

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.



LAB REPORT

on

INTERNET OF THINGS

Submitted by

MD SURAJ KUMAR(1BM20CS079)

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 **MD SURAJ KUMAR(1BM20CS079)**, 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)

Components Required :

Arduino Uno board - 1

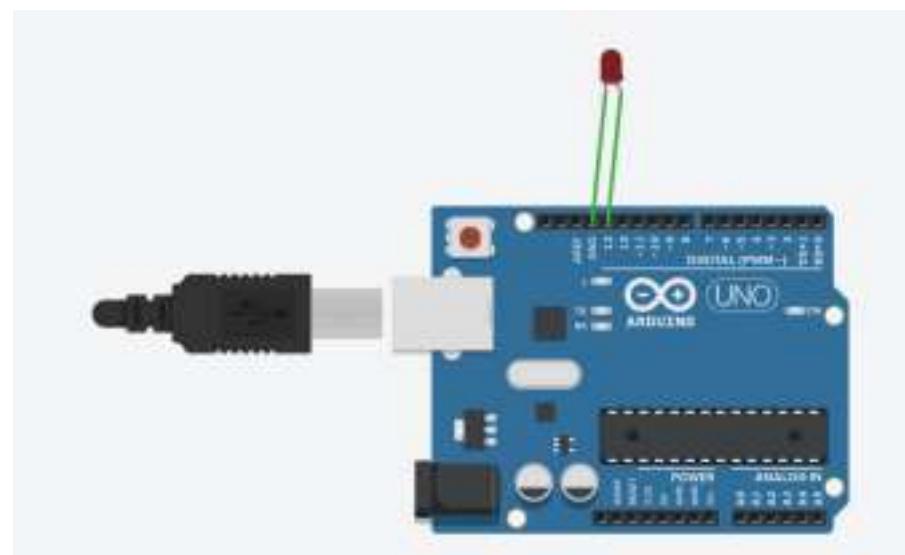
USB Cable - 1

LED - 1

Jumper wires

Pin connection :

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



Code :

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

Handwritten code :

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

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

Program no: 02

Program Title: TRAFFIC SIGNAL

Aim : To simulate traffic signal scenario using an arduino UNO board

Components Required :

Arduino Uno board - 1

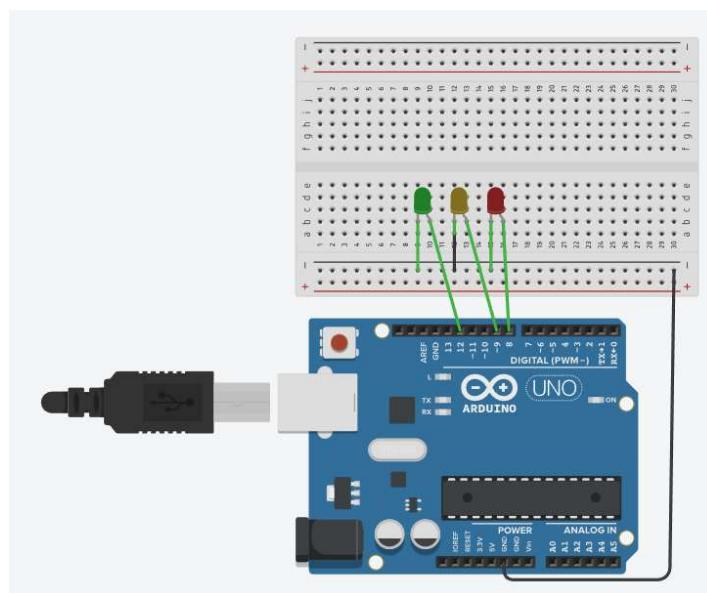
USB Cable - 1

LED - 3

Jumper wires

Pin Connections :

- LED(RED) positive pin to digital pin 13
- LED(Yellow) positive pin to digital pin 12
- LED(Green) positive pin to digital pin 11
- LED(ALL) negative pin to Ground



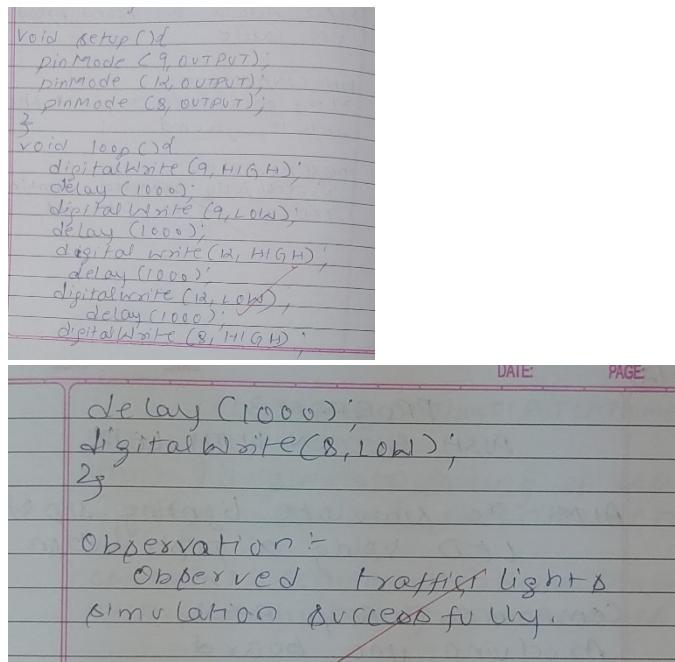
Code :

```
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);
}
```

Handwritten code :



Void setup() {
pinMode (9, OUTPUT);
pinMode (10, OUTPUT);
pinMode (8, OUTPUT);
}
void loop() {
digitalWrite (9, HIGH);
delay (1000);
digitalWrite (9, LOW);
delay (1000);
digitalWrite (10, HIGH);
delay (1000);
digitalWrite (10, LOW);
delay (1000);
digitalWrite (8, HIGH);
}

DATE: PAGE:

delay (1000);
digitalWrite (8, LOW);
2

Observation:-
Observed traffic lights
simulation successfully.

Observation: Each of the LEDs switch ON/OFF periodically.

Program no: 03

Program Title: **PUSH BUTTON**

Aim : To simulate LED blink using digital input.

Components Required :

Arduino Uno board - 1

USB Cable - 1

Breadboard - 1

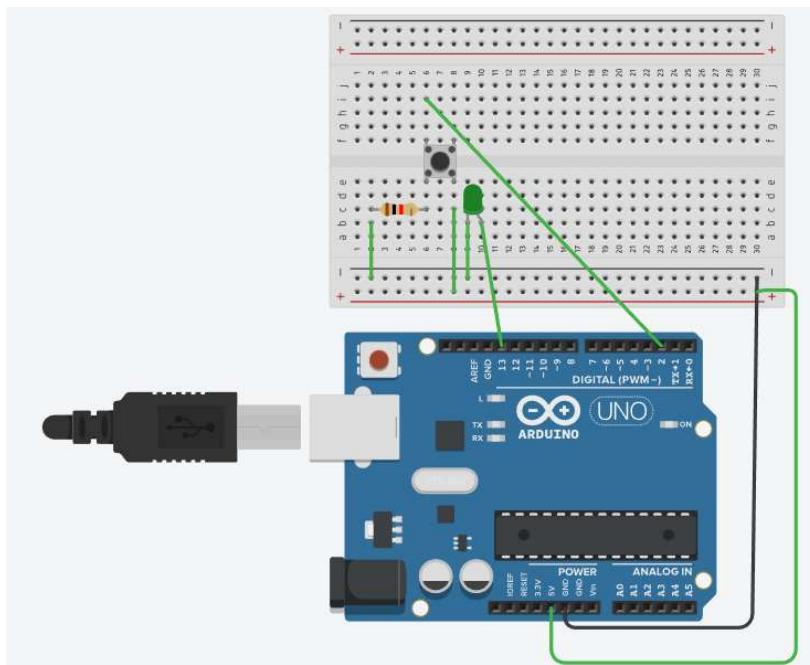
Push Button - 1

LED - 1

Jumper wires

Pin Connections :

- PUSH Button one pin to digital pin 2
- PUSH Button opposite pin of pin 2 to ground
- PUSH Button adjacent pin of ground 5v of arduino
- LED positive pin to digital pin 13
- LED negative pin to Ground



Code :

```
int pushButtonPin = 2;
int ledPin = 13;
void setup()
{
    pinMode(pushButtonPin,INPUT);
    pinMode(ledPin,OUTPUT);
}
```

```

void loop() {
    int pushButtonState = digitalRead(pushButtonPin);
    if (pushButtonState == 1)
    {
        digitalWrite(ledPin, HIGH);
    }
    else
    {
        digitalWrite(ledPin, LOW);
    }
}

```

Handwritten code :

```

int bp=0
void setup() {
    pinMode(13, OUTPUT);
    pinMode(2, INPUT);
}

void loop() {
    bp = digitalRead(2);
    if (bp == HIGH)
        digitalWrite(13, HIGH);
    else
        digitalWrite(13, LOW);
}

```

Observation: The LED switches ON/OFF periodically when input is given through the push button.

Program no: **04**

Program Title: **LED FADE**

Aim : To demonstrate analog output with fading LED

Components Required :

Arduino Uno board - 1

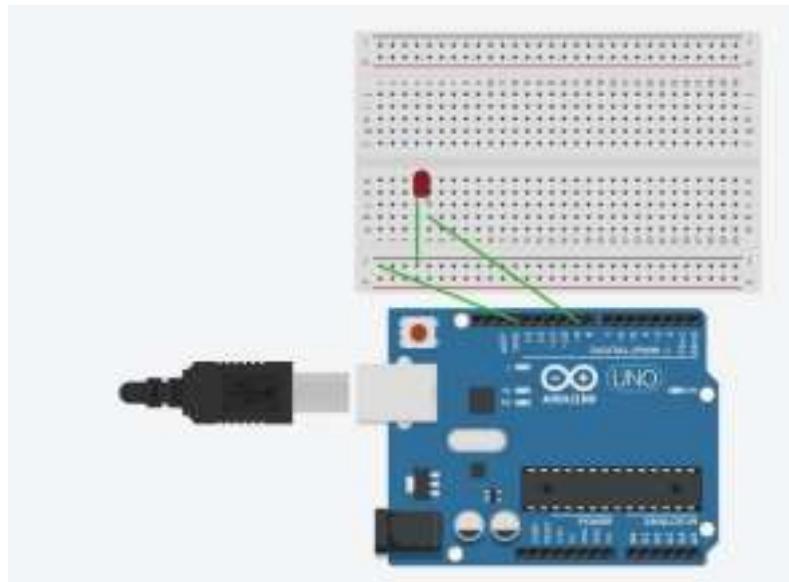
USB Cable - 1

LED - 1

Jumper wires

Pin Connections :

- LED positive pin to digital pin 9
- LED negative pin to Ground



Code :

```
const int LedPin = 9;  
void setup()  
{  
    pinMode(LedPin, OUTPUT);  
}  
void loop()  
{  
    for(int fade value = 0; fade value <= 255; fade value += 255);  
    {  
        analogWrite(LedPin, fade value);  
        delay(330);  
    }  
}  
  
for(int fade value = 255; fade value <= 0; fade value -= 255);
```

```
{  
analogWrite(LedPin, fade value);  
delay(330);  
}  
}
```

Handwritten code :

The image shows handwritten C code on lined paper. The code defines an integer variable `ledPin` as 9, sets up pins, and enters a loop. Inside the loop, it uses a `for` loop to fade the LED from 0 to 255. It then uses another `for` loop to fade it back down to 0. The `analogWrite` function is used to set the LED brightness, and `delay` functions are used to control the fade rate.

```
Code >  
int ledPin = 9;  
void setup() { }  
void loop()  
{  
    for(int fadeValue=0; fadeValue<=255;  
        fadeValue++)  
    {  
        analogWrite(ledPin, fadeValue);  
        delay(30);  
    }  
    for(int fv=255; fv>=0; fv-=5)  
    {  
        analogWrite(ledPin, fv);  
        delay(1);  
    }  
}
```

Observation: The LED gradually fades away.

Program no: 05

Program Title: **LED FADE WITH POTENTIOMETER**

Aim :To demonstrate analog output with fading LED using potentiometer

Components Required :

Arduino Uno board - 1

USB Cable - 1

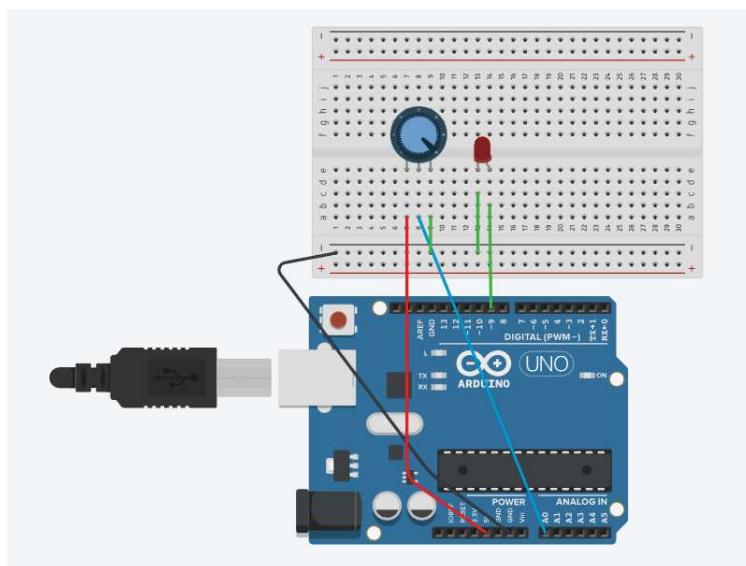
LED - 1

Potentiometer - 1

Jumper wires

Pin Connections :

- LED positive pin to digital pin 11
- LED negative pin to analog pin 10
- Potentiometer and LED negative pin to ground



Code :

```
const int LedPin = 11;  
void setup()  
{  
    pinMode(LedPin, OUTPUT);  
    Serial.begin(9600);  
}  
void loop()  
{  
    int fade value = analogRead(A0);  
    int brightness = map(fade value 0,1023,0,255)  
    analogWrite(LedPin, brightness);
```

```
Serial.print("analog");
Serial.println(fade value);
Serial.print("brightness");
Serial.println(brightness);
delay(2);
}
}
```

Handwritten code :

The handwritten code is a translation of the provided pseudocode into C++ for an Arduino. It includes setup and loop sections, initializes pin 9 as an output, reads analog input from pin A0, maps the value to a brightness range, and prints the analog value and brightness to the serial monitor.

```
Code:
1 int LED-PIN=9
void setup()
{
    Serial.begin(9600);
    pinMode(LED-PIN, OUTPUT);
}
void loop()
{
    int analogValue = analogRead(A0);
    int BrightnessD = map(analogValue, 0, 1023, 255);
    analogWrite(LED-PIN, BrightnessD);
    Serial.print("Analog");
    Serial.print(analogValue);
    Serial.print("Brightness");
    Serial.print(BrightnessD);
    delay(100);
}
```

Observation: The LED gradually flows and fades according to potentiometer input.

Program no: 06

Program Title: **FADING OF 2 LEDs**

Aim : To demonstrate analog output with fading two LED

Components Required :

Arduino Uno board - 1

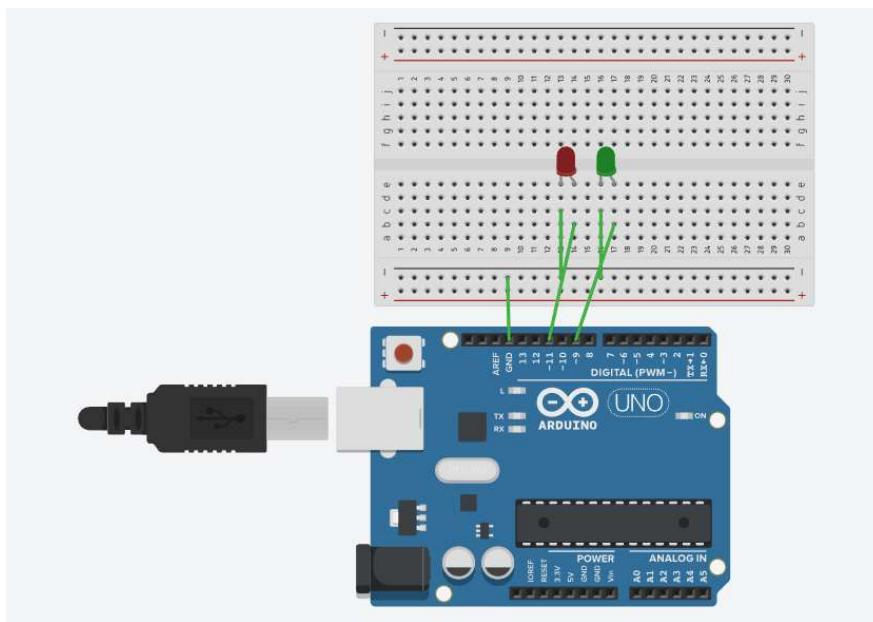
USB Cable - 1

LED - 2

Jumper wires

Pin Connections :

- LED green positive pin to digital pin 9
- LED red positive pin to digital pin 11
- Both LED's negative pin to Ground



Code :

```
const int LedPin1 = 9;  
const int LedPin2 = 11;  
void setup()  
{  
    pinMode(LedPin1,OUTPUT );  
    pinMode(LedPin2,OUTPUT );  
}  
void loop()  
{  
    for(int fade value = 0; fade value <= 255; fade value += 255);  
}
```

```

{
analogWrite(LedPin 1, fade value);
delay(100);
}
for(int fade value = 255; fade value >= 0; fade value -= 15);
{
analogWrite(LedPin 2, fade value);
delay(100);
}
for(int fade value = 255; fade value >= 0; fade value -= 15);
{
analogWrite(LedPin 1, fade value);
delay(1000);
}
for(int fade value = 255; fade value >= 0; fade value += 15);
{
analogWrite(LedPin 2, fade value);
delay(1000);
}
}
}

```

Handwritten code :

Coder
 int LED-PIN1=9;
 int LED-PIN2=6;
 void setup()
 {
 void loop()
 {
 int fv1=0;
 int fv2=255;
 while(fv1<=255 & fv2>=0)
 {
 analogWrite(LED-PIN-1,fv1);
 analogWrite(LED-PIN-2,fv2);
 fv1+=5;
 fv2-=5;
 delay(70);
 }
 fv1=255; fv2=0;

```
while (fr1>=0 && fr2<=255)
{
    analogWrite(LED-PIN-1, fr1);
    analogWrite(LED-PIN-2, fr2);
    fr1 -= 5;
    fr2 += 5;
    delay(70);
}
```

Observation: One LED flows and glows, while the other fades away simultaneously.

Program no: 07

Program Title: LED WITH LDR CONTROL

Aim :To demonstrate LED with LDR control

Components Required :

Arduino Uno board - 1

Resistor - 1

USB Cable - 1

LED - 1

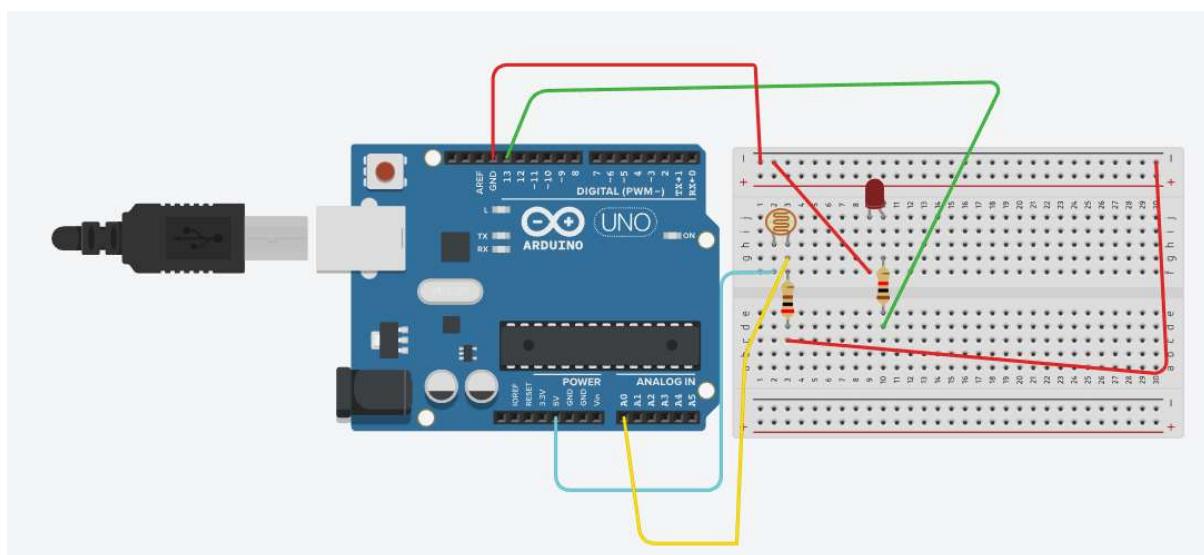
Breadboard -1

LDR - 1

Jumper wires

Pin Connections :

- LED positive pin to digital pin 13
- LED negative pin to ground
- Positive leg of LDR to A0
- Negative leg of LDR to 5V
- Resistor negative leg to ground
- Resistor positive leg to 5V



Code :

```
int value = 0;  
void setup()  
{  
    pinMode(11, OUTPUT);  
    pinMode(A0, INPUT);  
    Serial.begin(9600);  
}  
void loop()  
{  
    value = analogRead(A0);  
}
```

```

        if(value < 50 );
    {
        digitalWrite(11,HIGH)
        Serial.println("Light is ON");
        Serial.print(value);
    }
    else
    {
        digitalWrite(11,LOW)
        Serial.println("Light is OFF");
        Serial.print(value);
    }
}

```

Handwritten code :

Code

```

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

```

Observation: The LED glows when LDR is kept in the dark

Program no: 08

Program Title: TEMPERATURE SENSOR

Aim :To sense the temperature of the environment

Components Required :

Arduino Uno board - 1

USB Cable - 1

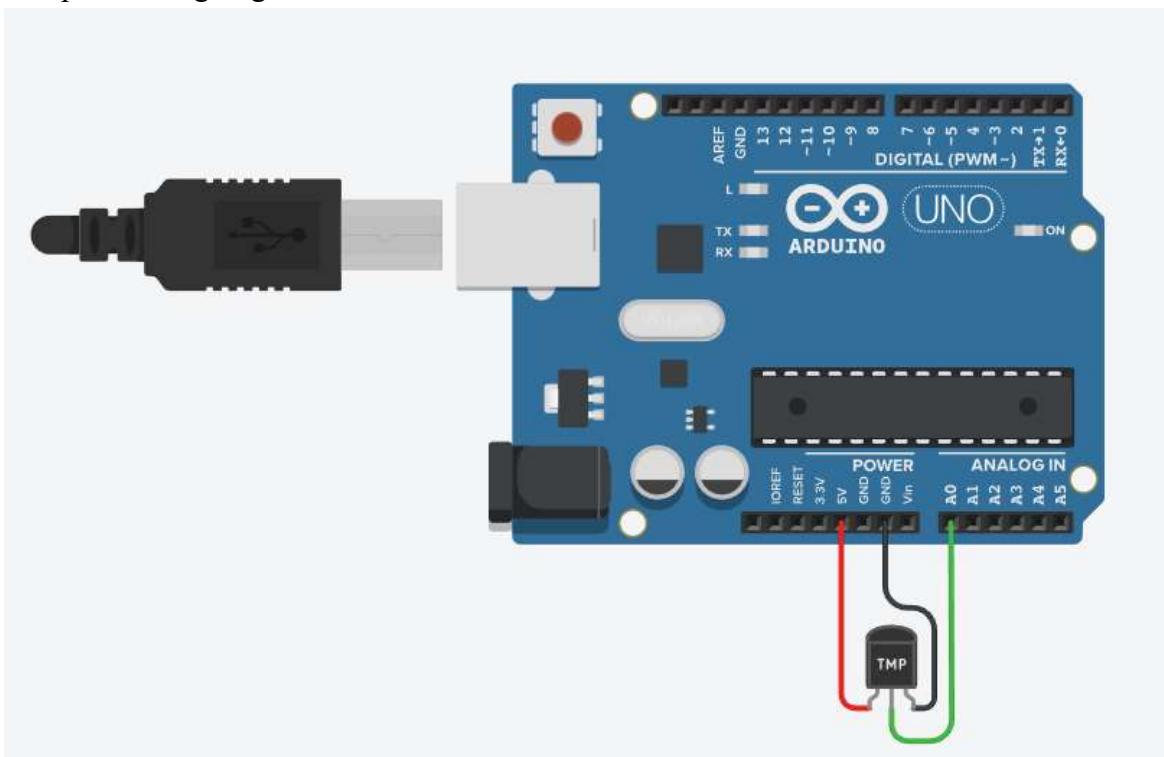
LED - 1

Temperature Sensor - 1

Jumper wires

Pin Connections :

- Center leg of Temperature sensor to A0.
- Right leg of Temperature sensor to 5V.
- Left leg of Temperature sensor to ground.
- Led positive leg to ground



Code :

```
int OutPin = 0;  
void setup()  
{  
    Serial.begin(9600);  
}  
void loop()  
{
```

```

int rowvoltage = analogRead(OutPin);
float millivolts = (rowvoltage/1024.0)*5000;
float celsius = millivolts/10;
Serial.print(celsius);
Serial.println(" degree celsius");
Serial.println((celsius*9)/5+32);
Serial.println(" degree Fahrenheit");
delay(100);
}

```

Handwritten code :

Code :-

```

int out_pin=10;
void setup()
{
    Serial.begin(9600);
}

void loop()
{
    int rawVoltage=analogRead(out-
    float mV = (rawVoltage/1024
    float celsius= mV/10;
    Serial.print(celsius);
    Serial.print((celsius*9)/5 +3
    Serial.println("degree fahren
    delay(1000);
}

```

~~Observation:- successfully observe
the calculation of temperature
in fahrenheit.~~

Observation: The current temperature in the environment is observed.

Program no: 09

Program Title: **ULTRASOUND SENSOR**

Aim : To measure the distance between an object using ultrasound sensor.

Components Required :

Arduino Uno board - 1

USB Cable - 1

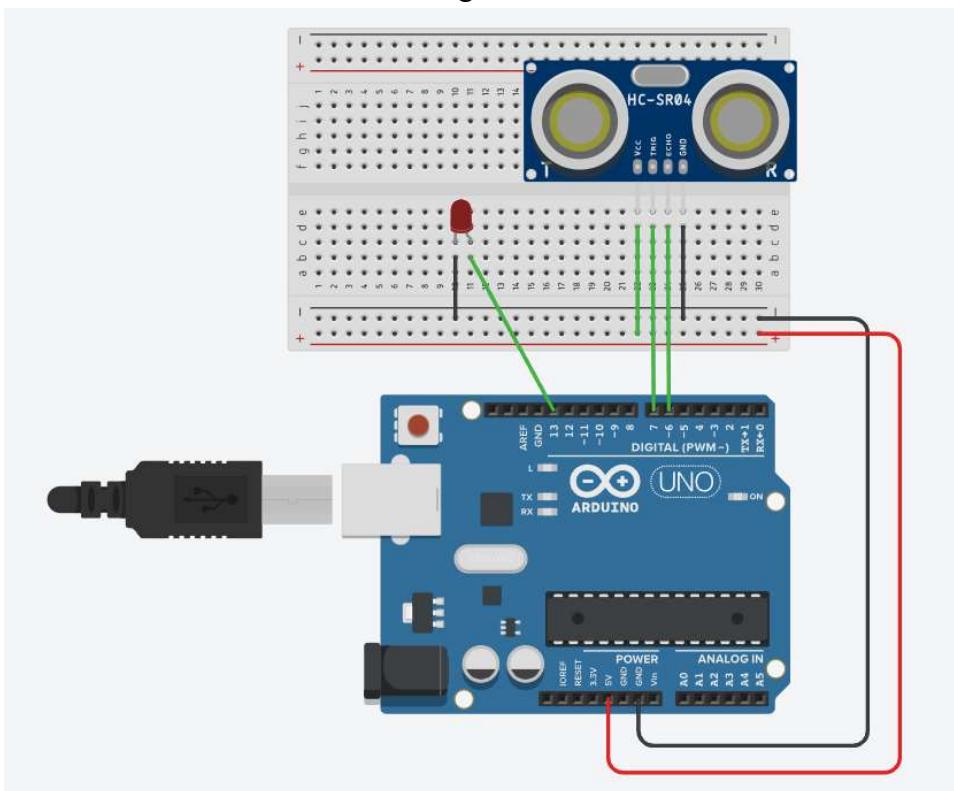
LED - 1

HC - SR04 - 1

Jumper wires

Pin Connections :

- Pin connected to digital Pin 7
- Pin connected to analog pin 6,
- VCC pin connected to 5v of arduino
- Ground of ultrasound connected to ground of arduino



Code :

```
const int pingPin = 7;
const int echoPin = 6;
void setup()
{
    Serial.begin(9600);
    pinMode(pingPin, OUTPUT);
    pinMode(echoPin, INPUT);
    pinMode(13,OUTPUT);
```

```
}

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);
    if(cm < 300)
    {
        digitalWrite(13,HIGH);
    }
    else
    {
        digitalWrite(13,LOW);
    }
    Serial.print(inches);
    Serial.print("in, ");
    Serial.print(cm);
    Serial.print("cm");
    Serial.println();
    delay(100);
}
long microsecondsToInches(long microseconds)
{
    return microseconds / 74 / 2;
}
long microsecondsToCentimeters(long microseconds)
{
    return microseconds / 29 / 2;
}
```

Handwritten code :

```
code:-  
int pingPin=9;  
int ePin=6;  
long dur,cm;  
int inch=0;  
void setup() {  
    pinMode(pingPin, OUTPUT);  
    pinMode(ePin, INPUT);  
    Serial.begin(9600);  
}  
void loop() {  
    digitalWrite(pingPin, LOW);  
    delayMicroseconds(2);  
    digitalWrite(pingPin, HIGH);  
    delayMicroseconds(10);  
    digitalWrite(pingPin, LOW);  
    dur = pulseIn(ePin, HIGH);  
    inch = dur / 7412;  
    cm = dur / 2912;  
  
    Serial.print("cm: ");  
    Serial.println(cm);  
}
```

Observation: The distance between various objects were observed.

Program no: 10

Program Title: **NIGHT LIGHT SIMULATION WITH HUMAN PRESENCE DETECTION**

Aim: Nightlight Simulation with Human Presence Detection.

Components Required :

Arduino Uno board - 1

USB Cable - 1

Resistor(110K) - 1

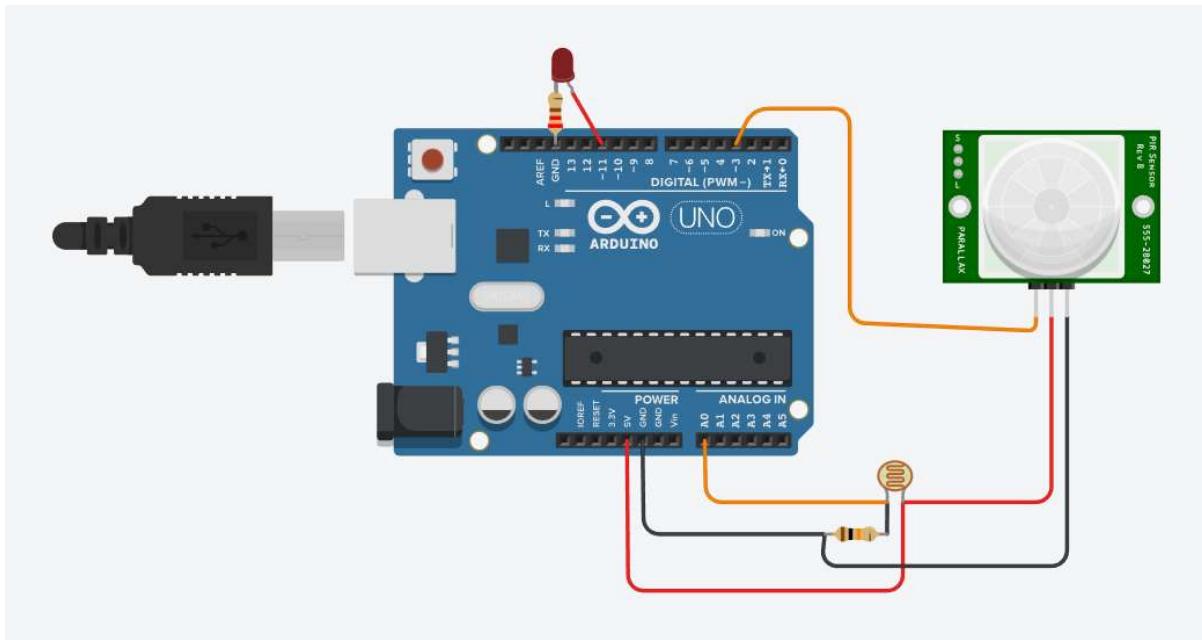
LED - 1

LDR- 1

Jumper wires

Pin Connections:

- One leg of LDR to 5V and another leg to Arduino Analog pin A0
- One leg of 110K register with that leg of LDR connected to A0
- Another leg of register to the ground
- Positive leg of LED to pin 11 and negative to GND
- Positive leg of PIR to 5V and negative leg to GND
- Output pin of PIR to digital pin 3



Code:

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

```
long unsigned int pause = 5000;
boolean lockLow = true;
boolean takeLowTime;
int pirPin = 3;
int ledPin = 11;
void setup()
{
    Serial.begin(9600);
    buadpinMode(11, OUTPUT);
    pinMode(pirPin, INPUT);
    pinMode(ledPin, OUTPUT);
    digitalWrite(pirPin, LOW);
    Serial.print("calibrating sensor ");
    for(int i = 0; i<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);
}
}

```

Handwritten code :

Code:-

```

int LDR=0;
int LDRVal=0
int lightPenitvity=500
int Ctime=30
long unsigned int lowIn;
long unsigned int pause=5000;
boolean lockLow=true;
boolean takeLowTime;
int pirPin=3
int ledPin=11;

```

```

void setup()
{
    Serial.println(9600);
    pinMode(11, OUTPUT);
    pinMode(GP1Pin, INPUT);
    pinMode(GP2Pin, OUTPUT);
    digitalWrite(GP1Pin, LOW);
    Serial.println("calibrating sensor");
    for (int i=0; i<calibrationTime; i++)
        Serial.print(".");
    delay(1000);
}

Serial.println("done");
Serial.println("sensor active");
delay(50);
}

void loop()
{
    LDRval = analogRead(CLDR);
    if (digitalRead(GP1Pin) == HIGH)
        LDRval < LightThreshold);
    digitalWrite(GP2Pin, HIGH);
    if (ClockLow)
        Serial.println("motion detected");
    Serial.print(millis() / 1000);
    Serial.println("sec");
    delay(50);
}
delay(100);
}

```

Observation: The LED glows while there is darkness when there is a human presence detected.

Program no: 11

Program Title: FIRE ALERT

Aim: Fire alarm simulation

Components Required :

Arduino Uno board - 1

USB Cable - 1

Flame Sensor - 1

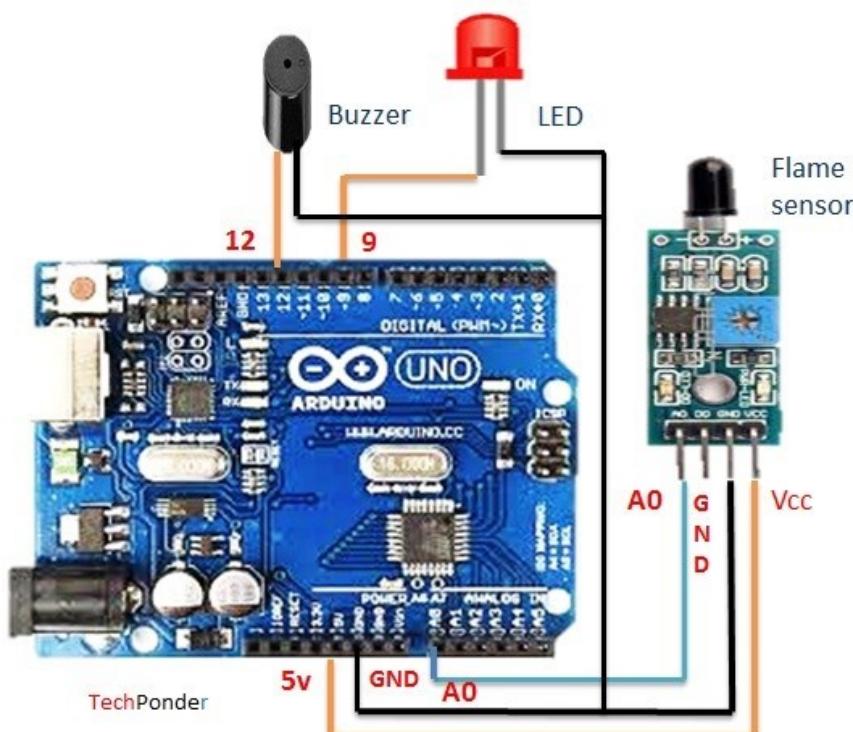
LED - 1

Buzzer - 1

Jumper wires

Pin Connections:

- Flame sensor interfacing to Arduino
- Flame sensor to Arduino 5V -> 5V
- ground -> ground
- A0 -> A0
- Led interfacing to Arduino
- LED +ve is connected to 9th pin of Arduino
- LED -ve is connected to ground 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



Code:

```
int sensorPin = A0;  
int sensorValue = 0;
```

```
int led = 9;  
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);  
}
```

Handwritten code :

```
Code >
#include <SoftwareSerial.h>
int sensorPin = A0;
int senVal = 0;
int led = 9;
int buzz = 12;
void setup(){
    pinMode(led,OUTPUT);
    pinMode(buzzer,OUTPUT);
    Serial.begin(9600);
}
void loop(){
    Serial.println("Flame sensor");
    senVal = analogRead(sensorPin);
    Serial.println(senVal);
    if (senVal < 100)
        {
            Serial.println("Fire detected");
            digitalWrite(led,HIGH);
            digitalWrite(buzzer,HIGH);
            delay(1000);
        }
}
```

```
digitalWrite(LED,LOW);
digitalWrite(buzzer,LOW);
delay(senVal);
}
```

Observation: The fire alarm goes off when fire or hotness is detected.

Program no: 12

Program Title: AUTOMATIC IRRIGATION CONTROLLER SIMULATION

Aim: Sensing the soil moisture and sprinkling the Water simulation

Components Required :

Arduino Uno board - 1

USB Cable - 1

Min Servo Motor - 1

Moisture Sensor - 1

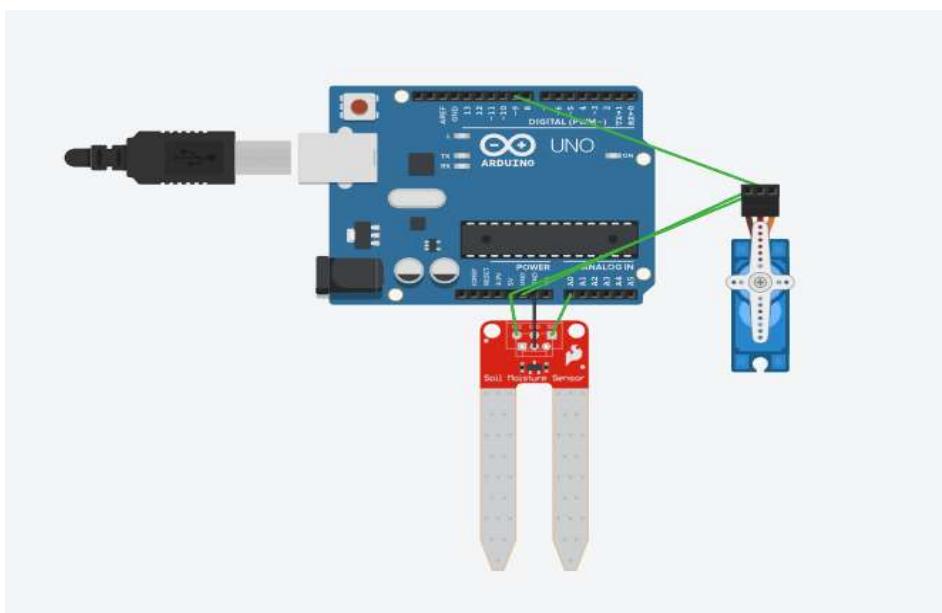
LED - 1

Buzzer - 1

Jumper wires

Pin Connections:

- 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



Code:

```
#include 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);  
    }  
}
```

Handwritten code :

```
Code :-  
int pop=0, penPin=A0, penVal=0;  
void setup(){  
    mySerial.attach(9);  
    serial.begin(9600);  
}  
void loop(){  
    penVal = analogRead(penPin);  
    serial.println(penVal);  
    if (penVal > 500){  
        for (pop=0; pop<=180; pop+=1){  
            mySerial.write(pop);  
            delay(15);  
        }  
    }  
}
```

```
for (pop=180; pop>=0; pop-=1){  
    mySerial.write(pop);  
    delay(15);  
}  
delay(100);  
}
```

Observation: The water sensor goes off when the lack of moisture content is sensed.

Program no: 13

Program Title: **RFID READER AND ACCESS CONTROL**

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

Components Required :

Arduino Uno board - 1

USB Cable - 1

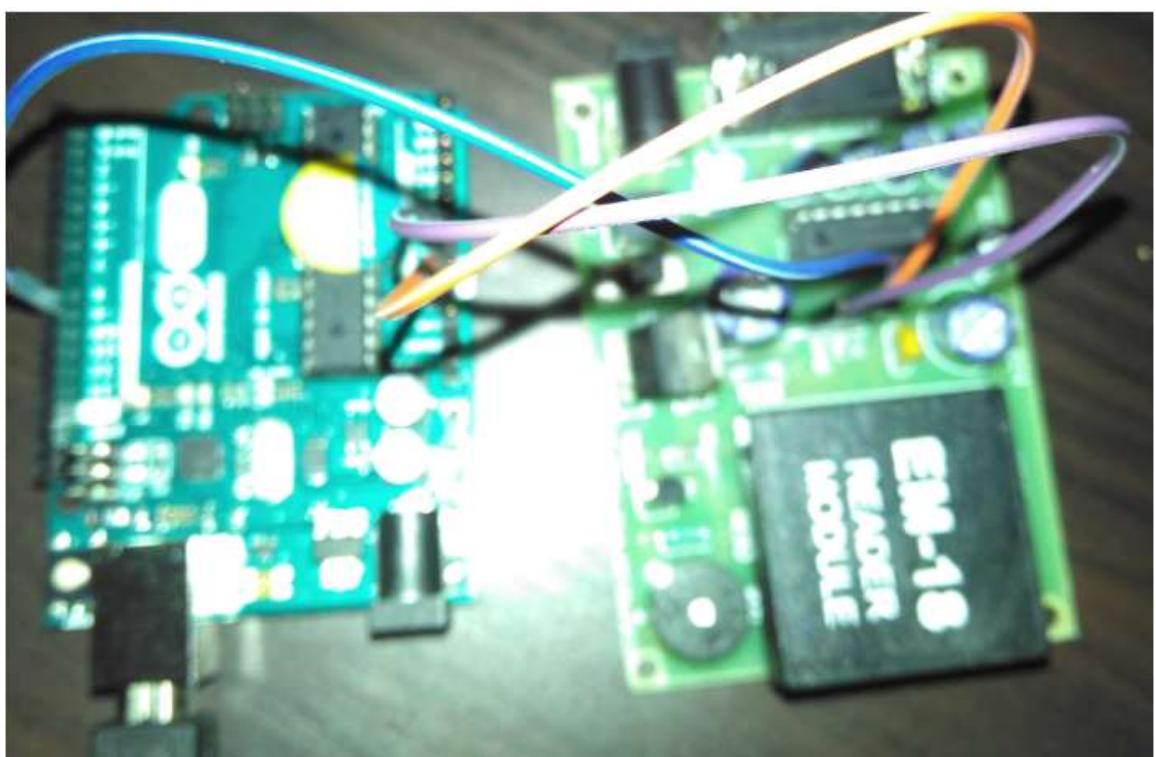
RFID reader module - 1

RFID tags - 1

Jumper wires

Pin Connections:

- Arduino ground to RFID ground
- TX(RFID) to 9(RX) arduino
- VCC USB Port to 12 RFID reader



Code:

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

void loop()
```

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

Handwritten 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: The RFID card is detected when brought closer to the RFID reader.

Program no: 14

Program Title: ACCESS BASED ON RFID TAG

Aim: To demonstrate the use of the RFID tag for access control

Components Required :

Arduino Uno board - 1

USB Cable - 1

RFID reader EM -18 - 1

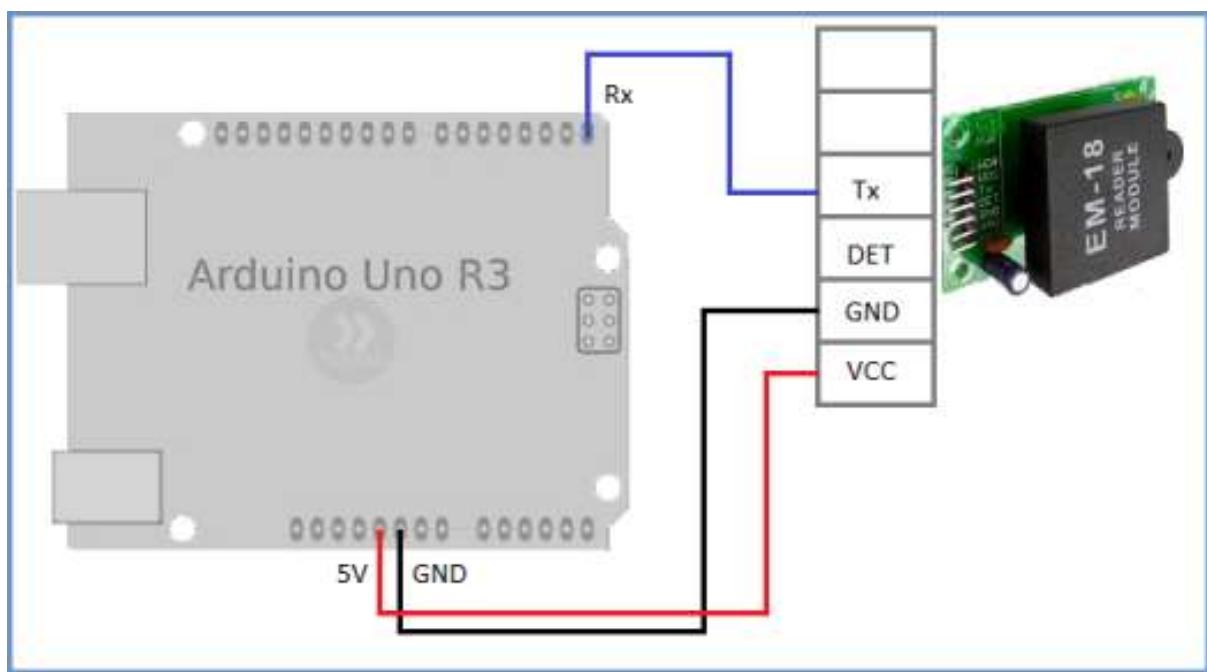
RFID tags - 1

RFID Tag Reader - 1

Jumper wires

Pin Connections :

- 5V to VCC of base Tag
- Ground of Arduino to ground of the base of the tag
- RX of arduino to TX of EM-18



Code :

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

```
char input[12];
int count = 0;
input[] character array
char input[12];
int count = 0;
boolean flag = 0;
void setup {}

{
Serial.begin(9600);
Monitor
Serial.begin(9600);
pinMode(LEDPIN, OUTPUT);
}

void loop {}

{
if(MySerial.available());
{
count=0;
while(MySerial.available()&& count<12)
{
input[count]=Myserial.read();
Serial.Write(input[count]);
count++;
delay(5);
}
if(count==12)
{
flag=1;
while(count=12; && flag!=0)
{
```

```
if(input[count]==flag[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++)
{
    input[count]= 'F'
}

count=0;
}
```

Handwritten code :

```
#include <SoftwareSerial.h>
SoftwareSerial mySerial(9,10);
#define LED 12
char tag[12];
char input[12];
int count = 0;

boolean flag = 0;
void setup()
{
  Serial.begin(9600);
  Monitor
  mySerial.begin(9600);
  pinMode(CLED, OUTPUT);
}
void loop()
{
  if (mySerial.available())
  {
    count = 0;
    while (mySerial.available() && count < 12)
    {
      input[count] = mySerial.read();
      Serial.write(input[count]);
      count++;
      delay(5);
    }
  }
}
```

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

Observation: Access is allowed when the card designated is recognized by the RFID tag reader.

Program no: 15

Program Title: HOME AUTOMATION(COMMAND PROMPT)

Aim: To establish the working of home automation.

Components Required :

Arduino Uno board - 1

USB Cable - 1

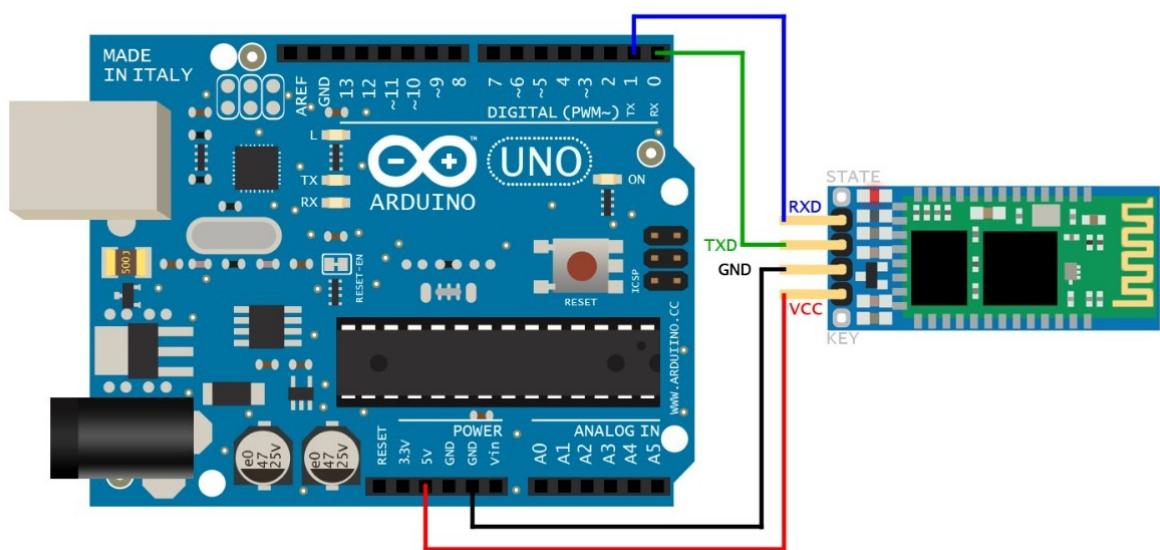
Bluetooth model - 1

Breadboard - 1

Jumper wires

Pin Connections :

- LED positive pin to digital 13
- LED negative pin to ground
- Bluetooth TX to arduino RX
- Bluetooth RX to arduino TX
- Bluetooth ground pin to ground
- Bluetooth VCC pin to VCC(5V)



Code :

```
#include SoftwareSerialBTSerial(10, 11);
void setup()
{
  Serial.begin(9600);
  Serial.println("Enter AT commands:");
  BTSerial.begin(38400);
}
void loop()
{
  Serial.write(BTSerial.read());
  BTSerial.write(Serial.read());
}
```

Handwritten code :

```
#include <SoftwareSerial.h>
SoftwareSerial BT(10,11);
void setup()
{
    Serial.begin(9600);
    Serial.println("Enter AT commands");
    BT.begin(38400);
}
void loop()
{
    Serial.write(BT.read());
    BT.write(Serial.read());
}
```

Observation: The Username and password for an automated home unit is set up successfully.

Program no: 16

Program Title: **HOME AUTOMATION(LED CONTROL)**

Aim: To see the working of LED control with the help of an android mobile application.

Components Required :

Arduino Uno board - 1

USB Cable - 1

Bluetooth module - 1

Android Phone - 1

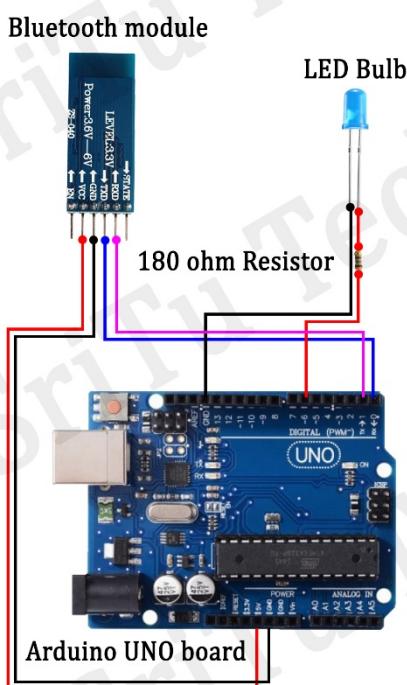
Breadboard - 1

Jumper wires

Arduino Bluetooth App

Pin Connections :

- LED positive pin to digital 13
- LED negative pin to ground
- Bluetooth TX to arduino RX
- Bluetooth RX to arduino TX
- Bluetooth ground pin to ground
- Bluetooth VCC pin to VCC(5V)



Code :

```
#defineledPin 13
int state = 0;
void setup()
{
    pinMode(ledPin, OUTPUT);
    digitalWrite(ledPin, LOW);
    Serial.begin(38400);
```

```

        }
        void loop()
    {
        if(Serial.available() > 0)
            if (state == '0')
        {
            digitalWrite(ledPin, LOW);
            Serial.println("LED: OFF");
            state = 0;
        }
        else if (state == '1')
        {
            digitalWrite(ledPin, HIGH);
            Serial.println("LED: ON");
            state = 0;
        }
    }

```

Handwritten code :

```

#define led 13
int state=0;
void setup()
{
    pinMode(led, OUTPUT);
    Serial.begin(38400);
}
void loop()
{
    if (Serial.available() > 0)
    {
        Serial.println("LED: OFF");
        state=0;
    }
    else if (state == '1')
    {
        digitalWrite (ledpin, HIGH);
        Serial.println ("LED: ON");
        state = 0;
    }
}

```

Observation: Using the username and password that was established in the previous experiment, the LED was switched ON/OFF using the android mobile application.

Program no: 17 Program Title: **HOME AUTOMATION(RELAY)**

Aim: To see the working of LED control with the help of an android mobile application.

Components Required :

Arduino Uno board - 1

USB Cable - 1

Bluetooth model - 1

Android Phone - 1

4 - Channel Relay - 1

Breadboard - 1

Jumper wires

Arduino Bluetooth App

Pin Connections :

- Output 1 to Pin 2 (Arduino Board)
- Output 2 to Pin 3
- Output 3 to Pin 4
- Output 4 to Pin 5
- 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

Code :

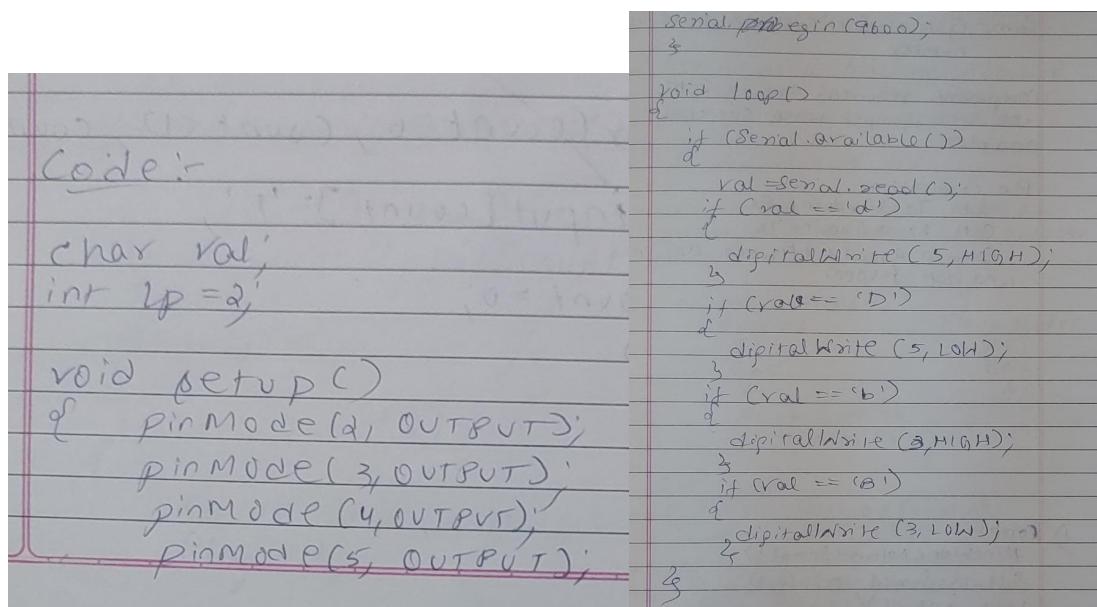
```
char val;  
  
int ledpin = 2;  
void setup()  
{  
    pinMode(ledpin = 2, OUTPUT);  
    pinMode(ledpin = 3, OUTPUT);  
    pinMode(ledpin = 4, OUTPUT);  
    pinMode(ledpin = 5, OUTPUT);  
    Serial.begin(9600);  
}  
void loop()  
{  
    if( Serial.available()  
    {  
        ;  
    }  
    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);
}
}
}

```

Handwritten code :



Observation: Home automation commands when connected through relay.

Program no: **18** Program Title: **PROGRAMMING ARDUINO WITH GSM MODULE**

Aim: To Program Arduino with GSM Module

Components Required :

Arduino Uno board - 1

USB Cable - 1

GSM module - 1

Android Phone - 1

Airtel SIM card - 1

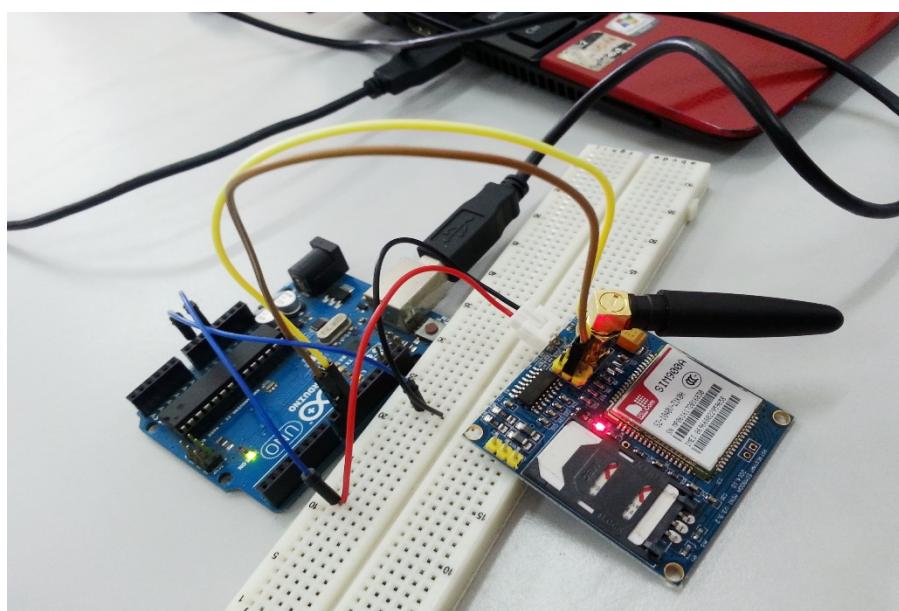
Breadboard - 1

Flame Sensor(For flame sensor program) - 1

Jumper wires

Pin Connections:

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



1. GSM Module: Call to a particular number

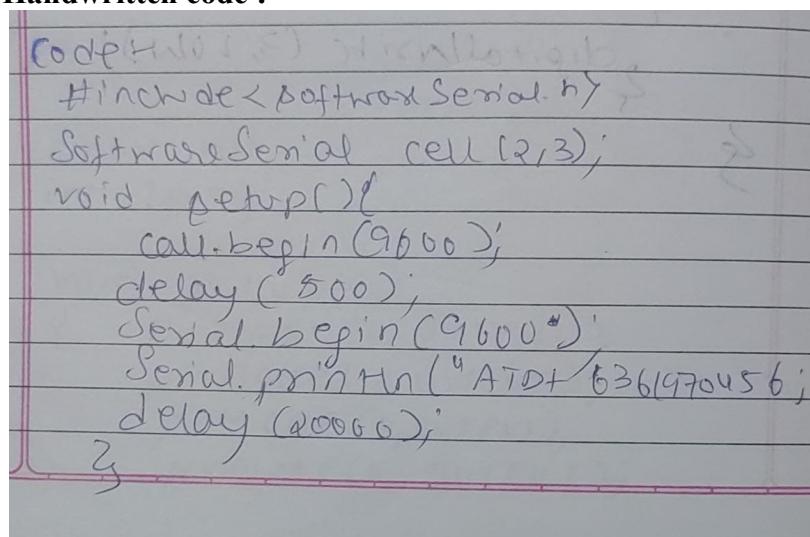
Aim : Call using Arduino and GSM Module – to a specific mobile number inside the program.

Code :

```
#include SoftwareSerialcell(2,3);  
void setup()  
{
```

```
cell.begin(9600);
delay(500);
Serial.begin(9600);
Serial.println("CALLING.....");
cell.println("ATD+919538433364;");
}
void loop()
{}
```

Handwritten code :



The image shows handwritten code on lined paper. At the top, it says "Code Handwritten" and "Date: 10/10/2023". The code itself starts with "#include <SoftwareSerial.h>" followed by "SoftwareSerial cell(2,3);". The "void setup()" section contains "cell.begin(9600);", "delay(500);", "Serial.begin(9600)", and "Serial.println("ATD+6361970456;"). There is also a "delay(20000);". The entire code is enclosed in curly braces {}, with one brace on each side of the code block.

```
#include <SoftwareSerial.h>
SoftwareSerial cell(2,3);
void setup()
{
    cell.begin(9600);
    delay(500);
    Serial.begin(9600);
    Serial.println("ATD+6361970456");
    delay(20000);
}
```

Observation: Using a specific number calls were made and received by the GSM module.

2. 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:

Arduino to Flame Sensor

5V to VCC

GND to GND

Code :

```
#include SoftwareSerialcell(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+916361970456;");
        delay(10000);
        cell.println("ATH");
    }
}
```

Handwritten code :

```
Code)-  
#include <SoftwareSerial.h>  
SoftwareSerial(2,3);  
void setup(){  
    call.begin(9600);  
    delay(500);  
    Serial.begin(9600);  
}  
void loop(){  
    int val = analogRead(A0);  
    Serial.println(val);  
    delay(1000);
```

```
if (val < ) {  
    Serial.println("calling ...");  
    cell.println("ATDT 6361920456");  
    delay(1000);  
    cell.println("ATH");  
}
```

~~Otp serv
21/1/23~~

Observation: Calls were made by the GSM module to a specific number when fire/flame was detected.

3. Sending and Receiving Message

Aim:

- 1) Send SMS using Arduino and GSM Module – to a specific mobile number inside the program
- 2) Receive SMS using Arduino and GSM Module – to the SIM card loaded in the GSM Module. Program: Note: According to the code, messages will be sent and received when ‘s’ and ‘r’ are pressed through the serial monitor respectively.

Code :

```
#include 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)
        Serial.write(mySerial.read());
}

void SendMessage()
{
    mySerial.println("AT+CMGF=1");
    delay(1000);
```

```

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

```

Handwritten code :

Code :-

```

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

void setup() {
  mySerial.begin(9600);
  Serial.begin(9600);
  delay(100);
}

void loop() {
  if (Serial.available() > 0)
  { write((Serial.read()))
    if ('s' == write)
      send();
    else if ('r' == write)
      receive();
  }
  if (mySerial.available())
    Serial.write(mySerial.read());
}

void send()
{
  mySerial.println("AT+CMGF =1");
  delay(1000);
}

```

```

mySerial.println("AT+CMGS=1"+  

    "916361930456\"\\r\"");
delay(1000);
mySerial.println("I am SMS");
delay(1000);
mySerial.println((char)26);
delay(1000);
}

```

Observation: Messages were sent and received to and from the GSM module.

4. Controlling LED through received messages:

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

Pin Connections:

- Attach LED to pin 13 and GND.

Code :

```

#include SoftwareSerialcell(2,3);

voidreadfn()

{
    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();
}

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...");
        }
    }
}
```

Handwritten code :

```
Code:  
#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);  
    cell.begin(9600);  
    cell.println("AT");  
    delay(1000);  
    read();  
    cell.println("AT+CNMI=1,2,0,0,0");  
}  
  
void loop(){  
    if (cell.available())  
    {  
        String msg = cell.readString();  
        Serial.println(msg);  
        if (msg.indexOf("SWITCH ON") > 0)  
        {  
            digitalWrite(13, HIGH);  
        }  
        else if (msg.indexOf("SWITCH OFF") > 0)  
        {  
            digitalWrite(13, LOW);  
        }  
        else  
        {  
            Serial.println("nothing to do...");  
        }  
    }  
}
```

Observation: Received messages through Arduino and GSM Module to control Switching ON/OFF the LED