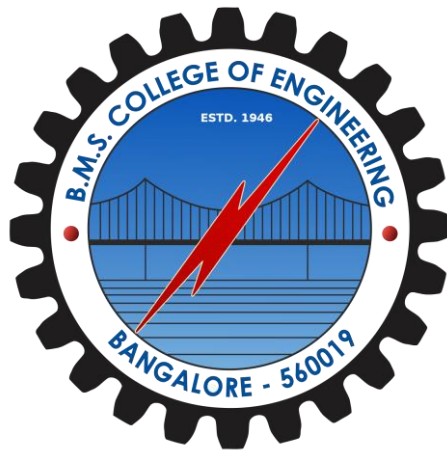


IOT LAB MANUAL-2020



Name: Sakshi Srivastava

USN: 1BM18CS090

CONSOLIDATED LAB PROGRAMS

Sakshi Srivastava

1BM18CS090

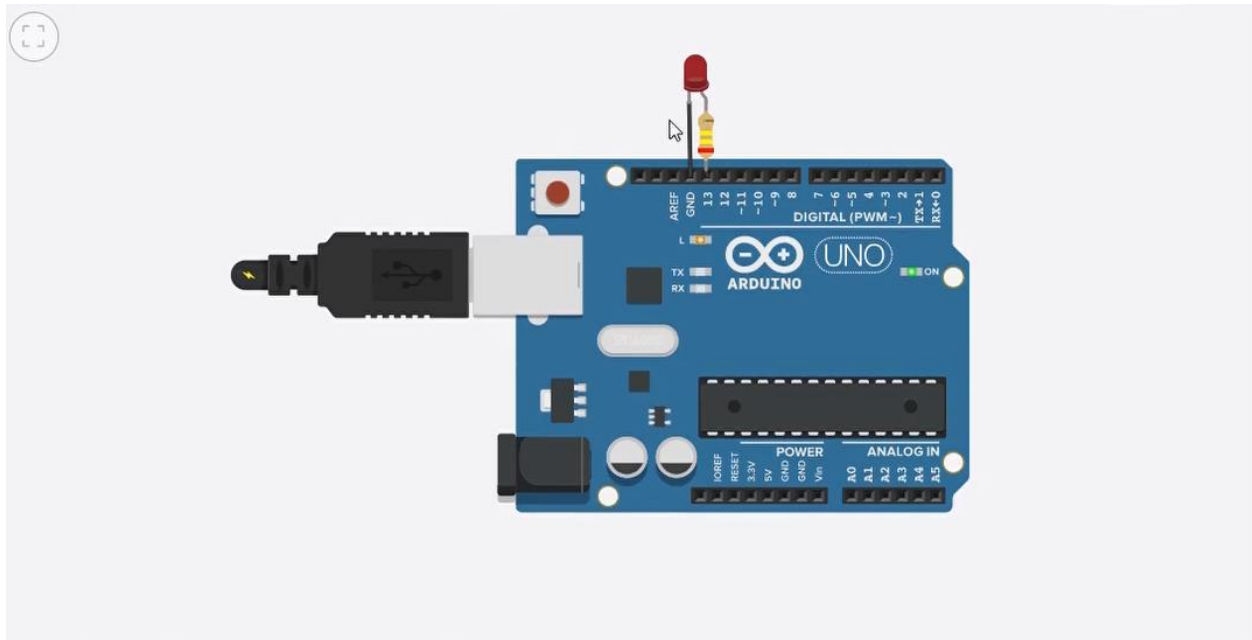
PROGRAM TITLE: BLINKING LED

Aim: TO TURN ON THE LED FOR A SECOND AND TURN IT OFF REPEATEDLY

Hardware Required:

- Arduino Board
- LEDs

Circuit Diagram:



Write-Up:

NAME: Sakshi Srivastava USN: IBM18CS090		Date: 16/9/2020
Expt. No. 1	Page No. 1	

LED BLINKING.

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

Hardware Required:

- Arduino Board
- LEDs.

CODE:

```
// Pin 13 has an LED connected on most Arduino boards.

void setup()
{
    // initialise the digital pin as an output.
    pinMode(13, OUTPUT);
}

void loop()
{
    digitalWrite(13, HIGH); // turns the LED on.
    delay(1000);           // wait for a second.
    digitalWrite(13, LOW);  // turns the LED off.
    delay(1000);           // wait for a second.
}
```

CODE:

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

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

Observation /Output:

LED is blinking.

Sakshi Srivastava

1BM18CS090

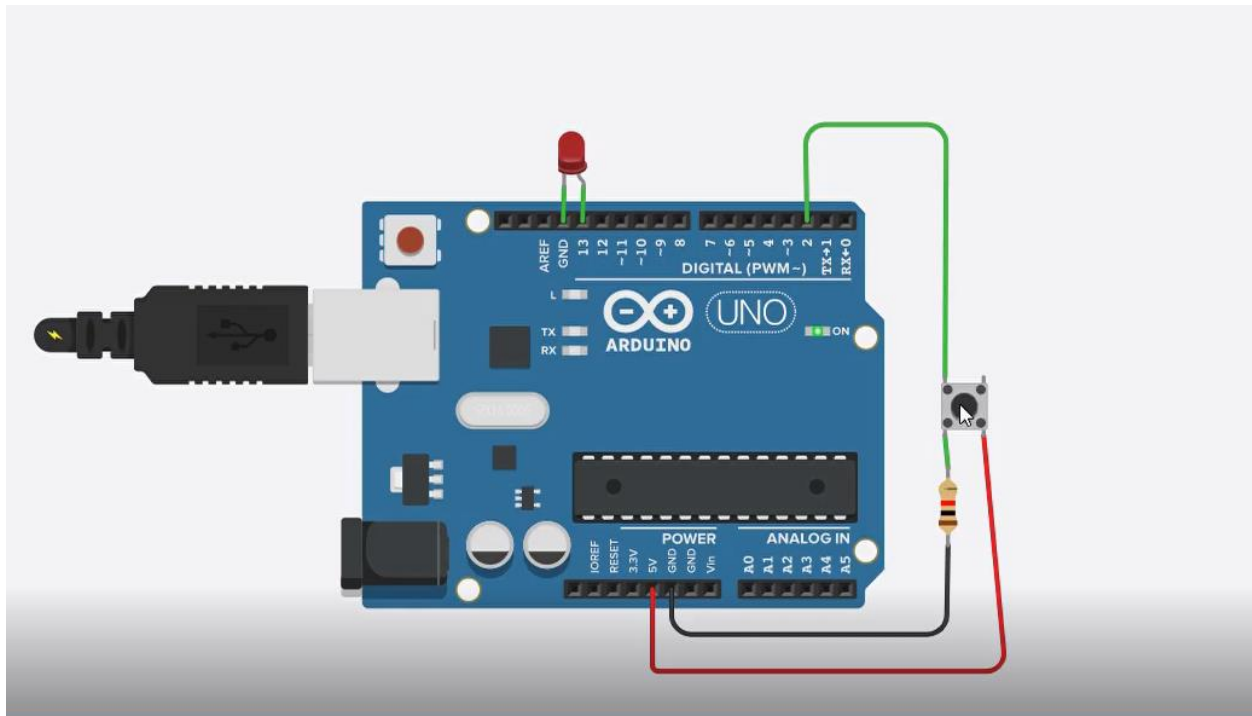
PROGRAM TITLE: LED USING PUSHBUTTON

Aim: DEMONSTRATE TO SHOW ON/OFF OF A LED USING PUSHBUTTON

Hardware Required:

- Arduino Board
- LED
- Pushbutton
- Resistor

Circuit Diagram:



Write-Up:

NAME: Lakshi Srivastava USN: IBM18C6090 Date 23/09/20

Expt. No. 3 Page No. 4

Program Title: LED Using Pushbutton.

Aim: Demonstrate to show ON/OFF of a LED using Pushbutton (Digital output)

→ Hardware Required:-

- LED
- Arduino Board
- Pushbutton, Resistor.

Code:

```
const int buttonPin = 2;
const int ledPin = 13;
int buttonState = 0;

void setup()
{
  pinMode(ledPin, OUTPUT);
  pinMode(buttonPin, INPUT);
}

void loop()
{
  buttonState = digitalRead(buttonPin);
  if(buttonState == HIGH)
```

Name: Lakshi Srivastava USN: IBM18C6090 Date 23/09/2020

Expt. No. 3 Page No. 5

```
}
  digitalWrite(ledPin, HIGH);
}
else
{ digitalWrite(ledPin, LOW);
}
}
```

CODE:

```
const int
buttonPin=2;

const int ledPin=13;
int buttonState=0;
void setup()
{
  pinMode(ledPin,OUTPUT); // declare LED as output
  pinMode(buttonPin, INPUT); // declare pushbutton as input
}

void loop()
{
  buttonState = digitalRead(buttonPin); // read input value
  if (buttonState== HIGH)
  {
    digitalWrite(ledPin, HIGH);
  }
  else
  {
    digitalWrite(ledPin, LOW);
  }
}
```

Observation /Output:

On/Off Of A Led Using Pushbutton

Sakshi Srivastava

1BM18CS090

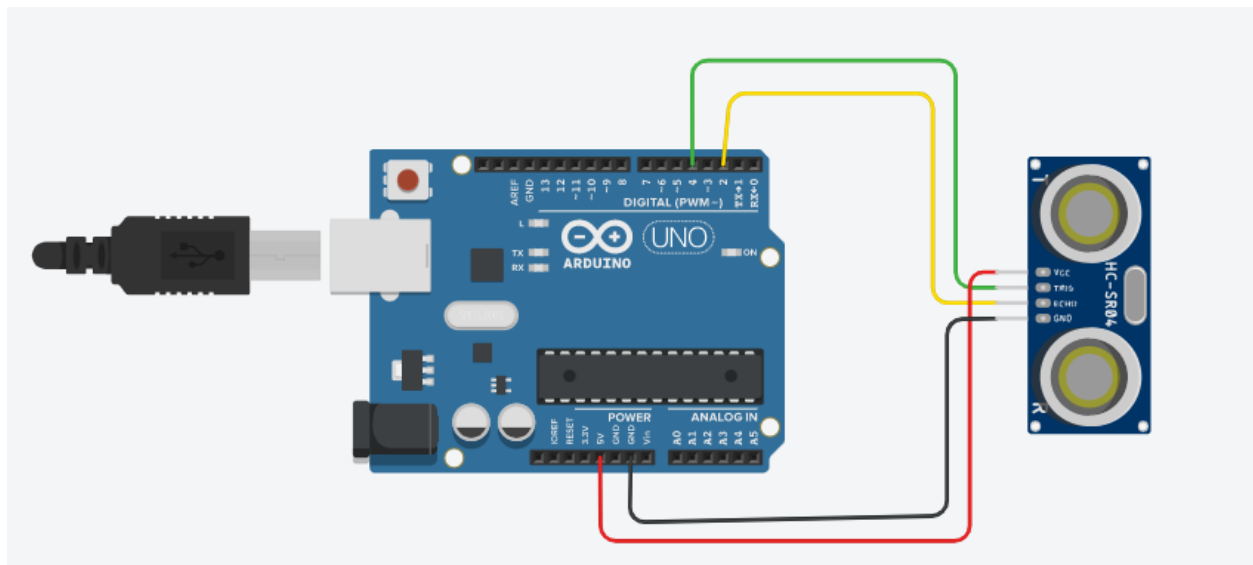
PROGRAM TITLE: DISTANCE MEASURING

Aim: DESIGN A SYSTEM TO MEASURE THE DISTANCE BETWEEN OBJECTS

Hardware Required:

- Arduino Board
- Ultrasonic Sensor

Circuit Diagram:



Write-Up:

Name: Sakshi Shivastava (16M18CS090) Date: 7/10/2020
Expt. No. 9 Page No. 13

Aim: Design a system to measure the distance between objects.

HARDWARE REQUIRED:

- Arduino - Board
- Ultrasonic sensor

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);
```

Expt. No. 9 Date: 7/10/2020
Page No. 14

```
  pinMode(echoPin, INPUT);
  duration = pulseIn(echoPin, HIGH);

  cm = (duration/2) / 29.1;
  inches = (duration/2) / 74;
  Serial.print("Distance: ");
  Serial.print(inches);
  Serial.print(" in ");
  Serial.print(cm);
  Serial.print(" cm");
  Serial.println();

  delay(250);
}
```

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);

  pinMode(echoPin, INPUT);
  duration = pulseIn(echoPin, HIGH);

  cm = (duration/2) / 29;
```

```
inches = (duration/2) / 74;  
Serial.print("Distance: ");  
Serial.print(inches);  
Serial.print("inches, ");  
Serial.print(cm);  
Serial.print("cm");  
Serial.println();  
  
delay(250);  
}
```

OUTPUT/OBSERVATION:

The distance is being measured.

Sakshi Srivastava
1BM18CS090

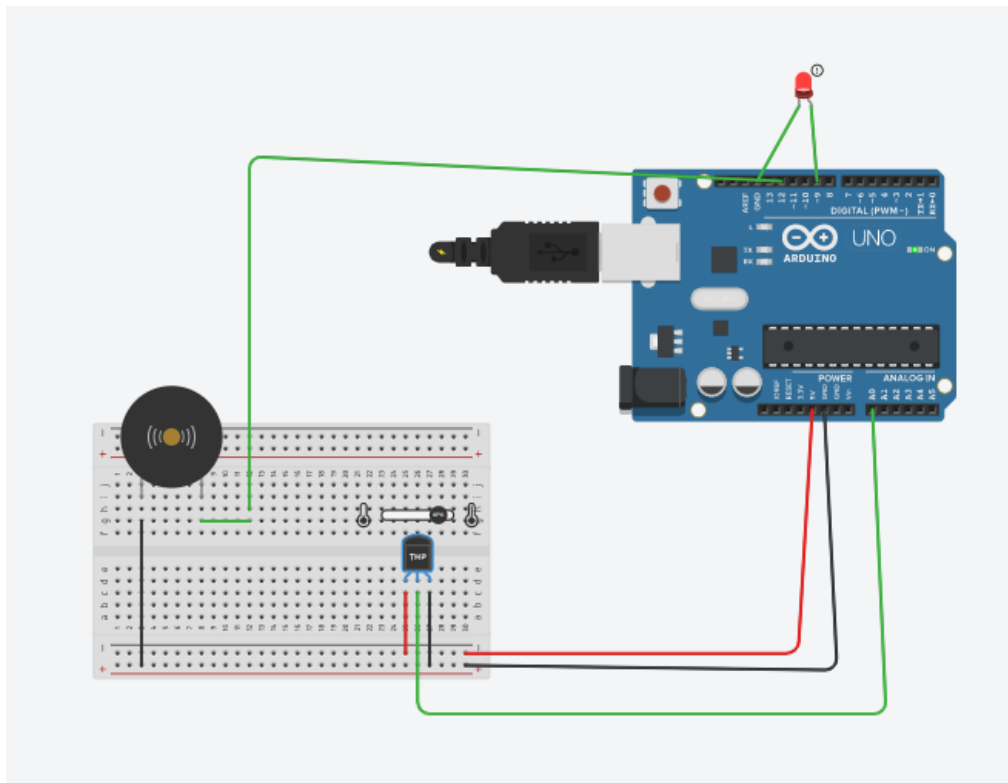
PROGRAM TITLE: FLAME SENSOR

Aim: DESIGN AN ALERT SYSTEM USING FLAME SENSOR

Hardware Required:

- Arduino Board
- LED
- Breadboard
- Temperature Sensor(LM35)
- Buzzer

Circuit Diagram:



Write-Up:

Name: Nakshi Srivastava (18M18CS090) Date: 7/10/2020
Expt. No. 10 Page No. 15

Aim: Design an alert system using flame sensor.

HARDWARE REQUIRED :-

- Arduino Board
- LED
- Breadboard
- Temperature sensor (LM35)
- Buzzer

CODE

```
const int temperaturePin = 0;
int buzzer = 12;
void setup()
{
  Serial.begin(9600);
  pinMode(buzzer, OUTPUT);
  pinMode(9, OUTPUT);
}
float getVoltage(int pin)
{
  return (analogRead(pin) * 0.004882814);
}
```

Expt. No. 10 Date: 7/10/2020
Page No. 16

```
void loop()
{
  float voltage, degreeC;
  voltage = getVoltage(temperaturePin);
  degreeC = (voltage - 0.5) * 100.0;
  digitalWrite(9, LOW);
  if (degreeC < 37)
  {
    Serial.print(degreeC);
    Serial.print("SAFE");
  }
  if (degreeC > 37)
  {
    Serial.print(degreeC);
    Serial.println("ALERT!!");
    digitalWrite(buzzer, LOW);
    digitalWrite(9, HIGH);
    tone(12, 1000, 100);
    delay(100);
  }
}
```

CODE:

```
const int
temperaturePin=0;

int buzzer = 12;

void setup()
{
  Serial.begin (9600);
  pinMode(buzzer, OUTPUT);
  pinMode(9,OUTPUT);
}
float getVoltage(int pin)
{
  return (analogRead(pin) * 0.004882814);
}
void loop()
{
  float voltage, degreesC;
  voltage = getVoltage(temperaturePin);//gets temp in 5v
  degreesC = (voltage-0.5)*100.0;//converts the temp to
celcius
  digitalWrite(9,LOW);
  if(degreesC < 37)
  {
    Serial.print(degreesC);
    Serial.println(" IT IS SAFE!");
  }
  if(degreesC > 37)
  {
    Serial.print(degreesC);
    Serial.println(" ALERTTTTTTT!");
    digitalWrite(buzzer, LOW);
    digitalWrite(9,HIGH);
    tone(12, 10000,100);
    delay(100);
  }
}
```

OUTPUT/OBSERVATION:

The temperature is being measured and accordingly the message is being displayed after sensing it.

Sakshi Srivastava

1BM18CS090

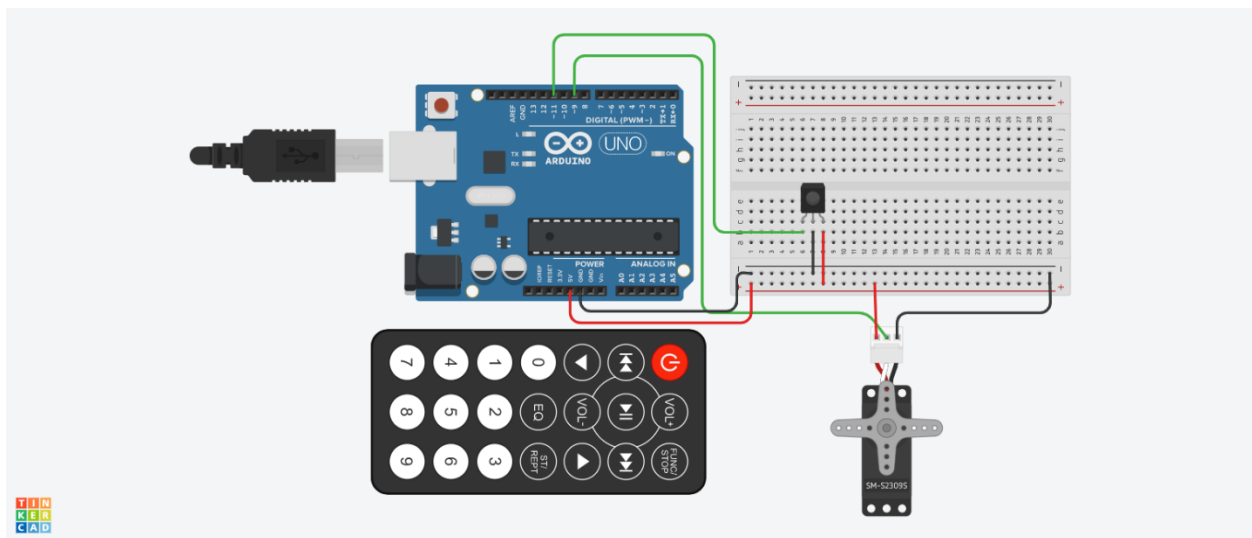
PROGRAM TITLE: IR REMOTE

Aim: DESIGN IR based SERVO MOTOR controller (Clockwise and counterclockwise Rotation of shaft)

Hardware Required:

- IR remote
- IR sensor
- Micro Servo
- Breadboard
- Arduino UNO

Circuit Diagram:



Write-Up:

SAKSHI SRIVASTAVA		16MNC090	Date <u>28 Oct</u>	Expt. No. <u>14</u>	Date
Expt. No. <u>14</u>		Page No. <u>22</u>		Page No. <u>23</u>	
<p>Aim: Design IR based SERVO Motor controller. (Clockwise and Counterclockwise rotation of shaft)</p> <p>Hardware Required:-</p> <ul style="list-style-type: none">→ Arduino Board→ Breadboard→ Micro Servo→ IR Sensor→ IR Remote <p>Code:-</p> <pre>#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(); }</pre>			<pre>void loop() { if (irrecv.decode(&results)) { switch (results.value) { case 0xFD00FF: myservo.attach(9); Serial.println("Stop"); break; case 0xFD609F: myservo.write(360); Serial.println("Clockwise"); break; case 0xFD20DF: myservo.write(-360); Serial.println("Counterclockwise"); break; default: Serial.println("Unrecognised code received: 0x"); Serial.println(results.value, HEX); break; } } irrecv.resume(); }</pre>		

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/Observation:

Start
Counter Clockwise
Clockwise
Unrecognized code received:
0xFFFFFFFF

Sakshi Srivastava

1BM18CS090

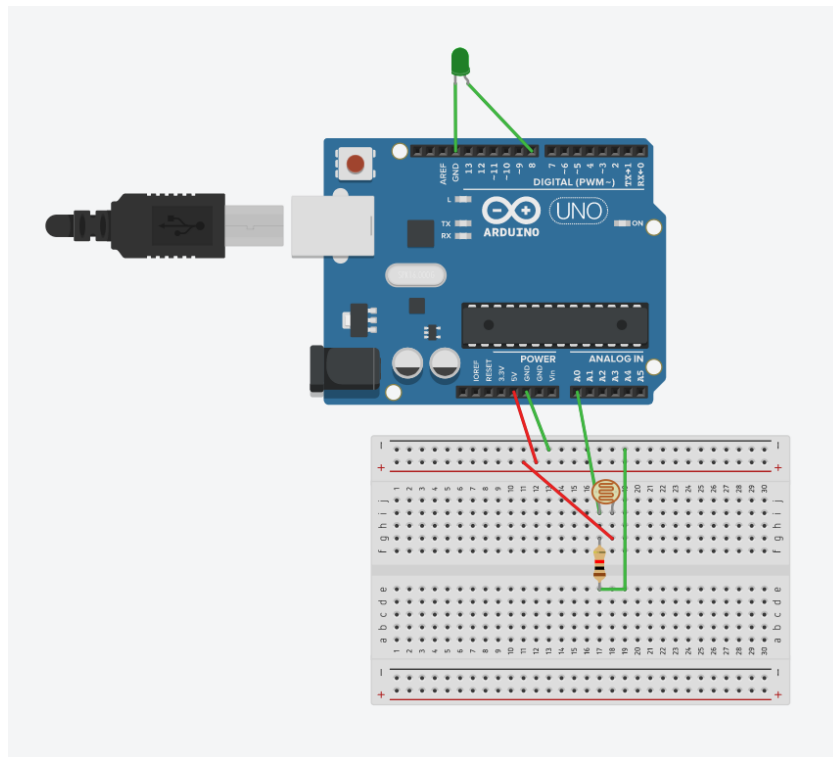
PROGRAM TITLE: LDR-NIGHT LIGHT SIMULATION

Aim: DEMONSTRATE AND SHOW ON/OFF OF A LED USING LDR-NIGHT LIGHT SIMULATION

Hardware Required:

- Arduino Board
- LED
- Photoresistor
- Resistor
- Breadboard

Circuit Diagram:



Write-Up:

Name: Sakshi Srivastava	Date: 30/9/2020
Expt. No. 6	Page No. 8

Aim: Demonstrate to show ON/OFF of a LED using LDR - Night Light simulation.

Hardware Required:-

- LED
- Arduino Board
- Photoresistor
- Resistor
- Breadboard

CODE:

```
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);
```

Expt. No. 6	Date: _____
	Page No. 9


```
if(ldrStatus <= 10)
{
  digitalWrite(ledPin, HIGH);
  Serial.println("LDR is DARK, LED is ON");
}
else
{
  digitalWrite(ledPin, LOW);
  Serial.println("-----");
}
}
```

CODE:

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

OUTPUT/OBSERVATION:

LED is ON.

Sakshi Srivastava

1BM18CS090

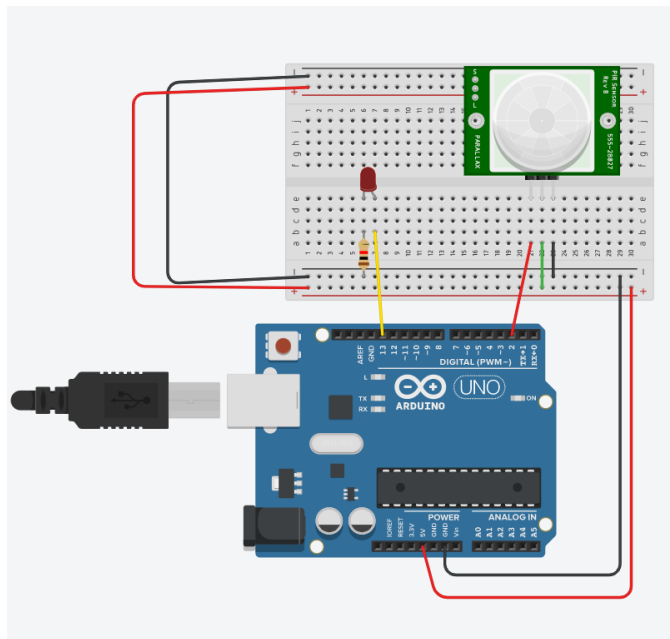
PROGRAM TITLE: PIR SENSOR

Aim: DEMONSTRATE AND SHOW THE WORKING OF PIR SENSOR

Hardware Required:

- Arduino Board
- LED
- Photoresistor
- Resistor

Circuit Diagram:



Write-Up:

Name: Sakshi Srivastava	USN: BM18C6090	Date:
Expt. No. 7	Page No. 10	

AIM: Demonstrate to show the working of PIR sensor.

Hardware Required:
Arduino board
LED
Photoresistor
Breadboard.

Code:

```
int sensorState = 0;  
void setup()  
{  
  pinMode(2, INPUT);  
  pinMode(13, INPUT);  
  Serial.begin(9600);  
}  
  
void loop()  
{  
  sensorState = digitalRead(2);  
  if(sensorState == HIGH)  
  {  
    digitalWrite(13, HIGH);  
    Serial.println("sensor activated!");  
  }  
  else
```

Expt. No. 7	Date:
Page No. 11	

```
}  
  digitalWrite(13, LOW);  
  Serial.println("sensor deactivated!");  
}  
  delay(10);  
}
```


CODE:

```
int sensorState = 0;

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

void loop()
{
    // read the state of the sensor/digital input
    sensorState = digitalRead(2);
    // check if sensor pin is HIGH. if it is, set the
    // LED on.
    if (sensorState == HIGH)
    {
        digitalWrite(13, HIGH);
        Serial.println("Sensor activated!");
    } else
    {
        digitalWrite(13, LOW);
        Serial.println("Sensor Deactivated!");
    }
    delay(10); // Delay a little bit to improve simulation performance
}
```

OUTPUT/OBSERVATION:

Sensor is activated.

1BM18CS090

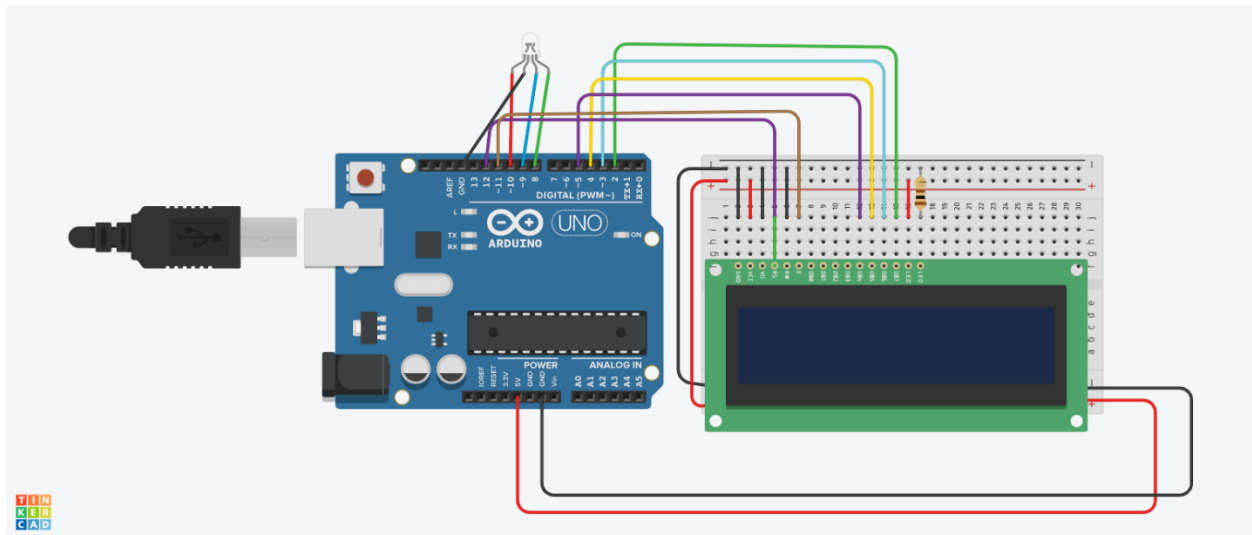
PROGRAM TITLE: RGB LED AND LCD

Aim: DESIGN A DISPLAY SYSTEM TO PRINT RED, BLUE AND GREEN COLORS (RGB LED and LCD)

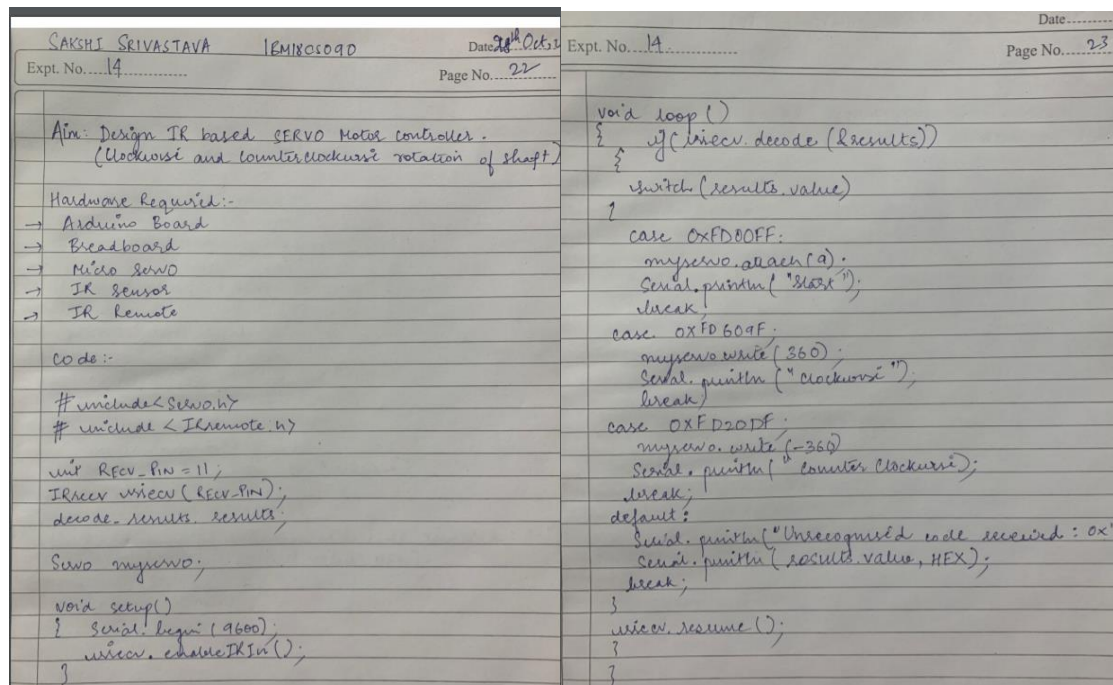
Hardware Required:

- Wires
- LCD
- LED
- Breadboard
- Arduino UNO

Circuit Diagram:



Write-Up:



CODE:

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

```
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
//Parameters: (rs, enable, d4, d5, d6, d7)
```

```
int red_light_pin = 10;
```

```
int green_light_pin = 8;
```

```
int blue_light_pin = 9;
```

```
void setup() {
```

```
  pinMode(red_light_pin, OUTPUT);
```

```
  pinMode(green_light_pin, OUTPUT);
```

```
  pinMode(blue_light_pin, OUTPUT);
```

```
}
```

```
void loop() {
```

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

```
  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(255, 255, 255); // White
lcd.print("WHITE");
delay(1000);
lcd.clear();
}
void RGB_color(int red_light_value, int
green_light_value, int blue_light_value)
{
    analogWrite(red_light_pin, red_light_value);
    analogWrite(green_light_pin,
green_light_value);
    analogWrite(blue_light_pin, blue_light_value);
}

```

OBSERVATION/OUTPUT

Displays the colour on the LCD.

Sakshi Srivastava
1BM18CS090

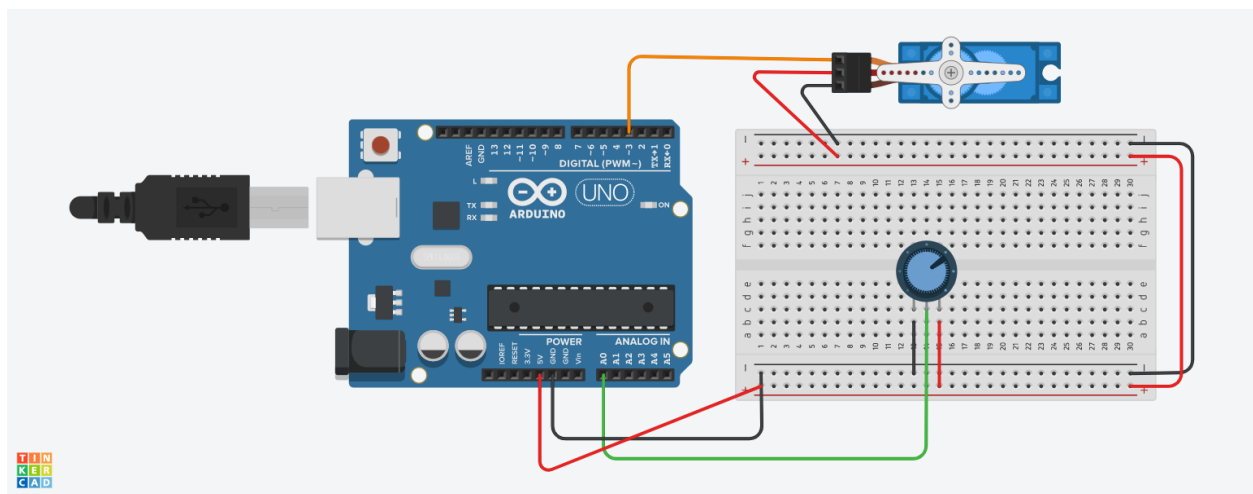
PROGRAM TITLE: SMART IRRIGATION

Aim: DESIGN A SMART IRRIGATION SYSTEM (Potentiometer, Servo Motor shaft)

Hardware Required:

- Wires
- Potentiometer
- Micro Servo
- Breadboard
- Arduino UNO

Circuit Diagram:



Write-Up:

Sakshi Srivastava 18M18CSD90 Date.....

Expt. No. 15 Page No.....

SMART IRRIGATION 4th Nov, 2020

```
#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)
  {
    for (pos = 0; pos <= 180; pos += 1)
      myservo.write(pos);
      delay(15);
    }
    for (pos = 180; pos >= 0; pos -= 1)
      myservo.write(pos);
      delay(15);
    }
    delay(1000);
  }
```

Teacher's Signature :

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 3 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/Observation:

716
 634
 450

Sakshi Srivastava

1BM18CS090

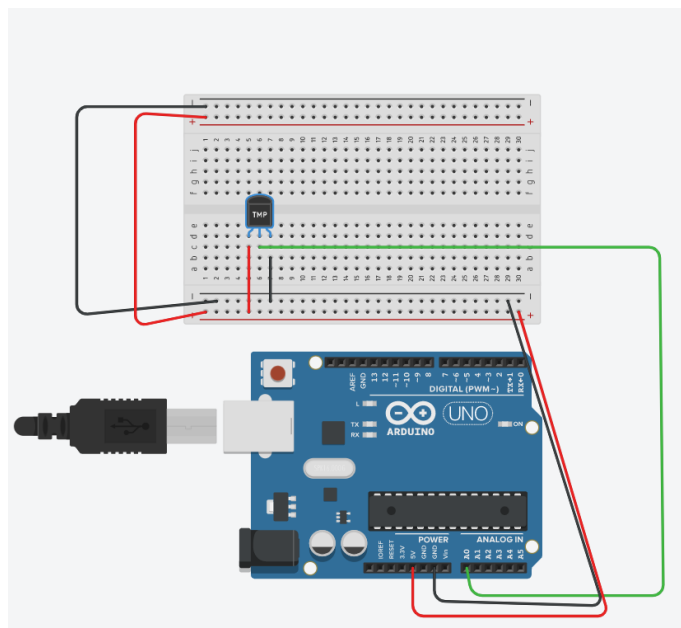
PROGRAM TITLE: TEMPERATURE SENSOR(LM35)

Aim: DEMONSTRATE AND MAKE YOUR OWN TEMPERATURE SENSOR BY ARDUINO AND LM35 SENSOR

Hardware Required:

- Arduino Board
- Breadboard
- LM35 Temperature Sensor

Circuit Diagram:



Write-Up:

Name: SAKSHI SRIVASTAVA		Date: 30/9/2022
Expt. No. 8	\	Page No. 12

Aim: Demonstrate and make your own temperature sensor by Arduino and LM35 sensor.

Hardware Required:-

- Arduino board
- Breadboard
- LM35 Temperature sensor

CODE:

```
float temp;
int tempPin = 0;

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

void loop()
{
  temp = analogRead(tempPin);
  temp = temp * 0.48828125;
  Serial.println("TEMPERATURE = ");
  Serial.println(temp);
  Serial.println("* C *");
  Serial.println();
  delay(1000);
}
```

CODE:

```
float temp;
int tempPin = 0;

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

void loop()
{
    temp = analogRead(tempPin);
    temp = temp * 0.48828125;
    Serial.println("TEMPERATURE = ");
    Serial.println(temp);
    Serial.println("*C");
    Serial.println();
    delay(1000);
}
```

OUTPUT/OBSERVATION:

Temperature is being measured.

Sakshi Srivastava

1BM18CS090

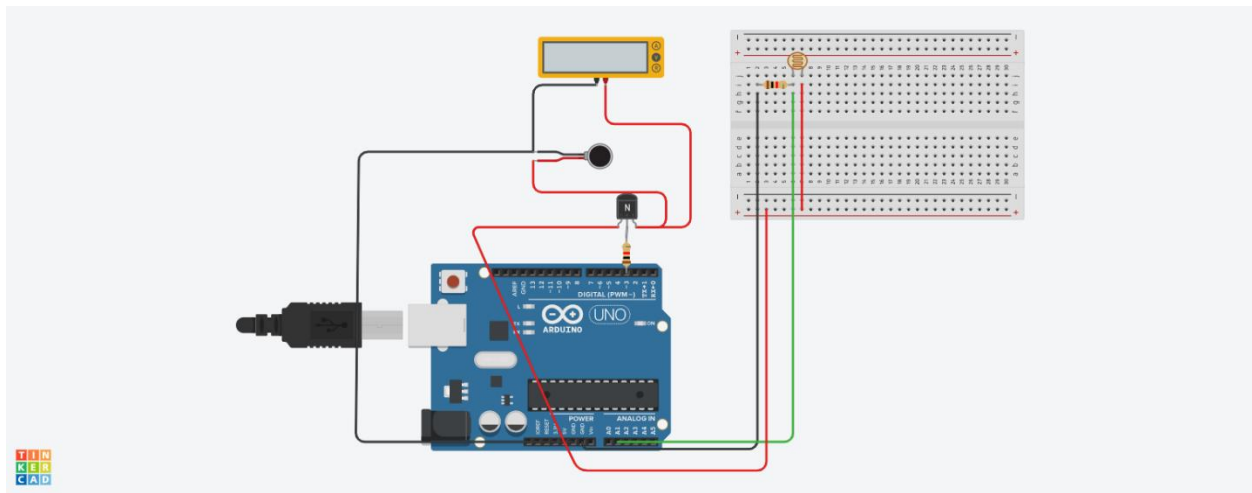
PROGRAM TITLE: VIBRATOR MOTOR

Aim: DESIGN AUTOMATED DAY INDICATOR SYSTEM(VIBRATOR MOTOR AND LDR)

Hardware Required:

- Photoresistor
- Resistor, LED
- Breadboard, Arduino UNO
- Multimeter
- npn resistor, Vibrator Motor

Circuit Diagram:



Write-Up:

NAME: Anshu Swastara USN: 18M131090 Date: 1st Oct 2020
Expt. No. 12 Page No. 19

Aim: Design automated day indicator system.
(Vibrator Motor and LDR)

HARDWARE REQUIRED:-

- Photoresistor
- Resistor
- Arduino UNO
- NPN resistor
- Multimeter
- Breadboard
- Vibrator Motor.

CODE:

```
int motorPin = 3;
int sensorPin = A1;
int threshold = 100;

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

Date:

Expt. No. 12 Page No. 20

```
void loop()
{
  int sensorValue = analogRead(sensorPin);
  Serial.println(sensorValue);

  if(sensorValue > threshold)
  {
    digitalWrite(motorPin, HIGH);
  }
  else
  {
    digitalWrite(motorPin, LOW);
  }
}
```

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);
    }
}
```

OBSERVATION/OUTPUT:

If sensor value>threshold it displays HIGH else LOW.

Sakshi Srivastava

1BM18CS090

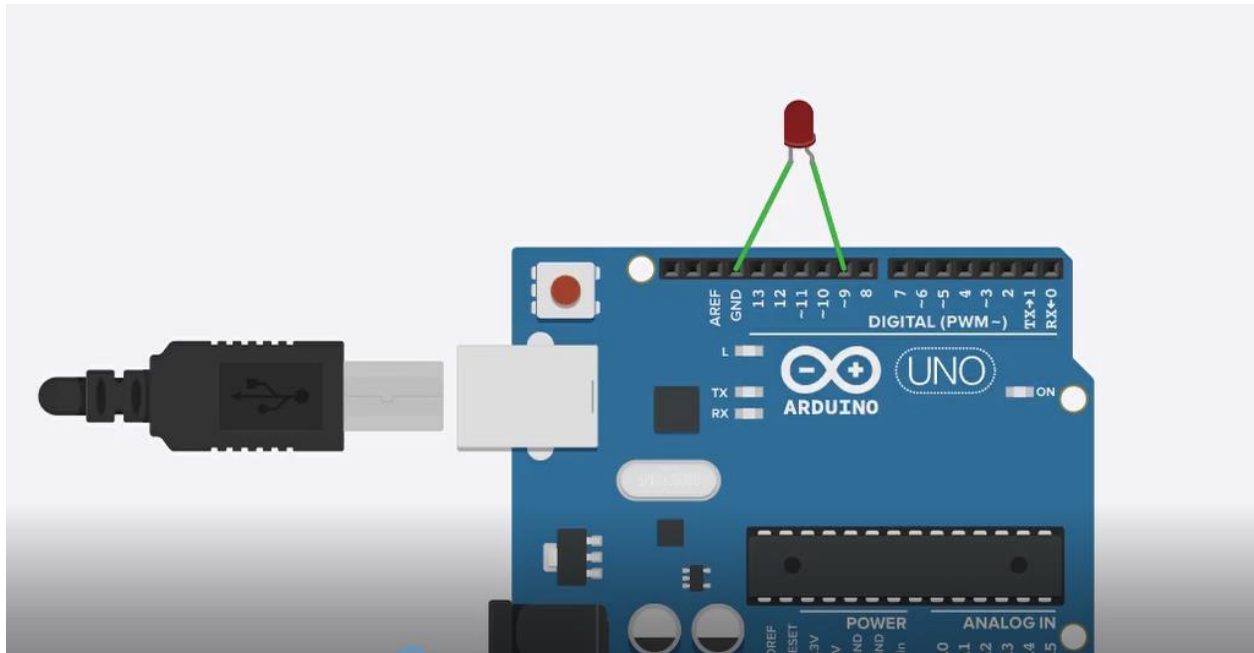
PROGRAM TITLE: LED FADING

Aim: DEMONSTRATE TO SHOW LED FADING(ANALOG OUTPUT)

Hardware Required:

- Arduino Board
- LED

Circuit Diagram:



Write-Up:

Name: Sakshi Sivastava	USN: 18M18CS090	Date: 23/09/2020
Expt. No. 4	Page No. 6	
Program Title: LED FADING		
Aim: Demonstrate to show LED Fading (Analog output)		
Hardware Required:		
<ul style="list-style-type: none">• Arduino board• LED		
<u>Code:-</u>		
<pre>int brightness = 0; void setup() { pinMode(9, OUTPUT); } void loop() { for(brightness = 0; brightness <= 255; brightness += 10) { analogWrite(9, brightness); delay(50); } for(brightness = 255; brightness >= 0; brightness -= 10) { analogWrite(9, brightness); delay(50); } }</pre>		

CODE:

```
int
brightness=0;

void setup()
```

```
{ pinMode(9, OUTPUT);  
}  
void loop()  
{  
  for (brightness =0;brightness <= 255; brightness +=10)  
  {  
    analogWrite(9, brightness);  
    delay(50);  
  }  
  for (brightness = 255;brightness >=0; brightness -=10)  
  {  
    analogWrite(9, brightness);  
    delay(50);  
  }  
}
```

OUTPUT/OBSERVATION:

LED is fading.

Sakshi Srivastava

1BM18CS090

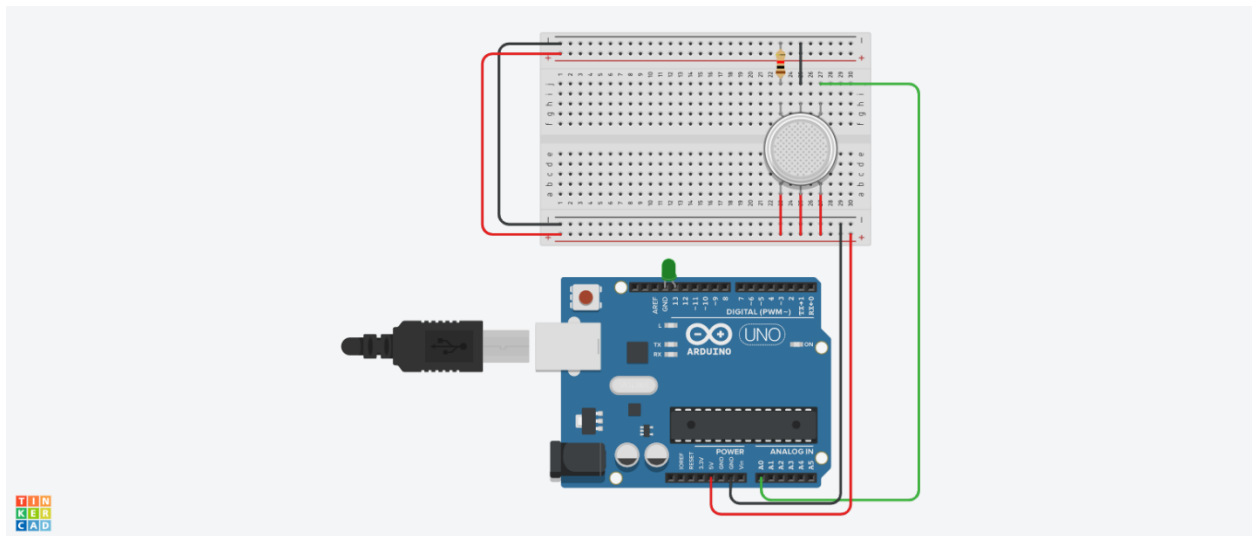
PROGRAM TITLE: GAS SENSOR

Aim: DESIGN A SMART GAS LEAKAGE INDICATOR SYSTEM (GAS SENSOR AND LED)

Hardware Required:

- Arduino Board
- LED
- Breadboard
- Gas Sensor
- Resistor

Circuit Diagram:



Write-Up:

NAME: Sakshi Mishra VCN: 18M18C0040 Date: 14th Oct, 2020
Expt. No. 11 Page No. 17

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

HARDWARE REQUIRED:

- Arduino Board
- Breadboard
- Gas sensor
- LED
- Resistors

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)
```

Teacher's Signature: _____

Expt. No. 11 Page No. 18

```
} digitalWrite(LED, HIGH);  
  Serial.print(sensorValue);  
  Serial.println(" - SMOKE DETECTED");  
  delay(sensorValue);  
}  
else  
{ digitalWrite(LED, LOW);  
  Serial.println(" sensor Value :");  
  Serial.println(sensorValue);  
}  
  delay(1000);  
}
```

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("Sensor
Value: ");

    Serial.println(sensorValue);
  }
  delay(1000);
}
```

OBSERVATION/OUTPUT:

The gas sensor checks whether there is a leakage or not.If yes, it displays SMOKE DETECTED otherwise it displays the sensor value.

Sakshi Srivastava
1BM18CS090

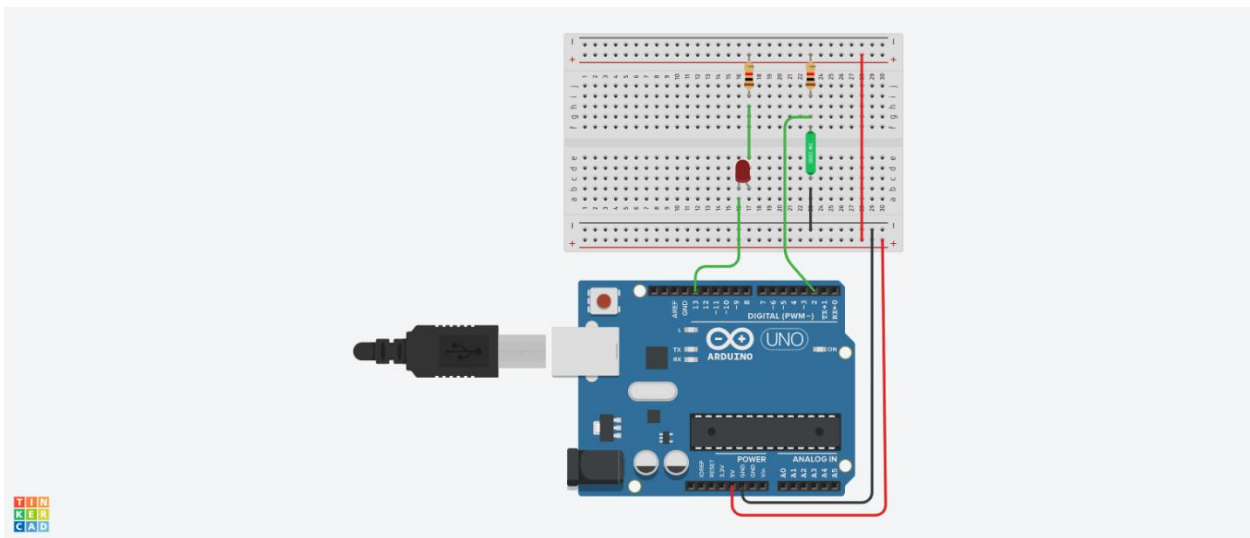
PROGRAM TITLE: TILT SENSOR

Aim: DESIGN A SMART PACKAGE HANDLING SYSTEM(TILT SENSOR AND LED)

Hardware Required:

- Tilt Sensor
- Resistor
- LED
- Breadboard
- Arduino UNO

Circuit Diagram:



Write-Up:

Expt. No.13.....	Page No. ...21.....
Aim: Design a smart package handling system (TILT SENSOR AND LED).	
HARDWARE REQUIRED:	
→ Tilt sensor	
→ Resistor	
→ LED.	
→ Breadboard	
→ Arduino Uno.	
<u>CODE.</u>	
<pre>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); }</pre>	
Teacher's Signature :	

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/Observation:

The LED light glows accordingly.

Sakshi Srivastava

1BM18CS090

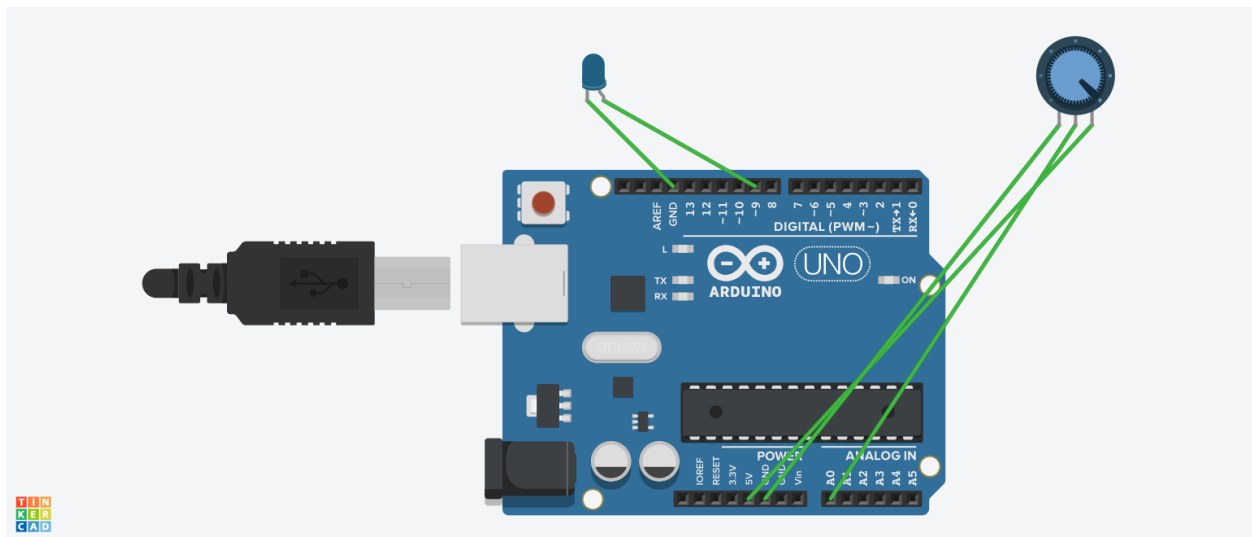
PROGRAM TITLE: LED FADING (USING POTENTIOMETER)

Aim: DEMONSTRATE LED FADING USING POTENTIOMETER

Hardware Required:

- Arduino Board
- LED
- Potentiometer

Circuit Diagram:



Write-Up:

NAME: Sakshi Srivastava	USN: 1BMIEC5090	Date: 23/9/2020
Expt. No. 5	Page No. 7	

PROGRAM TITLE: LED FADING (USING Potentiometer)

Aim: To demonstrate LED fading using potentiometer.

Hardware Required:-

- Arduino Board.
- LED.
- Potentiometer.

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);
}
```


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/OBSERVATION:

LED is fading.

Sakshi Srivastava

1BM18CS090

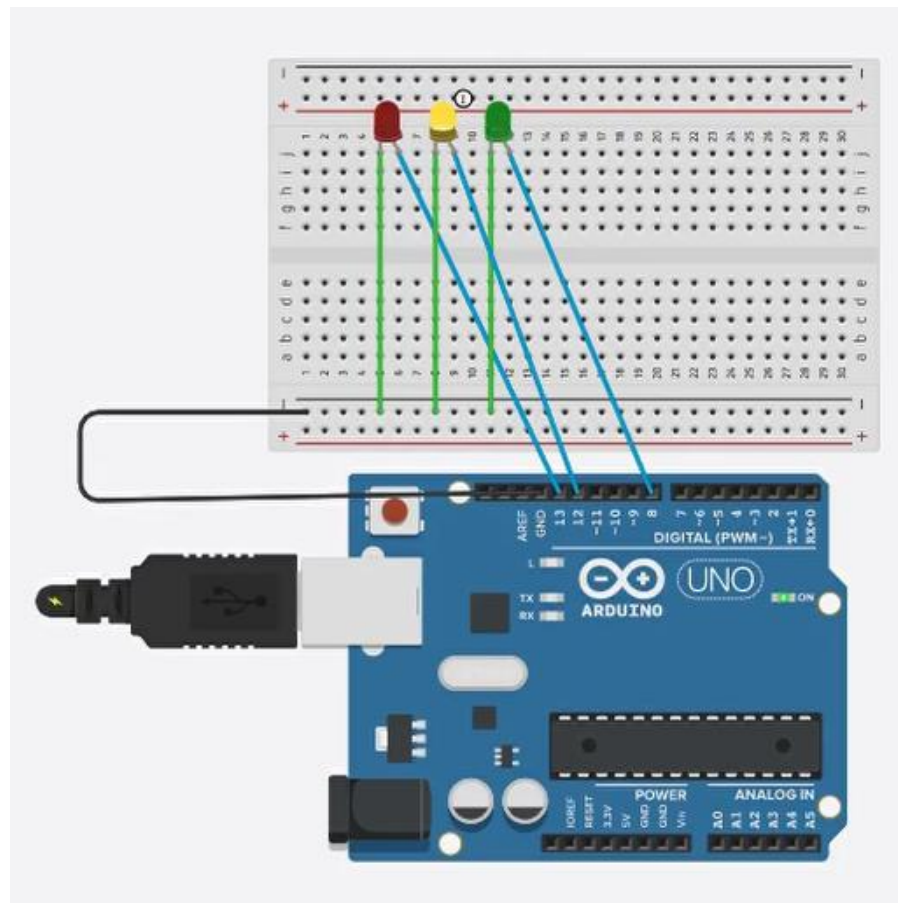
PROGRAM TITLE: TRAFFIC CONTROLLER

Aim: TRAFFIC SIGNAL STIMULATOR

Hardware Required:

- Arduino Board
- LEDs
- Breadboard

Circuit Diagram:



Write-Up:

NAME: Sakshi Srivastava		USN: 18M18CS090	Date: 14/9/2020
Expt. No. 2	Page No. 2		
<u>TRAFFIC CONTROLLER.</u>			
Aim: Traffic Signal Stimulator.			
Hardware Required:-			
<ul style="list-style-type: none">• Arduino Board• LEDs• Bread board.			
CODE:			
<pre>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); }</pre>			

```
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);
}
```

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);  
}
```

Observation /Output:

LEDs are blinking.