

# VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“GnanaSangama”, Belgaum -590014, Karnataka.



## LAB REPORT

# INTERNET OF THINGS

*Submitted by*

IMADH AJAZ BANDAY (1BM20CS059)

*in partial fulfillment for the award of the degree  
of  
BACHELOR OF ENGINEERING  
in  
COMPUTER SCIENCE AND ENGINEERING*



B.M.S. COLLEGE OF ENGINEERING  
(Autonomous Institution under VTU)  
BENGALURU-560019  
Oct 2022-Feb 2023

B. M. S. College of Engineering,  
Bull Temple Road, Bangalore 560019  
(Affiliated To Visvesvaraya Technological University, Belgaum)  
Department of Computer Science and Engineering



### **CERTIFICATE**

This is to certify that the Lab work entitled “INTERNET OF THINGS” carried out by **IMADH AJAZ BANDAY (1BM20CS059)**, who is a bonafide student of B.M.S. College of Engineering. It is in partial fulfillment for the award of Bachelor of Engineering in **Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum during the year 2022-23. The Lab report has been approved as it satisfies the academic requirements in respect of Internet of Things Lab - (20CS5PEIOT) work prescribed for the said degree.

Dr.K.Panimozhi  
Assistant Professor  
Department of CSE  
BMSCE, Bengaluru

Dr. Jyothi S Nayak  
Professor and Head  
Department of CSE  
BMSCE,  
Bengaluru

Program no: **01**

Program Title: **LED BLINK**

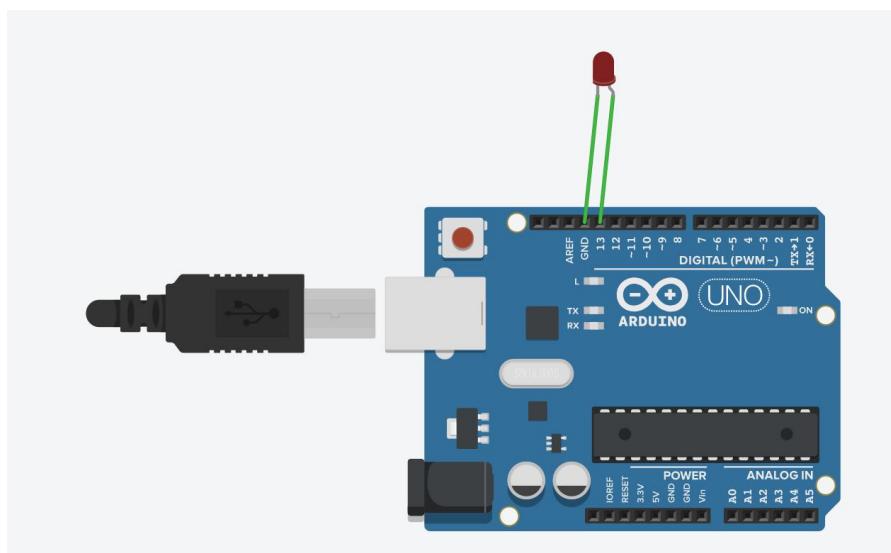
**Aim:** To control the LED using Arduino ( to turn ON/OFF LED)

**Hardware/components Required:**

- Arduino Uno board - 1
- USB Cable - 1
- LED - 1
- Jumper wires

**Circuit Diagram / Pin connection:**

- LED's positive leg is connected to digital pin 13.
- LED's negative leg is connected to ground.



## Handwritten code pic:

classmate  
Date 7/11/2021  
Page \_\_\_\_\_

Expt 1: Blink Program

Aim: To illustrate a blinking LED.

Components: Arduino uno Board, LED, jumper wires, bread board, USB cable.

Pin connections / Circuit description:

long end of LED to PIN13, short end to GND.

(code):

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

Observation: successfully observed blinking of LED at regular intervals.

**Code:**

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

**Observation:** LED switches ON/OFF periodically. Digital output visualization using Arduino Uno.

Program no: 02

Program Title: **Traffic Light Simulation**

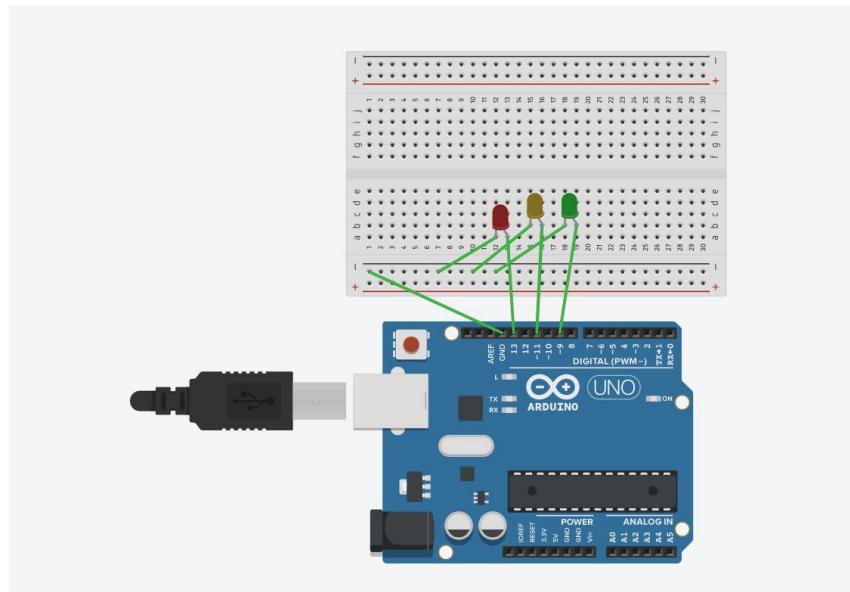
**Aim:** To simulate traffic lights by blinking of LEDs using Arduino Uno board

### **Hardware/components Required:**

- Arduino Uno board - 1
- USB Cable - 1
- LEDs - 3
- Jumper wires
- Bread Board

### **Circuit Diagram / Pin connection:**

- LED 1's positive leg is connected to digital pin 13, negative end to ground.
- LED 2's positive leg is connected to digital pin 13, negative end to ground.
- LED 3's positive leg is connected to digital pin 13, negative end to ground.



## Handwritten code pic:

Date: 3/11/2021  
Page

Exp 2: Traffic Simulation

Aim: To simulate traffic light simulation by blinking of LEDs using arduino uno board.

Components required: Arduino uno Board, USB cable, jumper wires, 3 LEDs, bread board.

Pin connection:

LED 1 - long end to Pin 9, short end to GND.  
LED 2 - long end to Pin 12, short end to GND.  
LED 3 - long end to Pin 8, short end to GND.

Code:

```
void setup() {
    pinMode(9, OUTPUT);
    pinMode(12, OUTPUT);
    pinMode(8, OUTPUT);
}

void loop() {
    digitalWrite(9, HIGH);
    delay(1000);
    digitalWrite(9, LOW);
    delay(1000);
    digitalWrite(12, HIGH);
    delay(1000);
    digitalWrite(12, LOW);
    delay(1000);
    digitalWrite(8, HIGH);
    delay(1000);
    digitalWrite(8, LOW);
    delay(1000);
}
```

(3) the author

184

~~Observation 3: successfully observed the traffic light simulation using LEDs.~~

January 1973 found under living in grassy  
raked out area

and at Nitrobus had nothing  
and at Nitrobus bus back  
up at Nitrobus waited half for bus and  
Nitrobus went away at Nitrobus bus stop

the 1st and last

(part 2) from sin

(P101T, 6) 12M 201

1000000000

↳ [View details](#) or [Edit](#)

(*Wheat, whole*) 11

*E. flava* as a light leucistic? (

W. C. Gandy, 1912, *Archaeological Survey of Canada*

623 for 400 weight transfers off forward deck  
with 100 weight transfers off aft deck

**Code:**

```
void setup()
{
    pinMode(11, OUTPUT);
    pinMode(9, OUTPUT);
    pinMode(13, OUTPUT);
}

void loop()
{
    digitalWrite(13, HIGH);
    delay(1000);
    digitalWrite(11, LOW);
    delay(1000);
    digitalWrite(9, LOW);
    delay(1000);

    digitalWrite(13, LOW);
    delay(1000);
    digitalWrite(9, HIGH);
    delay(1000);
    digitalWrite(11, LOW);
    delay(1000);

    digitalWrite(9, LOW);
    delay(1000);
    digitalWrite(11, HIGH);
    delay(1000);
    digitalWrite(13, LOW);
    delay(1000);

}
```

**Observation:** Successfully observed the traffic light simulation using LEDs.

Program no: 03

Program Title: **Push Button with LED**

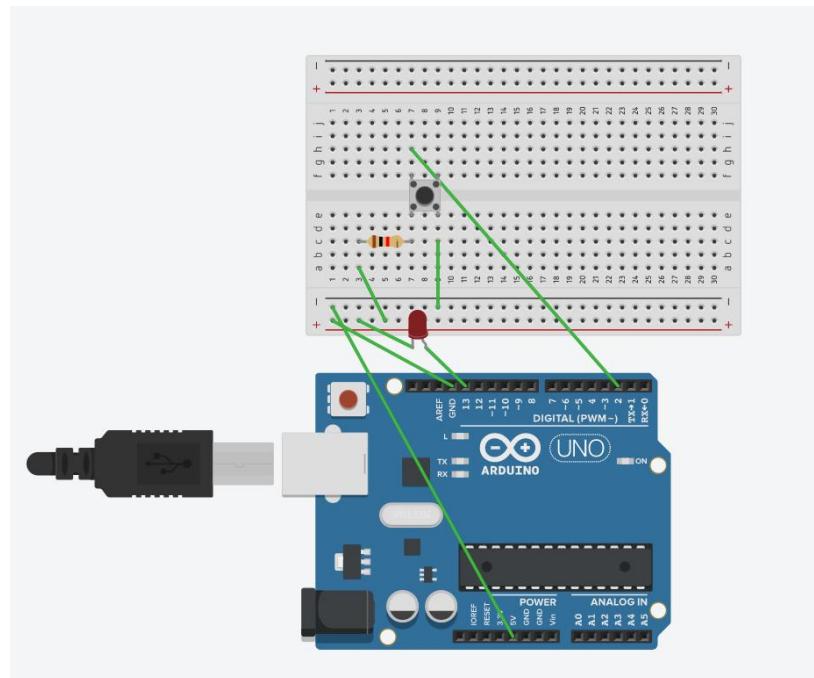
**Aim:** To simulate glowing of LEDs using a push button.

### **Hardware/components Required:**

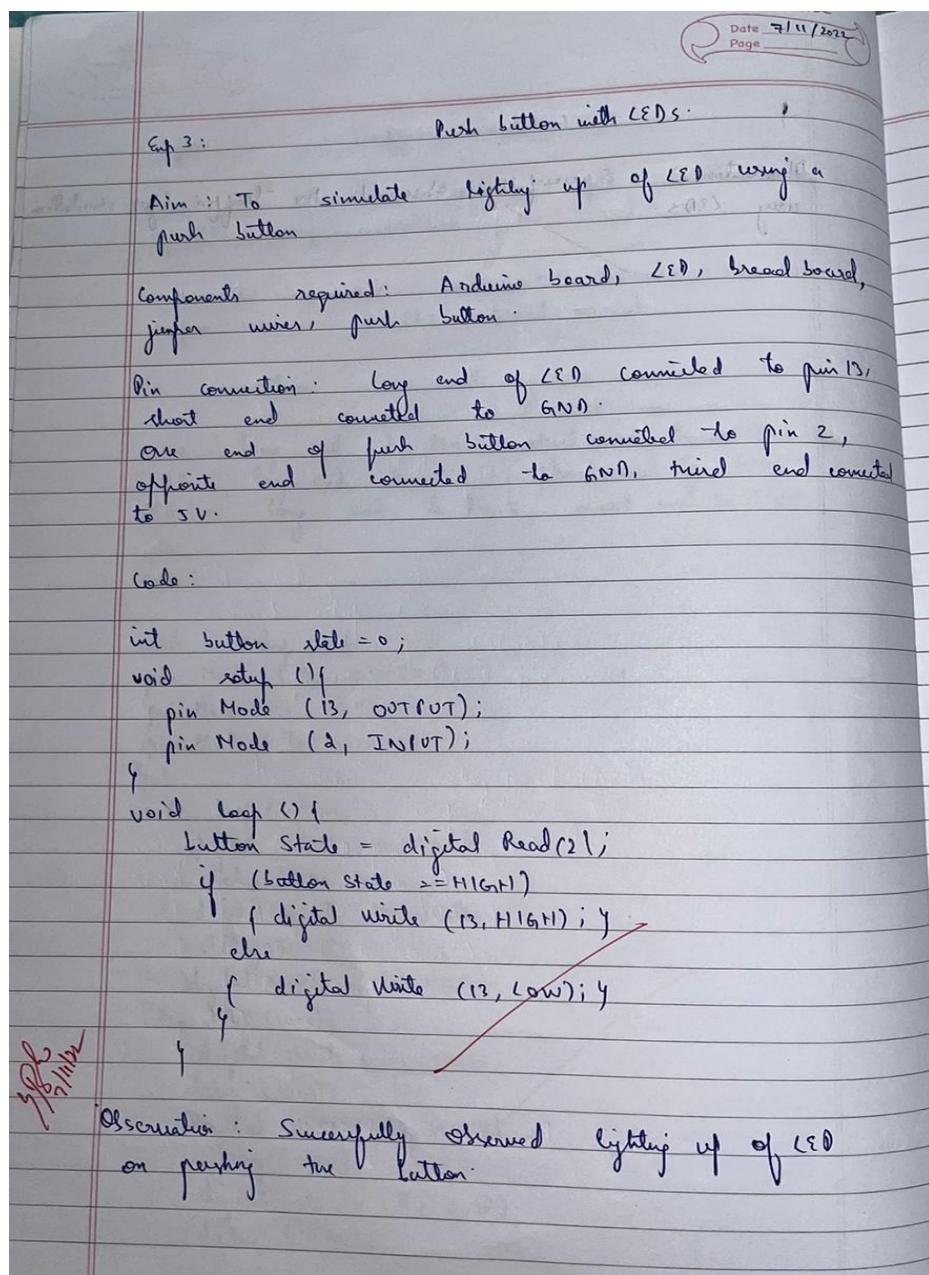
- Arduino Uno board - 1
- USB Cable - 1
- LED - 1
- Jumper wires
- Bread Board
- Push Button

### **Circuit Diagram / Pin connection:**

- LED's positive leg is connected to digital pin 13, negative end to ground.
- One end of push button connect to pin 2, opposite end connected to ground.
- Third end of push button connected to 5v supply.



### Handwritten code pic:



**Code:**

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

**Observation:** Successfully observed the glowing of LED on pushing the button.

Program no: 04

Program Title: **LED fading without Potentiometer**

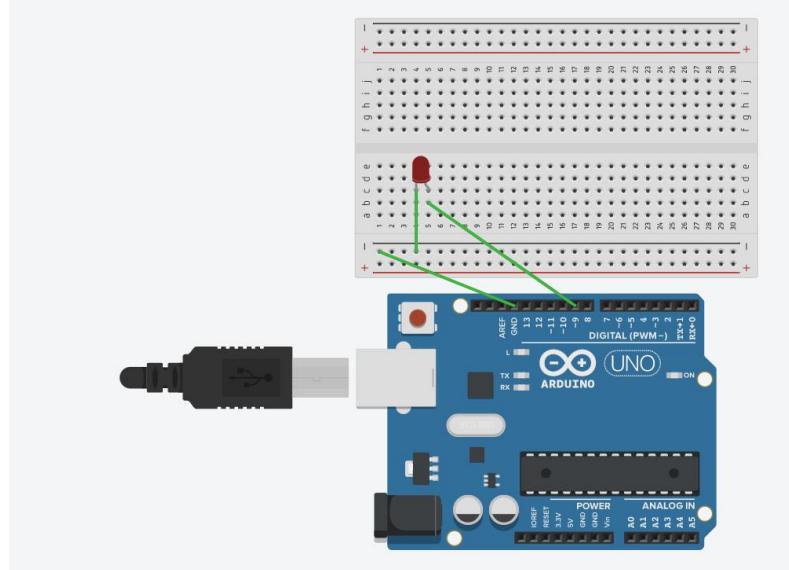
**Aim:** To simulate fading of LEDs without the use of potentiometer.

### **Hardware/components Required:**

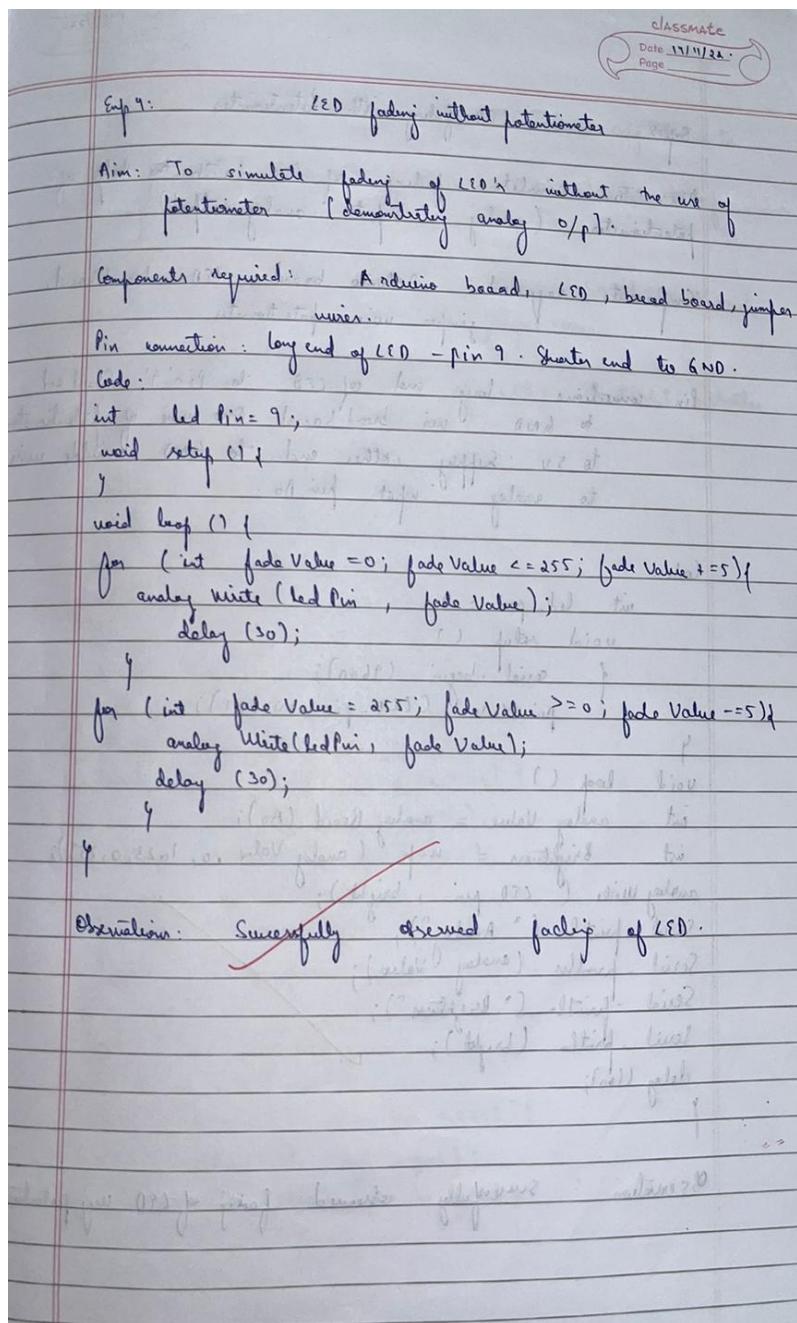
- Arduino Uno board - 1
- USB Cable - 1
- LED - 1
- Jumper wires
- Bread Board

### **Circuit Diagram / Pin connection:**

- LED's positive leg is connected to digital pin 9, negative end to ground.



## Handwritten code pic:



**Code:**

```
int ledPin = 9;

void setup() {
}

void loop() {

    for (int fadeValue = 0; fadeValue <= 255; fadeValue += 5) {
        analogWrite(ledPin, fadeValue);
        delay(30);
    }

    for (int fadeValue = 255; fadeValue >= 0; fadeValue -= 5) {
        analogWrite(ledPin, fadeValue);
        delay(30);
    }
}
```

**Observation:** Successfully observed the fading of LED without the use of potentiometer.

Program no: 05

Program Title: **LED fading with Potentiometer**

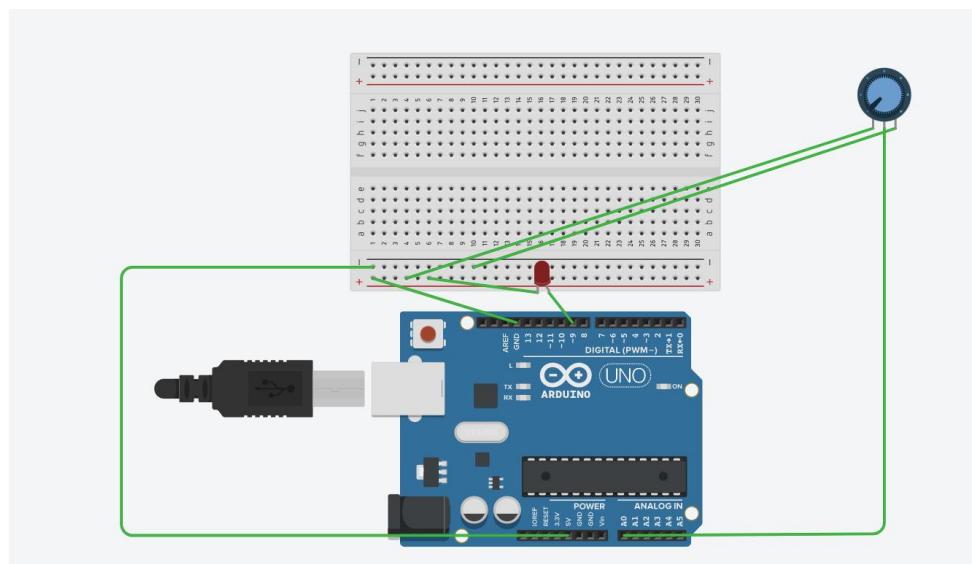
**Aim:** To simulate fading of LEDs without the use of potentiometer.

### **Hardware/components Required:**

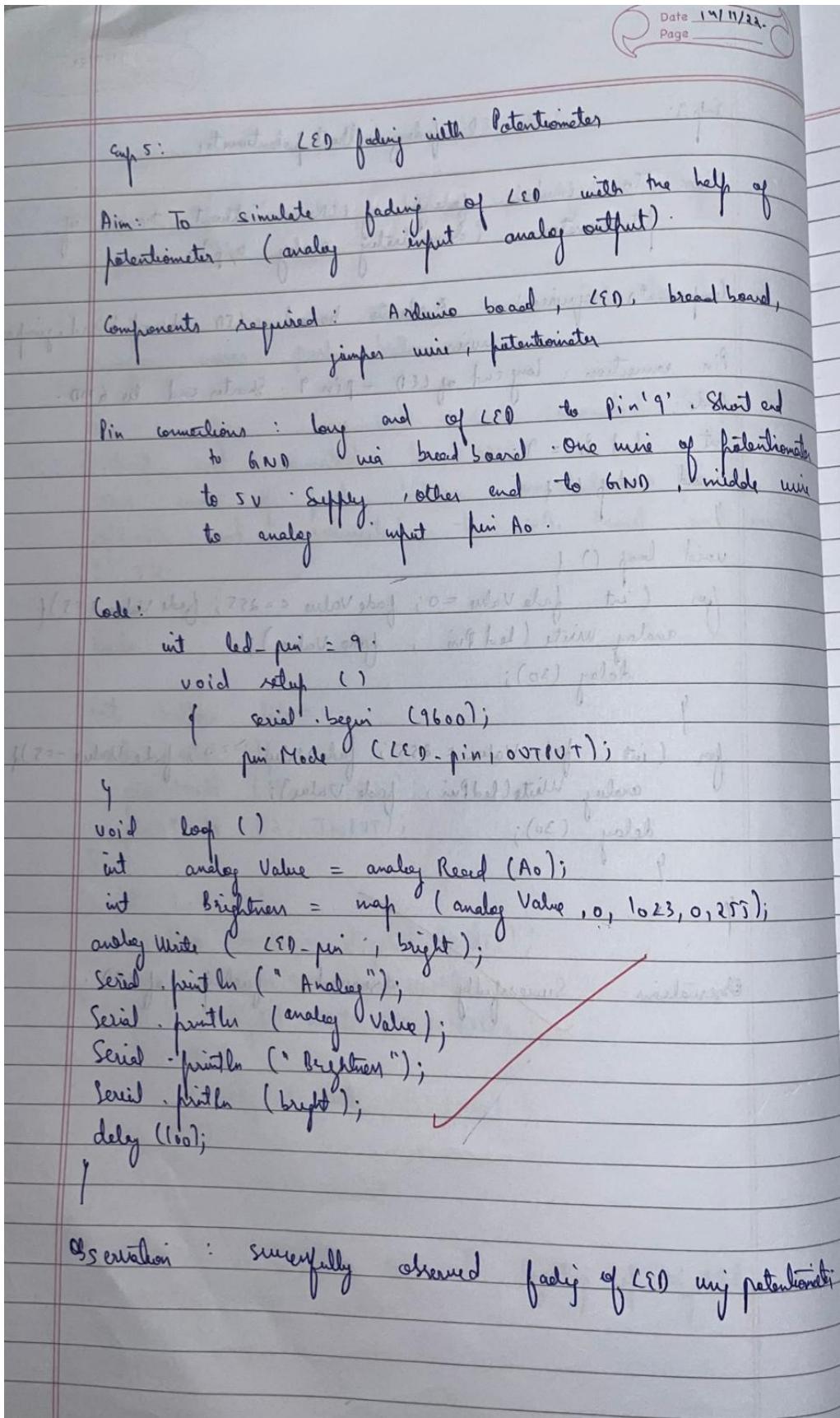
- Arduino Uno board - 1
- USB Cable - 1
- LED - 1
- Jumper wires
- Bread Board
- Potentiometer

### **Circuit Diagram / Pin connection:**

- LED's positive leg is connected to digital pin 9, negative end to ground.
- One end of potentiometer to 5v supply other end to ground.
- Middle wire of potentiometer to analog input pin A0.



## Handwritten code pic:



**Code:**

```
int ledPin = 9;
void setup()
{
Serial.begin(9600);
pinMode(ledPin,OUTPUT);
}

void loop()
{
int analogValue = analogRead(A0);
int brightness = map(analogValue,0,1023,0,255);
analogWrite(ledPin,brightness);
Serial.print("Analog");
Serial.print(analogValue);
Serial.print("brightness");
Serial.print(brightness);
delay(100);
}
```

**Observation:** Successfully observed the fading of LED by turning the shaft of the potentiometer.

Program no: **06**

Program Title: **Fading effect of two LEDs using two Potentiometers**

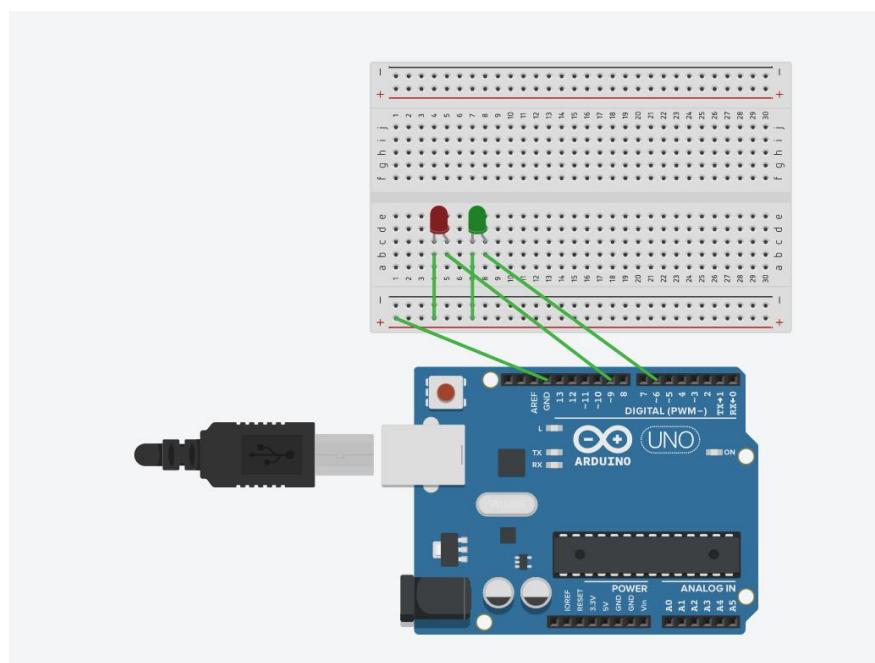
**Aim:** To simulate fading of two LEDs without the use of potentiometer.

**Hardware/components Required:**

- Arduino Uno board - 1
- USB Cable - 1
- LEDs - 2
- Jumper wires
- Bread Board

**Circuit Diagram / Pin connection:**

- LED 1's positive leg is connected to digital pin 9, negative end to ground.
- LED 2's positive leg is connected to digital pin 6, negative end to ground.



### Handwritten code pic:

CLASSMATE  
Date 14/10/22.  
Page \_\_\_\_\_

Exp 6: Fading Effect of 2 LEDs without using Potentiometer

Aim: To simulate fading of 2 LEDs without use of potentiometer.

Components required: 2 LED's, arduino board, breadboard, connecting cable, jumper wires.

Pin connections: Keep each of 2 LEDs to Pin 9, shorter end to GND via breadboard!

Code:

```
int pin 1 = 9;
int pin 2 = 6;
void setup () {
    }
void loop () {
    int fv1 = 0;
    int fv2 = 255;
    while (fv1 <= 255 && fv2 >= 0) {
        analogWrite (pin 1, fv1);
        analogWrite (pin 2, fv2);
        fv1 += 5;
        fv2 -= 5;
        delay (70);
    }
    fv1 = 255;
    fv2 = 0;
    while (fv1 >= 0 && fv2 <= 255) {
        analogWrite (pin 1, fv1);
        analogWrite (pin 2, fv2);
        fv1 -= 5;
        fv2 += 5;
        delay (70);
    }
}
```

**Code:**

```
int ledpin1 = 9;
int ledpin2 = 6;
void setup(){
}

void loop()
{
int fadevalue1 =0;
int fadevalue2 =255;

while(fadevalue1 <=255&& fadevalue2>=0)
{
analogWrite(ledpin1,fadevalue1);
analogWrite(ledpin2,fadevalue2);
fadevalue1 +=5;
fadevalue2 -=5;
delay(70);
}

fadevalue1 =255;
fadevalue2 =0;
while(fadevalue1>=0 && fadevalue2<=255)
{
analogWrite(ledpin1,fadevalue1);
analogWrite(ledpin2,fadevalue2);
fadevalue1 -=5;
fadevalue2 +=5;
delay(70);
}
}
```

**Observation:** Successfully observed the fading of two LEDs alternatively without using potentiometer.

Program no: **07**

Program Title: **LDR with LED**

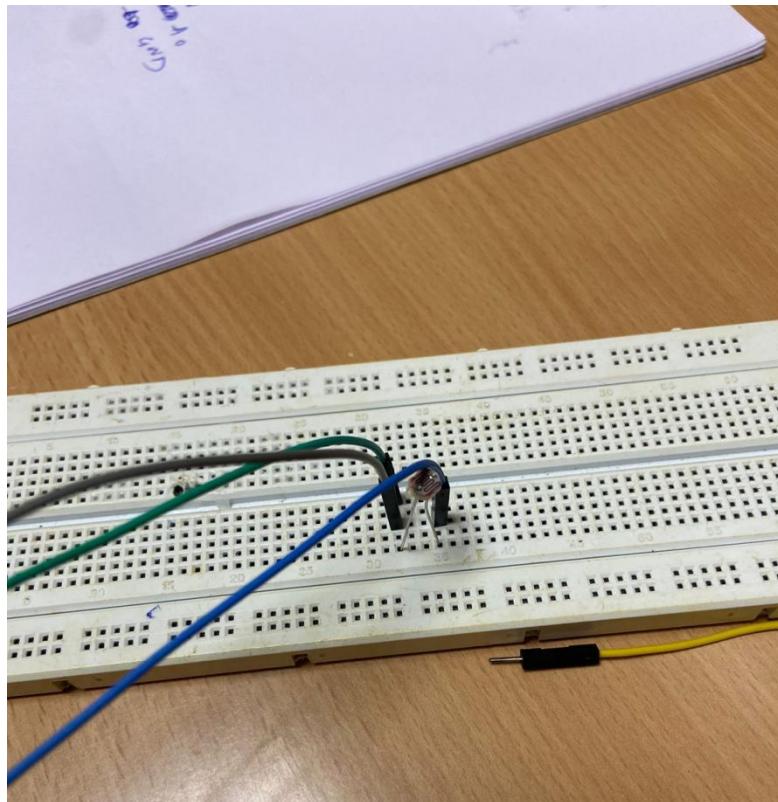
**Aim:** To observe the slow of LED when the LDR is covered.

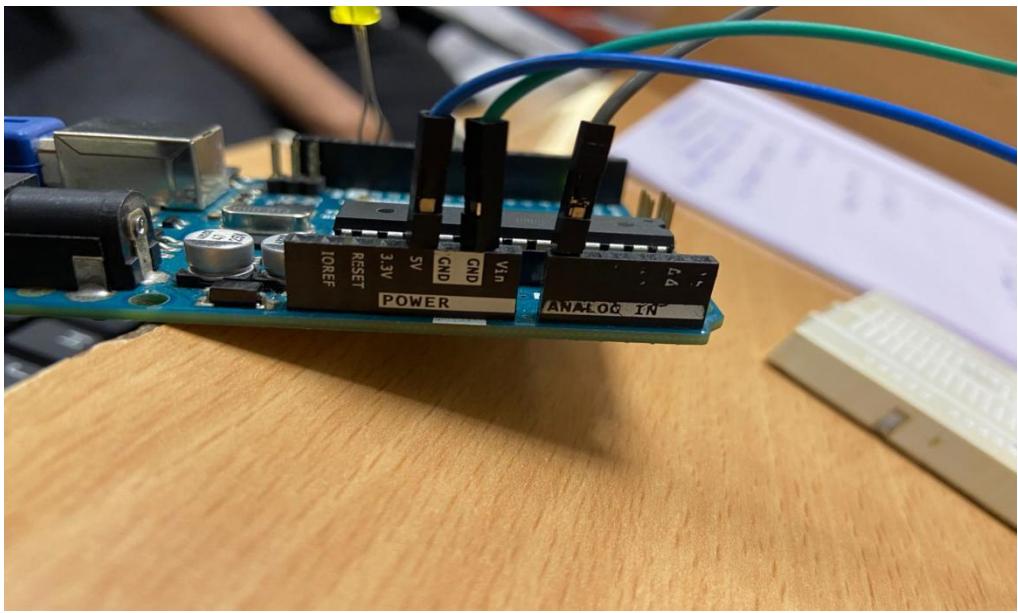
### **Hardware/components Required:**

- Arduino Uno board - 1
- USB Cable - 1
- LED - 1
- Jumper wires
- Bread Board
- LDR - 1

### **Circuit Diagram / Pin connection:**

- Connect shorter end of LED to ground, longer end of LED to pin 13
- Connect shorter end of LDR to A0 pin as well as ground via breadboard
- Connect longer end of LDR to 5v supply





### Handwritten code pic:

CLASSMATE  
Date: 19/11/22.  
Page.

Exp 7: LDR with LED.

Aim: To observe the glow of LED when LDR is covered.

Components required: Arduino uno board, breadboard, 4 jumper wires, LDR, LED, USB cable,

Pin connections: Connect LDR on breadboard to shorter end. Connect a wire to A0 pin and another to GND. From the longer end of the LDR connect to the 5V supply. Connect LED to pin 13 and GND.

Code:

```
int value = 0;  
void setup () {  
    pinMode (13, OUTPUT);  
    pinMode (A0, INPUT);  
}  
  
void loop () {  
    value = analogRead (A0);  
    if (value < 200)  
        {  
            digitalWrite (13, HIGH);  
            Serial.println ("Light on");  
            Serial.println ("Value");  
        }  
    else  
        {  
            digitalWrite (13, LOW);  
            Serial.println ("Light OFF");  
            Serial.println (value);  
        }  
}
```

**Code:**

```
int value =0;
void setup()
{
    Serial.begin(9600);
    pinMode(13,OUTPUT);
    pinMode(A0,INPUT);
}

void loop()
{
    value = analogRead(A0);
    if(value<200)
    {
        digitalWrite(13,HIGH);
        Serial.println("light on");
        Serial.println(value);
    }

    else
    {
        digitalWrite(13,LOW);
        Serial.println("light off");
        Serial.println(value);
    }
}
```

**Observation:** Successfully observed the glowing of LED when the LDR was covered.

Program no: **08**

Program Title: **LM35 Temperature Sensor**

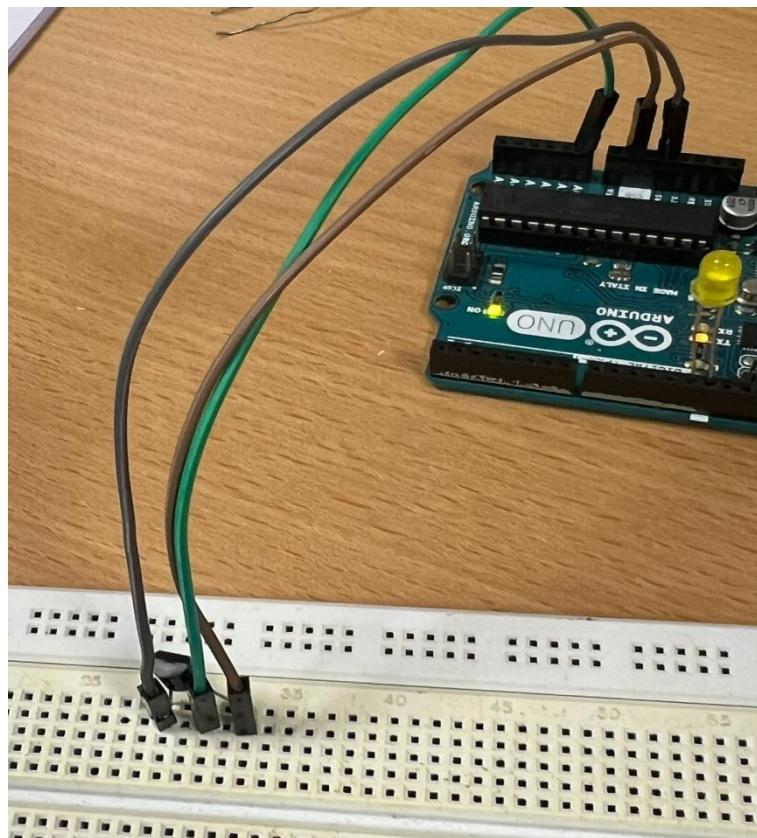
**Aim:** To observe the working of temperature sensor and convert to Fahrenheit

**Hardware/components Required:**

- Arduino Uno board - 1
- USB Cable - 1
- LM35 Temperature Sensor
- Jumper wires
- Bread Board

**Circuit Diagram / Pin connection:**

- Place the temperature sensor flat side facing us.
- Right end of LM35 will be grounded, left end will be connected to 5v supply.
- Middle end will be connected to A0 pin.



## Handwritten code pic:

CLASSMATE  
Date 17/11/22  
Page

Exp 8: LM35 temperature sensor

Aim: To observe the working of temperature sensor and calculation of temperature.

Components required: LM35 temperature sensor, 3 jumper wires, breadboard, arduino uno board.

Pin connection: Connect the temperature sensor flat side facing up. Right end will be grounded. Left side will be connected to 5V. Middle wire will be connected to A0 pin.

Code:

```
int op = 0;
void setup () {
    Serial.begin (9600);
}
void loop () {
    int raw voltage = analogRead (outputPin);
    float millivolt = (raw voltage / 1024.0) * 5000;
    float celsius = millivolt / 10;
    Serial.println (celsius);
    Serial.println ("degree celsius");
    Serial.println ((celsius * 9) / 5 + 32);
    Serial.println ("degree fahrenheit");
    delay (1000);
}
```

Observation: Successfully observed working of temperature sensor and calculation of temperature in fahrenheit.

*S.B.  
17/11/22*

**Code:**

```
int outputpin=0;

void setup()
{
Serial.begin(9600);
}
Void loop()
{
int rawvoltage=analogRead(outputpin);
float millivolts=(rawvoltage / 1024.0) * 5000;
float celsius - millivolts/10;
serial.print(celsius);
serial.print("degree celsius");
serial.print((celsius * 9)/5 + 32);
serial.print("degrees fahrenheit");
Delay(1000);
}
```

**Observation:** Successfully observed the calculation of temperature in Fahrenheit.

Program no: 9

**Program Title: Night Light Simulation with Human Presence Detection**

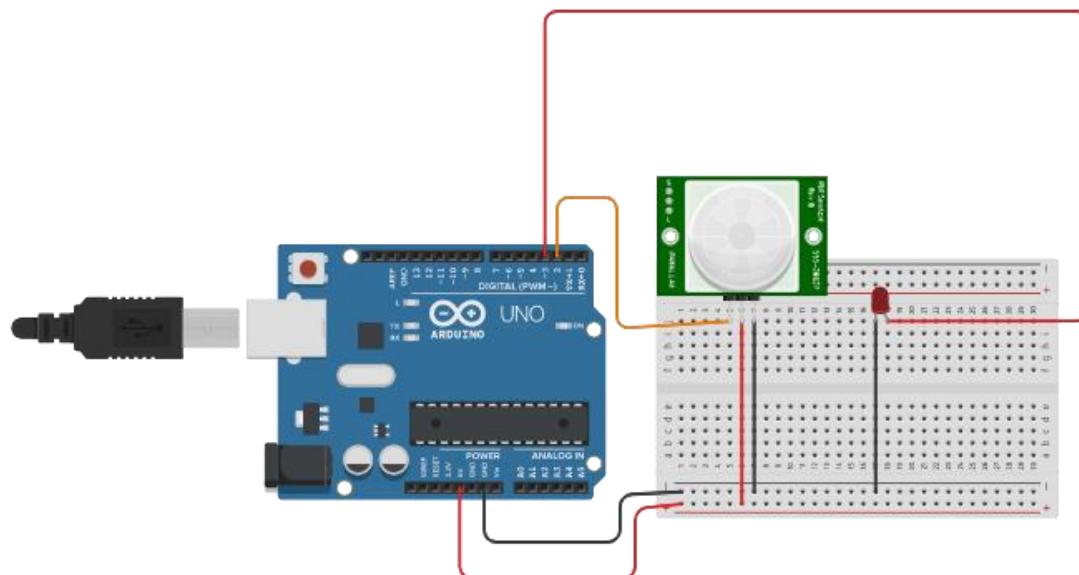
**Aim:** To observe night light simulation with human presence detection using LDR and PIR.

**Hardware/components Required:**

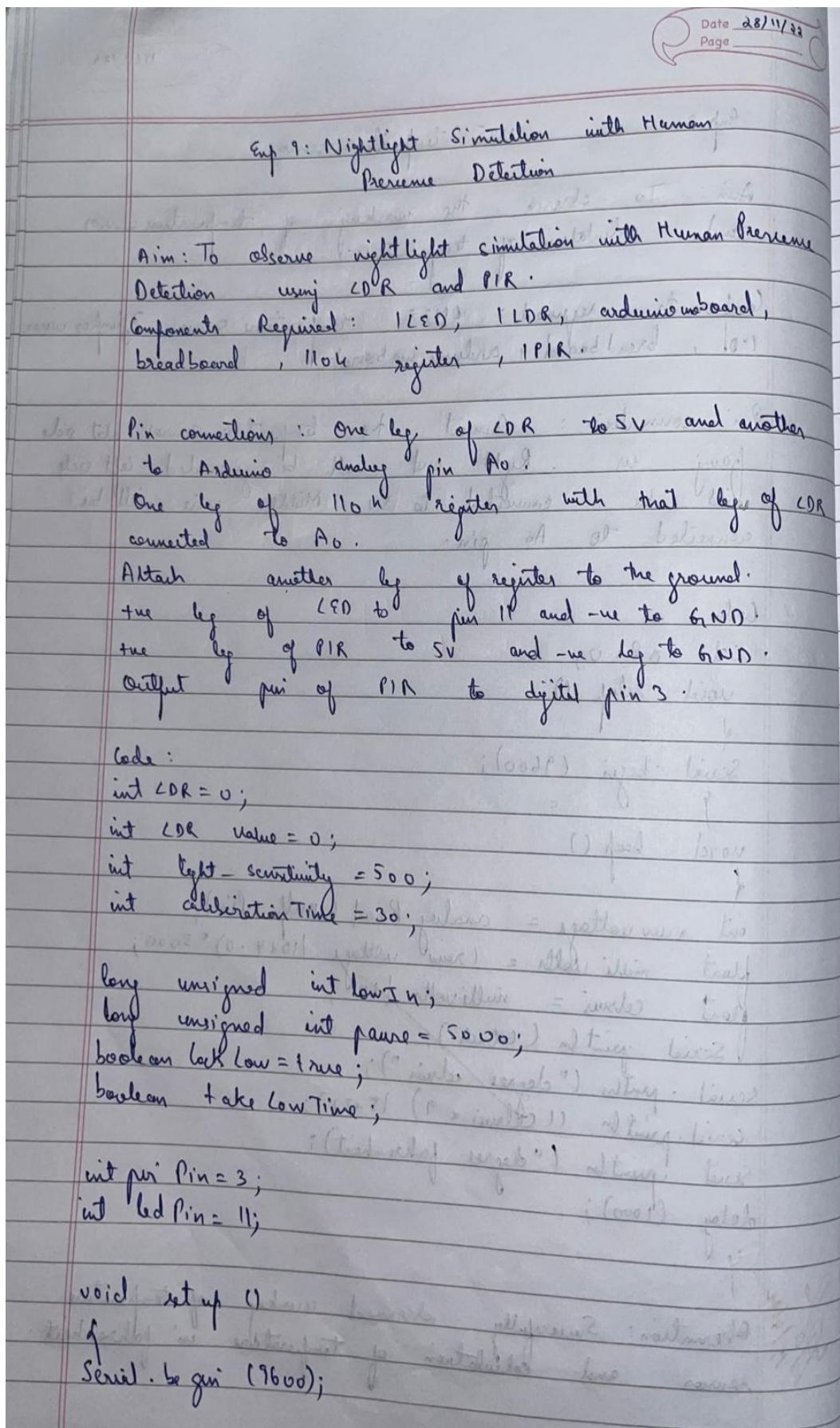
- Arduino Uno board - 1
- USB Cable - 1
- LEDs - 3
- Jumper wires
- Bread Board
- LDR - 1
- PIR - 1
- 110k Resistor

**Circuit Diagram / Pin connection:**

- Attach one leg of LDR to 5V and another leg to Arduino Analog pin A0
- Attach one leg of 110K register with that leg of LDR connected to A0
- Attach another leg of register to the ground
- Connect the positive leg of LED to pin 11 and negative to GND
- Connect positive leg of PIR to 5V and negative leg to GND
- Connect output pin of PIR to digital pin 3



### Handwritten code pic:



```
pinMode (11, OUTPUT);
pinMode (pinPin, INPUT);
pinMode (ledPin, OUTPUT);
digitalWrite (pinPin, LOW);

Serial . print ("Calibrating sensor");
for (int i = 0; i < Calibration Time; i++) {
    Serial . print (".");
    delay (1000);
}

Serial . println ("done");
Serial . println ("Sensor Active");
delay (50);

void loop () {
    LDRValue = analogRead (LDR);
    if (digitalRead (pinPin) == HIGH && LDRValue < light_sensitivity) {
        digitalWrite (ledPin, HIGH);
        if (darkLow) {
            darkLow = false;
            Serial . printn ("---");
            Serial . print ("motion detected at");
            Serial . print (millis () / 1000);
            Serial . println (" sec");
            delay (50);
        }
        takeLowTime = true;
    }
    if (digitalRead (pinPin) == low) | LDRValue >= light_sensitivity) {
        digitalWrite (ledPin, low);
        if (takeLowTime) {
            lowIn = millis ();
        }
    }
}
```

**Code:**

```
int LDR = 0;
int LDRValue = 0;
int light_sensitivity = 500;
int calibrationTime = 30;

long unsigned int lowIn;
long unsigned int pause = 5000;
boolean lockLow = true;
boolean takeLowTime;

int pirPin = 3;
int ledPin = 11;

void setup()
{
    Serial.begin(9600);
    pinMode(11, OUTPUT);

    pinMode(pirPin, INPUT);
    pinMode(ledPin, OUTPUT);
    digitalWrite(pirPin, LOW);

    Serial.print("calibrating sensor ");

    for(int i = 0; i<calibrationTime; i++){
        Serial.print(".");
        delay(1000);
    }
    Serial.println(" done");
    Serial.println("SENSOR ACTIVE");
    delay(50);
}

void loop()
{
    LDRValue = analogRead(LDR);
    Serial.println(LDRValue);
    if(digitalRead(pirPin) == HIGH && LDRValue < light_sensitivity){
        digitalWrite(ledPin, HIGH);
        if(lockLow){
            lockLow = false;
            Serial.println("---");
            Serial.print("motion detected at ");
            Serial.print(millis()/1000);
            Serial.println(" sec");
            delay(50);
        }
        takeLowTime = true;
    }
}
```

```

}

if(digitalRead(pirPin) == LOW || LDRValue >= light_sensitivity){
    digitalWrite(ledPin, LOW);
    if(takeLowTime){
        lowIn = millis();
        takeLowTime = false;
    }

    if(!lockLow&&millis() - lowIn> pause){

        lockLow = true;
        Serial.print("motion ended at ");
        Serial.print((millis() - pause)/1000);
        Serial.println(" sec");
        delay(50);
    }
    delay(100);
}
}

```

**Observation:** Successfully observed the glowing of LED when LDR was covered and motion is sensed by the PIR.

Program no: 10

Program Title: **Ultra Sonic Sensor**

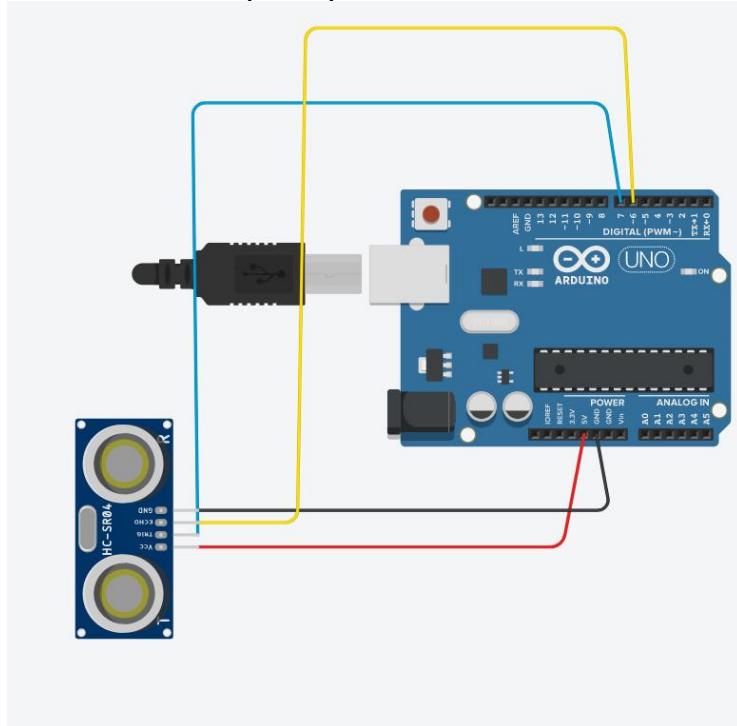
**Aim:** To observe and understand the working of the ultra sound sensor.

### **Hardware/components Required:**

- Arduino Uno board - 1
- USB Cable - 1
- Jumper wires
- Bread Board
- HC-SR04

### **Circuit Diagram / Pin connection:**

- Connect ground of HC-SR04 to ground of Arduino
- Connect Vcc of HC-SR04 to 5v supply of Arduino
- Connect trig pin to pin 7
- Connect echo pin to pin 6



### Handwritten code pic:

classmate  
Date 25/11/22  
Page

Exp 10: Ultra Sonic Sensor.

Aim: To find distance using Ultra sonic sensor.

Components Required : Arduino uno board, Ultra sonic sensor, jumper wires.

Pin connections: Vcc of ultrasonic to +5V  
Trig pin to pin 7 (digital)  
Echo pin to pin 6 (digital)  
GND to GND.

Code :

```
const int pingPin = 9;  
const int echoPin = 6;  
long duration;  
int inches;  
long cm;  
  
void setup ()  
{  
    pinMode (pingPin, OUTPUT);  
    pinMode (echoPin, INPUT);  
    Serial.begin (9600);  
}  
  
void loop ()  
{  
    digitalWrite (pingPin, LOW);  
    delayMicroseconds (1);  
    digitalWrite (pingPin, HIGH);  
    delayMicroseconds (10);  
    digitalWrite (pingPin, LOW);  
    duration = pulseIn (echoPin, HIGH);  
    inches = duration / 74 / 2;
```

**Code:**

```
const int pingPin = 7;
echoPin = 6;
void setup()
{
    Serial.begin(9600);
    pinMode(pingPin, OUTPUT);
    pinMode(echoPin, INPUT);
}
void loop()
{
    long duration, inches, cm;
    digitalWrite (pingPin, LOW);
    delayMicroseconds(2);
    digitalWrite (pingPin, HIGH);
    delayMicroseconds(10);
    digitalWrite (pingPin, LOW);
    duration = pulseIn(echoPin, HIGH);
    inches = microsecondsToInches(duration);
    cm = microsecondsToCentimeters(duration);
}

Long microsecondsToInches(long microseconds)
{
    return microseconds * 174 / 2;
}

Long microsecondsToCentimeters(long microseconds)
{
    return microseconds * 29 / 2;
}

Long microsecondsToCentimeters(long microseconds)
{
    return /29/2;
}
```

**Observation:** Successfully observed the distance of the obstacle from the sensor in cm.

Program no: **11**

Program Title: **Fire Alert**

**Aim:** To simulate the working of fire alarm using buzzer and LED.

**Hardware/components Required:**

- Flame sensor (Analogue Output)
- Arduino
- Bread board
- LED
- Buzzer
- Connecting wires

**Circuit Diagram / Pin connection:**

**Flame sensor interfacing to Arduino**

Flame sensor to Arduino

vcc -> vcc

gnd -> gnd

A0 -> A0

**Led interfacing to Arduino**

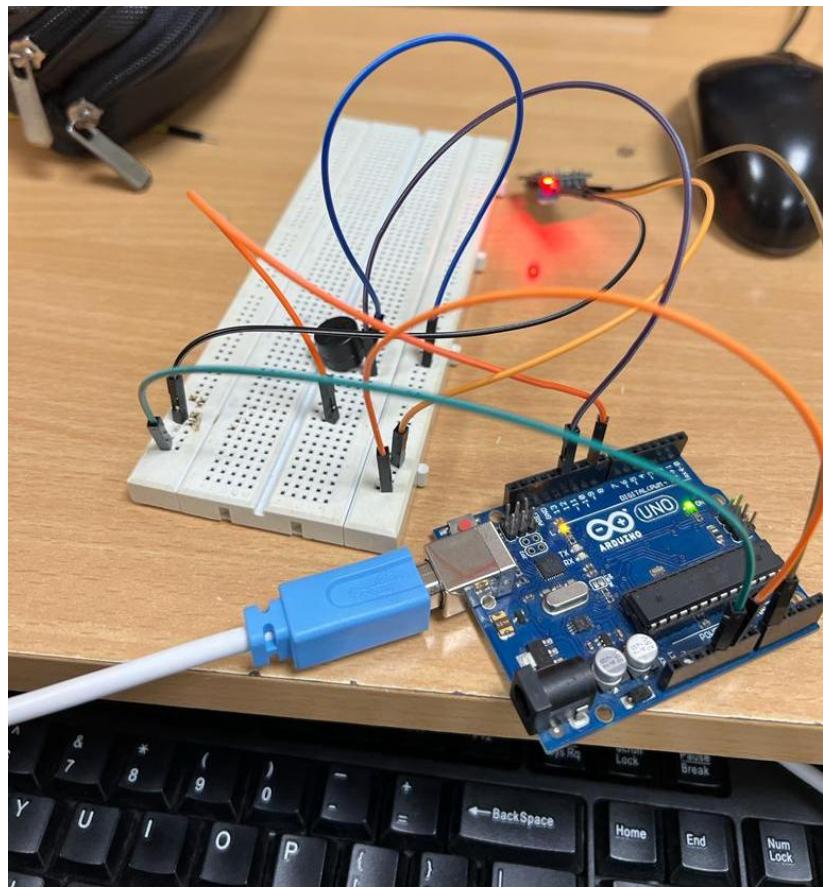
**LED +ve** is connected to **9th pin** of Arduino

**LED -ve** is connected to **gnd pin** of arduino

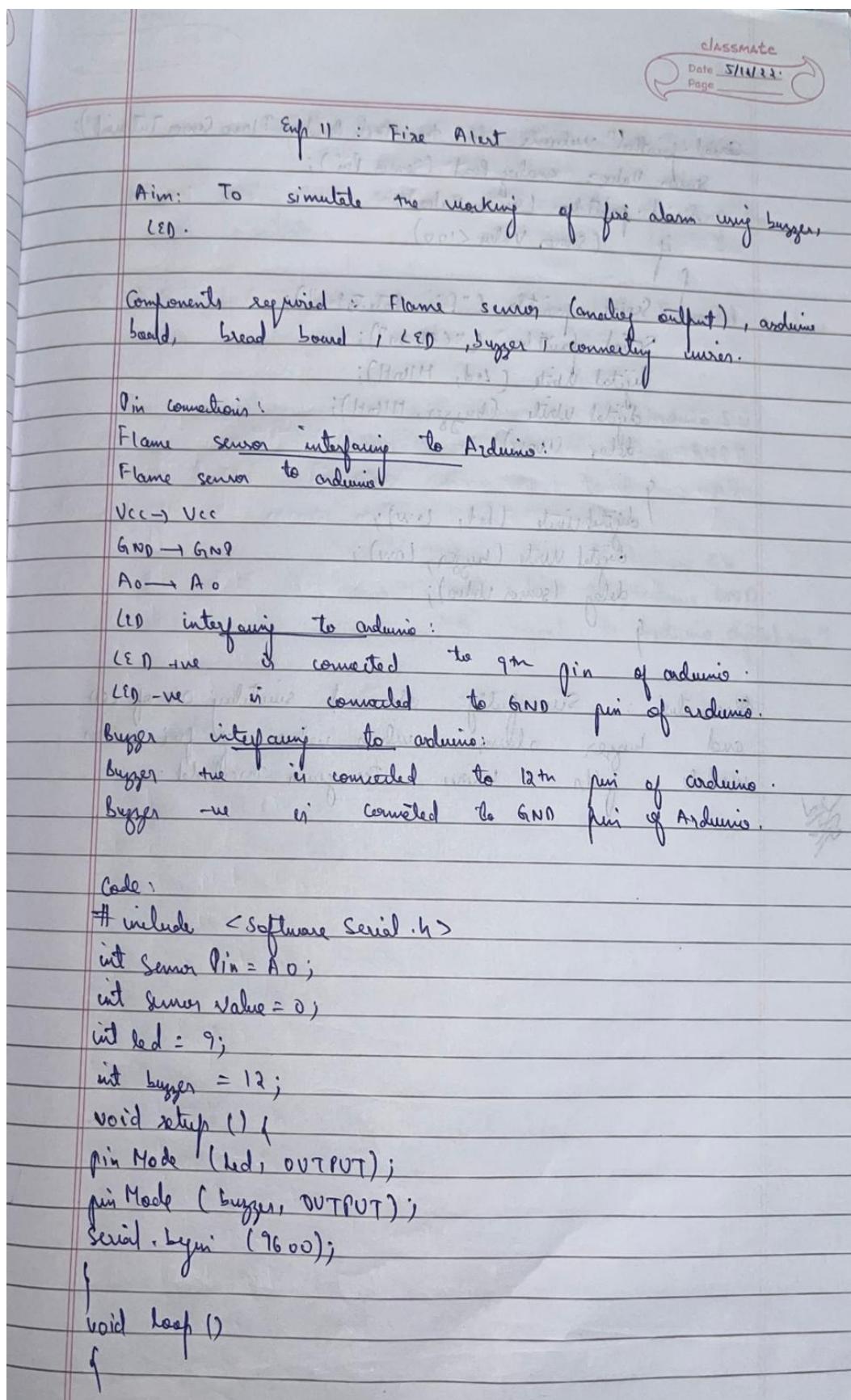
**Buzzer interfacing to Arduino**

**Buzzer +ve** is connected to **12th pin** of Arduino

**Buzzer -ve** is connected to **GND** pin of Arduino



## Handwritten code pic:



**Code:**

```
#include<SoftwareSerial.h>

int sensorPin = A0;
int sensorValue = 0;
int led = 9;
int buzzer = 12;

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

void loop()
{
  Serial.println("Welcome to TechPonder Flame Sensor Tutorial");
  sensorValue = analogRead(sensorPin);
  Serial.println(sensorValue);
  if (sensorValue < 100)
  {
    Serial.println("Fire Detected");
    Serial.println("LED on");
    digitalWrite(led,HIGH);
    digitalWrite(buzzer,HIGH);
    delay(1000);
  }
  digitalWrite(led,LOW);
  digitalWrite(buzzer,LOW);
  delay(sensorValue);
}
```

**Observation:** Successfully observed the blinking of LED and buzzing of buzzer when flame was detected by flame sensor.

Program no: 12

Program Title: **Automatic Irrigation Controller Simulation**

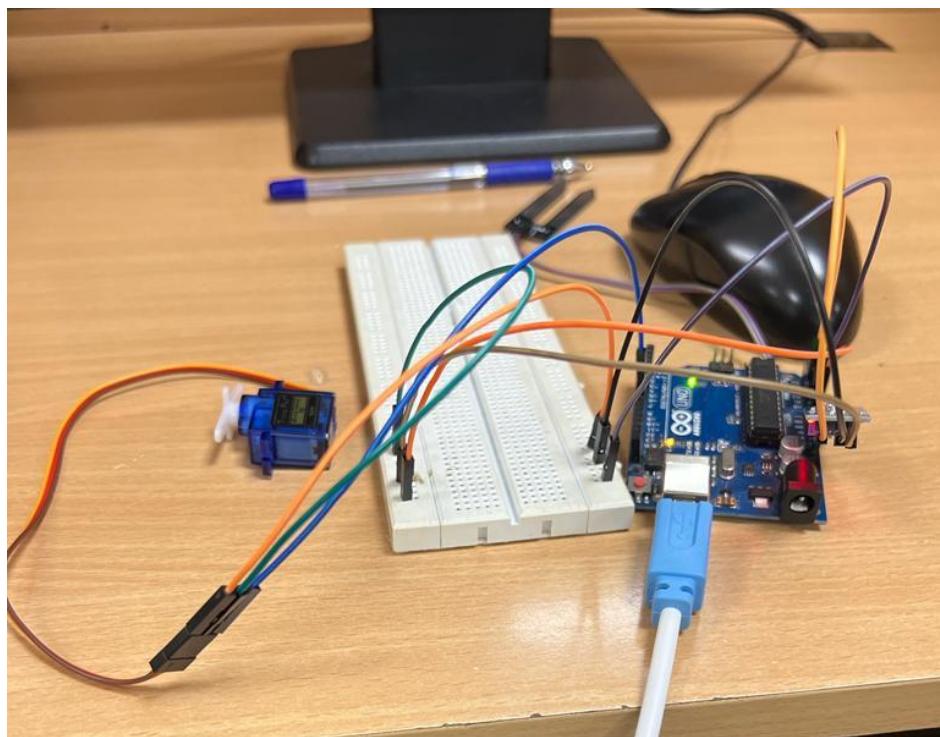
**Aim:** Sensing the soil moisture and sprinkling the water simulation.

**Hardware/components Required:**

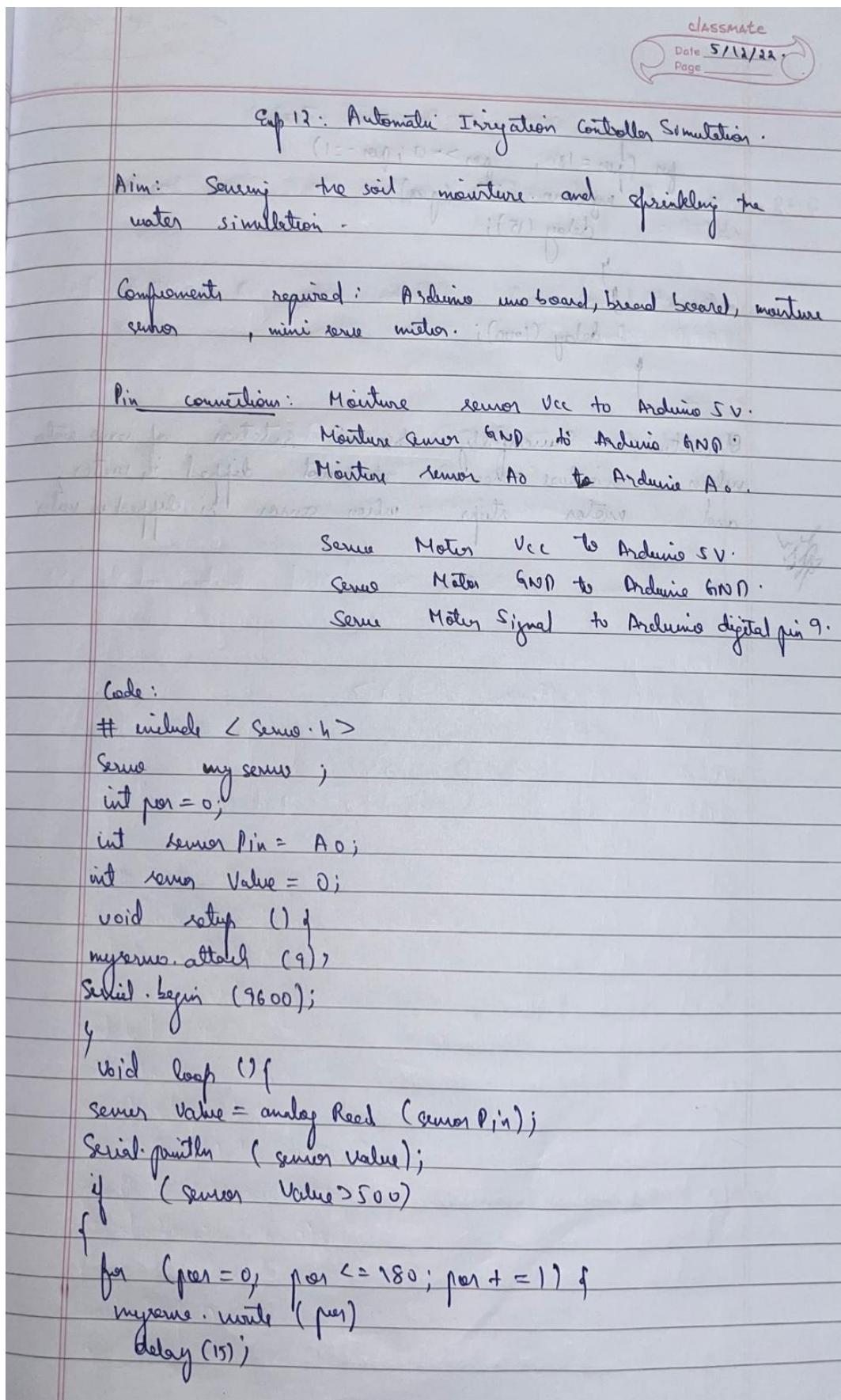
- Arduino Uno board - 1
- USB Cable - 1
- Moisture Sensor
- Mini Servo Motor
- Jumper wires
- Bread Board

**Circuit Diagram / Pin connection:**

- Moisture sensor VCC to Arduino 5V
- Moisture sensor GND to Arduino GND
- Moisture sensor A0 to Arduino A0
  
- Servo motor VCC to Arduino 5V
- Servo motor GND to Arduino GND
- Servo Motor Signal to Arduino digital pin 9



## Handwritten code pic:



for ( $\mu$ ns = 180;  $\mu$ ns >= 0;  $\mu$ ns -= 1)  
my servo. write ( $\mu$ ns);  
delay (15);  
  
delay (1000);

Observation: Successfully observed rotation of some rods when moisture sensor is not dipped in water, and water steps when sensor is dipped in water.

223

• Complex analysis of  $\log^2 z$  with  $z \neq 0$

1.  $\frac{1}{2} \times \frac{1}{2} = \frac{1}{4}$

1. What is the value?

$$d = \frac{1}{2} \cdot 10^3 \text{ m} = 500 \text{ m}$$

10月 26日 (金) 晴

b (1) (A) (1)

(a) ~~Reutter~~ ~~revised~~

*(1000) 1962-1963*

300-100-100

~~values~~ - values

1986-12

$\text{f} = \text{f}_0 + \text{f}_1$

Yester. evening

( $\mathcal{L}_\alpha$ )  $\rightarrow$  ( $\mathcal{L}_\beta$ )

**Code:**

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

int pos = 0;
int sensorPin = A0;
int sensorValue = 0;
void setup() {
  myservo.attach(9);
  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);
}
```

**Observation:** Successfully observed the rotation of motor when the moisture sensor is not in contact with water. And observed the stopping of motor when the moisture sensor is in contact with water.

Program no: **13**

Program Title: **RFID Reader**

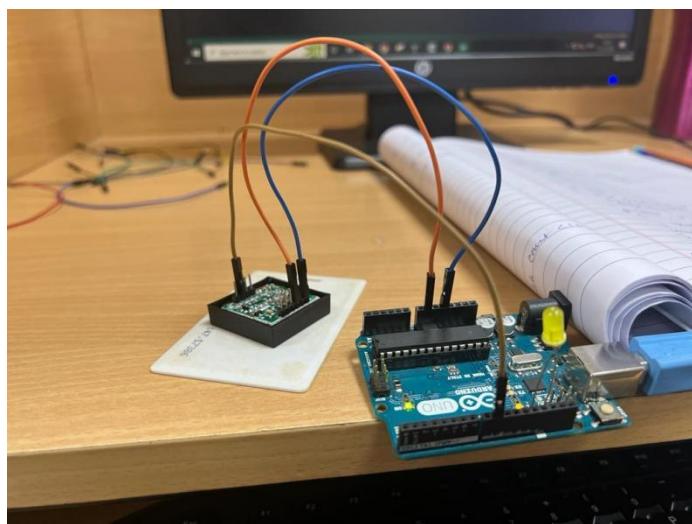
**Aim:** To count the number of RFID tags read by the RFID reader

### **Hardware/components Required:**

- Arduino Uno board - 1
- USB Cable - 1
- Jumper wires
- Bread Board
- RFID Reader module
- RFID Tags

### **Circuit Diagram / Pin connection:**

- Ground of RFID to ground of Arduino Uno board.
- TX of RFID to RX(9) of Arduino.



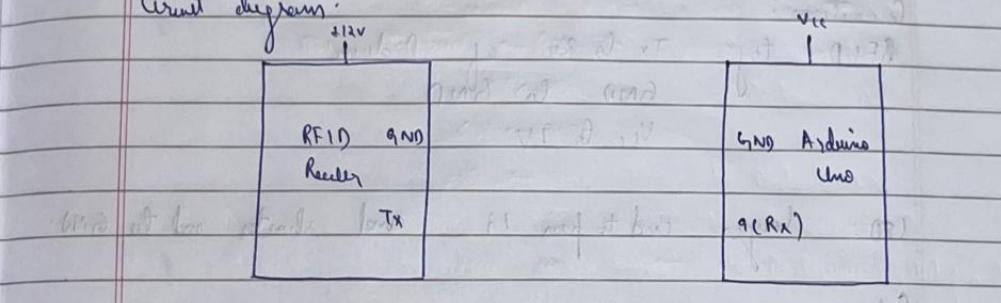
### **Handwritten code pic:**

## Exp 13. RFID Reader

Aim: To count number of RFID tags read by the RFID reader.

Components Required: Arduino uno board, RFID tag, RFID reader module.

Circuit diagram:



Code:

```

#include <SoftwareSerial.h>
SoftwareSerial mySerial(9,10);
void setup() {
    mySerial.begin(9600);
    serial.begin(9600);
}
void loop() {
    if (mySerial.available() > 0) {
        serial.write(mySerial.read());
    }
}
    
```

Observation: successfully able to read the RFID tag on the serial monitor.

**Code:**

```
#include <SoftwareSerial.h>
SoftwareSerial mySerial(9, 10);
void setup()
{
mySerial.begin(9600);
Serial.begin(9600); }
void loop()
{
if(mySerial.available()>0)
{
Serial.write(mySerial.read());
}
}
```

**Observation:** Was successfully able to read the RFID ID on the serial monitor.

Program no: 14

Program Title: **RFID Access control**

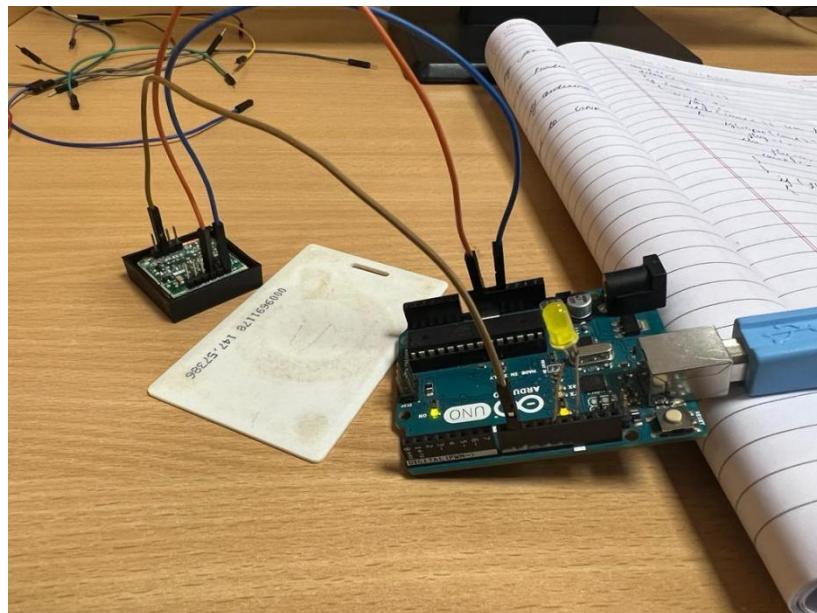
**Aim:** To demonstrate the use of RFID tags for access control.

### **Hardware/components Required:**

- Arduino Uno board - 1
- USB Cable - 1
- Jumper wires
- Bread Board
- RFID Reader
- RFID Tag

### **Circuit Diagram / Pin connection:**

- RFID TX to Rx of Arduino
- Ground of RFID reader to ground of Arduino
- VCC of RFID to 5v supply of Arduino
- Longer end of LED to pin 12 and shorter end to Ground



## Handwritten code pic:

Date 12/12/23  
Page

Expt 14: RFID access control

Aim: To demonstrate use of RFID tags for access control.

Components required: Arduino uno, LED, RFID reader, RFID tags.

Connections:

RFID tag: Tx to Rx of Arduino  
GND to GND  
Vcc to 9V

LED: longer end to pin 13 and shorter end to GND.

Code:

```
#include <SoftwareSerial.h>
#define ledPin 13
SoftwareSerial mySerial(9,10);
char tag[7] = "51009408C40F";
char input[12];
int count=0;
boolean flag=0;
void setup() {
  Serial.begin(115200);
  mySerial.begin(115200);
  pinMode(ledPin, OUTPUT);
}
void loop() {
  if (mySerial.available()) {
    count=0
    while (mySerial.available() && count<12) {
      input[count]=mySerial.read();
      count++;
    }
  }
}
```

```

Serial.write (input [count]);
count++;
delay (5);
if (count == 12) {
    count = 0;
    flag = 1;
    while (count < 12 && flag != 0)
        if (input [count] == key [count]) {
            flag = 1;
        } else {
            flag = 0;
            count++;
        }
    if (flag == 1) {
        Serial.println ("Access Allowed");
        digitalWrite (ledpin, HIGH);
        delay (2000);
        digitalWrite (ledpin, LOW);
    } else {
        Serial.println ("Access denied");
        digitalWrite (ledpin, LOW);
        delay (2000);
    }
    for (count = 0; count < 12; count++)
        if (input [count] == 'F')
            count = 0;
}

```

*Observation: Successfully received the string of 12 when the R key and message "Access Allowed" was printed on the serial monitor.*

**Code:**

```
#include<SoftwareSerial.h>
SoftwareSerial mySerial(9, 10);
#define LEDPIN 12
char tag[] ="3C0087D597F9";
char input[12];
int count = 0;
input[] character array
boolean flag = 0;

void setup()
{ Serial.begin(9600);
mySerial.begin(9600);
pinMode(LEDPIN,OUTPUT); //WRONG TAG INDICATOR
}
void loop()
{
if(mySerial.available())
{
count = 0;
while(mySerial.available() && count < 12)
{
input[count] = mySerial.read();
Serial.write(input[count]);
count++; // increment counter
delay(5);
}
if(count == 12) //
{
count =0; // reset counter varibale to 0
flag = 1;

while(count<12 && flag !=0)
{
if(input[count]==tag[count])
flag = 1; // everytime the values match, we set the flag variable
to 1
else
flag= 0;
count++;
}
}
if(flag == 1)
{
Serial.println("Access Allowed!");
digitalWrite(LEDPIN,HIGH);
delay (2000);
digitalWrite (LEDPIN,LOW);
```

```
    }
} else
{
    Serial.println("Access Denied");
    digitalWrite(LEDPIN,LOW);
    delay(2000);
}
for(count=0; count<12; count++)
{
    input[count]= 'F';
}
count = 0; // Reset counter variable
}
```

**Observation:** Successfully observed the glowing of LED when the RFID key and message “Access Allowed” was printed on the serial monitor.

Program no: **15**

Program Title: **Bluetooth Home Automation**

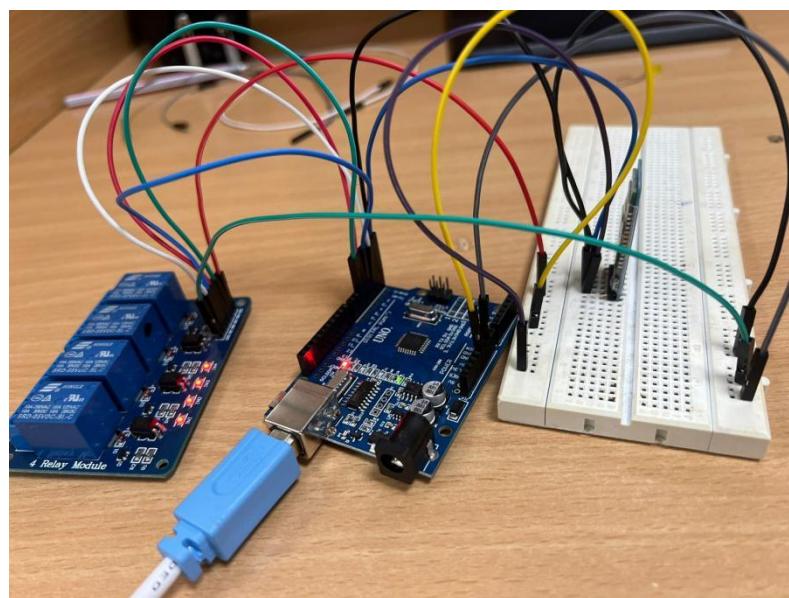
**Aim:** To control the working of relay through Android Mobile.

**Hardware Required:**

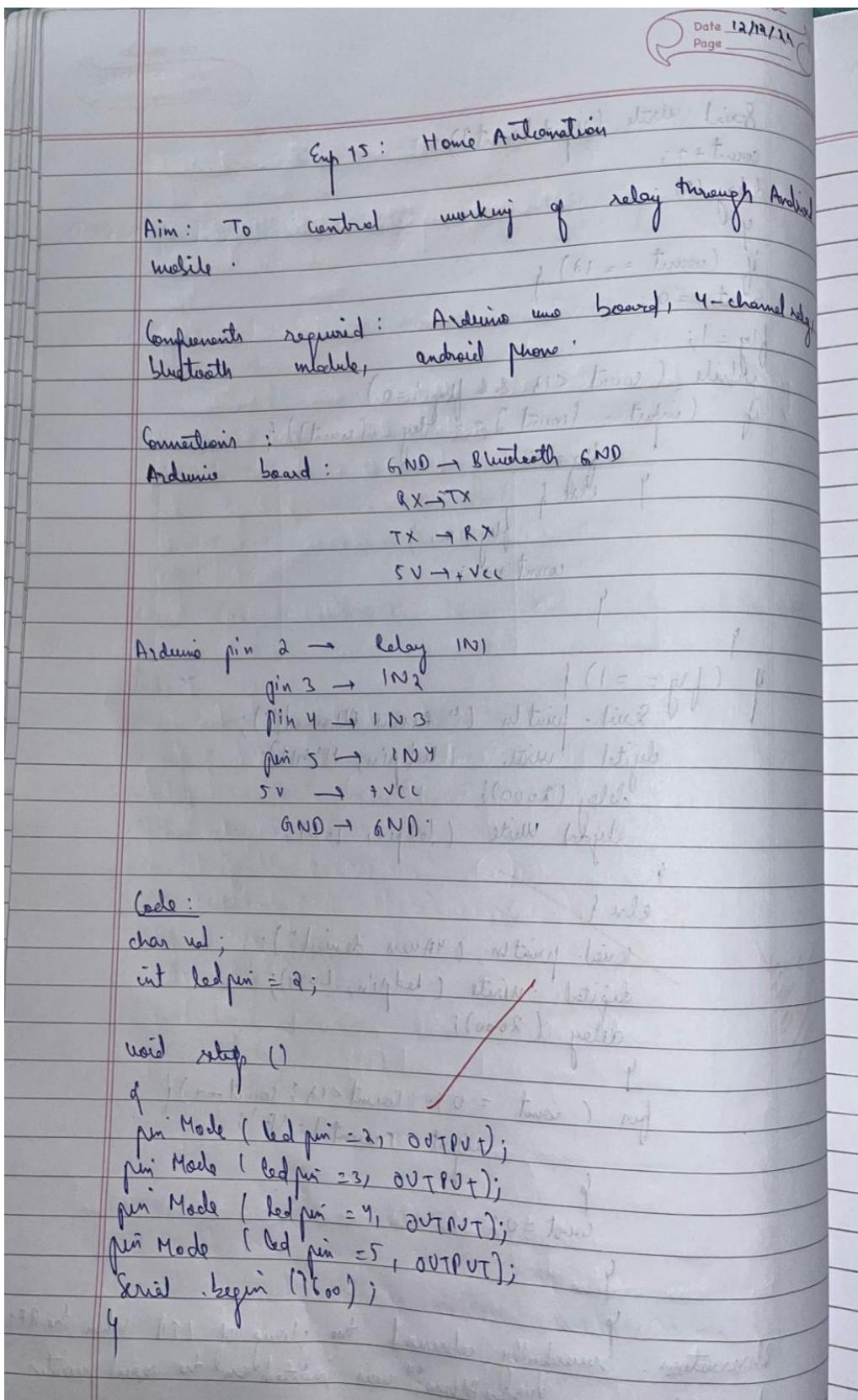
- Arduino
- 4-Channel relay
- Bluetooth Module
- Android phone

**Circuit Diagram / Pin connection:**

- Output 1 to Pin 4 ( Arduino Board )  
Output 2 to Pin 5  
Output 3 to Pin 6  
Output 4 to Pin 7
- Bluetooth Module Tx to Pin 0  
Bluetooth Module Rx to Pin 1
- VCC of Bluetooth & relay should be connected to Arduino 5V (through breadboard)
- GND of Bluetooth & relay should be connected to Arduino GND



## Handwritten code pic:



```
void loop()
{
    if (serial.available())
    {
        val = serial.read();
        serial.println(val);
        if (val == 'a')
            digitalWrite (ledpin = 2, HIGH);
        if (val == 'A')
            digitalWrite (ledpin = 2, LOW);
        if (val == 'b')
            digitalWrite (ledpin = 3, HIGH);
        if (val == 'B')
            digitalWrite (ledpin = 3, LOW);
        if (val == 'c')
            digitalWrite (ledpin = 4, LOW);
        if (val == 'C')
            digitalWrite (ledpin = 5, LOW);
        if (val == 'd')
            digitalWrite (ledpin = 4, HIGH);
    }
}
```

if (val == 'd')

digital write (led pin = 5, HIGH);

~~Observation:~~ Was successfully able to control the glowing  
of LEDs using mobile phone.

**Code:**

```
char val;
int ledpin = 2;

void setup()
{
    pinMode(ledpin = 2, OUTPUT);
    pinMode(ledpin = 3, OUTPUT);
    pinMode(ledpin = 4, OUTPUT);
    pinMode(ledpin = 5, OUTPUT);

    Serial.begin(9600);
}

void loop()
{
    if( Serial.available() )
    {
        ;
    }
    val = Serial.read();

    if( val == 'a' )
    {
        digitalWrite(ledpin = 2, HIGH);
    }

    if( val == 'A' )
    {
        digitalWrite(ledpin = 2, LOW);
    }

    if( val == 'b' )
    {
        digitalWrite(ledpin = 3, HIGH);
    }

    if( val == 'B' )
    {
        digitalWrite(ledpin = 3, LOW);    }

    if( val == 'C' )
    {
        digitalWrite(ledpin=4, LOW);
    }

    if( val == 'D' )
    {
```

```
    digitalWrite(ledpin=5, LOW);
}

if( val == 'c' )
{
    digitalWrite(ledpin = 4, HIGH);
}
if( val == 'd' )
{
    digitalWrite(ledpin = 5, HIGH);
}
}
```

**Observation:** Was successfully able to control the glowing of LEDs using mobile phone.

Program no: **16**

Program Title: **Working with GSM Module**

### **GSM Module: Call to a particular number**

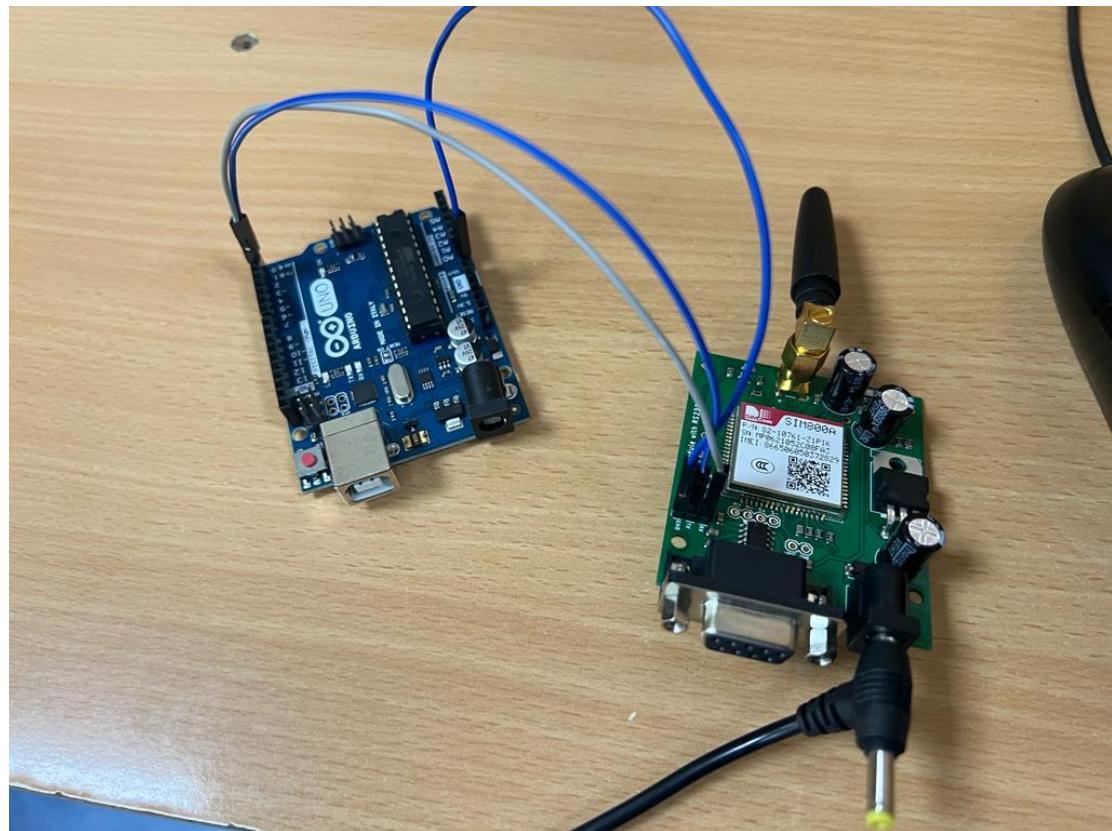
#### **Hardware/components Required:**

- Arduino Uno board - 1
- USB Cable - 1
- SIM900 GSM Module
- Jumper wires
- Bread Board

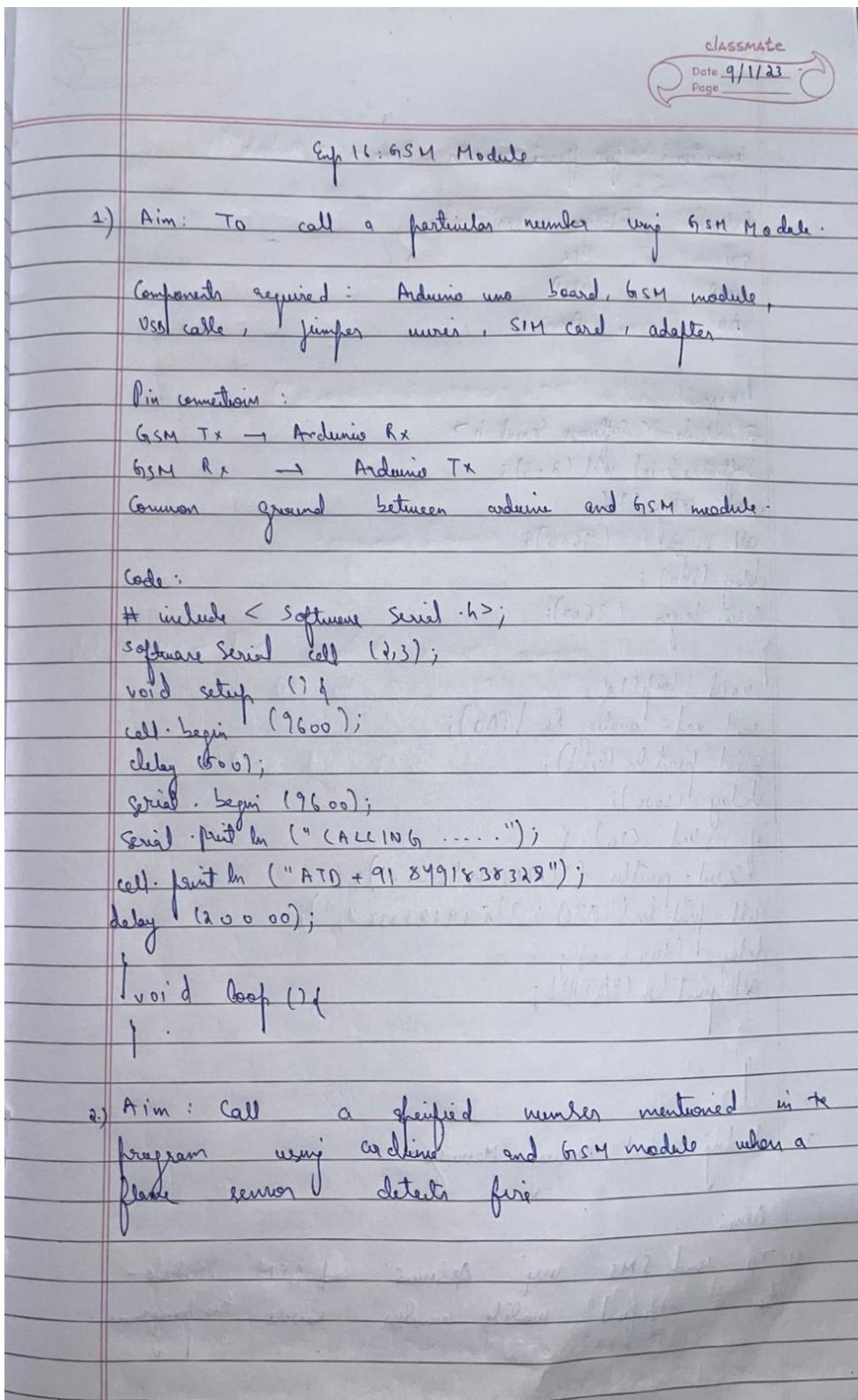
**Aim:** Call using Arduino and GSM Module – to a specified mobile number inside the program.

#### **Circuit Diagram / Pin connection:**

- GSM Tx → Arduino Rx (Here pin 2)
- GSM Rx → Arduino Tx. (Here pin 3)
- Make the ground common between Arduino and GSM modem.



## **Handwritten code pic:**



**Code:**

```
#include <SoftwareSerial.h>
SoftwareSerial cell(2,3); // (Rx, Tx)

void setup() {
cell.begin(9600);
delay(500);
Serial.begin(9600);
Serial.println("CALLING.....");
cell.println("ATD+9538433364;");
delay(20000);
}
void loop() {

}
```

**Call to a particular number on an alert**

**Aim:** Call a specified mobile number mentioned in the program using Arduino and GSM Module when a flame sensor detects “fire”.

**Connections for flame sensor:**

| <b>Arduino</b> | <b>Flame Sensor</b> |
|----------------|---------------------|
| 5V             | VCC                 |
| GND            | GND                 |
| A0             | A0                  |

**Handwritten code pic:**

connections for flame sensor

Ardueino      Flame sensor  
5V              Vcc  
GND              GND  
A0              Av.

Program:

```
#include <SoftwareSerial.h>
SoftwareSerial mySerial(2,3);
void setup() {
    mySerial.begin(9600);
    delay(500);
    Serial.begin(9600);
}
void loop() {
    int val = analogRead(A0);
    Serial.println(val);
    delay(1000);
    if (val < 50) {
        Serial.println("CALLING 918491838328");
        mySerial.print("AT&T<CR><LF>");
        delay(100000);
        mySerial.print("AT&T<CR><LF>");
    }
}
```

3. Sending and Receiving Message.

Aim:

- 1) To send SMS via Arduino and GSM Module - to a specified mobile number directly the program.

**Code:**

```
#include <SoftwareSerial.h>
SoftwareSerial cell(2,3);
void setup() {
cell.begin(9600);
delay(500);
Serial.begin(9600);
}
void loop() {
Int val=analogRead(A0);
Serial.println(val);
delay(1000);
if (val<50)
{
Serial.println("CALLING.....");
cell.println("ATD+919742980606;");
delay(10000);
cell.println("ATH"); // Attention Hook Control
}
}
```

**Sending and Receiving Message**

**Aim:**

- 1) Send SMS using Arduino and GSM Module – to a specified mobile number inside the program
- 2) Receive SMS using Arduino and GSM Module – to the SIM card loaded in the GSM Module.

**Handwritten code pic:**

- 7) Receive SMS using Arduino and GSM Module - to two SIM card loaded in the GSM Module.

Program:

```
#include <SoftwareSerial.h>
SoftwareSerial mySerial(2,3);
void setup()
{
    mySerial.begin(9600);
    Serial.begin(9600);
    delay(100);
}
void loop()
{
    if (Serial.available() > 0)
        switch (Serial.read())
    {
        case 'S':
            SendMessage();
            break;
        case 'R':
            ReceiveMessage();
            break;
    }
    if (mySerial.available() > 0)
        mySerial.write(mySerial.read());
}
void SendMessage()
{
    mySerial.println("AT+CMGF=1");
    delay(1000);
    mySerial.println("AT+QMGR=\"+918491838328\"|1");
    delay(1000);
}
```

```

mySerial.print("I am SMS from GSM Module");
delay(1000);
mySerial.println((char)26);
delay(1000);
void RecvMsg() {
    mySerial.println("AT+CNMI=1,1,0,0,0");
    delay(1000);
}

```

#### 4. Controlling LED through received message:

Aim:

The received message through Arduino and GSM Module to control switch ON/OFF the LED.

Connection: Attach LED to pin 13 and GND.

Program:

```

#include <SoftwareSerial.h>
SoftwareSerial mySerial(2, 3);
void readSerial() {
    if (mySerial.available() > 0) {
        while (mySerial.available() > 0) {
            Serial.write(mySerial.read());
        }
    }
    void setup() {
        pinMode(13, OUTPUT);
        Serial.begin(9600);
    }
}

```

**Code:**

```
#include <SoftwareSerial.h>
SoftwareSerial mySerial(2, 3);

void setup()
{
mySerial.begin(9600); // Setting the baud rate of GSM Module
Serial.begin(9600); // Setting the baud rate of Serial Monitor (Arduino)
delay(100);
}

void loop()
{
if (Serial.available()>0)
switch(Serial.read())
{
case 's':
SendMessage();
break;
case 'r':
RecieveMessage();
break;
}

if (mySerial.available()>0)
Serial.write(mySerial.read());
}

void SendMessage()
{
mySerial.println("AT+CMGF=1");
delay(1000); // Delay of 1000 milli seconds or 1 second
mySerial.println("AT+CMGS=\\"+919742980606\\r");

delay(1000);
mySerial.println("I am SMS from GSM Module");

delay(100);
mySerial.println((char)26);
delay(1000);
}

Void RecieveMessage()
{
mySerial.println("AT+CNMI=2,2,0,0,0");
delay(1000);
}
```

## Controlling LED through received messages:

### Aim:

Use received message through Arduino and GSM Module to control Switching ON / OFF the LED.

**Connection:** Attach LED to pin 13 and GND.

### Handwritten code pic:

The handwritten code is a C++ program for an Arduino. It includes a function to handle incoming messages and a main loop to check for messages and control an LED. The code is annotated with comments explaining its purpose.

```
#include <SoftwareSerial.h>
SoftwareSerial cell(2,3);
void read() {
    if (cell.available()) {
        while (cell.available()) {
            Serial.write(cell.read());
        }
    }
}
void setup() {
    pinMode(13,OUTPUT);
    Serial.begin(9600);
}
```

Annotations in the code:

- mySerial.print("I am SMS from GSM Module");
- delay(1000);
- mySerial.println((char)26);
- delay(1000);
- void RecvMsg()
- {
- mySerial.print("AT+CMGR=1,0,0,0");
- delay(1000);
- }
- 4. Controlling LED through received messages:
- Aim:  
Use received message through Arduino and GSM Module  
to control switch ON/OFF the LED.
- Connection: Attach LED to pin 13 and GND.
- Program:

```
cell.begin(9600);
cell.print("AT");
delay(1000);
Serial.println("AT+CMII=1,2,0,0,0");
y.
void loop() {
if (cell.available()) {
String message = cell.readString();
Serial.print(message);
if (message.indexOf("SWITCH ON") > 0)
digitalWrite(13,HIGH);
else if (message.indexOf("SWITCH OFF") > 0)
digitalWrite(13,LOW);
else
Serial.println("Nothing to do...");
```

**Code:**

```
#include <SoftwareSerial.h>
SoftwareSerial cell(2,3);

void readfn()
{
    if (cell.available()) {
        while (cell.available()) {
            Serial.write(cell.read());
        }
    }
}

void setup() {
    pinMode(13,OUTPUT);
    Serial.begin(9600);
    cell.begin(9600);
    cell.println("AT");
    delay(1000);
    readfn();
    //New SMS alert
    cell.println("AT+CNMI=1,2,0,0,0");
}

void loop() {
    if(cell.available())
    {
        String message =cell.readString();
        Serial.println(message);
        if(message.indexOf("SWITCH ON")>0)
        {
            digitalWrite(13,HIGH);
        }
        else if(message.indexOf("SWITCH OFF")>0)
        {
            digitalWrite(13,LOW);
        }
        else
        {
            Serial.println ("Nothing to do...");
        }
    }
}
```