

## Introduction:-

### Arduino installation:-

2.2.1

Windows 10 and latest

- Go to arduino website
- then search for latest version
- Click download for windows
- Install Arduino-IDE using executable file exe file.

### =) Features:-

- \* Arduino programming is a simplified version of C++, which makes the learning process easy
- \* The Arduino IDE is used to control the functions of boards. It further sends the set of specifications to the microcontroller.
- \* Arduino does not need an extra-board or piece to load new code.

- \* Arduino can read analog and digital input signals
- \* The hardware and software platform is easy to use and implement.

### Microprocessor :

A processing device implemented on a single chip is called a microprocessor. A microprocessor is the most crucial component of computer or any other computing device. Because it is entirely responsible for processing data based on instructions to produce information.

Intel 8085 or 8086 processing chips are the examples of microprocessors.

### Microcontroller :

A microcontroller is an electronic system which consists of a processing element, a small memory (RAM, ROM, EPROM) I/O ports, etc. on a single chip. Therefore, a microcontroller is a tiny resemblance of a microcomputer.

It is a quite small and low-cost electronic device which is used in several electronic appliances as the main functioning device.

Arduino UNO is not a microprocessor. It is a microcontroller board. It is based on the ATmega328P. It has 14 digital input / output pins (of which 6 can be used as PWM outputs), 6 analog input, a 16 MHz ceramic resonator, a USB connection, a power jack, a ICSP header and a reset button.

20/11/2021

ZAB-

①

## LED Blinking

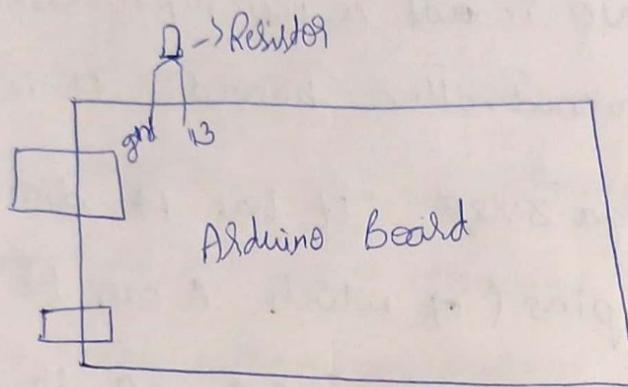
Aims - Turns on an LED on for one second, then off for one second, repeatedly.

Components / Hardware Required :-

- Arduino Board
- LED's

Circuit:

Diagram:



Code:-

```
// Pin 13 has an LED connected on most Arduino boards  
int led = 13;
```

```
void setup()
```

```
{  
    // initialize the digital pin as an output:  
    pinMode(led, OUTPUT);  
}
```

```
void loop()
```

```
{  
    digitalWrite(led, HIGH);  
    delay(1000);  
    digitalWrite(led, LOW);  
    delay(1000);  
}
```

## ② Observation:-

LFD turns ON.

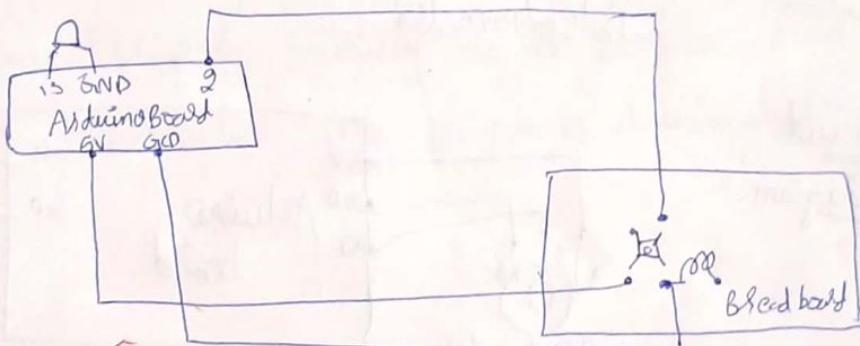
## ② LED ON/OFF Using Pushbutton

Aim:- Turn on LED ON/OFF using a Pushbutton.

Hardware Required:-

- A Arduino Board
- LED
- Push button

Circuit diagram:-



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

```

Observation:-

on pushing pushbutton Led glows and releasing it  
stop glowing.

③

### LED Toggling using Potentiometer

Aim:- To control the brightness of an LED using a potentiometer

Hardware

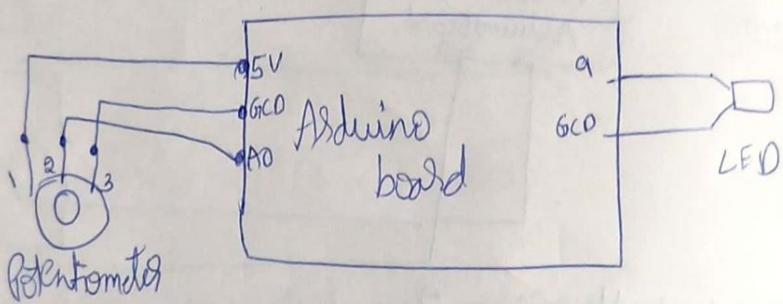
Required:-

: Arduino Board

: LED

Potentiometer

Circuit  
diagram:-



Code:-

```
const int analogInPin = A0;
const int analogOutPin = 9;
int sensorValue = 0;
int outputValue = 0;

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

```

void loop() {
    sensorValue = analogRead(analogInPin);
    outputValue = map(sensorValue, 0, 1023, 0, 255);
    analogWrite(analogOutPin, outputValue);
    Serial.print("Sensor Value:");
    Serial.println(outputValue);
    delay(2);
}

```

### Observation:-

→ LED's glows brightly initially as we turn the  
 Note of the potentiometer LED's brightness diminishes.

8th  
20/11

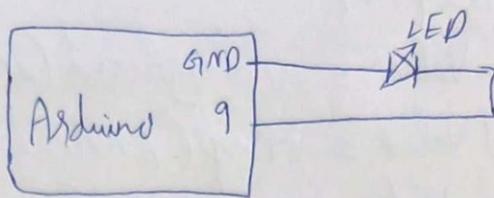
### ④ Fade potentiometer:-

Aim:- Constructing a circuit on fading of LED without  
 potentiometer

### Components:-

- Arduino board
- LED
- USB

→ Circuit diagram:-



→ Code:-

```
int led = 9;  
int brightness = 0;  
int fadeAmount = 5;  
  
void setup() {  
    pinMode(led, output);  
}  
  
void loop() {  
    analogWrite(led, brightness);  
    brightness = brightness + fadeAmount;  
  
    if (brightness <= 0 || brightness >= 255) {  
        fadeAmount = -fadeAmount;  
    }  
    delay(3);  
}
```

Observation:-

After uploading the program LED automatically  
Start fading.

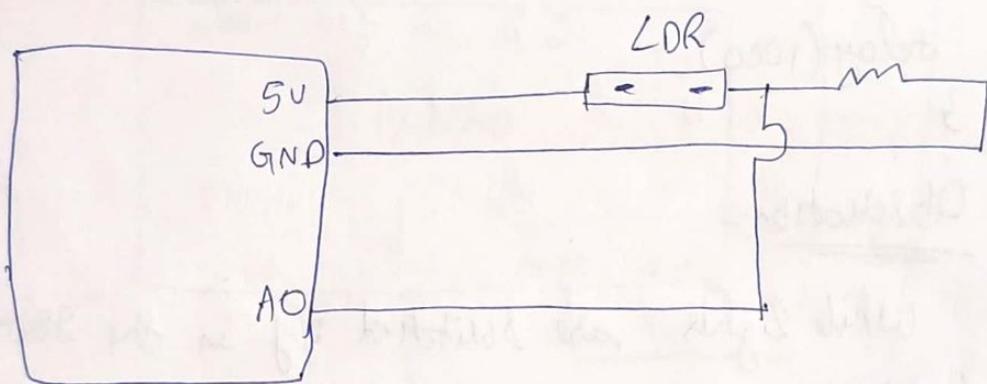
## (5) Night light simulation:-

Aim:- Simulating a night light using LDR and PIR.

Hardware Required :-

- 1 LED
- 1 LDR
- 110K Resistor

Circuit diagram:-



=> Connection:-

=> Code:-

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

```
void setup()
```

```
{
```

~~```
    Serial.begin(9600);  
    pinMode(11, Output);
```~~

```
3
```

```
void loop()
```

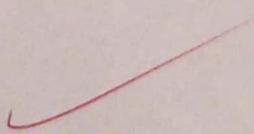
```
{
```

```
    LDRValue = analogRead(LDR);
```

```
Serial.println(LDR Value);  
delay(50);  
if (LDR Value < light-sensitivity)  
{  
    digitalWrite(11, High);  
}  
else  
{  
    digitalWrite(11, Low);  
}  
delay(1000);  
}
```

### Observation:-

While lights are switched off in the room, LED should switch ON, when lights are switched on in the room, LED should switch off immediately.



⑥

## PIR with Arduino UNO:

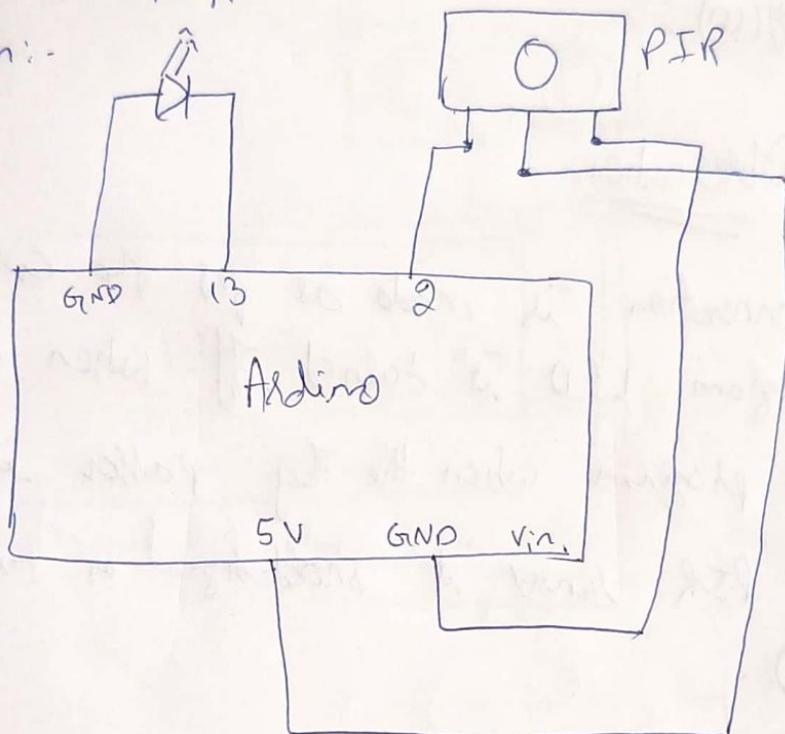
Aim:-

PIR with Arduino UNO

Hardware

- Arduino board
- bread
- PIR

Circuit diagram:-



Code:-

```
int sensorState = 0;  
void setup()  
{  
    pinMode(2, INPUT);  
    pinMode(13, OUTPUT);  
    Serial.begin(9600);  
}  
void loop()  
{  
    sensorState = digitalRead(2);  
    if (sensorState == HIGH) {  
        digitalWrite(13, HIGH);  
    }
```

```
Serial.println("Sensor activated!");  
} else {  
    digitalWrite(13,LOW);  
}  
delay(10);  
}  
}
```

→ Observation:-

Connection is made as per the circuit diagram. LED is turned off when the upload the program when the object passes in front of the PIR sensor it should signal or turn on LED.

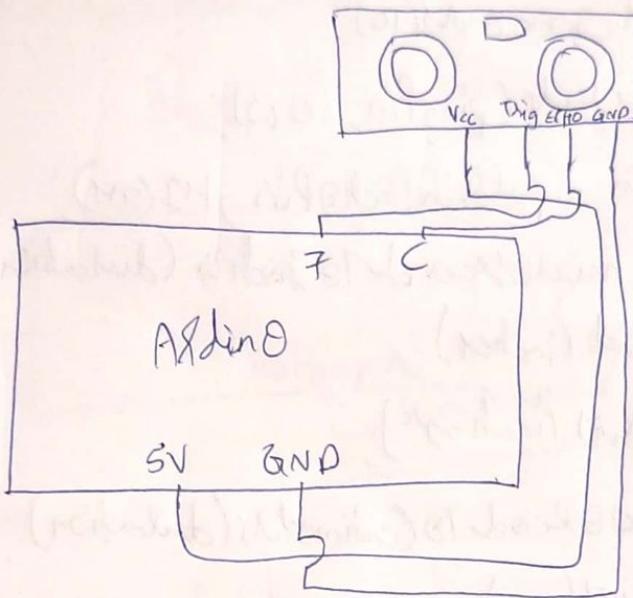
# ⑦ Ultrasonic with Arduino Uno

Aim:- To measure the distance

Hardware required:-

- Arduino board
- wires
- ultrasonic sensor

Circuit diagram:-



Code:-

```
const int pingPin = 7;
```

```
const int echoPin = 6;
```

Ultrasonic Sensor

```
void setup()
```

```
{
```

```
Serial.begin(9600);
```

~~```
pinMode(pingPin, OUTPUT);
```~~~~```
pinMode(echoPin, INPUT);
```~~

```
}
```

```
void loop()
```

```
{
```

log long duration, inches, cm;

digital write (pingPin, low);

delayMicroseconds(2);

digitalWrite(pingPin, HIGH);

delayMicroseconds(10);

digitalWrite(pingPin, LOW);

duration = pulseIn(#echoPin, HIGH);

inches = microsecondsToInches(duration);

Serial.print(inches);

Serial.print("inches");

cm = microsecondsToCentimeters(duration);

Serial.print(cm);

Serial.println("cm");

}

longMicrosecondsToInches(longMicroseconds)

{

return microseconds / 74 / 2;

}

longMicrosecondsToCm(longMicroseconds)

{

return microseconds / 29 / 5;

}

Observation:-

Sept 27/11/23

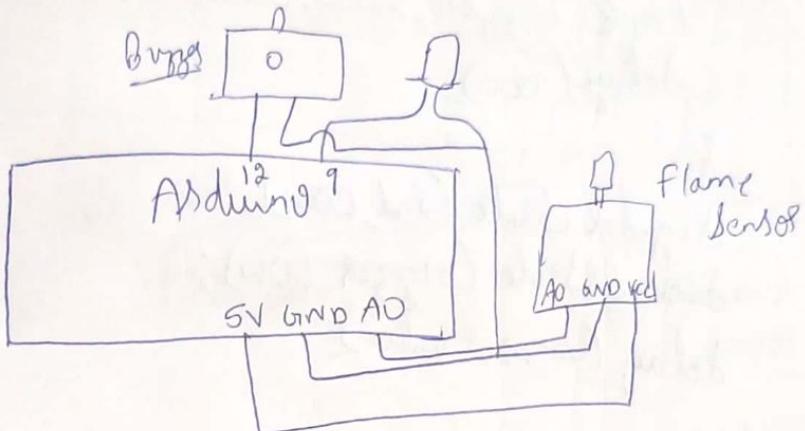
Connection made as per the circuit diagram  
The object finds it detects and reduce the distance  
of the object.

(8) Fire Alert  
Aim:- To design and implement fire alarm system using flame sensor and buzzer.

Hardware required :-

Arduino board  
LED's  
Connecting wires

Circuit diagram :-



Code:-

```
#include <SoftwareSerial.h>
#include <NewPing.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()
{
    sensorValue = analogRead(sensorPin);
    Serial.println(sensorValue);
```

if (SensorValue < 100)

{

Serial.println("Fire Detected");

Serial.println("led on");

digitalWrite(led, HIGH);

tone(buzzer, 2000, 1000);

delay(1000);

}

digitalWrite(led, LOW);

digitalWrite(buzzer, LOW);

delay(SensorValue);

}

### Observation:-

Buzzers is made high pitch ~~sound~~ sound when  
it detect flame to the flame sensor.

①

## Moisture sensor

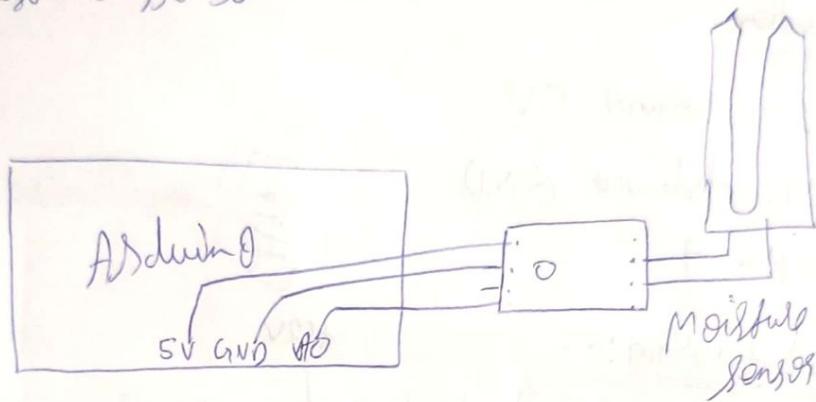
Aim:- To find the Moisture level in the Sand

Hardware required:

- Arduino Board

- Moisture Sensor

Circuit diagram :-



Code:-

```

int sensorPin = A0;
int sensorValue = 0;

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

void loop() {
  sensorValue = analogRead(sensorPin);
  Serial.println(sensorValue);
  delay(1000);
}
  
```

Observation:-

The soil moisture sensor could determine the moisture level present in the surrounding sand.

gauri  
11/12/23

(10)  
(8) 12/23

## RFID tag :-

Reading the code present on RFID tag:

The following code will read the code present on RFID tag and print it in serial monitor.

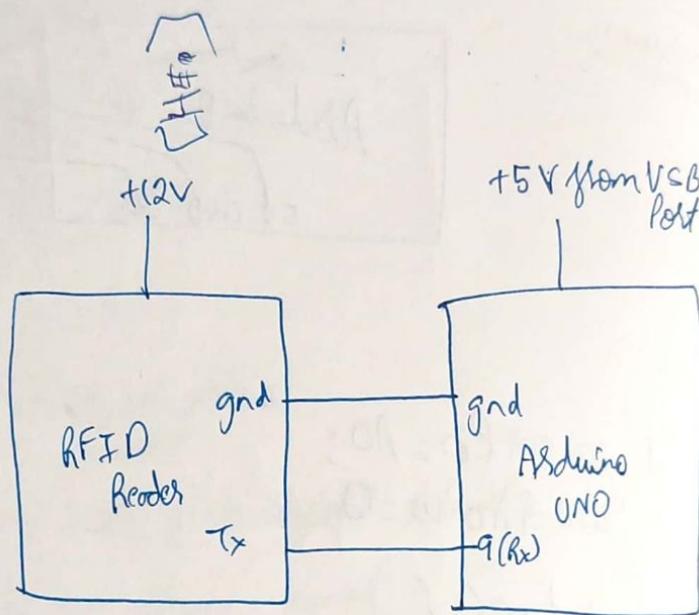
Connection:-

5V - Arduino 5V

GND - Arduino GND

Tx - pin 9

Circuit diagram:-



Code:-

```
#include<SoftwareSerial.h>
SoftwareSerial mySerial(9,10);
int count = 0;
char input[12];
boolean flag = 0;
void setup()
{
    Serial.begin(9600);
    mySerial.begin(9600);
}
void loop()
```

```
2 count = 0;  
while (mySerial.available() && count < 12)  
{  
    input [count] = mySerial.read();  
    count++;  
    delay(5);  
}  
3 Serial.println(input);
```

3 Conclusion:-

The o/p consists of 12 characters ASCII data, where first 10 bits will be the tog number and last 2 bits will be XOR result of the tog number which can be used for error collection.

Actual count

Aim: Access control through RFID.

11

Connection:-

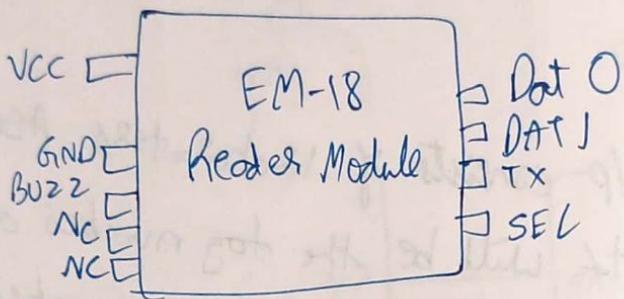
5V - Arduino 5V

GND - Arduino GND

Tx-pin = 9

Led-pin = 12

Circuit diagram:-



Code:-

```
#include <SoftwareSerial.h>
SoftwareSerial mySerial(9,10);
#define LEDPIN 12
char tag[] = "53002920D087";
char input[12];
int count=0;
character array;
boolean flag=0;
void setup()
{
    Serial.begin(9600);
    mySerial.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)
```

```
{
```

```
            count = 0;
```

```
            flag = 1;
```

```
            while (count <= 12 && flag != 0)
```

```
{
```

```
                if (input[count] == flag [count])
```

```
                    flag = 1;
```

```
                else
```

```
                    flag = 0;
```

```
                count++;
```

```
}
```

```
}
```

$y(\text{flag} == 1)$

2

```
Serial.println("Access Allowed");
digitalWrite(LEDPIN, HIGH);
delay(2000);
digitalWrite(LEDPIN, LOW);
```

3

else

4

```
Serial.println("Access Denied");
digitalWrite(LEDPIN, LOW);
delay(2000);
```

5

```
for(count=0; count < 12; count++)
```

6

```
input[count] = 'f';
```

7

```
count = 0;
```

8

9

Observation:-

When the HFID Tag no matches with it allows access and an appropriate message is displayed the ID number else access is denied.

23/2/23

## Detecting temperature via temp sensor

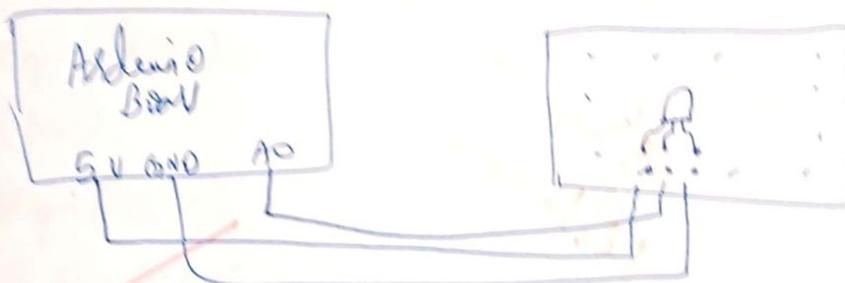
Aim:- Using a temperature sensor (LM35) to detect temperature in degrees and fahrenheit.

Components required :-

- Arduino Board
- Temperature sensor [LM35]
- Bread board

Circuit

diagram:-



Code:-

```
int outputPin = A0;  
void setup ()  
{  
    Serial.begin(9600);  
}  
void loop ()  
{
```

```
    int rawVoltage = analogRead(outputPin);
```

```
    float millivolts = (rawVoltage / 1024.0) * 5000;
```

```
    float celcius = millivolts / 10;
```

S. dial. plnt (Celsius);

S. dial. plnt ("Degrees (Celsius,")";

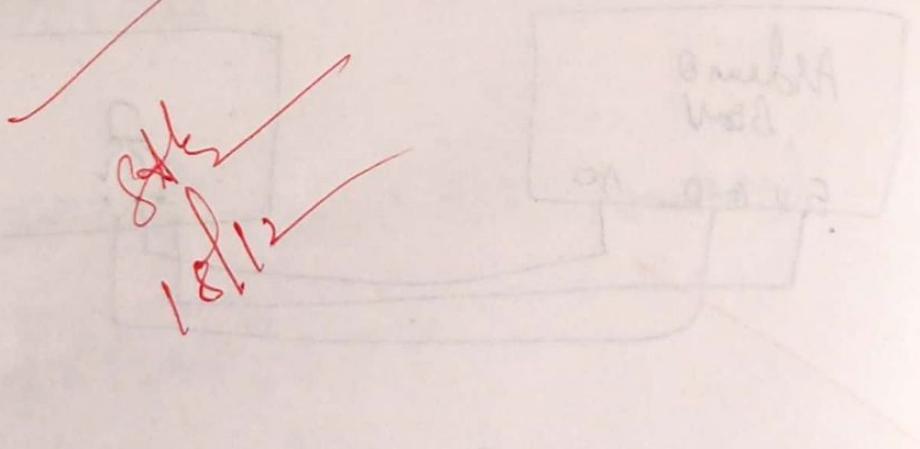
S. dial. plnt ((Celsius \* 9) / (5 + 32));

S. dial. plnt ln ("Degree Fahrenheit");

3

Observation:-

The temperature was used and observed in degrees and Fahrenheit.



⑬  
8/1/24

## GSM Module:

### Call to Particular Number

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

#### Program:-

```
#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 + 8277094848;");
    delay(20000);
}
```

```
void loop()
{}
```

Output:-

Calling

⑭

Call to a particular member on Alert :-

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

Connection for flame sensor :-

Arduino

5V

GND

AO

Flame Sensor

VCC

GND

AO

Program :-

```
#include <SoftwareSerial.h>
SoftwareSerial cell(2,3);
void setup()
{
    cell.begin(9600);
    delay(500);
    Serial.begin(9600);
}
void loop()
{
    initial = analogRead(A0);
    Serial.println(val);
    delay(1000);
    if (val < 50)
        Serial.println ("Calling").
```

cell. plinth ("ATH 91 97429 80606 3")

clay (1000)

cell. plinth ("ATH")

}

}

Off:-

Calling:-

(15)

## Sending and Receiving Messages

Aim:- Send and Receive SMS using Arduino and GSM module to a specified mobile number inside the programs.

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':
            send Message();
            break;

        case 'R':
            receive Message();
            break;
    }
}
```

g (mySerial.available())

Serial.write (mySerial.read())

}

void sendMessage ()

{

mySerial.println ("AT+CMGR=1");

delay (1000);

mySerial.println ("AT+CMGS=1"+91972780606);

delay (1000);

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

delay (100);

mySerial.println ((char) 26);

delay (1000);

}

void receiveMessage ()

{

mySerial.println ("AT+CNMI = 2,0,0,0");

delay (100);

}

# ⑯ Controlling LED through Received message

Aim:- Use received msg's through Arduino and GSM Module to control switching ON/OFF the led.

Connection:- Attach LED to pin 13 and GND.

Program:

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

```
    pinMode(13,OUTPUT);
    Serial.begin(9600);
    cell.begin(9600);
    cell.println("AT");
    delay(1000);
    read();
    cell.println("AT+CNMI=1,1,0,0,0");
```

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.. ");~~

}

}

}

~~guruji  
2/2/24~~