

IOT LAB REPORT

Name – Suman kumar pal

USN - 1BM18CS111

Program No. – 01

Program Title – LED Blinking

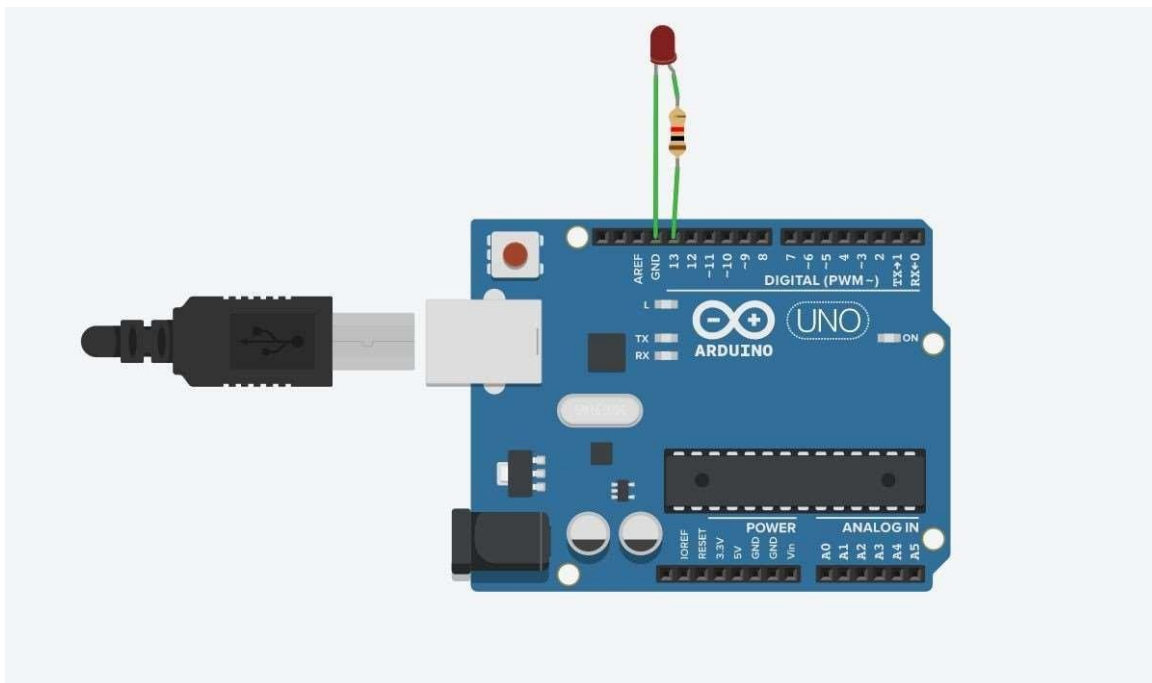
AIM

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

HARDWARES REQUIRED

- Arduino Board
- LEDs

CIRCUIT DIAGRAM



WRITE-UP

IOT

SUMAN KUMAR PAL
18M18CS111

Blink:

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

void loop()
{
    digitalWrite(13, HIGH);
    delay(1000);
    digitalWrite(13, LOW);
    delay(1000);
}
```

B

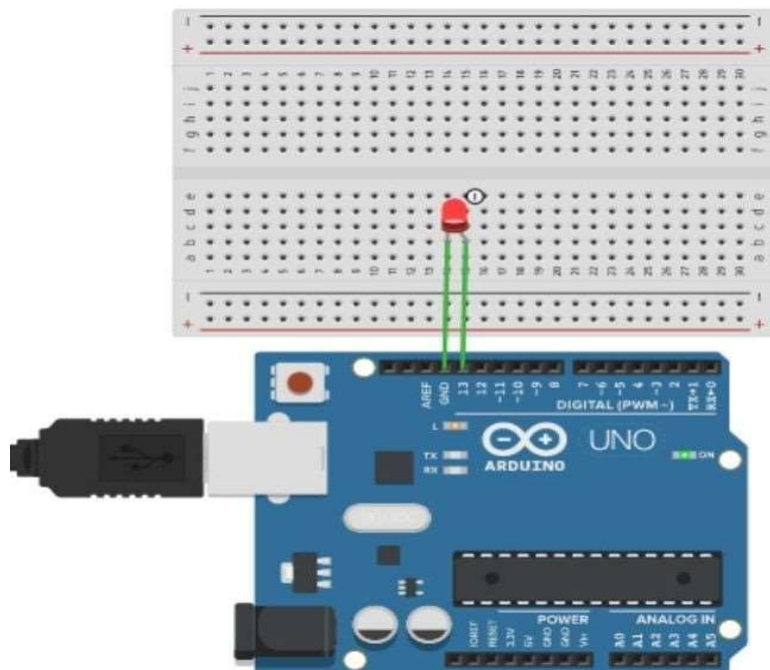
CODE

```
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

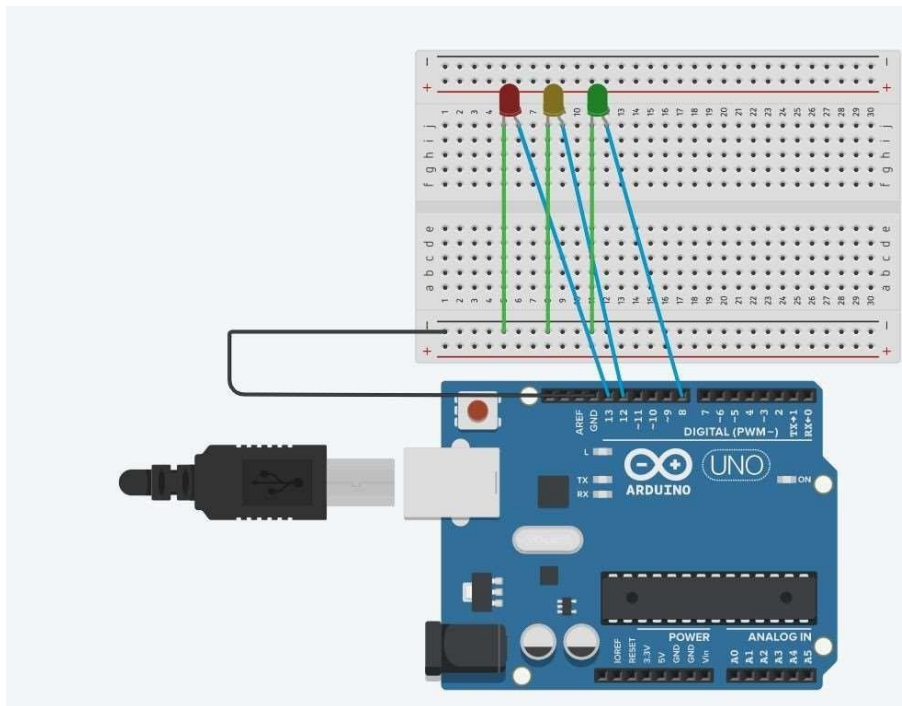
AIM

Traffic Signal Simulator.

HARDWARES REQUIRED

- Arduino Board
- LEDs
- Breadboard

CIRCUIT DIAGRAM



WRITE-UP

IOT

SUMAN KUMAR Patel
18M18CS111

Traffic Light

```
int Green = 2 ;
int Yellow = 3 ;
int Red = 4 ;
int DelayGreen = 1000 ;
int DelayYellow = 1000 ;
int DelayRed = 1000 ;

void Setup()
{
  pinMode (Green, Output);
  pinMode (Yellow, Output);
  pinMode (Red, Output);
}

void loop()
{
  Redlight();
  delay (DelayRed);
  Yellowlight();
  delay (DelayYellow);
  Greenlight();
  delay (DelayGreen);
}

void Greenlight()
{
  digitalWrite (Green, High);
  digitalWrite (Yellow, Low);
  digitalWrite (Red, Low);
}

void Redlight()
{
  digitalWrite (Red, High);
  digitalWrite (Green, Low);
  digitalWrite (Yellow, Low);
}

void Yellowlight()
{
  digitalWrite (Yellow, High);
  digitalWrite (Red, Low);
  digitalWrite (Green, Low);
}
```

R

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

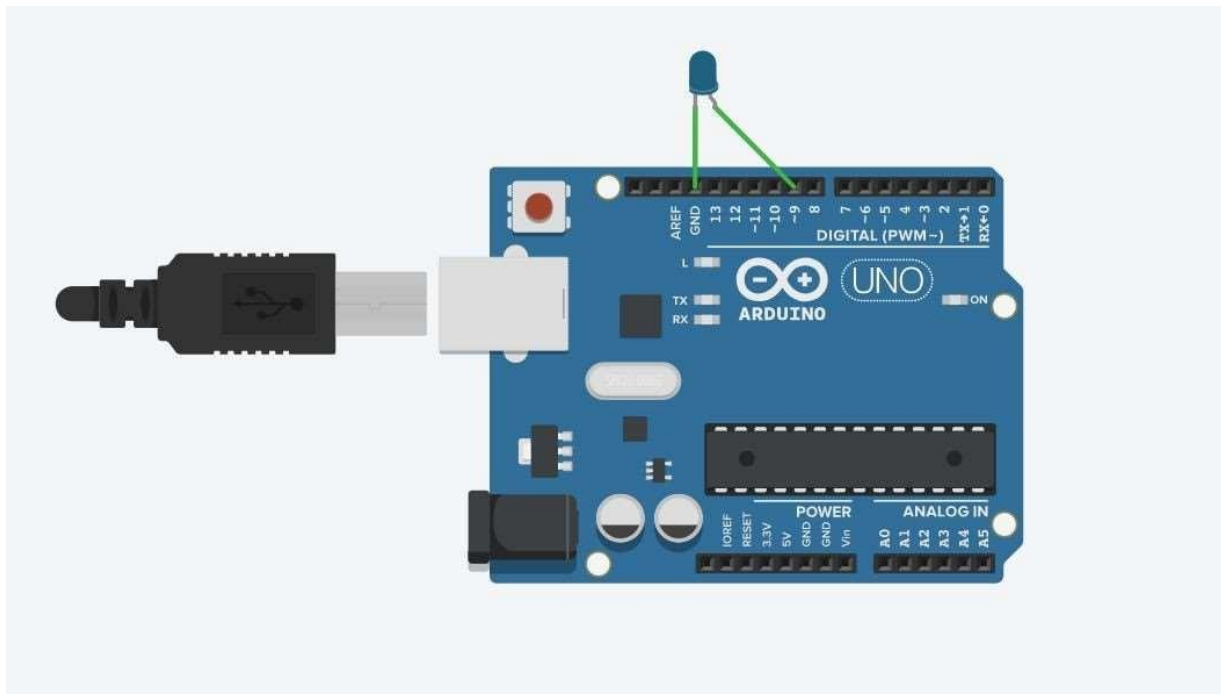
AIM

Demonstrate to show LED fading.

HARDWARES REQUIRED

- Arduino Board
- LED bulb

CIRCUIT DIAGRAM



WRITE-UP

Jade Suman Kumar Patel

```
void setup()
{
    pinMode(2, OUTPUT);
}

void loop()
{
    for (int Jade = 0; Jade <= 255; Jade += 5)
    {
        analogWrite(2, Jade);
        delay(50);
    }

    for (int Jade = 255; Jade >= 0; Jade -= 5)
    {
        analogWrite(2, Jade);
        delay(50);
    }
}
```

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

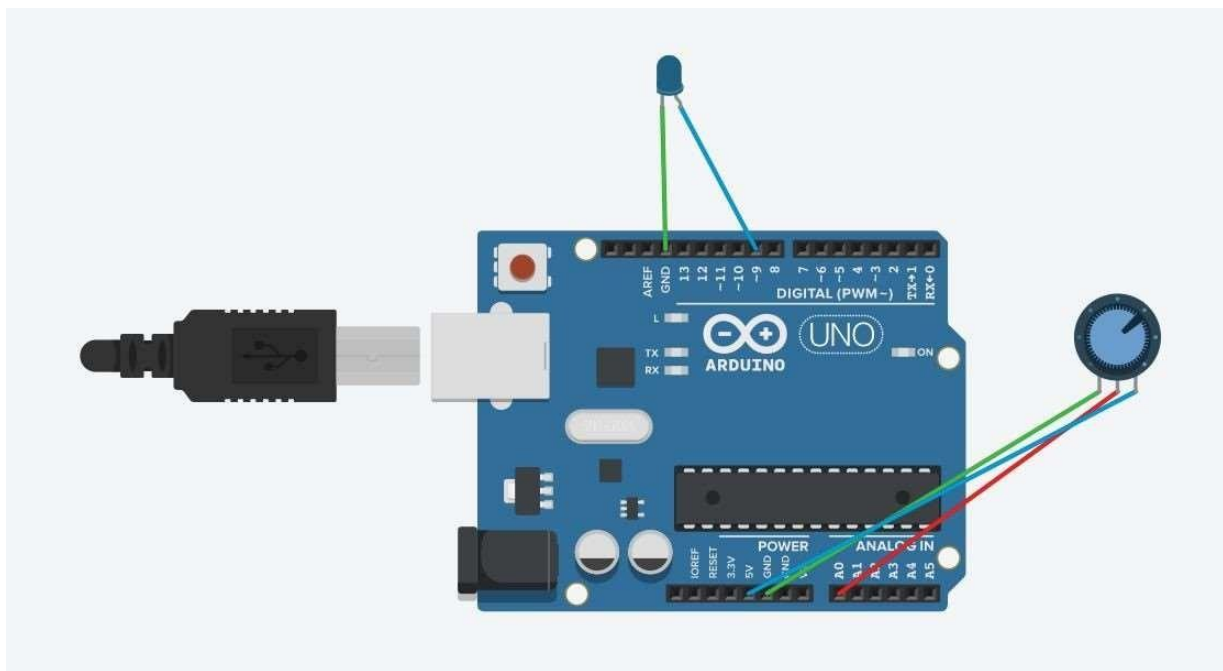
AIM

Demonstrate to show LED fading(analog output).

HARDWARES REQUIRED

- Arduino Board
- LED bulb
- Potentiometer

CIRCUIT DIAGRAM



WRITE-UP

Suman Kumar Pal

Brightness

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("\n analog Value");

Serial.print(analog);

Serial.print("\n brightness Value");

Serial.print(brightness);

}

CODE

```
int LED_PIN = 9;

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

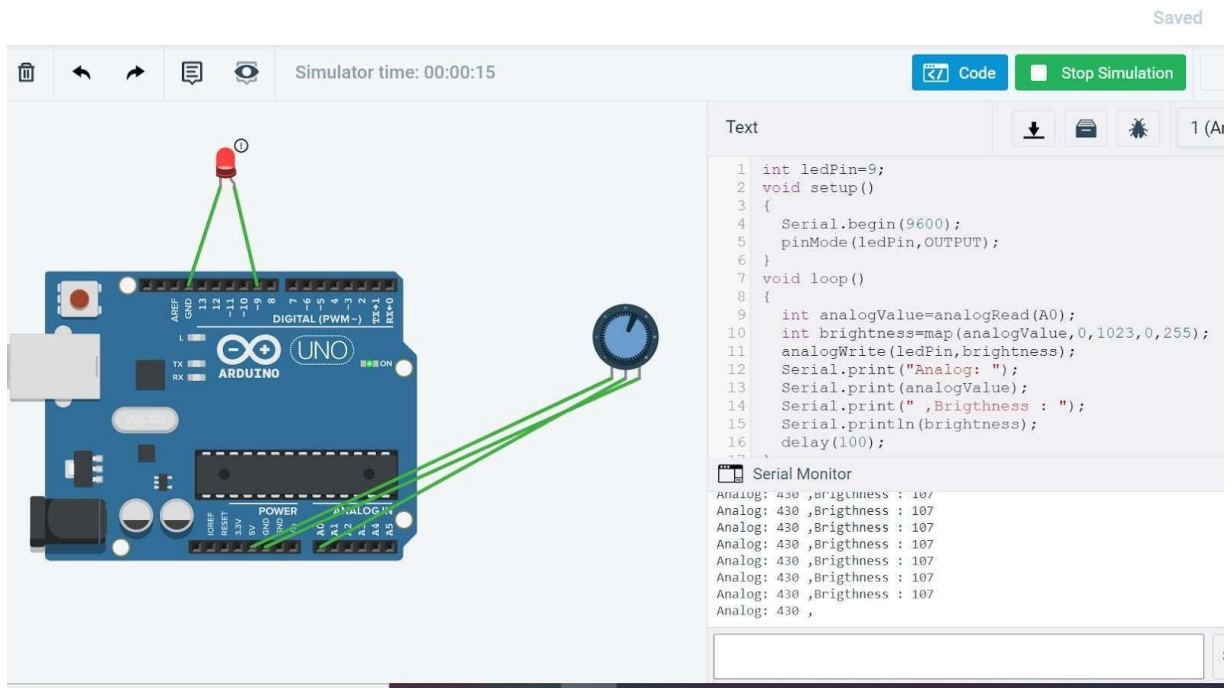
void loop()
{

    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);
}
```

$$\}$$

OUTPUT

Fading of LED with potentiometer.



Program No. – 05

Program Title – ON/OFF LED using Push Button

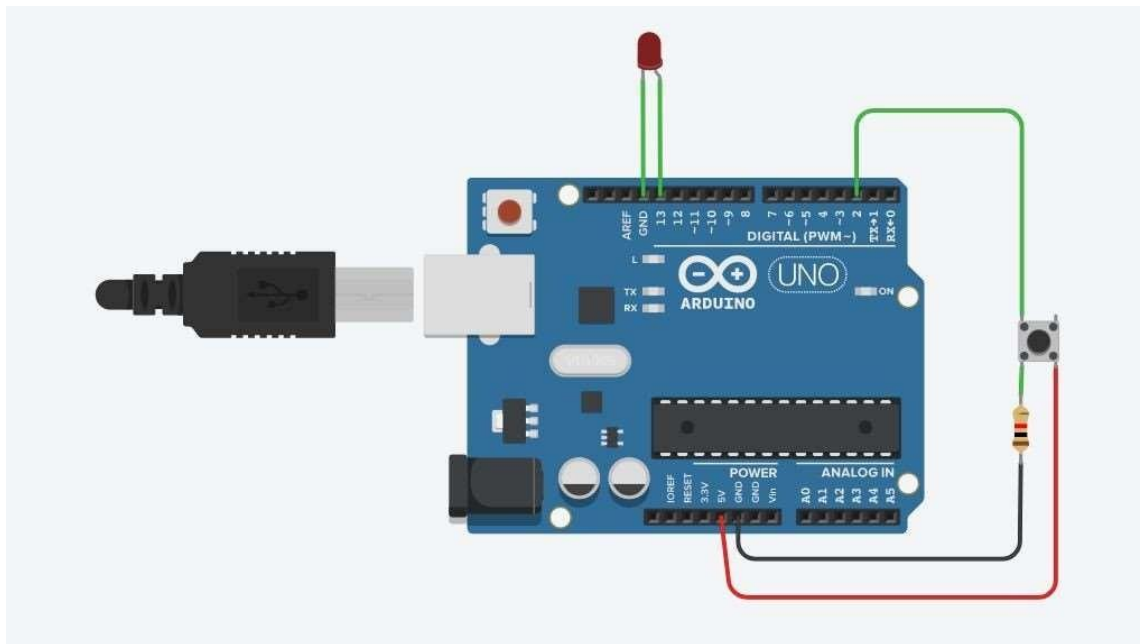
AIM

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

HARDWARES REQUIRED

- Arduino Board
- LED bulb
- Push Button
- Resistor

CIRCUIT DIAGRAM



WRITE-UP

Push Suman Kumar Pal

```
int button State = 0;
void Setup() {
    pinMode(15, OUTPUT);
    pinMode(12, OUTPUT);
}
void loop() {
    button State = digitalRead(2);
    if (button State == HIGH) {
        digitalWrite(15, HIGH);
    }
    else {
        digitalWrite(15, LOW);
    }
}
```

```
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

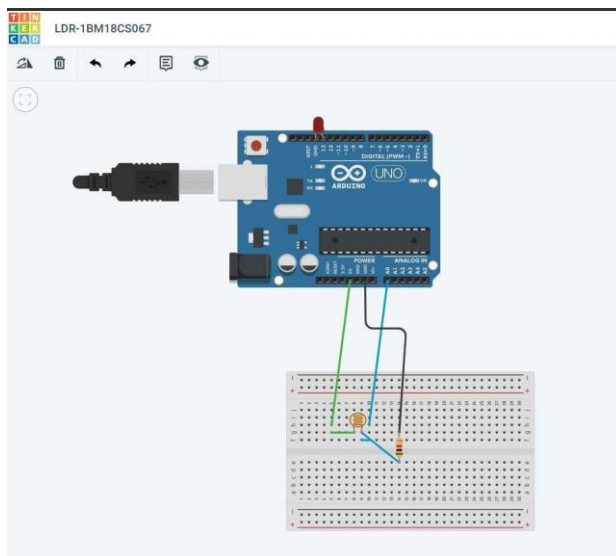
AIM

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

HARDWARES REQUIRED

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

CIRCUIT DIAGRAM



WRITE-UP

```
const int ledPin = 8;
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 Low, on");
    }
    else
    {
        digitalWrite(ledPin, LOW);
        Serial.println("LDR is High, off");
    }
}
```

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)
    {
```

```

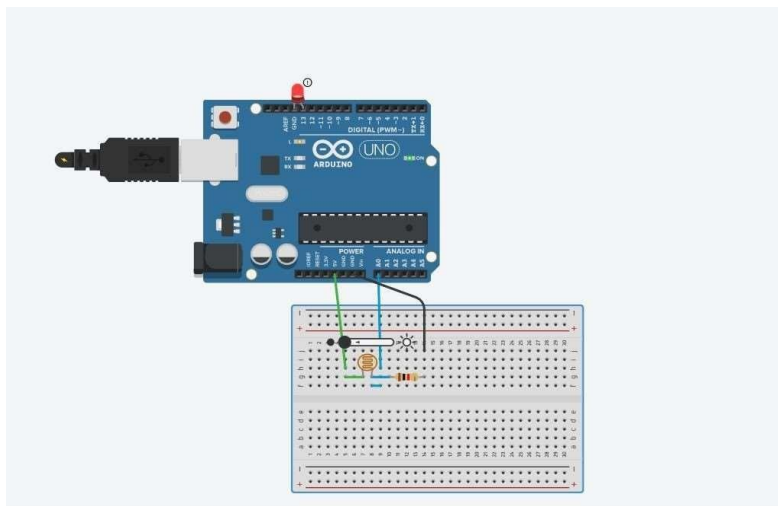
    digitalWrite(ledPin, HIGH);
}
else Serial.println("LDR is DARK, LED is ON");
{

    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

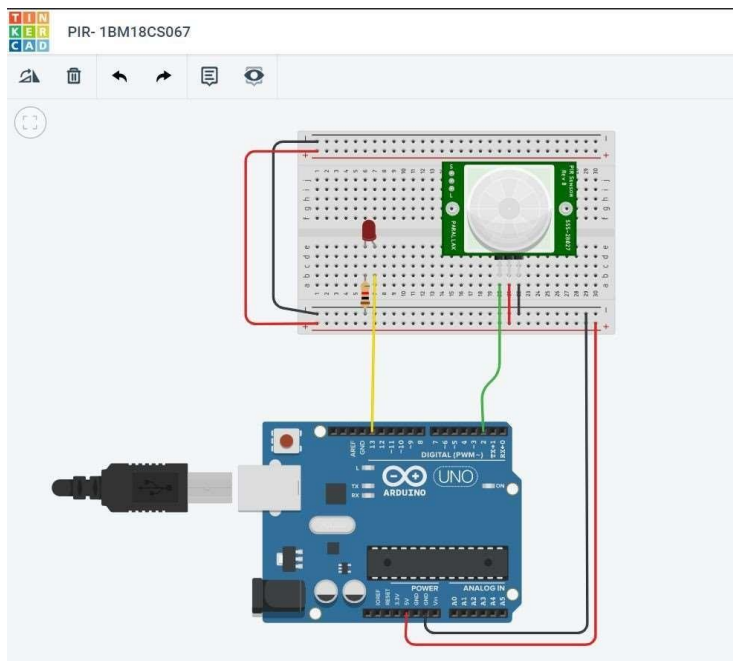
AIM

Demonstrate to show working of PIR sensor.

HARDWARES REQUIRED

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

CIRCUIT DIAGRAM



WRITE-UP

Pin Suman Kumar Pal

```
int SenSor State = 0;
void Setup() {
    pinMode(2, Input);
    pinMode(13, Output);
    Serial.begin(9600);
}

void loop() {
    SenSor State = digitalRead(2);
    if (SenSor State == HIGH) {
        digitalWrite(13, HIGH);
        Serial.println("Sensor activated");
    } else {
        digitalWrite(13, LOW);
    }
    delay(100);
}
```

CODE

```
int sensorState = 0;

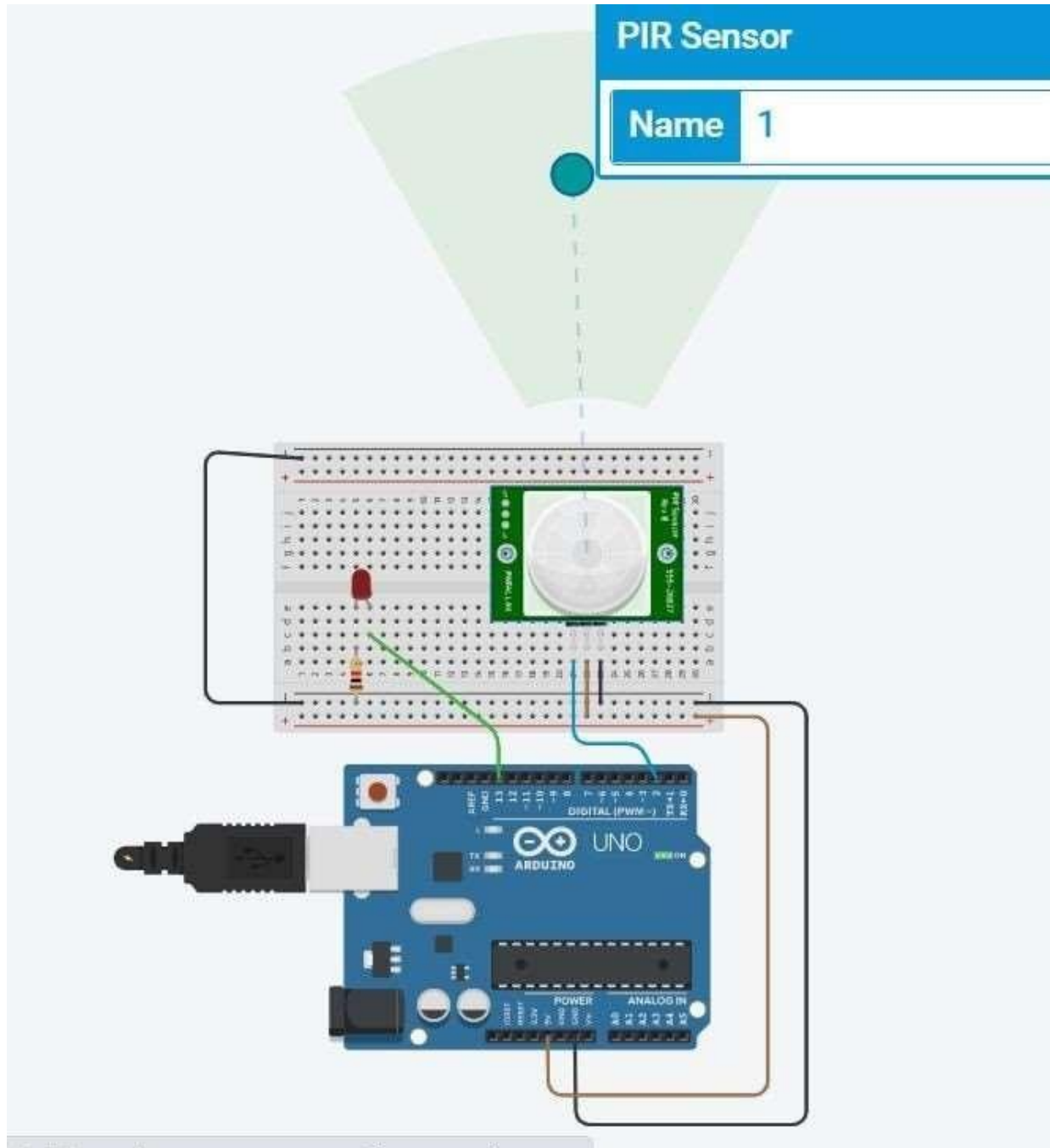
void setup()
{
  pinMode(2, INPUT);
  pinMode(13, OUTPUT);
  Serial.begin(9600);
}

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

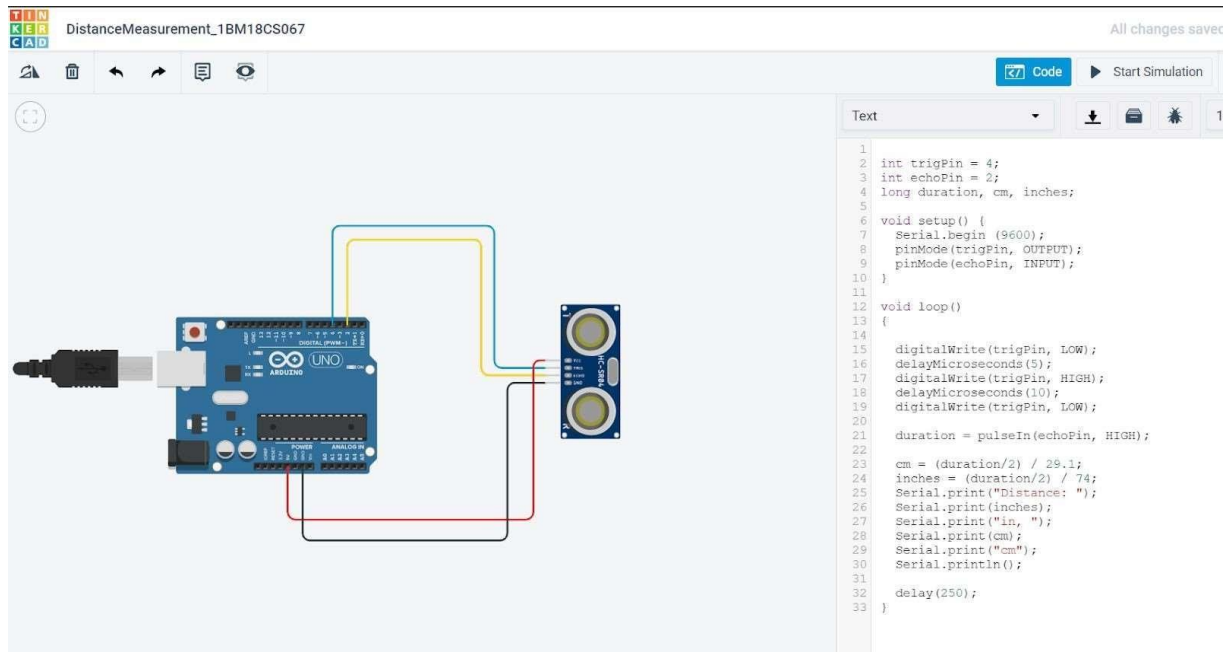
AIM

Design a system to measure the distance between objects.

HARDWARES REQUIRED

- Arduino Board
- Ultrasonic sensor HC-SR04

CIRCUIT DIAGRAM



WRITE-UP

```
const int PingPin = 7;
void Setup() {
    Serial.begin(9600);
}
void loop() {
    long duration, inch, cm;
    pinMode(PingPin, OUTPUT);
    digitalWrite(PingPin, LOW);
    delayMicroseconds(2);
    digitalWrite(PingPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(PingPin, LOW);
    pinMode(7, INPUT);
    duration = pulseIn(7, HIGH);
    inch = duration / 147;
    cm = inch * 2.54;
    Serial.print(" = cm");
    Serial.println(cm);
    Serial.print(" = inch");
    Serial.println(inch);
    delay(500);
}
```

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.

Week 4

circ bit design object distance | circle aeagn Ayazing rpran-e n J

6

Z

g tinkercad.com/things/2CZcGFFDcZN - object-distance-/editel

D @ @ object distance

ML


6

6

Simulator time: 00:00:07

Ultrasonic Distance Sensor

Name 1



Text

```
1 int echoPin = 2;
2 long elula than . cm, Inches ;

6 Serial.begin (9600);
7 pinMode(trigPin, OUTPUT);

14 digitalWrite(trigPin, nor
15 delayMicroseconds(5);
16 digitalWrite(trigPin, HIGH);

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
Distance: 52inches, 133cm
Distance: 52inches, 133cm

Send Clear
```

Type here to search



ENG

15:53

07-10-2020

Program No. – 09

Program Title – Fire Alarm using flame Sensor

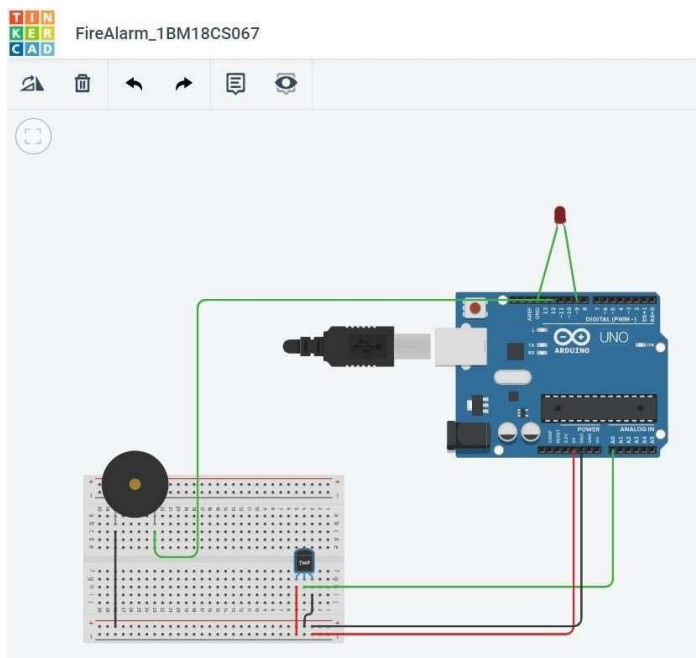
AIM

Design an alert system using a flame sensor.

HARDWARES REQUIRED

- Arduino Board
- Piezo
- Temperature Sensor
- Breadboard small

CIRCUIT DIAGRAM



WRITE-UP

```
Temp.
int tempIn = A0;
float temp = 0;
int Buz = 12;
void setup()
{
  Serial.begin(9600);
  pinMode(Buz, OUTPUT);
  pinMode(18, OUTPUT);
}

void loop()
{
  temp = analogRead(tempIn);
  temp = temp * 0.4828125;
  Serial.println(temp);
  delay(1000);
  if(temp > 25)
  {
    digitalWrite(Buz, HIGH);
    digitalWrite(18, HIGH);
    delay(500);
  }
  else { digitalWrite(Buz, LOW);
        digitalWrite(18, LOW);
  }
}
```

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.

The screenshot shows a Tinkercad workspace with an Arduino Uno R3 connected to a flame sensor and a buzzer. The flame sensor is connected to the Arduino's A0 pin, and the buzzer is connected to digital pin 12. The code in the 'Code' block is as follows:

```
1 const int temperaturePin = 0;
2 int buzzer = 12;
3
4 void setup()
5 {
6   Serial.begin (9600);
7   pinMode (buzzer, OUTPUT);
8   pinMode (9, OUTPUT);
9 }
10 float getVoltage(int pin)
11 {
12   return (analogRead(pin) * 0.004882814);
13 }
14 void loop()
15 {
16   float voltage, degreesC;
```

The 'Serial Monitor' shows the following output:

```
17.87 SAFE!
17.87 SAFE!
30.08 SAFE!
44.73 ALERT!
44.73 ALERT!
44.73 ALERT!
44.73 ALERT!
44.73 ALERT!
```

Program No. – 10

Program Title – Gas Sensor

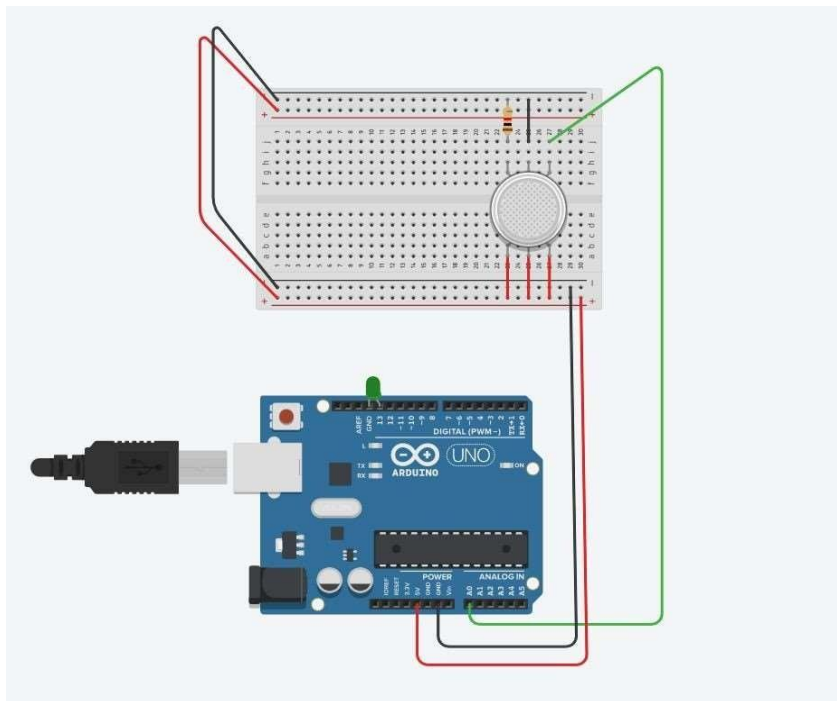
AIM

To design a smart gas leakage indicator system.

HARDWARES REQUIRED

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

CIRCUIT DIAGRAM



WRITE-UP

Date: / /

ind motor Pin = 3

ind sensor Pin =

At: int threshold = 400

void setup {

{

pinMode(indMotorPin,

OUTPUT);

Serial.begin(9600);

}

void loop {

int sensorValue =

analogRead(indSensorPin);

Serial.println(sensorValue);

if (sensorValue > threshold)

{
digitalWrite(indMotorPin, HIGH);

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);
```

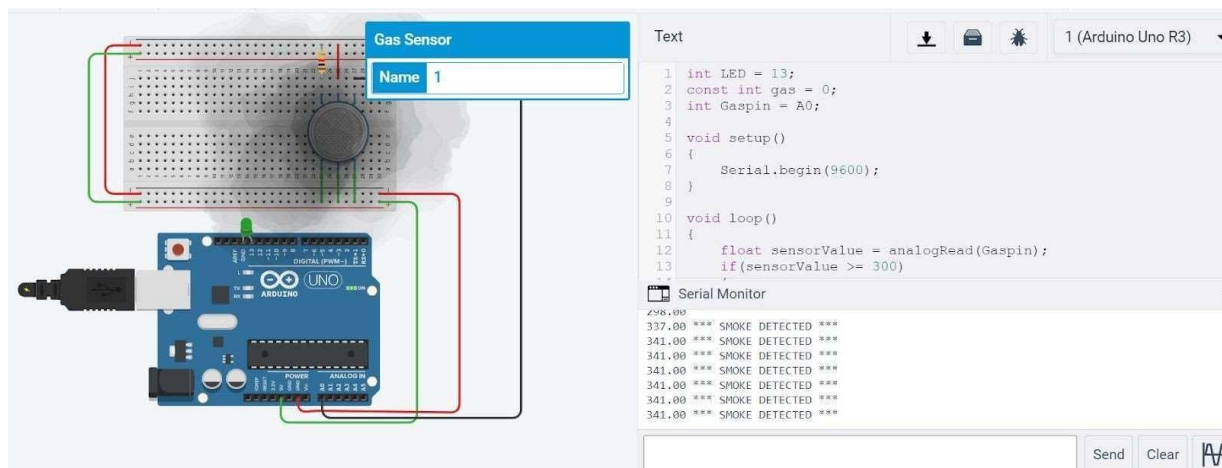
}

```
delay(1000);
```

}

OUTPUT

Designed a smart gas leakage indicator system.



Program No. – 11

Program Title – Vibration motor and LDR

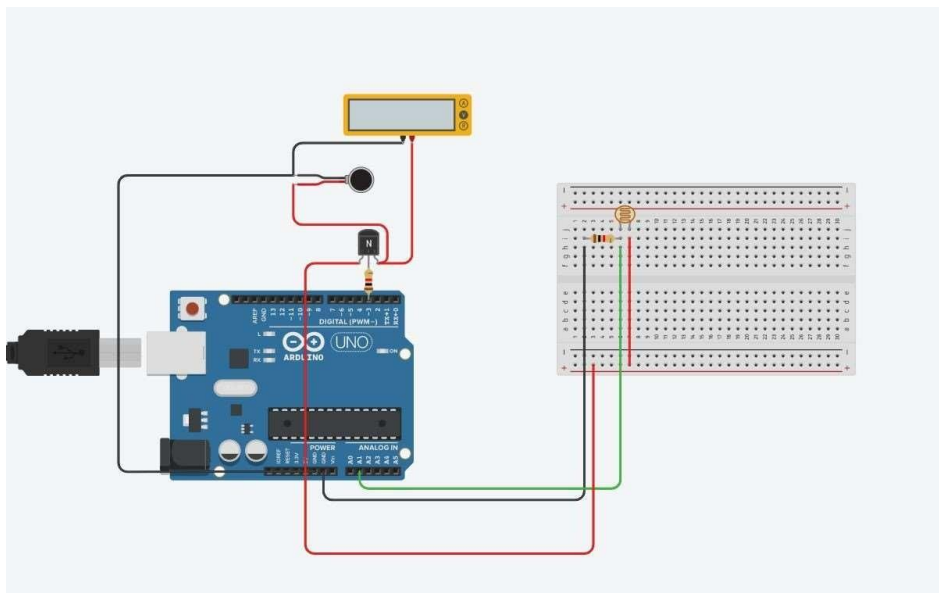
AIM

To design an automated day indicator system.

HARDWARES REQUIRED

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

CIRCUIT DIAGRAM



WRITE-UP

Date: / /

ind motor Pin = 3

ind sensor Pin =

At: int threshold = 400

void setup {

{

pinMode(ind motor Pin,

output

Serial.begin(9600);

}

void loop {

int sensor Value =

analogRead(ind sensor Pin);

Serial.println(sensor Value);

if (sensor Value > threshold)

{
digitalWrite(ind motor Pin, HIGH);

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);

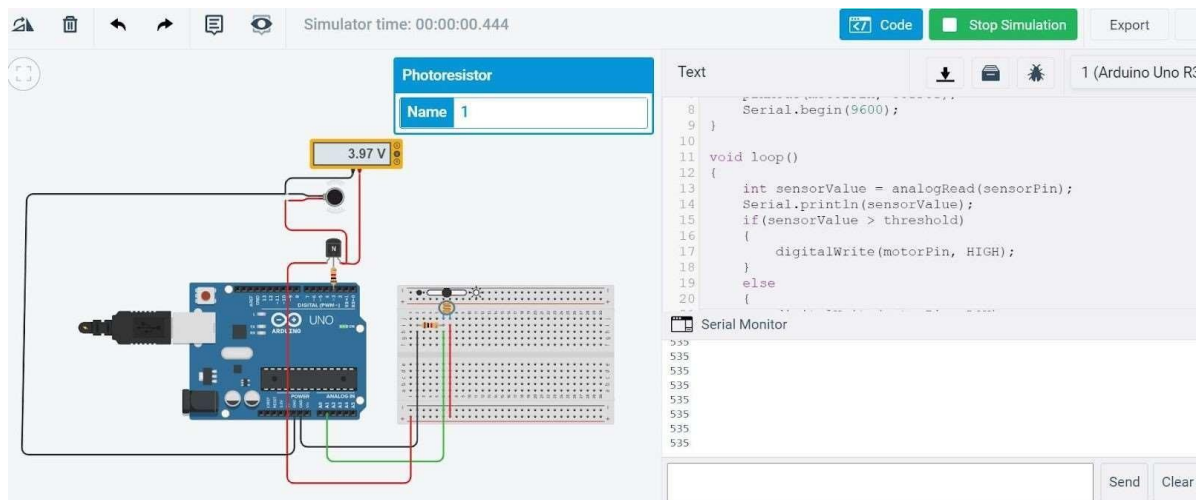
    }

}

```

OUTPUT

Designed an automated day indicator system.



Program No. – 12 Program

Title – Tilt Sensor

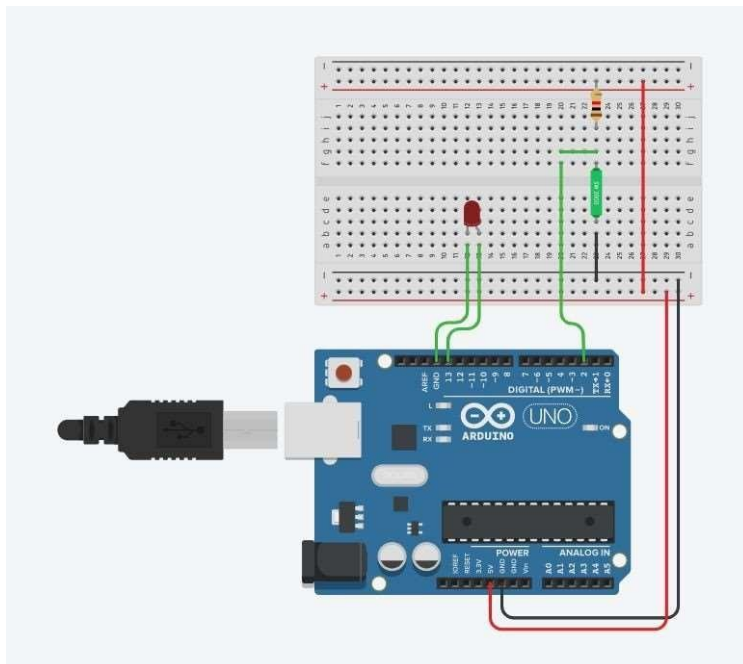
AIM

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

HARDWARES REQUIRED

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

CIRCUIT DIAGRAM



WRITE-UP

```
int bit = 2;
int led = 13;

void setup() {
  pinMode(bit, INPUT);
  pinMode(led, OUTPUT);
}

void loop() {
  int reading;
  reading = digitalRead(bit);
  if (reading)
    digitalWrite(led, HIGH);
  else
    digitalWrite(led, LOW);
}
```

CODE

```
int tilt = 2;

int led = 13;

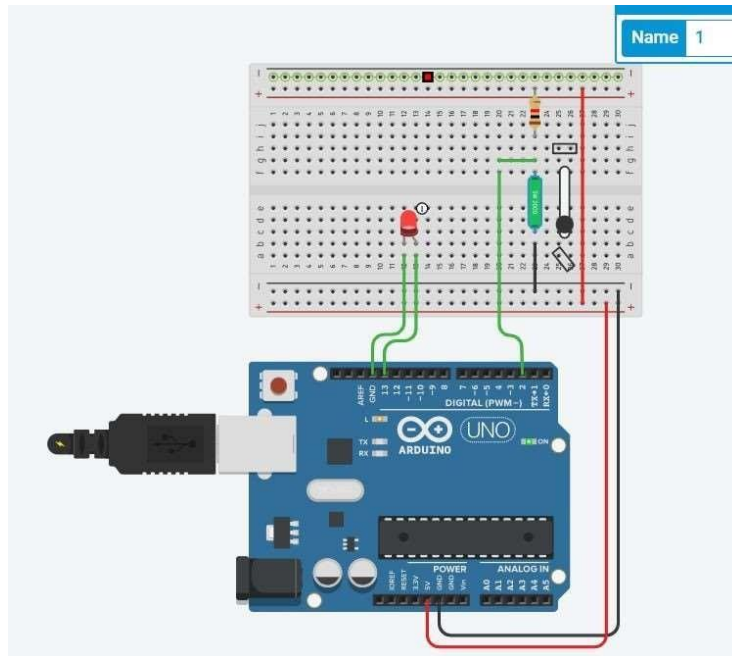
void setup()
{
    pinMode(tilt, INPUT);
    pinMode(led, OUTPUT);
}

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

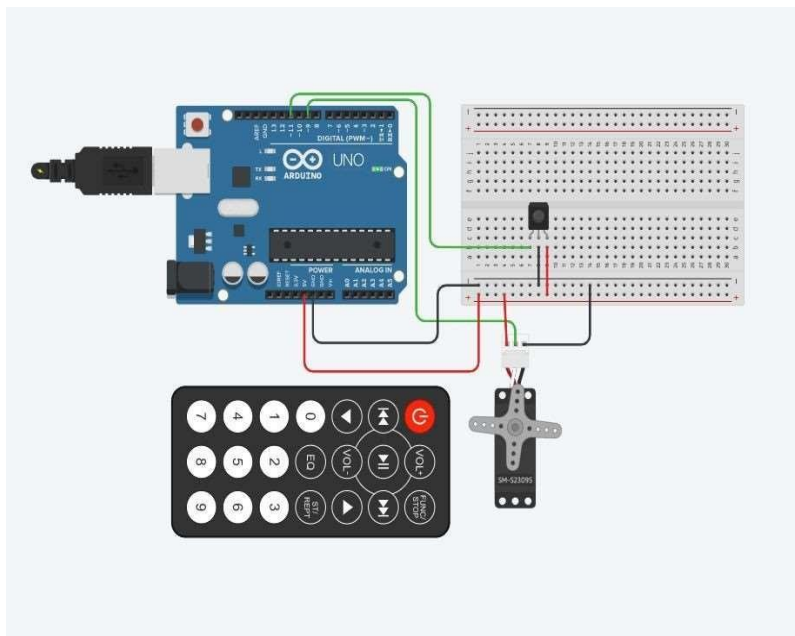
AIM

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



WRITE-UP

Date

```
#  
#  
#include <Servo.h>  
#include <IRremote.h>  
int RECV_PIN = 11;  
IRrecv irrecv(RECV_PIN);  
decode_results results;  
Servo myServo;  
void setup() {  
  Serial.begin(9600);  
  pinMode(RECV_PIN, INPUT);  
}  
void loop() {  
  if (irrecv.decode(&results)) {  
    Serial.println("Received: ");  
    for (int i = 0; i < results.value; i++) {  
      Serial.print(results.values[i]);  
      Serial.print(" ");  
    }  
    Serial.println();  
    myServo.write(results.value);  
    Serial.println("Servo position: ");  
    Serial.println(results.value);  
  }  
}
```

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**

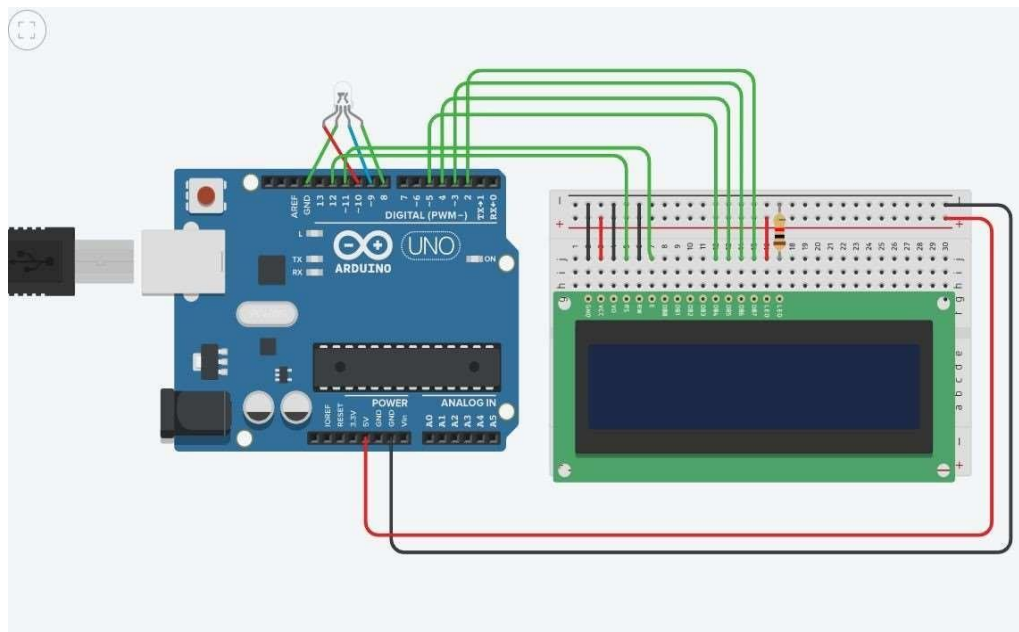
AIM

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



WRITE-UP

Date: / /

14

```
#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(8, OUTPUT);
```

```
  pinMode(9, OUTPUT);
```

```
}
```

```
void loop()
```

```
{
```

```
  lcd.print("Red & Green");
```

```
  delay(1000);
```

```
  lcd.clear();
```

```
  lcd.setCursor(0, 0);
```

```
  lcd.print("Red");
```

```
  delay(1000);
```

```
  lcd.clear();
```

```
  lcd.setCursor(0, 0);
```

```
  lcd.print("Blue");
```

```
  delay(1000);
```

```
  lcd.setCursor(0, 0);
```

```
  lcd.print("White");
```

```
  delay(1000);
```

```
  lcd.clear();
```

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();

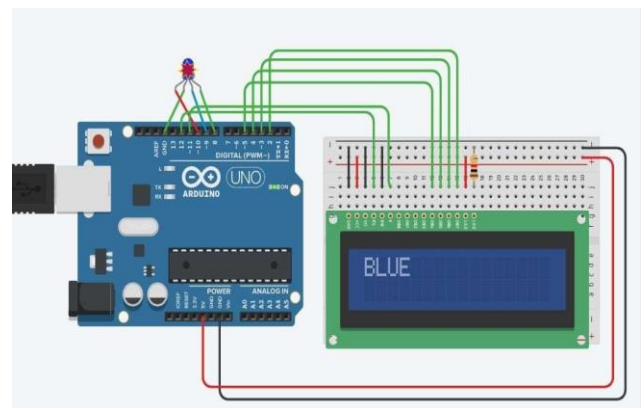
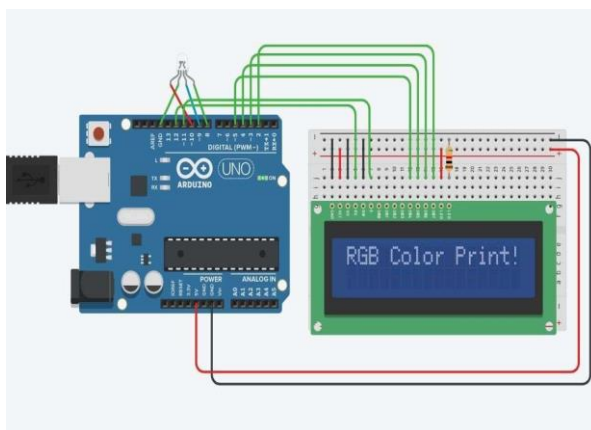
}

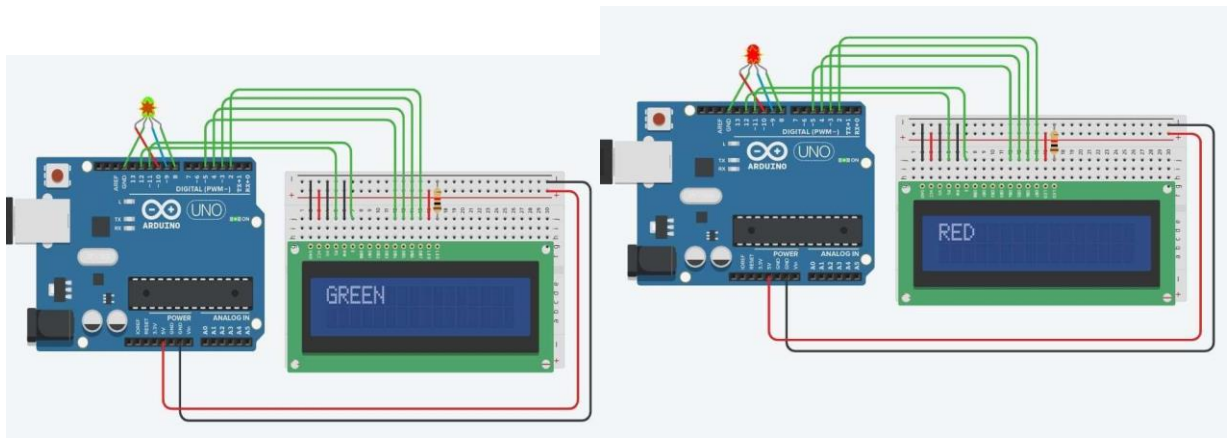
void RGB_color(int red_value, int green_value, int blue_value)
{
    analogWrite(red,red_value);
    analogWrite(green,green_value);
    analogWrite(blue,blue_value);
}

```

OUTPUT

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





Program No. – 15

Program Title – **Smart irrigation system**

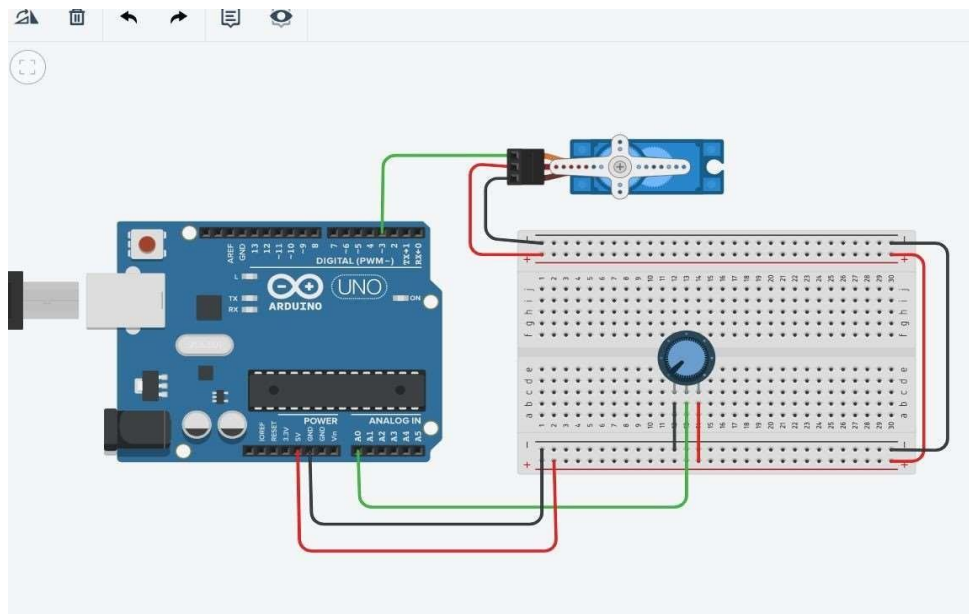
AIM

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

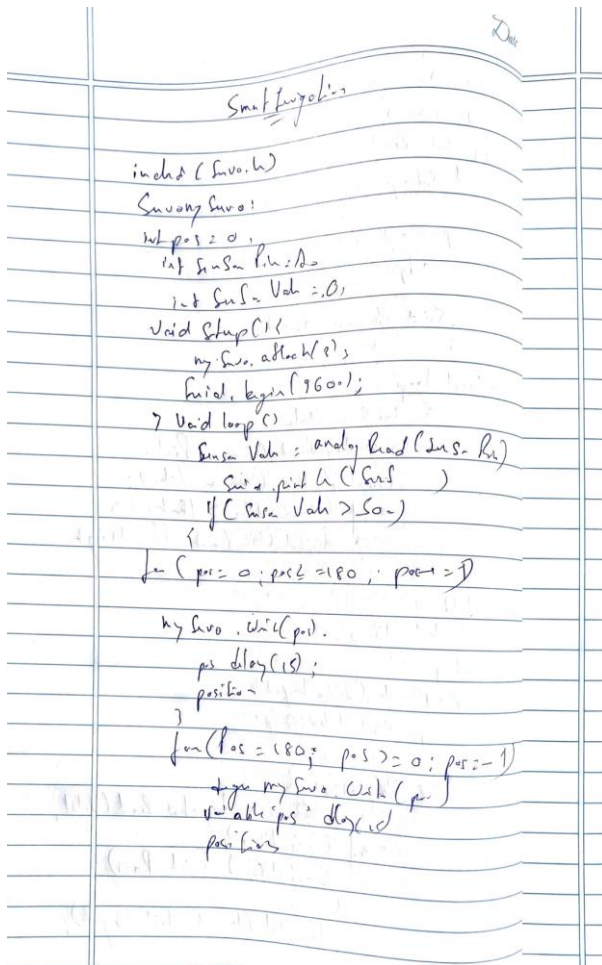
HARDWARES REQUIRED

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

CIRCUIT DIAGRAM



WRITE-UP



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).

