

**School of Engineering and Technology**

**Internet of Things**

*(Practical File)*

**18BCS-0IT32L**

***Submitted To:***

***Submitted By:***

Name: ASHVIN

Roll Number: 170BTCCSE063

Branch: Computer Science and Engg.

Year/Sem: 3rd Year – 6th Semester

Faculty: Mr. Antim Dev Mishra

Experiment 1

**Aim:** Write a program to test active and passive sensors

**Apparatus:**

* Arduino UNO
* Motion Sensor (Active Sensor)
* Ultrasonic Sensor (Passive Sensor)

**Theory:**

**Motion Sensor (Active Sensor)**

A motion detector is a device that detects moving objects, particularly people. Such a device is often integrated as a component of a system that automatically performs a task or alerts a user of motion in an area. They form a vital component of security, automated lighting control, home control, energy efficiency, and other useful systems



Passive infrared (PIR) sensors are sensitive to a person's skin temperature through emitted black-body radiation at mid-infrared wavelengths, in contrast to background objects at room temperature. No energy is emitted from the sensor, thus the name passive infrared. This distinguishes it from the electric eye for instance (not usually considered a motion detector), in which the crossing of a person or vehicle interrupts a visible or infrared beam. These devices can detect objects, people, or animals by picking up one's infrared radiation.

**Arduino UNO**

The Arduino Uno is a microcontroller board based on the ATmega328. It has 14 digital input/output pins (of which six can be used as PWM outputs), six analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started.

The Arduino Uno differs from all preceding boards because it does not use the FTDI USB-to-serial driver chip. Instead, it features the ATmega8U2 programmed as a USB-to-serial converter. Revision 2 of the Arduino Uno board has a resistor pulling the 8U2 HWB line to ground, making it easier to put into DFU mode.

|  |  |
| --- | --- |
| **Features** |  |
| * 14 digital I/O pins (six of which provide PWM output) * 3.3 V supply generated by an on-board regulator * Six analog input pins * 32 KB of flash memory | * Can supply 40 mA of DC current per pin * 16 MHz clock speed * Code example from Arduino website to help get started |

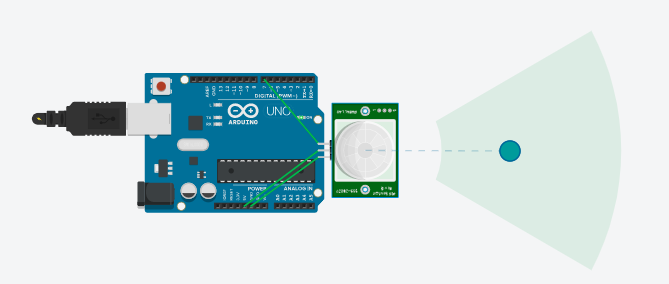
**Ultrasonic Sensor (Passive Sensor)**

Ultrasonic sensors work by emitting sound waves at a frequency too high for humans to hear. They then wait for the sound to be reflected back, calculating distance based on the time required. This is similar to how radar measures the time it takes a radio wave to return after hitting an object.

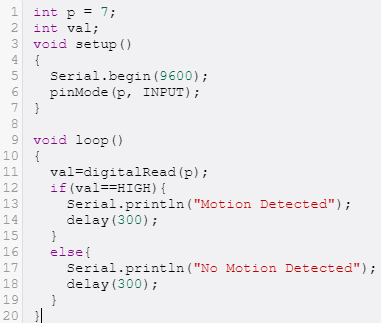


**Interfacing Active Sensor:**

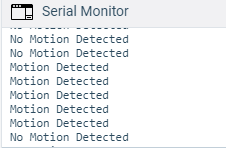
**Circuit:**



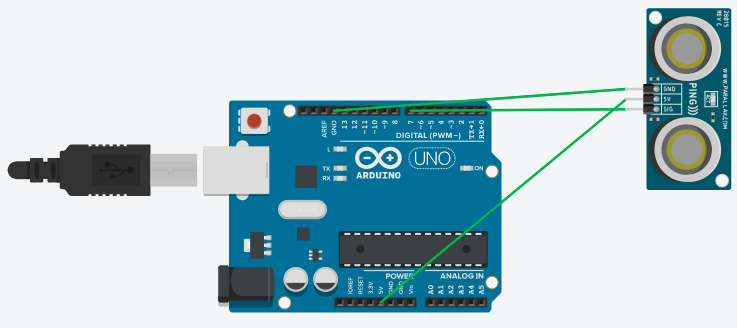
**Code:**



**Result:**

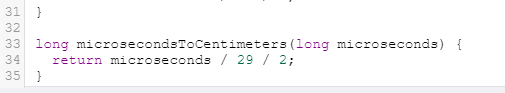


**Interfacing Passive Sensor:**

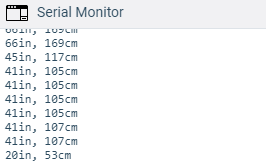


**Code:**





**Output:**



Experiment 2

**Aim:** Write a program to test actuators

**Apparatus:**

* Arduino UNO
* Servo Motor

**Theory:**

**Arduino UNO**

The Arduino Uno is a microcontroller board based on the ATmega328. It has 14 digital input/output pins (of which six can be used as PWM outputs), six analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started.

The Arduino Uno differs from all preceding boards because it does not use the FTDI USB-to-serial driver chip. Instead, it features the ATmega8U2 programmed as a USB-to-serial converter. Revision 2 of the Arduino Uno board has a resistor pulling the 8U2 HWB line to ground, making it easier to put into DFU mode.

|  |  |
| --- | --- |
| **Features** |  |
| * 14 digital I/O pins (six of which provide PWM output) * 3.3 V supply generated by an on-board regulator * Six analog input pins * 32 KB of flash memory | * Can supply 40 mA of DC current per pin * 16 MHz clock speed * Code example from Arduino website to help get started |

**Servo Motor**

A servo motor is an electrical device which can push or rotate an object with great precision. If you want to rotate and object at some specific angles or distance, then you use servo motor. It is just made up of simple motor which run through servo mechanism. If motor is used is DC powered then it is called DC servo motor, and if it is AC powered motor then it is called AC servo motor. We can get a very high torque servo motor in a small and light weight packages. Doe to these features they are being used in many applications like toy car, RC helicopters and planes, Robotics, Machine etc.

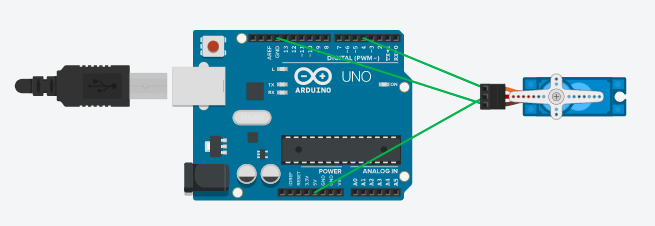


It consists of three parts:

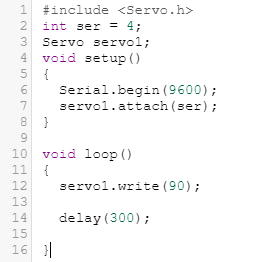
1. Controlled device
2. Output Sensor
3. Feedback system

It is a closed loop system where it uses positive feedback system to control motion and final position of the shaft. Here the device is controlled by a feedback signal generated by comparing output signal and reference input signal.

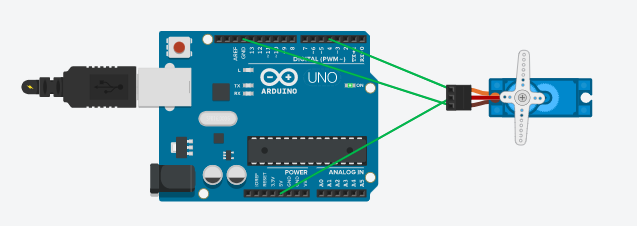
**Circuit Diagram:**



**Code:**



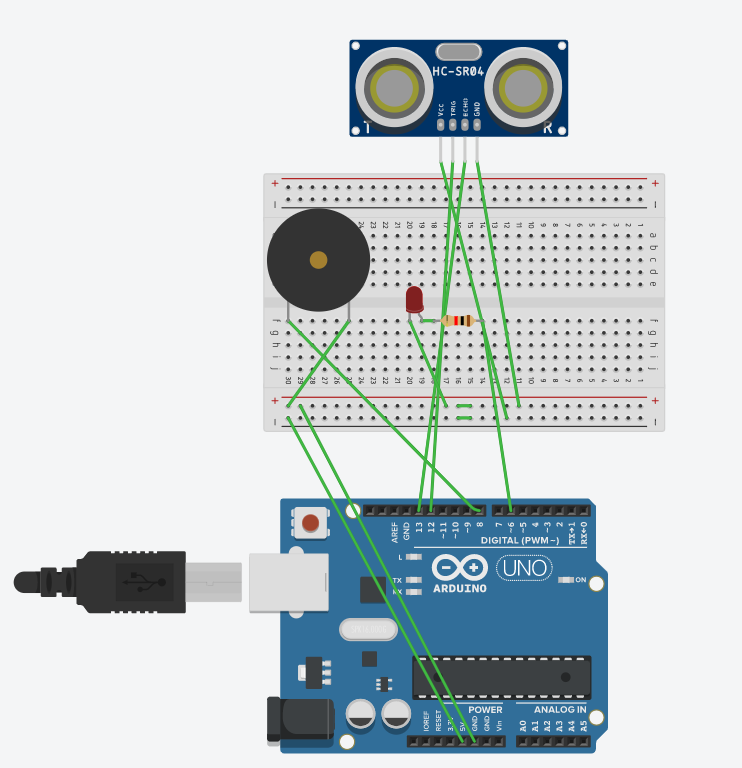
**Result:**



Experiment 3

**Aim:** WAP in Arduino to turn on the buzzer and LED (Red) if a person is less than 3 feet from the billing counter of a shop.

Circuit Diagram-



Code

#define trigPin 12

#define echoPin 13

int buzzer = 8;

int ledPin= 6;

int duration, distance;

void setup() {

Serial.begin (9600);

pinMode(trigPin, OUTPUT);

pinMode(echoPin, INPUT);

pinMode(buzzer, OUTPUT);

pinMode(ledPin, OUTPUT);

}

void loop() {

digitalWrite(trigPin, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin, LOW);

duration = pulseIn(echoPin, HIGH);

distance = (duration/2) / 29.1;

if (distance >= 213 || distance <= 0)

{

Serial.println("No personDetected");

digitalWrite(buzzer,LOW);

digitalWrite(ledPin,LOW);

}

else {

Serial.println("personDetected \n");

Serial.print("distance= ");

Serial.print(distance);

tone(buzzer,400);

digitalWrite(ledPin,HIGH);

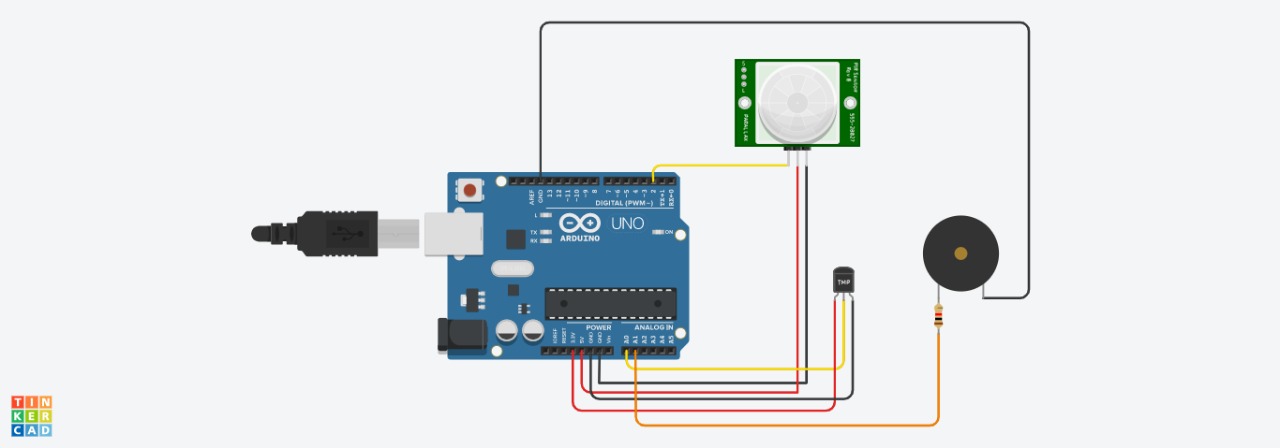
}

}

Experiment 4

**Aim:** WAP in Arduino to count the no. of person enter in a hospital, record their body temperature and generate an alert through buzzer if body temp is more than 98 F.

Circuit Diagram-



int pirPin = 2;

int temppin= A0;

float temp=0;

int Buzz=A1;

int counter = 0;

int state;

int laststate = HIGH;

void setup(){

Serial.begin(9600);

pinMode(pirPin, INPUT);

pinMode(Buzz,OUTPUT);

}

void loop(){

int state = digitalRead(pirPin);

if ( state != laststate) {

temp=analogRead(temppin);

temp=temp\*0.48828125;

Serial.println(temp);

delay(1000);

if (temp>30)

{digitalWrite(Buzz,HIGH);}

else

{digitalWrite (Buzz,LOW);}

counter=counter+1;

Serial.println(counter);

}

else{

laststate = state;

delay(0);

}

}

Experiment 5

**Aim:** Write an IOT program in Tinkercad to develop a project on environment applications using sensors and smart boards.

**Apparatus:**

* Arduino UNO
* Grove - Gas Sensor(MQ2)
* Buzzer

**Working Principle:**

**Theory:**

**Buzzer**



A buzzer or beeper is an audio signalling device, which may be mechanical, electromechanical, or piezoelectric. Typical uses of buzzers and beepers include alarm devices, timers, and confirmation of user input such as a mouse click or keystroke.

**Grove - Gas Sensor(MQ2)**



The Grove - Gas Sensor(MQ2) module is useful for gas leakage detection (home and industry). It is suitable for detecting H2, LPG, CH4, CO, Alcohol, Smoke or Propane. Due to its high sensitivity and fast response time, measurement can be taken as soon as possible. The sensitivity of the sensor can be adjusted by potentiometer.

**Arduino UNO**



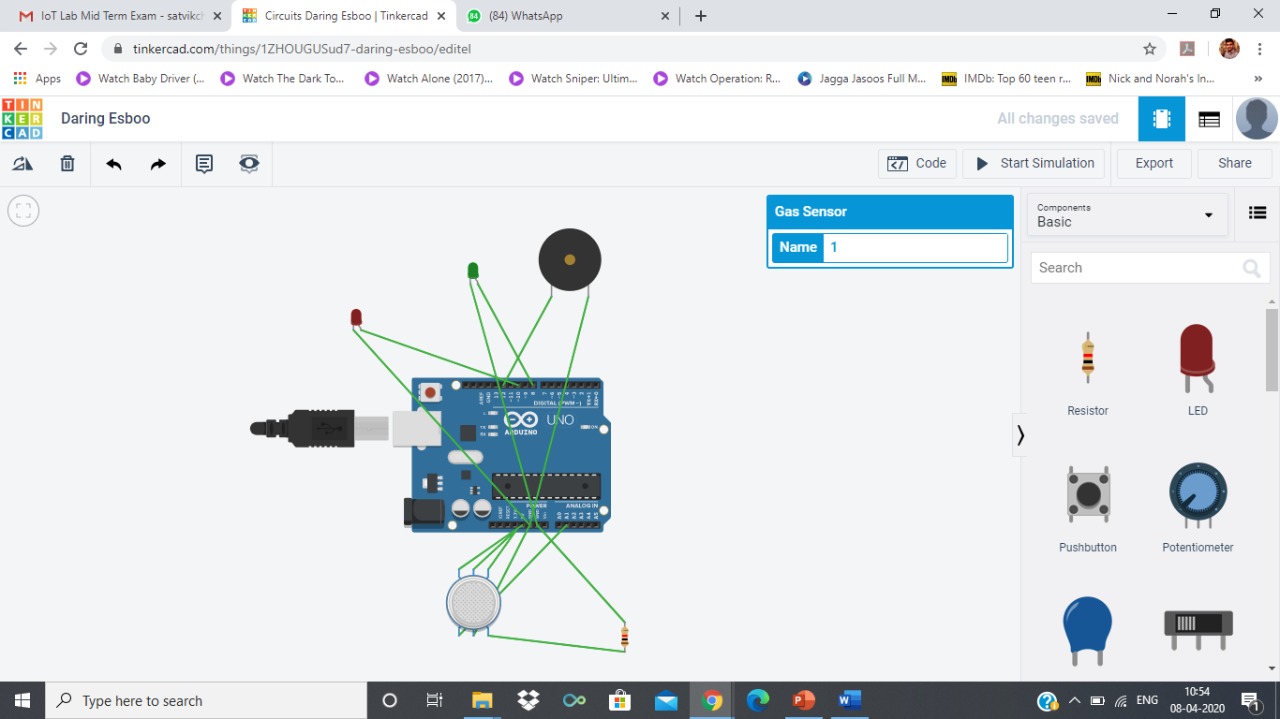
The Arduino Uno is a microcontroller board based on the ATmega328. It has 14 digital input/output pins , six analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started.

The Arduino Uno differs from all preceding boards because it does not use the FTDI USB-to-serial driver chip. Instead, it features the ATmega8U2 programmed as a USB-to-serial converter. Revision 2 of the Arduino Uno board has a resistor pulling the 8U2 HWB line to ground, making it easier to put into DFU mode.

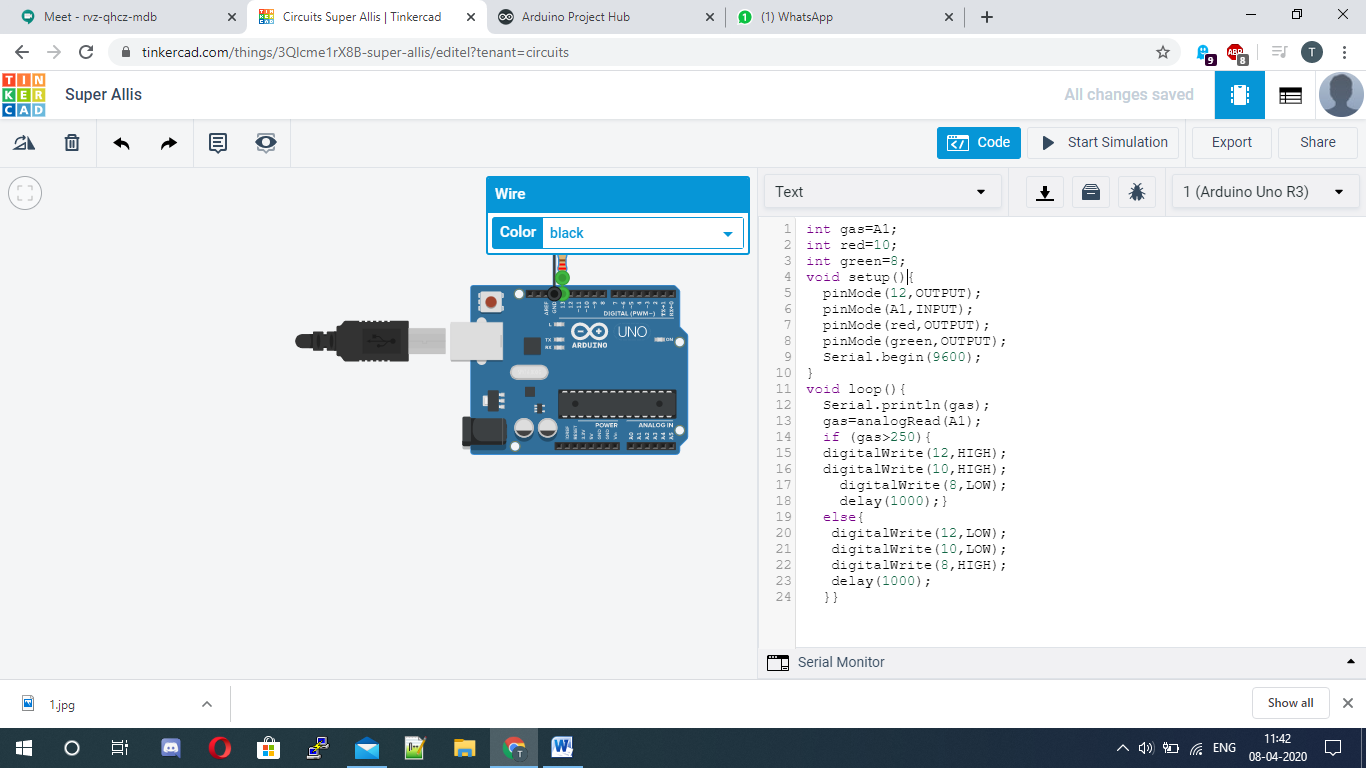
|  |  |
| --- | --- |
| **Features** |  |
| * 14 digital I/O pins (six of which provide PWM output) * 3.3 V supply generated by an on-board regulator * Six analog input pins * 32 KB of flash memory | * Can supply 40 mA of DC current per pin * 16 MHz clock speed * Code example from Arduino website to help get started |

**Interfacing Sensor:**

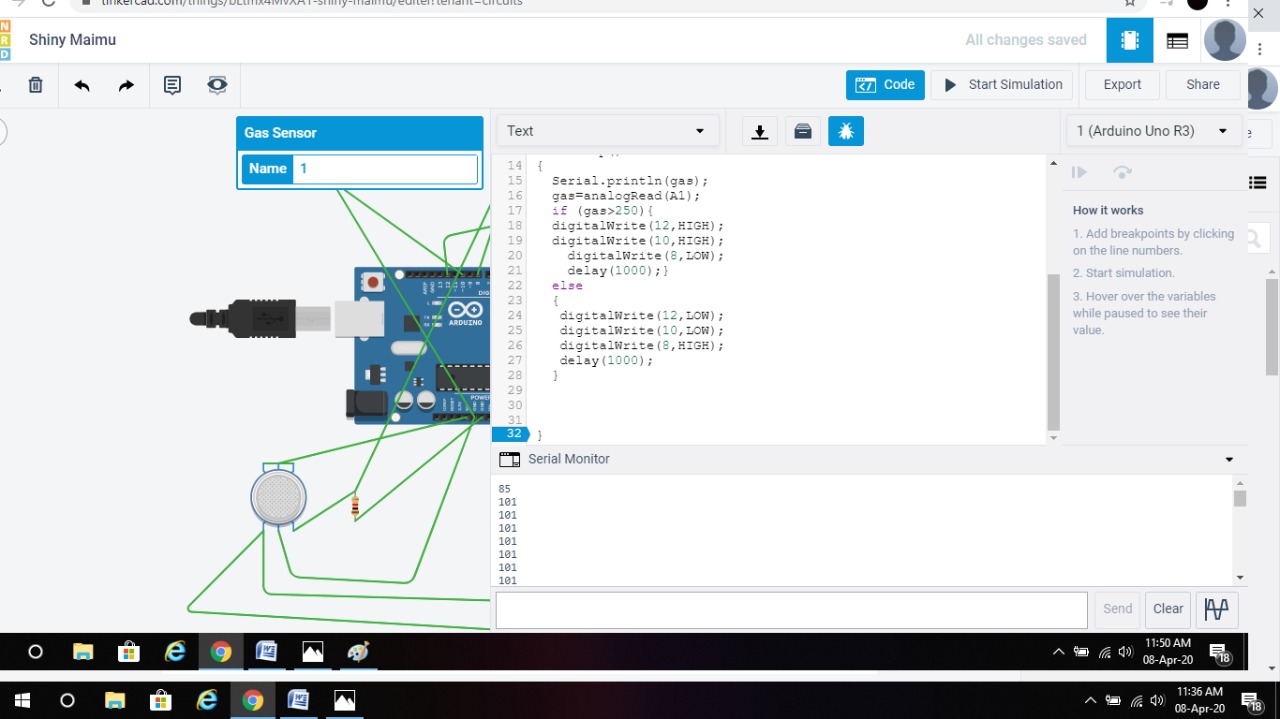
**Circuit:**



**Code:**



**Result:**



Experiment 6

**Aim:** Write a program in Arduino to generate alert through buzzer if (gas sensor values > 500) .

int val=A1;

void setup(){

pinMode(7,OUTPUT);

pinMode(A1,INPUT);

Serial.begin(9600);}

void loop(){

Serial.println(val);

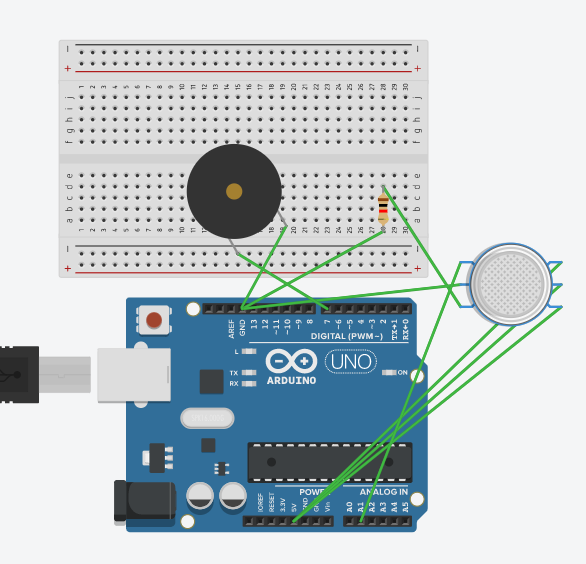
val=analogRead(A1);

if(val>500){

digitalWrite(7,HIGH); }

else{

digitalWrite(7,LOW);} }



Experiment 7

**Aim:** Write a program in Arduino to generate alert through buzzer if gas sensor values greater than the threshold value 250 then red LED high and green LED low and buzzer tone 1000 else red LED LOW green LED high and no buzzer tone.

int redLED = 11;

int greenLED = 10;

int buzzer = 13;

int smokeSense = A0;

int sensorThres = 300;

void setup()

{

pinMode(redLED, OUTPUT);

pinMode(greenLED, OUTPUT);

pinMode(buzzer, OUTPUT);

pinMode(smokeSense, INPUT);

Serial.begin(9600);}

void loop()

{

int analogSensor = analogRead(smokeSense);

Serial.print("Pin A0: ");

Serial.println(analogSensor);

if (analogSensor > sensorThres)

{

digitalWrite(redLED, HIGH);

digitalWrite(greenLED, LOW);

tone(buzzer, 1000, 200);

}

else{

digitalWrite(redLED, LOW);

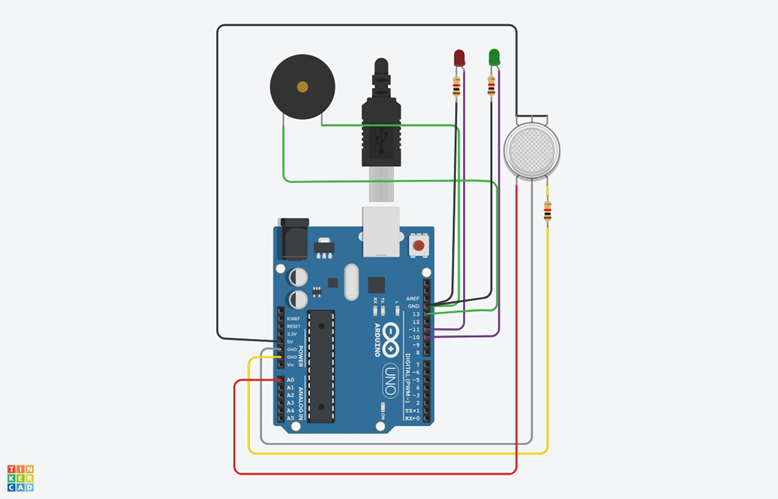
digitalWrite(greenLED, HIGH);

noTone(buzzer);

}

delay(100);

}



Experiment 8

**Aim:** Write a program in Arduino to turn light using PIR and relay.

int p = 7;

int val;

int l = 12;

void setup(){

Serial.begin(9600);

pinMode(p, INPUT);

pinMode(l, OUTPUT);}

void loop(){

val=digitalRead(p);

if(val==HIGH){

Serial.println("Motion Detected");

digitalWrite(l, HIGH);

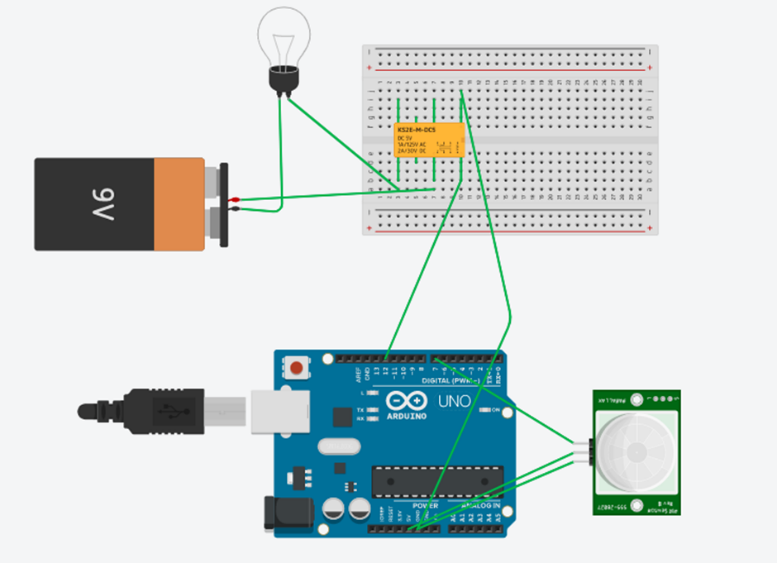
delay(300);

digitalWrite(l, LOW); }

else{

Serial.println("No Motion Detected");

delay(300); }}



Experiment 9

**Aim:** Aim: WAP in Tinkercad to develop earthquake detector using tilt sensor, LCD, Buzzer and Arduino Uno.

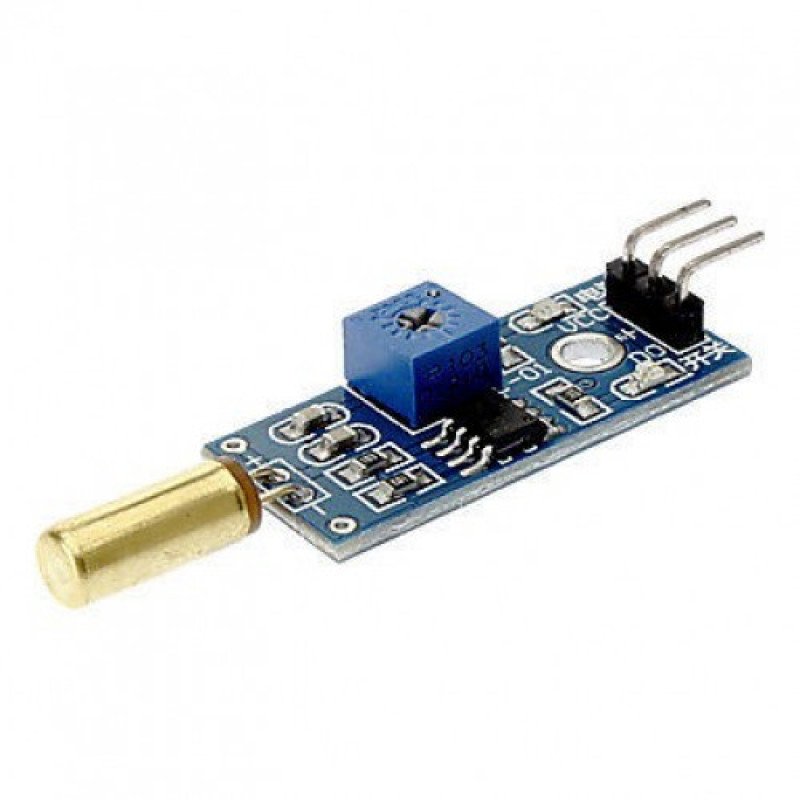
**Apparatus:**

* Tilt Sensor
* Arduino Uno
* Buzzer

**Theory:**

**Tilt Sensor**

A **tilt sensor** is an instrument that is used for measuring the **tilt** in multiple axes of a reference plane. **Tilt sensors** measure the **tilting** position with reference to gravity and are used in numerous applications. They enable the easy detection of orientation or **inclination.**



**Arduino UNO**

The Arduino Uno is a microcontroller board based on the ATmega328. It has 14 digital input/output pins (of which six can be used as PWM outputs), six analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started.

The Arduino Uno differs from all preceding boards because it does not use the FTDI USB-to-serial driver chip. Instead, it features the ATmega8U2 programmed as a USB-to-serial converter. Revision 2 of the Arduino Uno board has a resistor pulling the 8U2 HWB line to ground, making it easier to put into DFU mode.



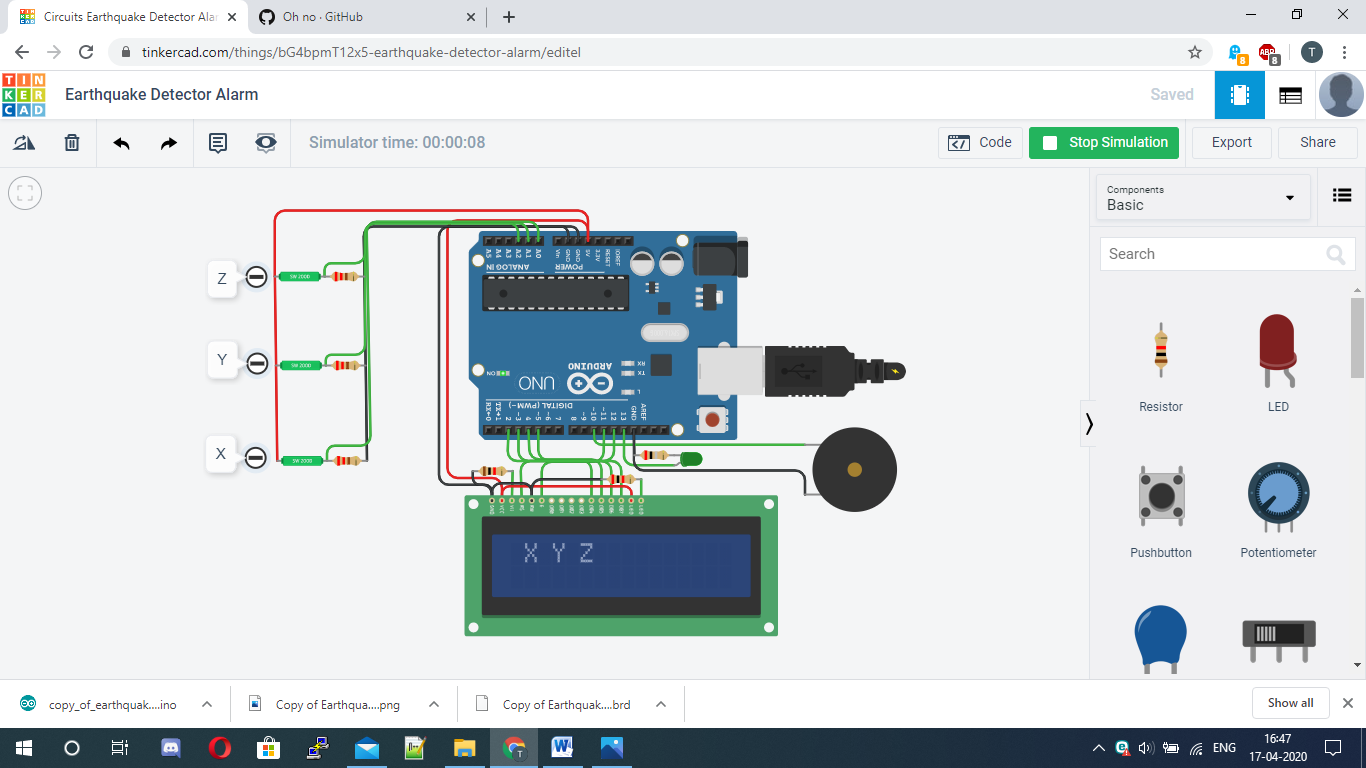
|  |  |
| --- | --- |
| **Features** |  |
| * 14 digital I/O pins (six of which provide PWM output) * 3.3 V supply generated by an on-board regulator * Six analog input pins * 32 KB of flash memory | * Can supply 40 mA of DC current per pin * 16 MHz clock speed * Code example from Arduino website to help get started |

**Buzzer**

A **buzzer** or beeper is an audio signalling device, which may be mechanical, electromechanical, or piezoelectric (piezo for short). Typical uses of **buzzers** and beepers include alarm devices, timers, and confirmation of user input such as a mouse click or keystroke.



**Circuit:**



**Code:**

#include<LiquidCrystal.h>

LiquidCrystal lcd(12,11,5,4,3,2);

#define buzzer 10

#define led 13

#define x A0

#define y A1

#define z A2

int xsample=0;

int ysample=0;

int zsample=0;

long start;

int buz=0;

#define samples 50

#define maxVal 20

#define minVal -20

#define buzTime 5000

void setup(){

lcd.begin(16,2);

Serial.begin(9600);

delay(1000);

lcd.print("EarthQuake ");

lcd.setCursor(0,1);

lcd.print("Detector ");

delay(2000);

lcd.clear();

lcd.print("Calibrating.....");

lcd.setCursor(0,1);

lcd.print("Please wait...");

pinMode(buzzer, OUTPUT);

pinMode(led, OUTPUT);

buz=0;

digitalWrite(buzzer, buz);

digitalWrite(led, buz);

for(int i=0;i<samples;i++){

xsample+=analogRead(x);

ysample+=analogRead(y);

zsample+=analogRead(z);}

xsample/=samples;

ysample/=samples;

zsample/=samples;

delay(3000);

lcd.clear();

lcd.print("Calibrated");

delay(1000);

lcd.clear();

lcd.print("Device Ready");

delay(1000);

lcd.clear();

lcd.print(" X Y Z ");}

void loop(){

int value1=analogRead(x);

int value2=analogRead(y);

int value3=analogRead(z);

int xValue=xsample-value1;

int yValue=ysample-value2;

int zValue=zsample-value3;

lcd.setCursor(0,1);

lcd.print(xValue);

lcd.setCursor(6,1);

lcd.print(yValue);

lcd.setCursor(12,1);

lcd.print(zValue);

delay(100);

if(xValue < minVal || xValue > maxVal || yValue < minVal || yValue > maxVal || zValue < minVal || zValue > maxVal){

if(buz == 0)

start=millis();

buz=1; }

else if(buz == 1) {

lcd.setCursor(0,0);

lcd.print("Earthquake Alert ");

if(millis()>= start+buzTime)

buz=0;}

else{

lcd.clear();

lcd.print(" X Y Z ");}

digitalWrite(buzzer, buz);

digitalWrite(led, buz);

Serial.print("x=");

Serial.println(xValue);

Serial.print("y=");

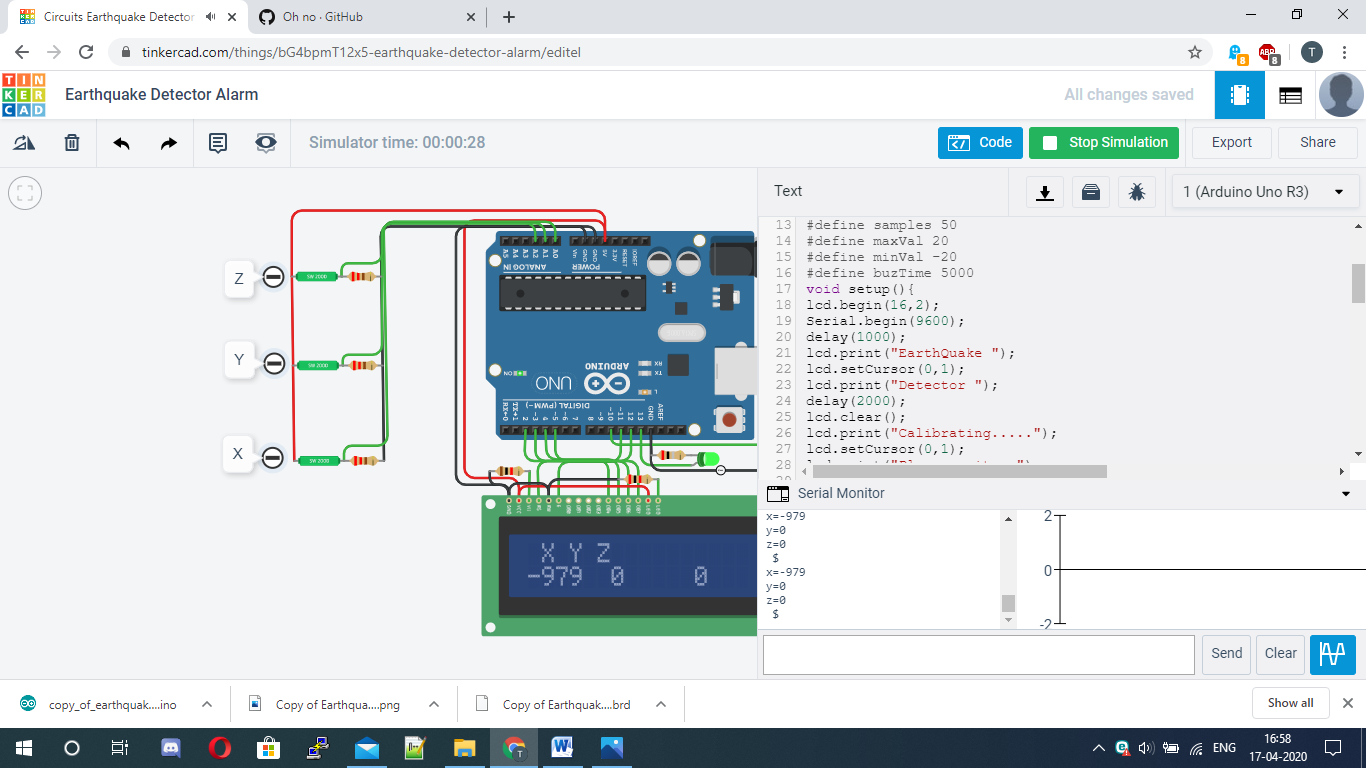
Serial.println(yValue);

Serial.print("z=");

Serial.println(zValue);

Serial.println(" $");}

**Result:**



Experiment 10

**Aim:** Aim: WAP in Tinkercad to develop a weather station on tinkercad using RGB, LDR, LM36, Buzzer and LCD.

**Apparatus:**

* LDR
* LM36
* Arduino Uno
* Buzzer

**Theory:**

**LDR**

A Light Dependent Resistor (also known as a photoresistor or **LDR**) is a device whose resistivity is a function of the incident electromagnetic radiation. Hence, they are light-sensitive devices. They are also called as photoconductors, photoconductive cells or simply photocells.



**Arduino UNO**

The Arduino Uno is a microcontroller board based on the ATmega328. It has 14 digital input/output pins (of which six can be used as PWM outputs), six analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started.

The Arduino Uno differs from all preceding boards because it does not use the FTDI USB-to-serial driver chip. Instead, it features the ATmega8U2 programmed as a USB-to-serial converter. Revision 2 of the Arduino Uno board has a resistor pulling the 8U2 HWB line to ground, making it easier to put into DFU mode.



|  |  |
| --- | --- |
| **Features** |  |
| * 14 digital I/O pins (six of which provide PWM output) * 3.3 V supply generated by an on-board regulator * Six analog input pins * 32 KB of flash memory | * Can supply 40 mA of DC current per pin * 16 MHz clock speed * Code example from Arduino website to help get started |

**Buzzer**

A **buzzer** or beeper is an audio signalling device, which may be mechanical, electromechanical, or piezoelectric (piezo for short). Typical uses of **buzzers** and beepers include alarm devices, timers, and confirmation of user input such as a mouse click or keystroke.

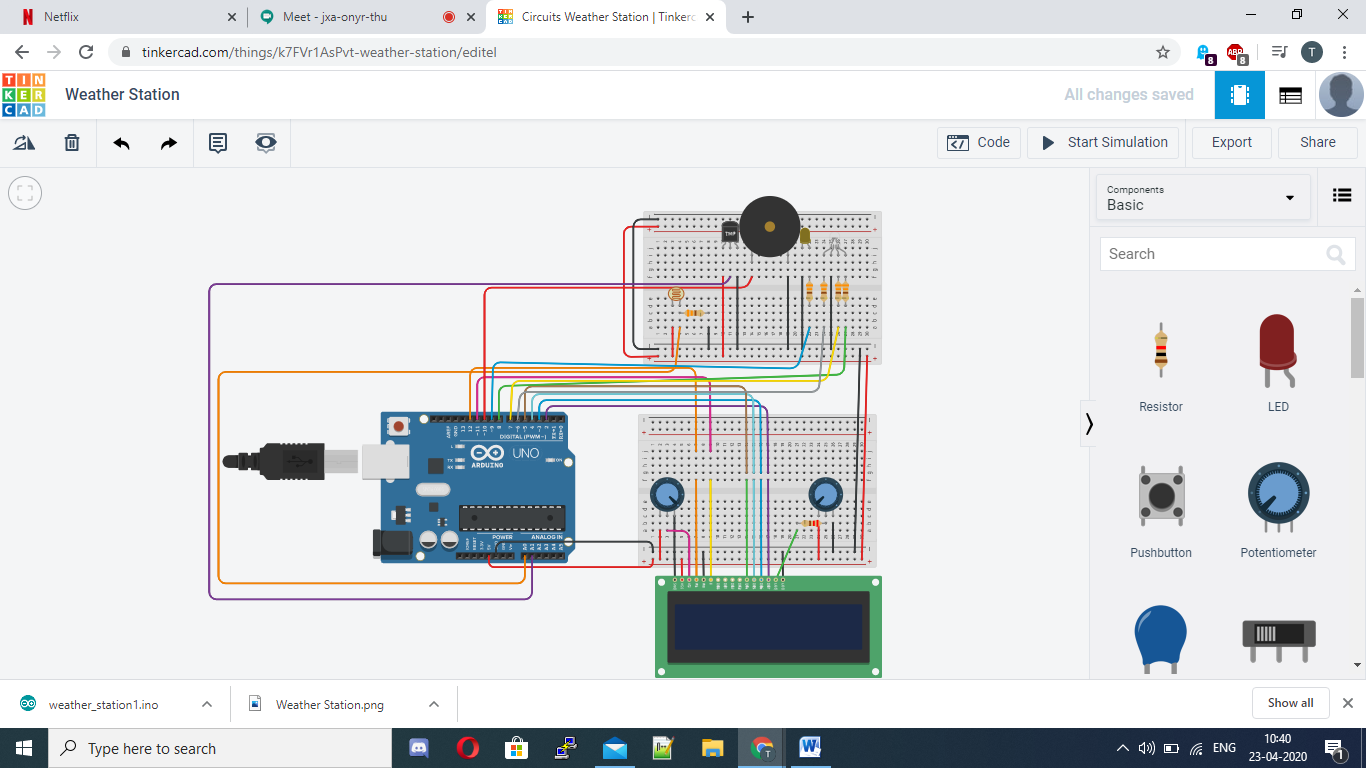


**LM36**

The **TMP36 temperature sensor** is an easy way to measure **temperature** using an **Arduino**! The **sensor** can measure a fairly wide range of **temperature** (-50°C to 125°C), is fairly precise (0.1°C resolution), and is very low cost, making it a popular choice.



**Circuit:**



**Code:**

#include <LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

int wLED = 9;

int lightLevel = 0;

int LDR = A0;

int TMP = A1;

int rLED = 6;

int bLED = 7;

int gLED = 8;

int sensorPin = A1;

void setup() {

Serial.begin(9600);

lcd.begin (16, 2);

pinMode(wLED, OUTPUT);

pinMode(rLED, OUTPUT);

pinMode(bLED, OUTPUT);

pinMode(gLED, OUTPUT);}

void loop() {

int reading = analogRead(sensorPin);

float voltage = reading \* 5.0;

voltage /= 1024.0;

Serial.print(voltage); Serial.println(" volts");

float temperatureC = (voltage - 0.5) \* 100 ;

lightLevel = analogRead(LDR);

Serial.println(lightLevel);

if (lightLevel < 10) {

digitalWrite(wLED, HIGH);

} else {

digitalWrite(wLED, LOW);}

lcd.clear();

lcd.setCursor(0, 0);

if (temperatureC < 2) {

digitalWrite(bLED, HIGH);

digitalWrite(rLED, LOW);

lcd.print("Cold Weather");

lcd.setCursor(0, 1);

lcd.print(temperatureC);

tone(10, 260);

} else if (temperatureC >= 2 && temperatureC < 45){

digitalWrite(bLED, LOW);

digitalWrite(rLED, LOW);

lcd.setCursor(0, 0);

lcd.print("Normal Weather");

noTone(10);

} else {

digitalWrite(rLED, HIGH);

digitalWrite(bLED, LOW);

lcd.setCursor(0, 0);

lcd.print("Hot Weather");

lcd.setCursor(0, 1);

lcd.print(temperatureC);

tone(10, 260);}

delay (500);}

**Result:**

