);  
    delay(5000);  
  
    digitalWrite(13, LOW);  
    digitalWrite(12, HIGH);  
    digitalWrite(11, LOW);  
    delay(5000);  
  
    digitalWrite(13, LOW);  
    digitalWrite(12, HIGH);  
    digitalWrite(11, HIGH);  
    delay(5000);  
}
```

## 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.

**Name – Sai Sriram Vemparala**

Program No. – 03

Program Title – LED fading without potentiometer

---

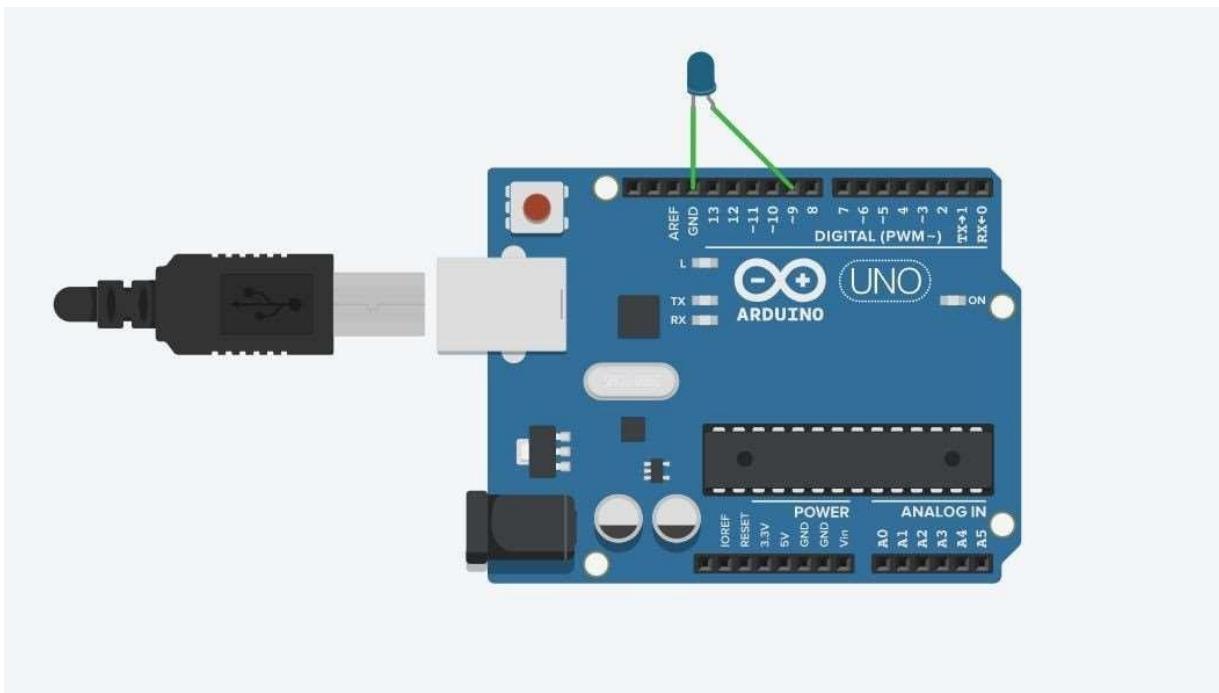
## **AIM**

Demonstrate to show LED fading.

## **HARDWARES REQUIRED**

- Arduino Board
- LED bulb

## **CIRCUIT DIAGRAM**



## WRITE-UP

PROGRAM No:- 3

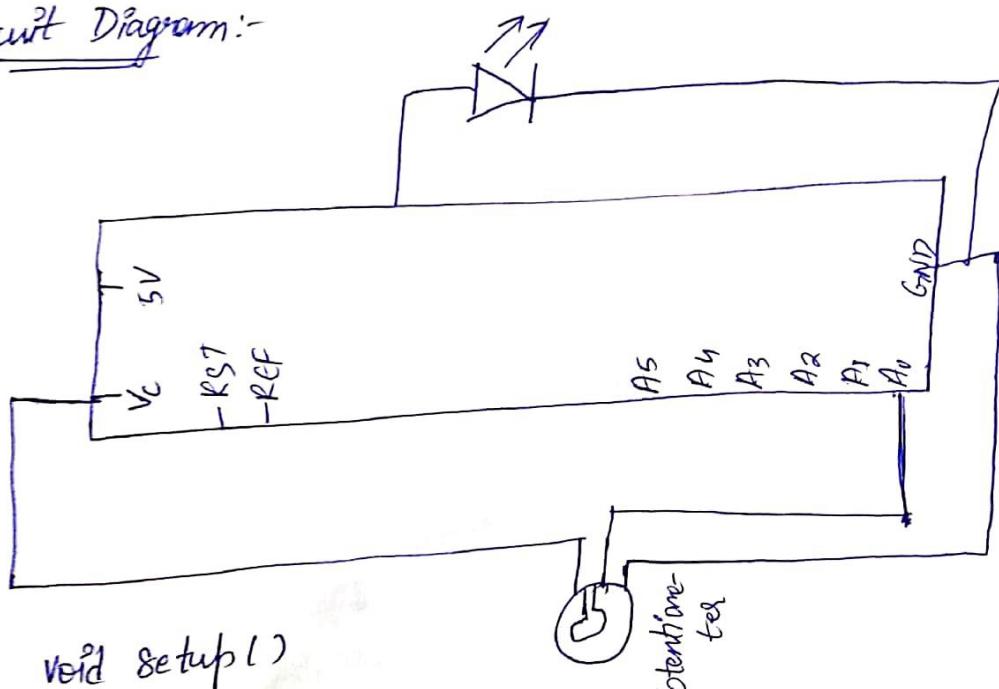
PROGRAM TITLE :- FADING LED USING POTENTIOMETER

Sai Sriram.V  
13M18CS140

AIM:- Demonstrate to show LED fading (Analog output)

Hardware Requirements:- ARDUINO Board, LCD, connecting wire, Resistor.

Circuit Diagram:-



Code:-

```
void setup() {  
  Serial.begin(9600);  
  pinMode(9, OUTPUT);  
}  
  
void loop() {  
  int analog = analogRead(A0);  
  int brightness = map(analog, 0, 1023, 0, 255);  
  analogWrite(9, brightness);  
  Serial.print("Analog value ");  
  Serial.print(analog);  
  Serial.print("\n Brightness value ");  
  Serial.print(brightness);  
}
```

?

③

## 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.

**Name – Sai Sriram Vemparala**

Program No. – 04

Program Title – LED fading with potentiometer

---

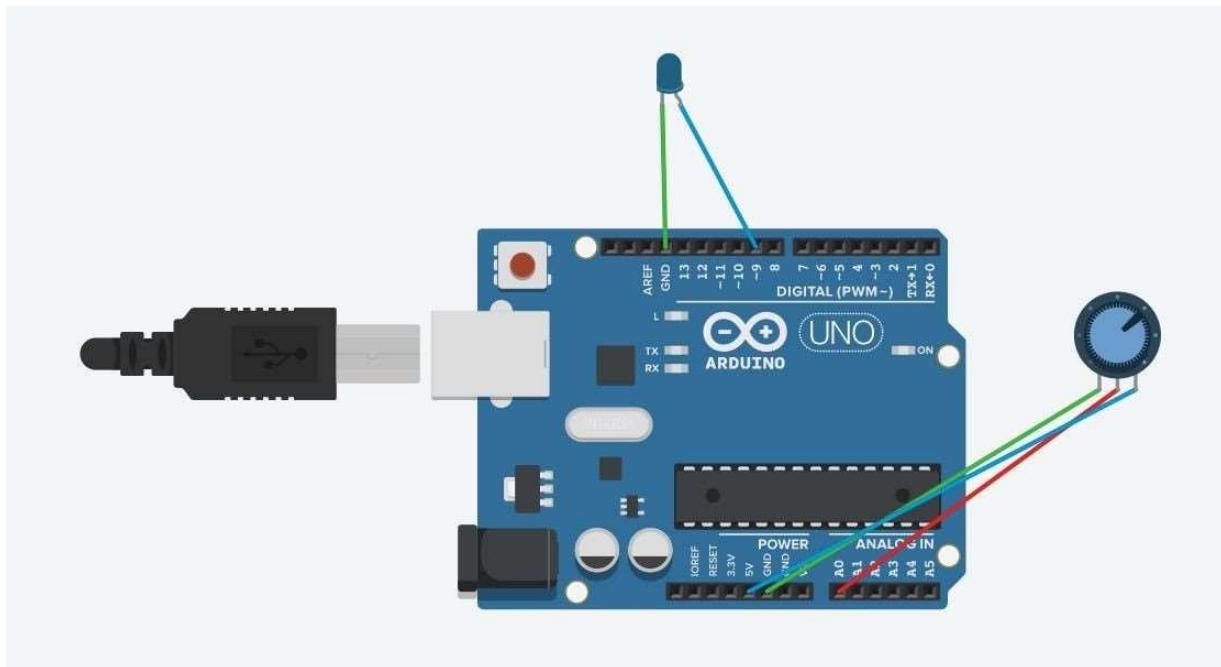
## **AIM**

Demonstrate to show LED fading(analog output).

## **HARDWARES REQUIRED**

- Arduino Board
- LED bulb
- Potentiometer

## **CIRCUIT DIAGRAM**



## WRITE-UP

Sai Sriram. V  
IBM I&C S14D

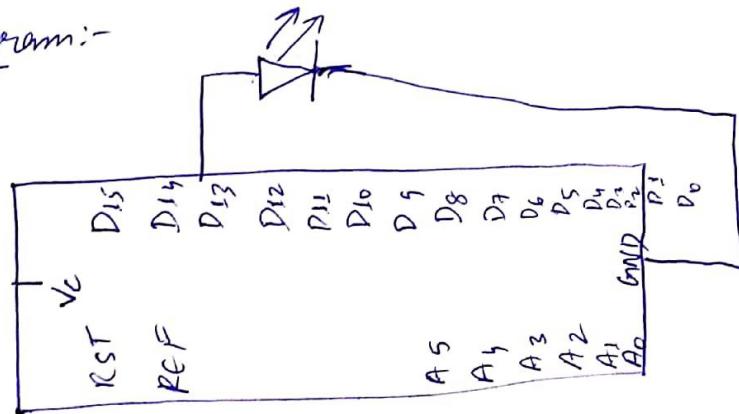
Program No. 4

Program Title : Fading LED

Aim:- Demonstrate to show Fading LED.

HARDWARE REQUIRED:- Arduino Board, LED, Connecting wire, Resistor.

Circuit Diagram:-



CODE:-

```
int brightness=0;
void setup()
{
    pinMode(11, OUTPUT);
}

void loop()
{
    for(brightness=0; brightness<=255; brightness+=2)
    {
        analogWrite(11, brightness);
        delay(30);
    }
    for(brightness=255; brightness>=0; brightness-=2)
    {
        analogWrite(11, brightness);
        delay(30);
    }
}
```

(Q)

## CODE

```
int LED_PIN = 9;

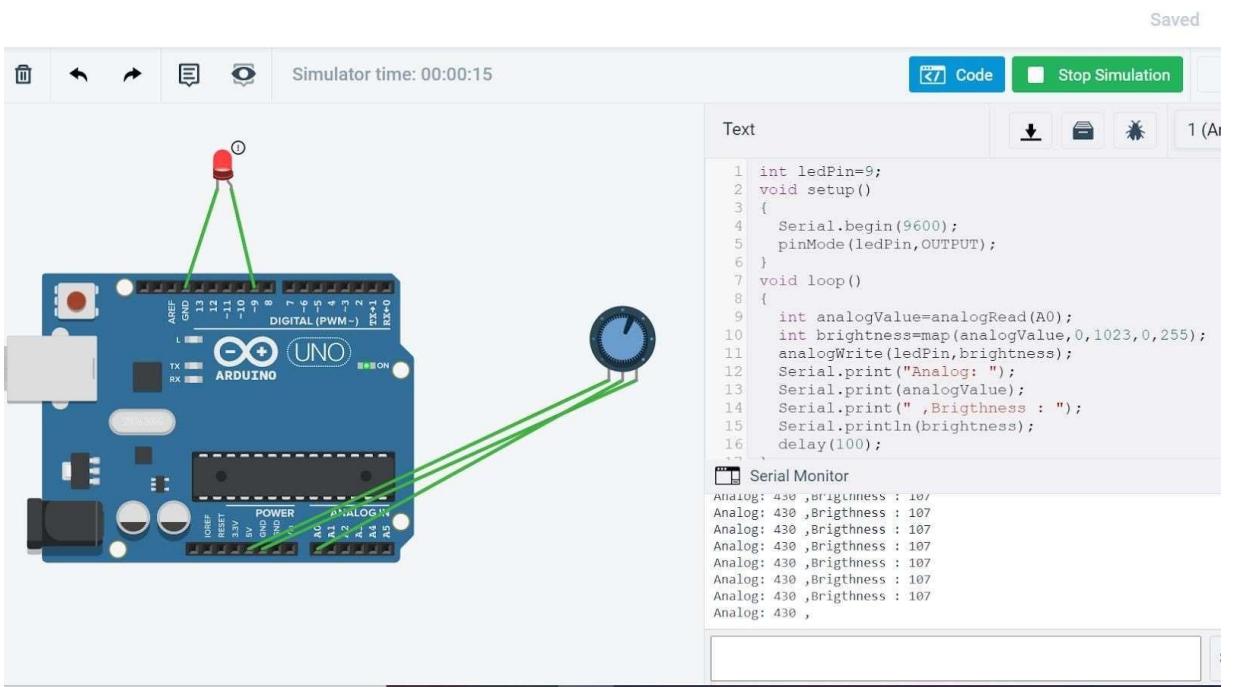
void setup()
{
    Serial.begin(9600);
    pinMode(LED_PIN, OUTPUT);
}

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);
```

```
    delay(100);  
}
```

## OUTPUT

## Fading of LED with potentiometer.



**Name – Sai Sriram Vemparala**

Program No. – 05

Program Title – ON/OFF LED using Push Button

---

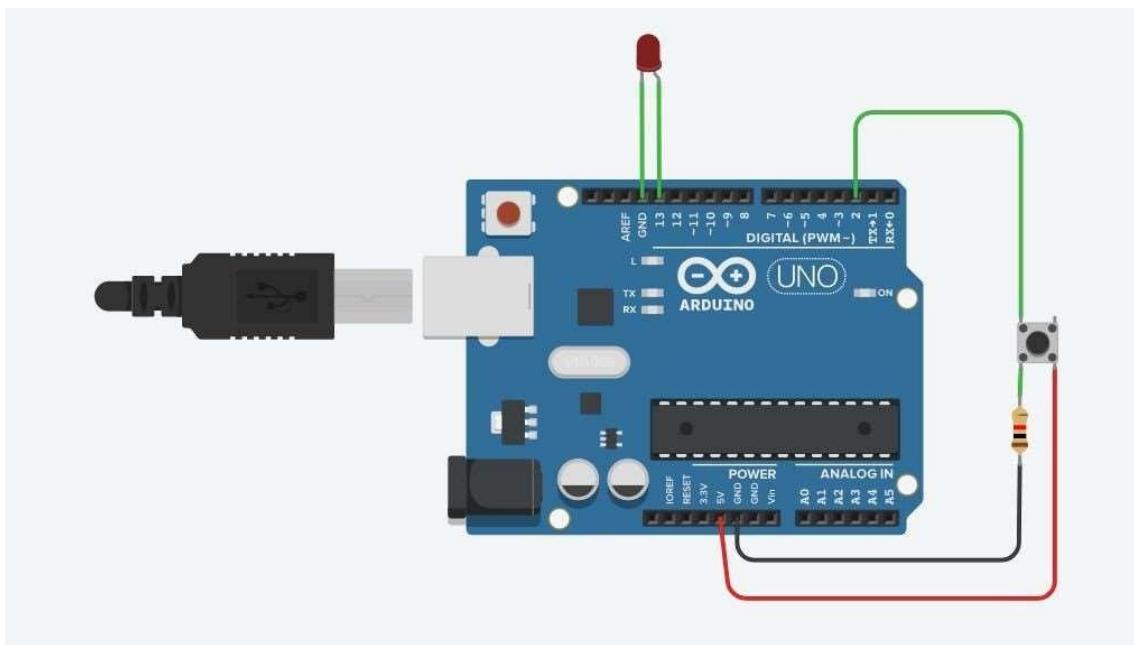
## **AIM**

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

## **HARDWARES REQUIRED**

- Arduino Board
- LED bulb
- Push Button
- Resistor

## **CIRCUIT DIAGRAM**



## WRITE-UP

Program No: 5

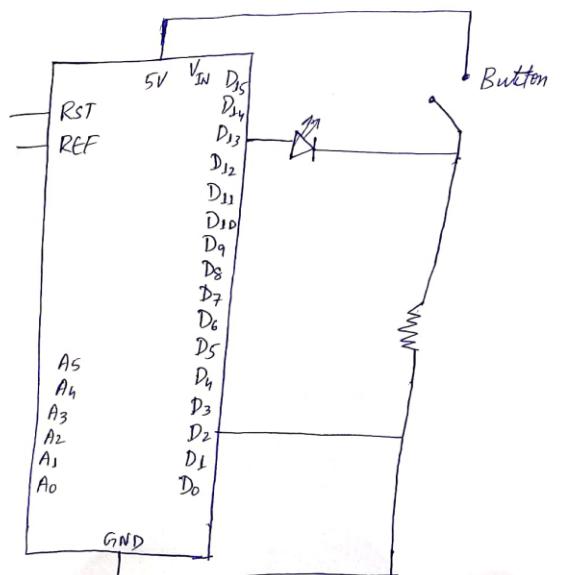
Sai Sudham V  
1BM18CS140

Program Title: LED Using Push Button

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

Hardware Requirement: Arduino Board, LCD, Push Button, Resistor.

Circuit Diagram:



(5)

CODE:-

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

(6)

## **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).

**Name –Sai Sriram**

**Vemparala Program**

No. – 06 Program Title

– LDR

---

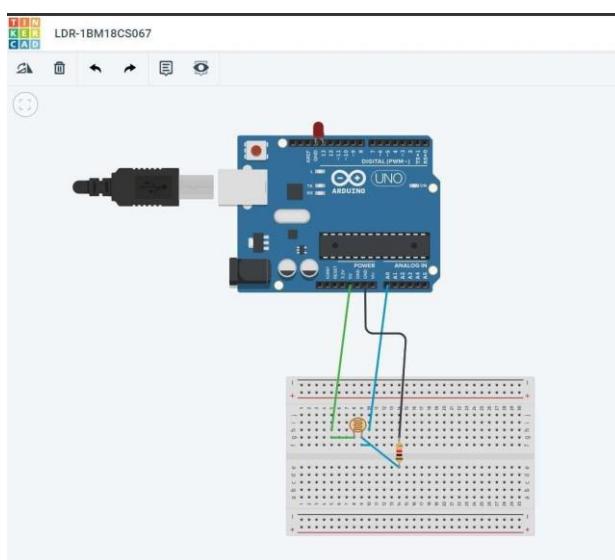
## **AIM**

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

## **HARDWARES REQUIRED**

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

## **CIRCUIT DIAGRAM**



## WRITE-UP

Program No.:-6 - ON/OFF Using LDR

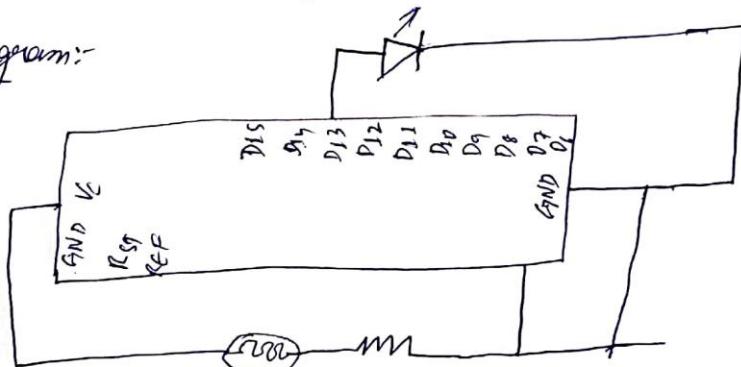
Sai Sriram.V

1BM18CS140

Aim:- Demonstrate to show ON/OFF of a LED using LDR - NIGHT LIGHT SIMULATOR.

Hardware Requirements:- Arduino board led, breadboard, resistor, connecting wire, photoresistor.

Circuit Diagram:-



Code:-

```
const int ledPin = 10;
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("----");
    }
}
```

Sai Sriram

## 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(" -----");

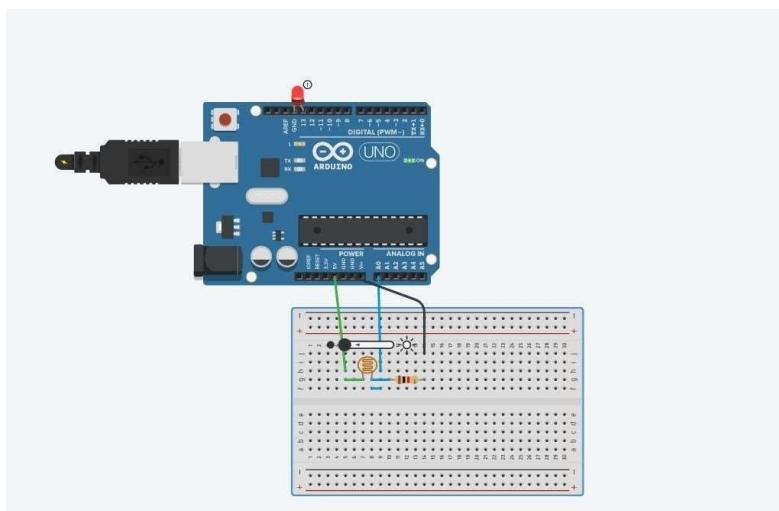
}

}

```

## OUTPUT

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



**Name – Sai Sriram**

**Vemparala Program No.**

– 07 Program Title – PIR

---

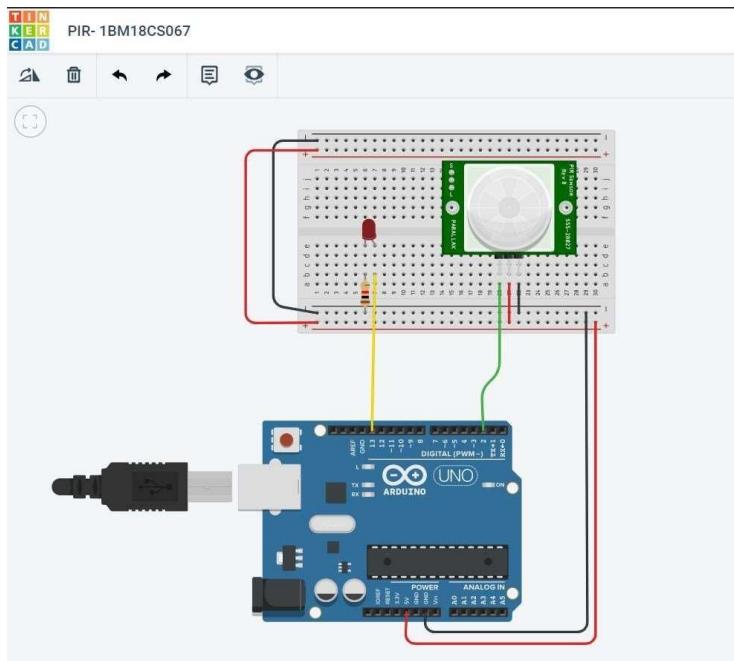
## **AIM**

Demonstrate to show working of PIR sensor.

## **HARDWARES REQUIRED**

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

## **CIRCUIT DIAGRAM**



## WRITE-UP

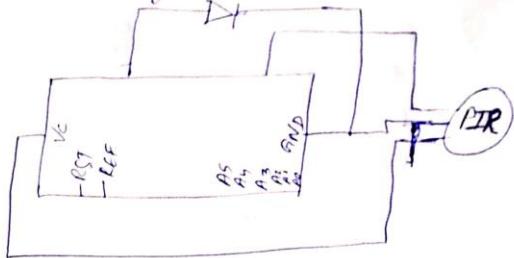
Program No. 7 → PIR SENSOR

828bit.com.V  
1BMHCS160

Aim:- Demonstrate the working of a passive infrared sensor.

Hardware Requirements:- Arduino Board, LED, resistor, bread board, connecting wires, PIR sensor.

Circuit Diagram:



Code:-

```
int led = 13;  
int sensor = 2;  
int state = LOW;  
int val = 0;  
void setup()  
{  
    pinMode(led, OUTPUT);  
    pinMode(sensor, INPUT);  
    serial.begin(9600);  
}  
  
void loop()  
{  
    val = digitalRead(sensor);  
    if (val == HIGH)  
    {  
        digitalWrite(led, HIGH);  
        delay(10);  
        if (state == LOW)  
        {  
            serial.println("Motion detected!");  
            state = HIGH;  
        }  
    }  
    else  
    {  
        digitalWrite(led, LOW);  
        delay(10);  
        if (state == HIGH)  
        {  
            serial.println("Motion stopped");  
            state = LOW;  
        }  
    }  
}
```

## CODE

```
int sensorState = 0;

void setup()
{
    pinMode(2, INPUT);
    pinMode(13, OUTPUT);
    Serial.begin(9600);
}

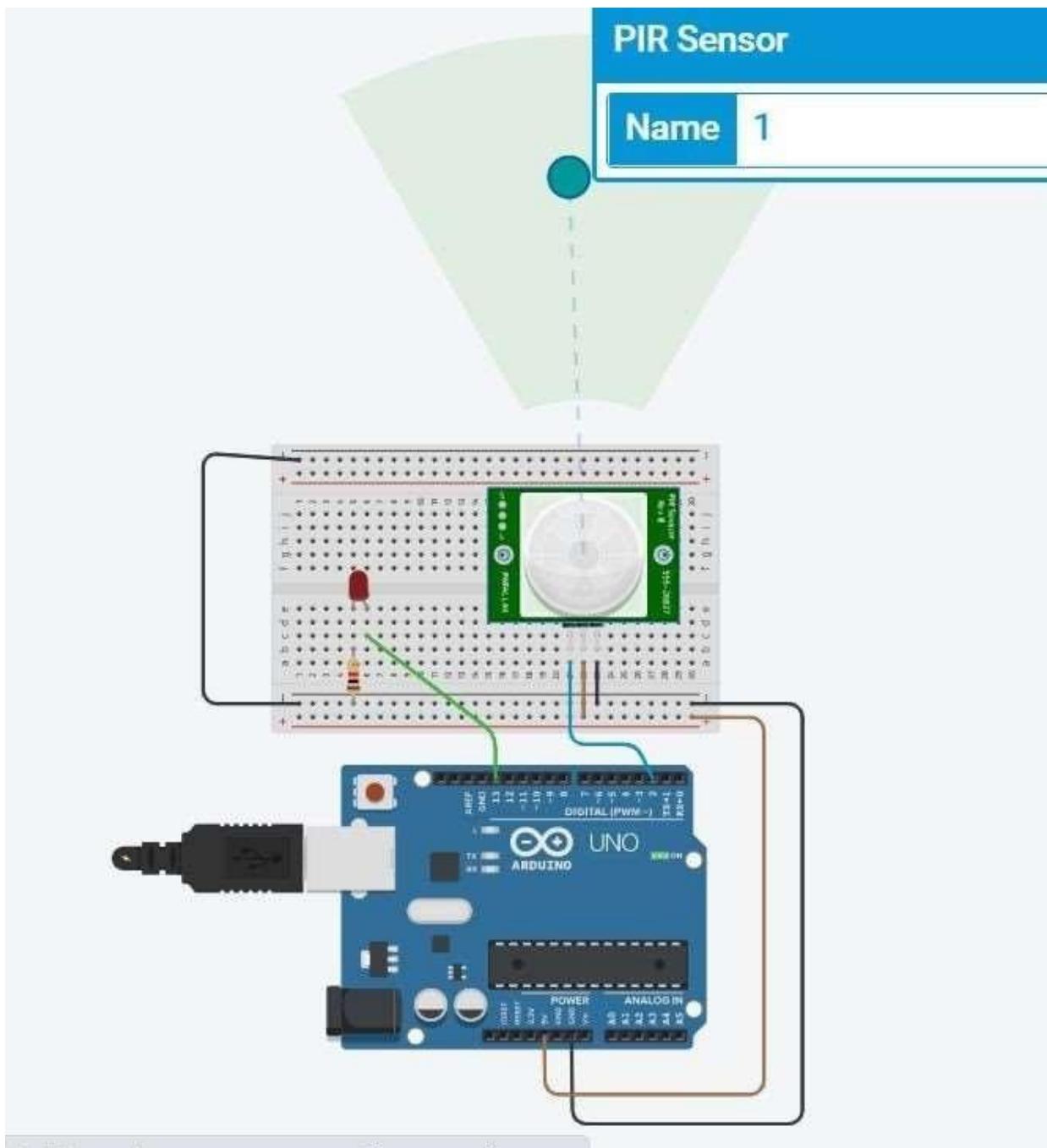
void loop()
{
    sensorState = digitalRead(2);

    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.



**Name – Sai Sriram Vemparala**

Program No. – 08

Program Title – Distance Measurement using ultrasonic sensor

---

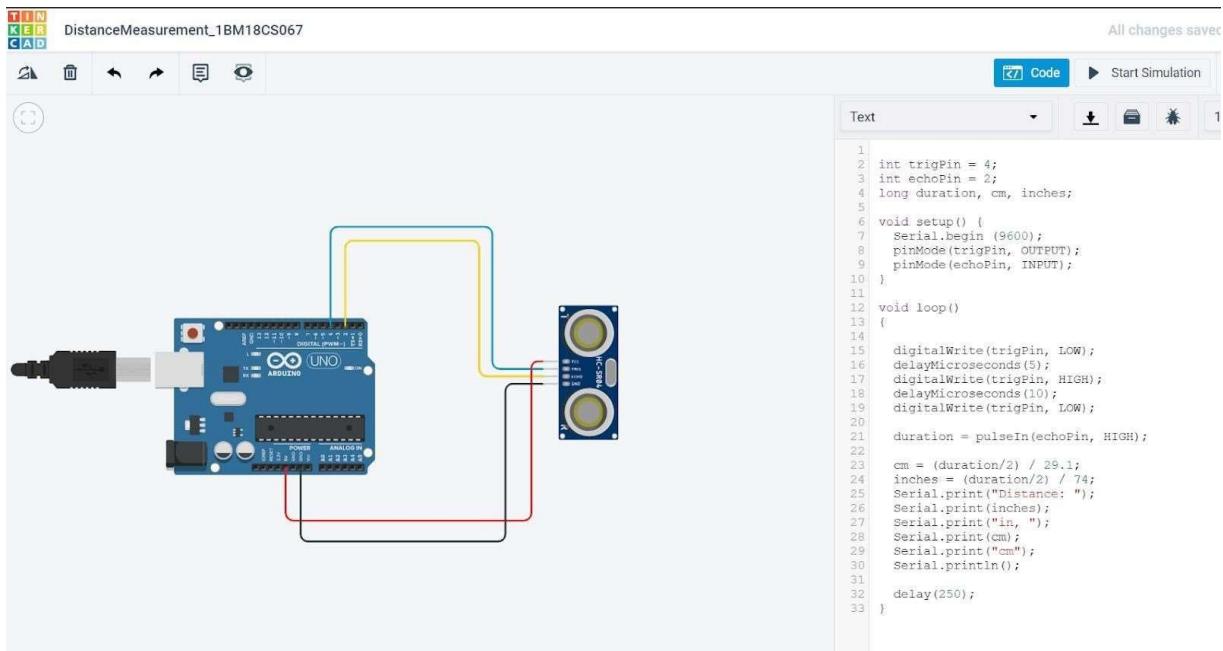
## AIM

Design a system to measure the distance between objects.

## HARDWARES REQUIRED

- Arduino Board
- Ultrasonic sensor HC-SR04

## CIRCUIT DIAGRAM



## WRITE-UP

IOT Program -8

Name: Sri Srinivas V  
USN: JBM18CS140

Program -8 :- OBJECT DISTANCE Measurement

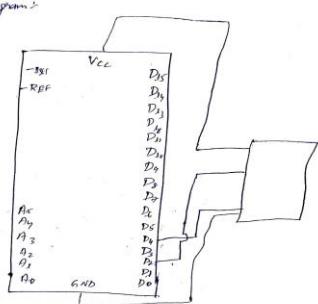
Aim:- Design a system to measure the distance b/w objects.

Hardware Requirements:- → Arduino  
→ Ultrasonic Distance Sensor

Code:-

```
int trigPin = 4;
int echoPin = 2;
long distance, cm, inches;
void setup()
{
    Serial.begin(9600);
    pinMode(trigPin, OUTPUT);
    pinMode(echoPin, INPUT);
}
void loop()
{
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(5);
    digitalWrite(trigPin, LOW);
    duration = pulseIn(echoPin, HIGH);
    cm = (duration/2)/29.1;
    inches = (duration/2)/74;
    Serial.print("Distance : ");
    Serial.print(cm);
    Serial.print("Inches");
    Serial.print(inches);
    Serial.print("cm");
    delay(250);
}
```

Circuit Diagram:-



## **CODE**

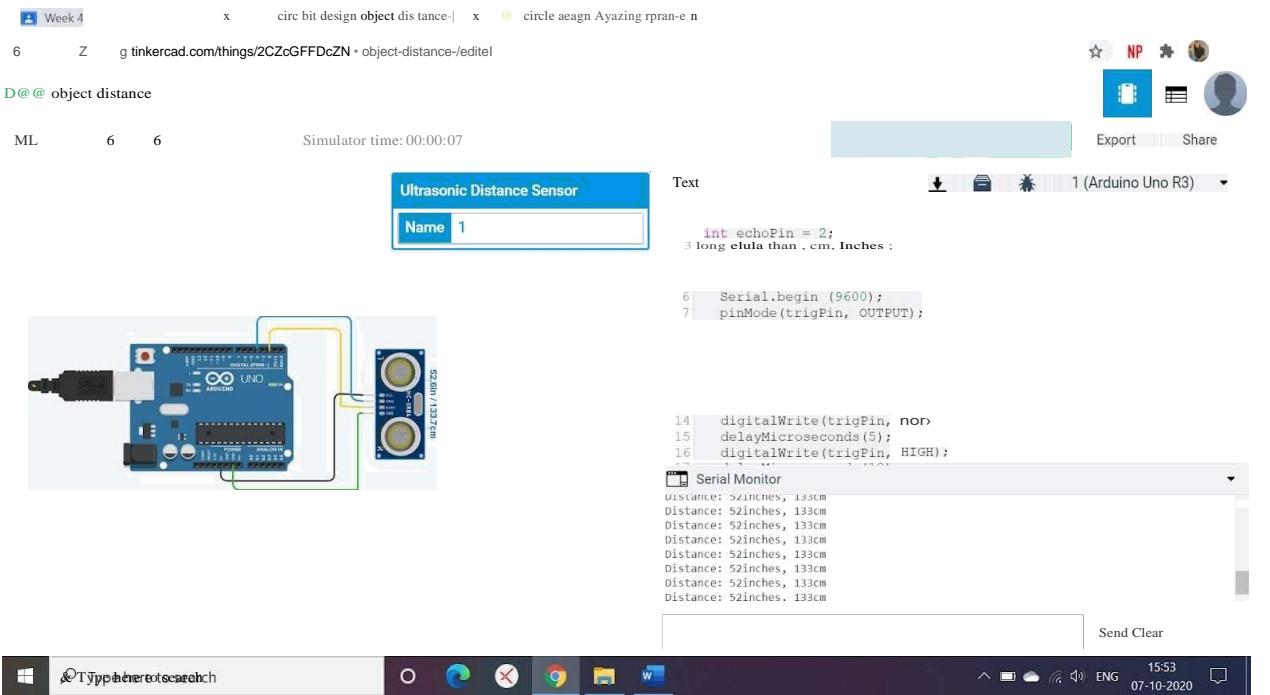
```
int trigPin = 4;  
int echoPin = 2;  
long duration, cm, inches;  
void setup() {  
    Serial.begin (9600);  
    pinMode(trigPin, OUTPUT);  
    pinMode(echoPin, INPUT);  
}  
void loop()  
{
```

```
\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.



**Name –Sai Sriram Vemparala**

Program No. – 09

Program Title – Fire Alarm using flame Sensor

---

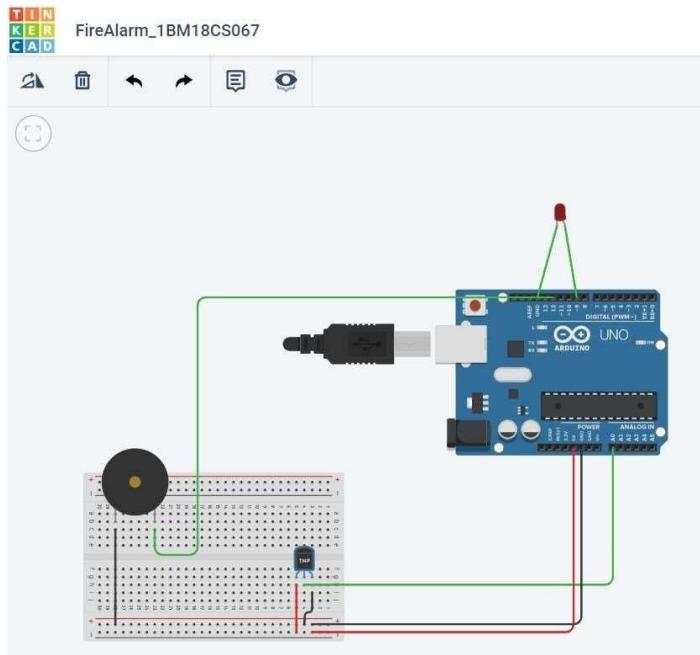
## **AIM**

Design an alert system using a flame sensor.

## **HARDWARES REQUIRED**

- Arduino Board
- Piezo
- Temperature Sensor
- Breadboard small

## **CIRCUIT DIAGRAM**



## WRITE-UP

IOT-Program 9

Sai Sriram.V  
1BM18CS160

Program 9:- Alert System

Aim:- Design an alert system using flame sensor (use temp sensor for experiments in Thinkschool)

Hardware Requirements:-  
→ Arduino  
→ Piezo  
→ Temperature Sensor  
→ LED.

Code:-

```
int temperaturePin = 0;  
int buzzer = 12;  
void setup()  
{  
    serial.begin(9600);  
    pinMode(buzzer, OUTPUT);  
    pinMode(9, OUTPUT);  
}  
  
float getVoltage(int pin)  
{  
    return (analogRead(pin) + 0.004882 * 84);  
}  
  
void loop()  
{  
    float voltage, degrees;  
    voltage = getVoltage(temperaturePin);  
    degreesC = (voltage - 0.5) * 100.0;  
    digitalWrite(9, LOW);  
    if (degreesC > 37)  
    {  
        serial.print(degreesC);  
        serial.println("ALERT!");  
    }  
}
```

```
digitalWrite(buzzer, HIGH);  
digitalWrite(9, HIGH);  
tone(12, 10000, 100);  
delay(200);  
}  
else if (degreesC < 37)  
{  
    serial.print("degrees");  
    serial.print("SANE!");  
    delay(200);  
}
```

## 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)
```

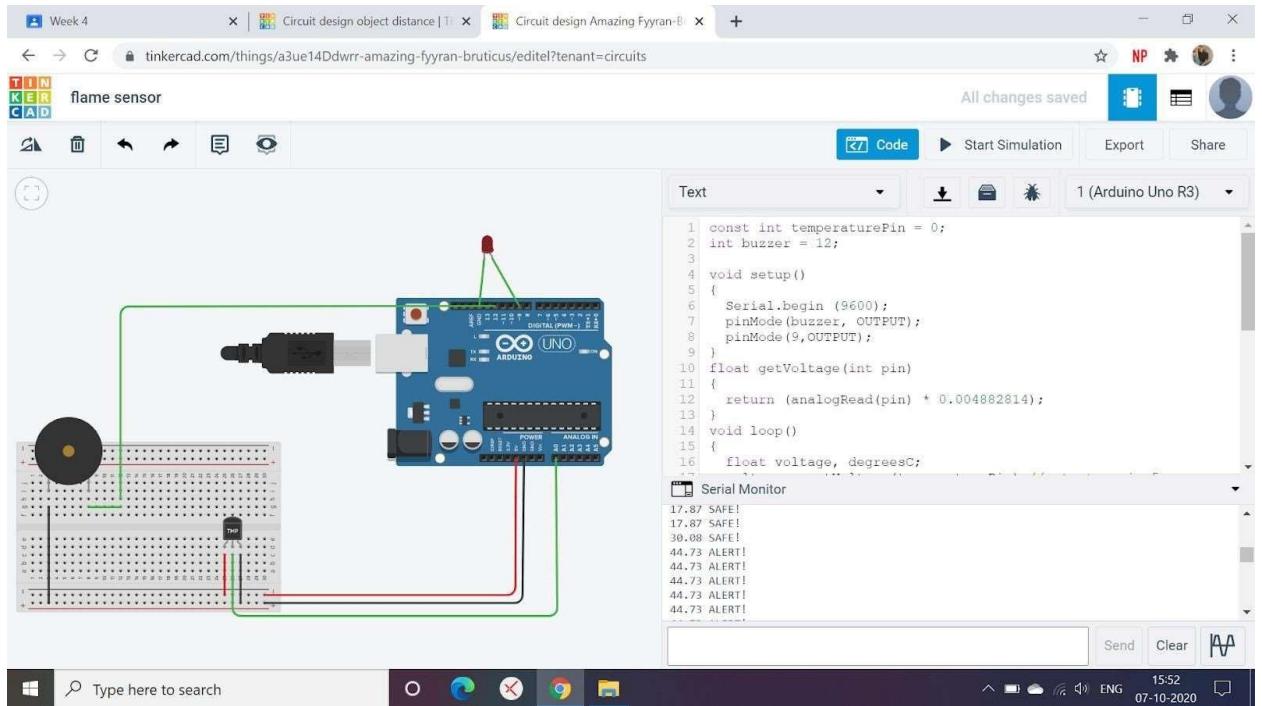
```
{  
    Serial.print(degreesC);  
    Serial.println(" SAFE!");  
}  
  
}
```

```
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.



**Name – Sai Sriram Vemparala**

Program No. – 10

Program Title – Gas Sensor

---

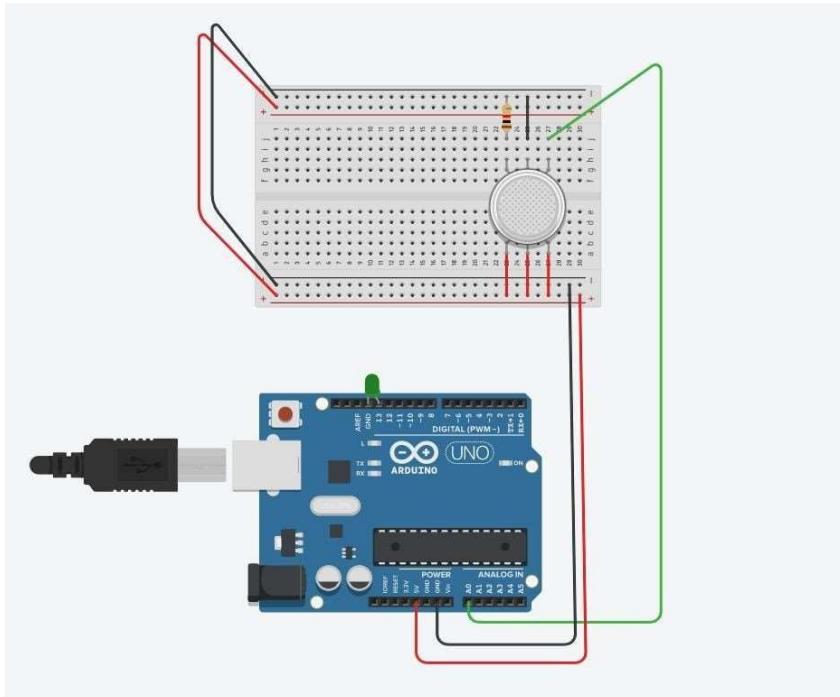
## **AIM**

To design a smart gas leakage indicator system.

## **HARDWARES REQUIRED**

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

## **CIRCUIT DIAGRAM**



## **WRITE-UP**

IOT - Program No. 10

Sai Sriram V  
JBM18CS160

GAS Sensor [ Smart Gas leakage Indicator ]

- Aim :- Design a smart gas leakage indicator system (gas sensor and LED).

Hardware Requirements :-

- Arduino
- Gas Sensor
- Bread Board
- Resistor
- LED.

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("Sensor Value : ");  
        Serial.println("Gas Detected");  
        delay(sensorValue);  
    }  
    else {  
        digitalWrite(led, LOW);  
        Serial.println("Sensor Value : ");  
        Serial.println("No Gas Detected");  
        delay(1000);  
    }  
}
```

W. Sri Ram

## 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);

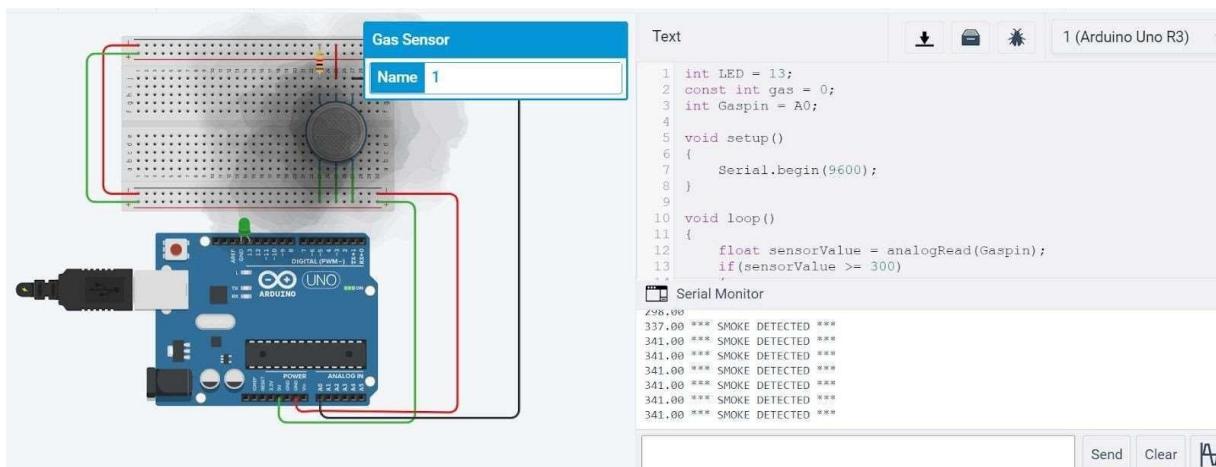
}

delay(1000);

}
```

## OUTPUT

Designed a smart gas leakage indicator system.



## **Name – Sai Sriram Vemparala**

## Program No. – 11

## Program Title – Vibration motor and LDR

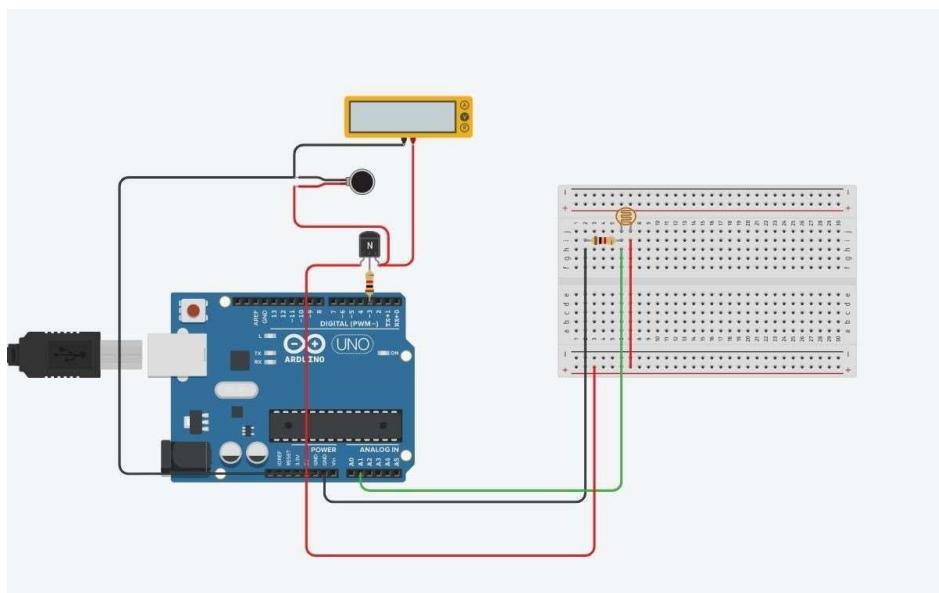
AIM

To design an automated day indicator system.

## **HARDWARES REQUIRED**

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

## CIRCUIT DIAGRAM



## **WRITE-UP**

IOT Program - II

Sap program.v

IBMSICS140

Program Title :- Day indicator system.

Aim :- Design an automated day indicator (Vibration motor and LDR).

Hardware Required:

- Arduino
- BreadBoard
- LDR
- NPN transistor
- Multimeter
- Resistors

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);  
  }  
}
```

*(Ankitam)*

## 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

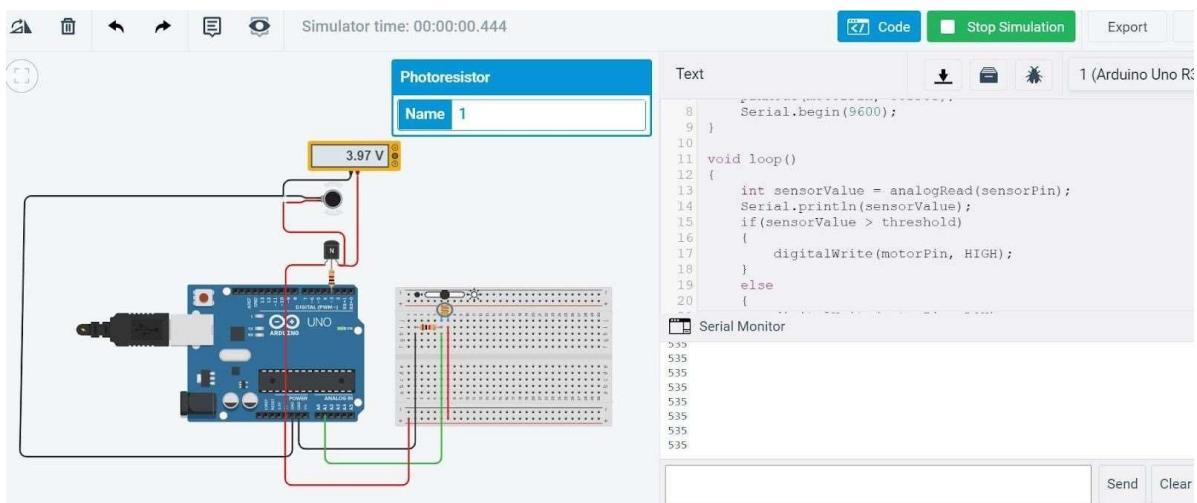
{



digitalWrite(motorPin, LOW);
```

## OUTPUT

Designed an automated day indicator system.



**Name – Sai Sriram**

**Vemparala Program No.**

– 12 Program Title – Tilt

Sensor

---

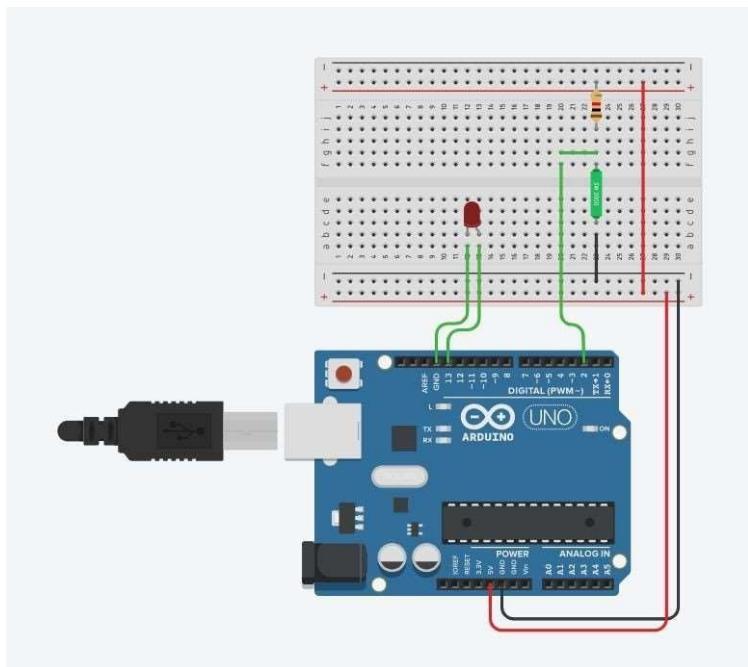
## **AIM**

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

## **HARDWARES REQUIRED**

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

## **CIRCUIT DIAGRAM**



## **WRITE-UP**

IOT Program - 12

Sai Sriram .V

1BM18CS140

Program Title :- TILT sensor .

AIM :- Design a smart package handling system  
(Tilt sensor and LED).

Hardware Required:- → Arduino Board  
→ Bread board.  
→ LEDs  
→ Tilt sensor  
→ Resistor

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);
}
```

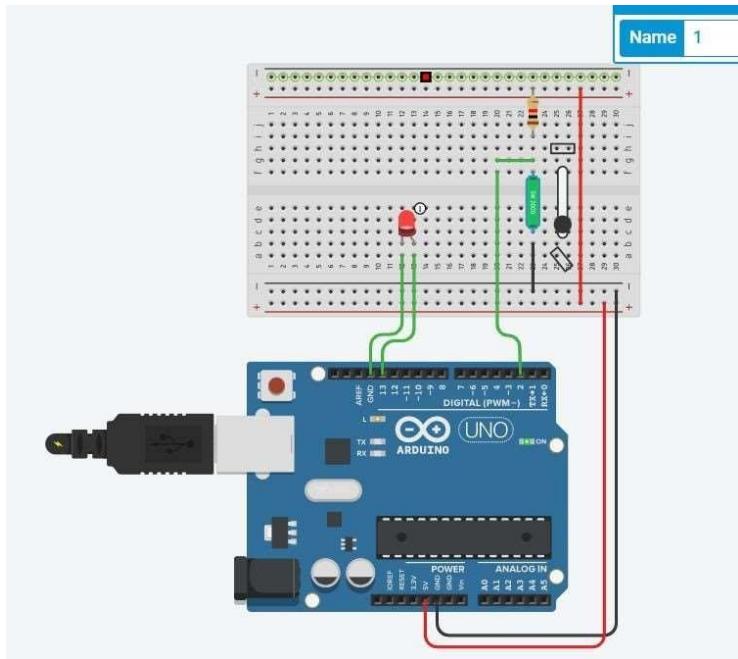
Sai Sriram

## 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.



**Name – Sai Sriram Vemparala**

Program No. – 13

Program Title – IR based SERVO Motor controller

---

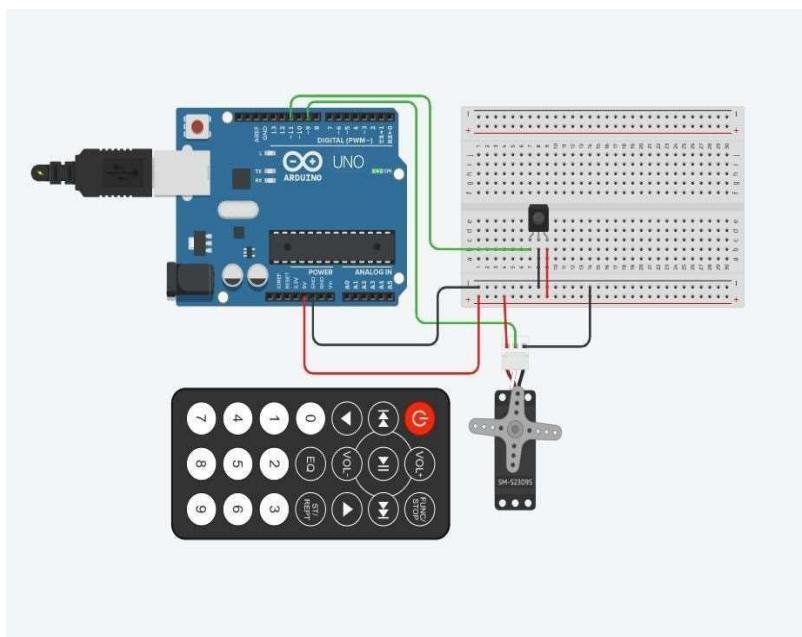
## **AIM**

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**



## WRITE-UP

JOT Program 13

Srinivas V  
IBM SCS 11e

Title :- IR based Servo motor controller

Aim :- Design IR based servo motor controller. (Determine clockwise and counter-clockwise rotation of shaft).

Hardware Requirement :-  
→ Arduino Board  
→ Breadboard  
→ IR Sensor  
→ IR Remote  
→ Micro Servo.

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()

P.T.O

```
void loop()
{
  if (irrecv.decode(&results))
  {
    switch (results.value)
    {
      case 0xFOODFF:
        myServo.attach(9);
        Serial.println("start...");
        break;
      case 0xFD20DE:
        myServo.write(-360);
        Serial.println("counter clockwise");
        break;
      default:
        Serial.print("Unrecognized code received
          : 0x");
        Serial.println(results.value, HEX);
        break;
    }
    irrecv.resume();
  }
}
```

## 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.



The screenshot shows the Arduino Serial Monitor window. The title bar says "Serial Monitor". The main area displays the following text:

```
Starting..
Clockwise..
Clockwise..
Counter Clockwise..
Counter Clockwise..
Unrecognized code received: 0xFD48B7
```

**Name – Sai Sriram Vemparala**

**Program No. – 14**

**Program Title – RGB Led and LCD**

---

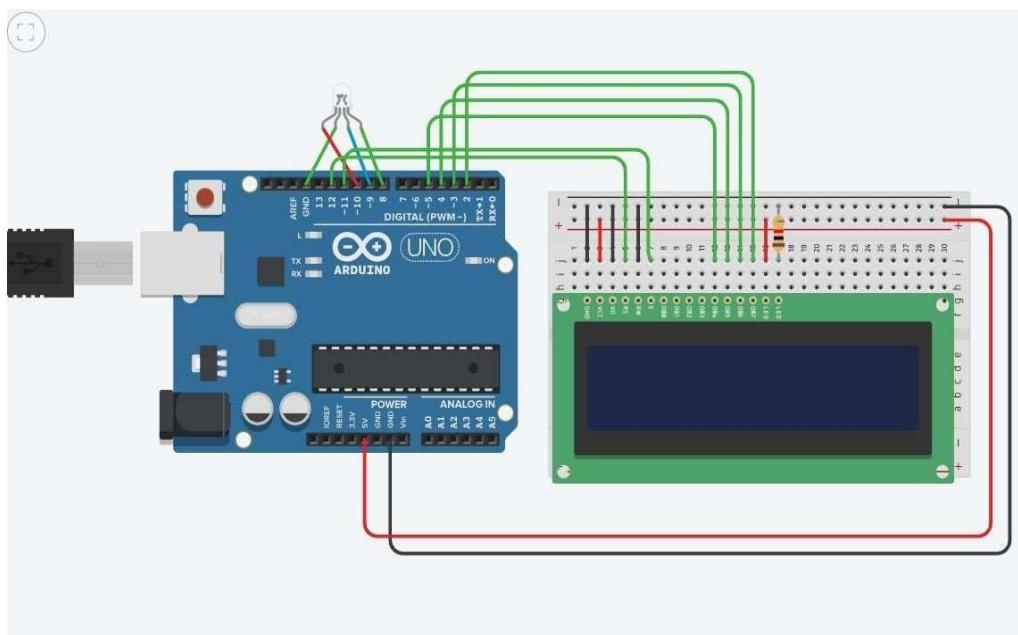
## **AIM**

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**



## WRITE-UP

IOT Program 14

Sai Sriram V  
1BM18CS110

Program Title:- RGB LED and LCD

AIM:- Design a display system to print the RED BLUE and green colors (RGB led and LCD).

Hardware required:-

- Arduino Board
- Breadboard Small
- LCD 16x2
- RGB LED
- RGB Register.

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()
{
  led.setRGB(0,0);
  led.print("RGB Color Print!");
  delay(1000);
  led.clear();
}
```

V. Balaji  
10

Sai Sriram V  
1BM18CS110

```
RGB_color(255,0,0);
led.print("RED");
delay(1000);
led.clear();
```

```
RGB_color(0,255,0);
led.print("GREEN");
delay(1000);
led.clear();
```

```
RGB_color(0,0,255);
led.print("BLUE");
delay(1000);
led.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);
}
```

## 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);

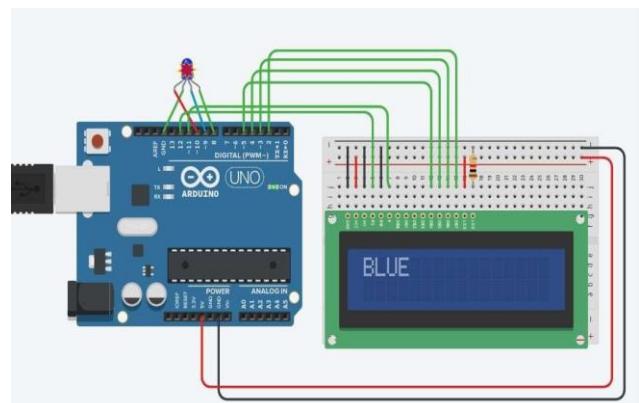
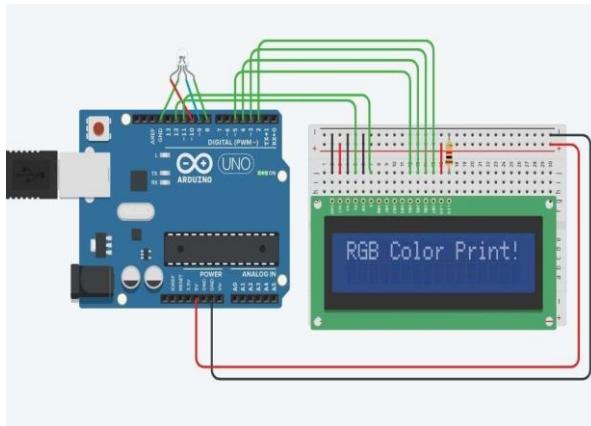
analogWrite(green,green_value);

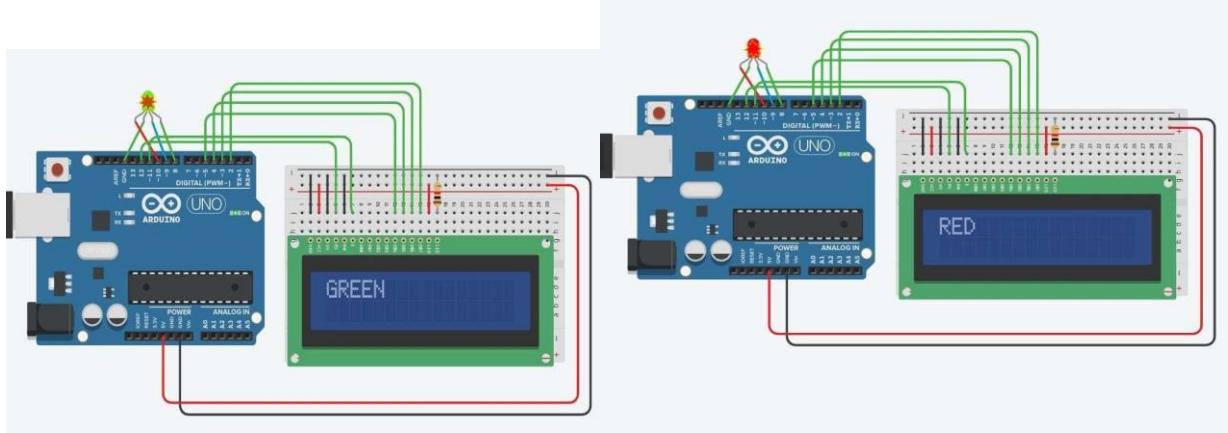
analogWrite(blue,blue_value);
}

```

## OUTPUT

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





**Name – Sai Sriram Vemparala**

Program No. – 16

Program Title – **Smart irrigation system**

---

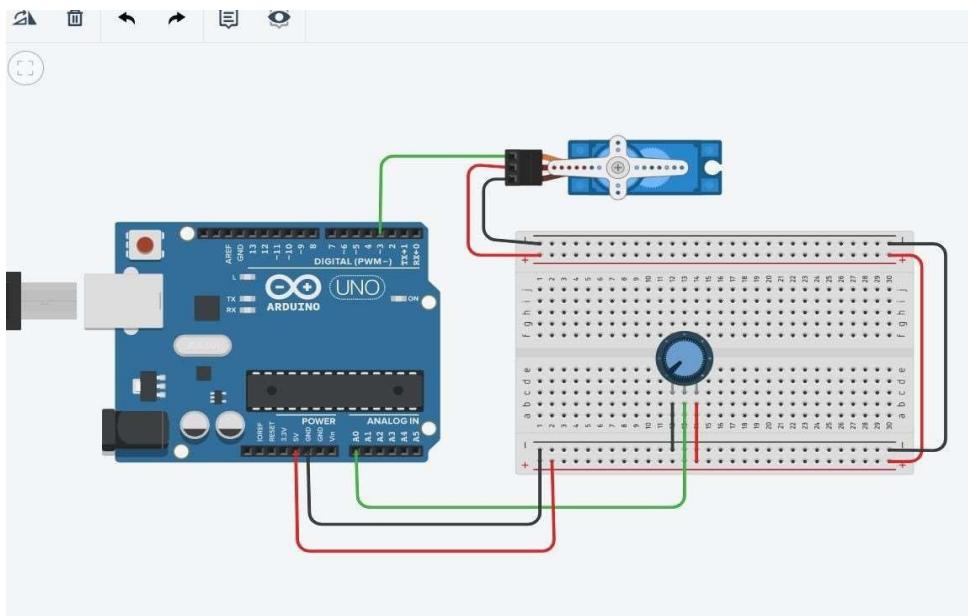
## **AIM**

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

## **HARDWARES REQUIRED**

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

## **CIRCUIT DIAGRAM**



## WRITE-UP

J.7 Program 15	Program 1	Program 16
<u>Program 7.16</u> Smart Navigation System	Program 1	Program 16
After Design a smart navigation system (Infrared sensor, servo motor shaft)		
<u>Hardware Requirements</u>	→ Arduino → Breadboard → Infrared sensor → Servo motor shaft	
<u>Code:</u>	#include <Servo.h>  Servo myservo;  int pos = 0; int sensorPin = A0; int sensorValue = 0;  void setup() { myservo.attach(3); Serial.begin(9600); }  void loop() { sensorValue = analogRead(sensorPin); Serial.println(sensorValue); if (sensorValue > 500) if (pos < 180) pos++; myservo.write(pos); delay(15); }	for (pos=180; pos=0; pos=0) { myservo.write(pos); delay(15); }  delay(1000); }

## 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 0 degrees
            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).

