**IOT BASED ENVIRONMENTAL MONITORING SYSTEM**

**PROJECT TITLE : ENVIRONMENTAL MONITORING**

**COLLEGE NAME : ST.MOTHER THERESA ENGINEERING COLLEGE**

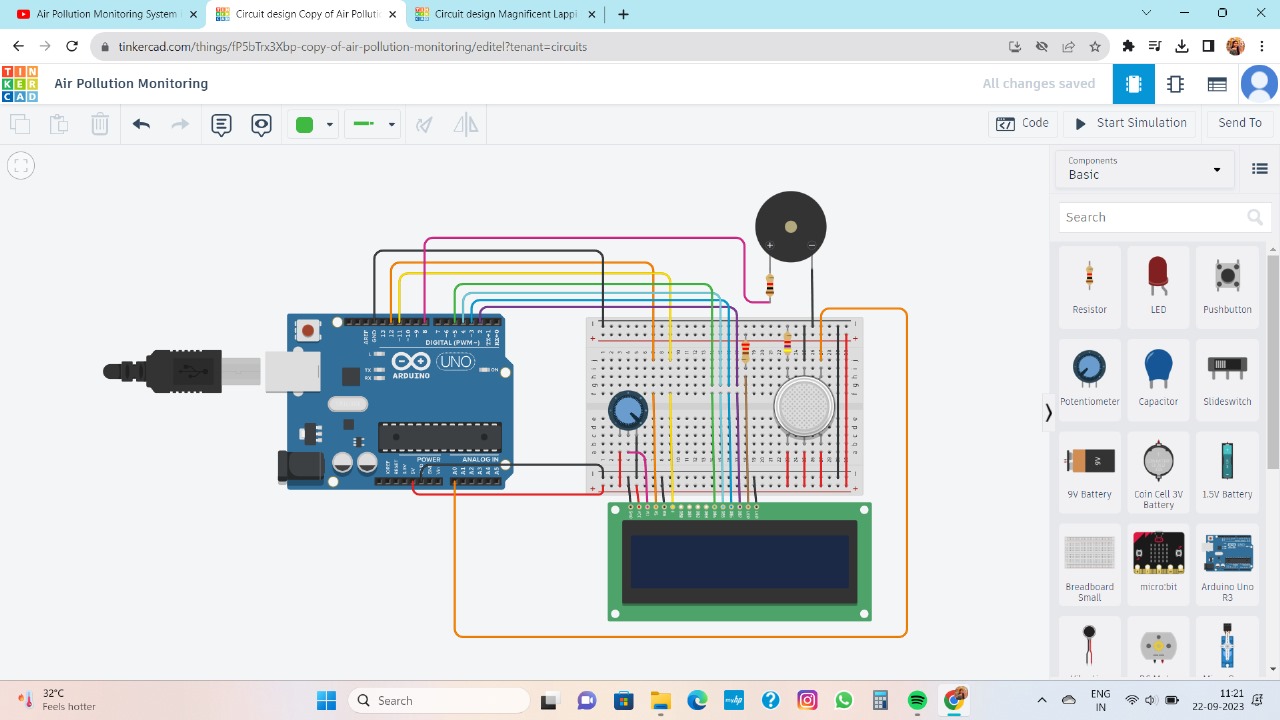
**COLLEGE CODE : 9530**

**NAME : R.SHARMILA**

**REG NO. : 953021106064**

**TEAM NAME : PROJ\_201034\_TEAM\_1**

**CIRCUIT DIAGRAM**

****

**METHODOLOGY USED WITH FLOWCHART**

**Start**

**Pollution is detected by gas sensor**

**Stop**

**When pollution comes**

**Through this circuit we can reduced pollution.**

**This circuit will monitor the amount of pollution in the air.**

**LCD displays the air is really polluted or not**.

**CODE**

#include <LiquidCrystal.h>

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

int pin8 = 8;

int analogPin = A0;

int sensorValue = 0;

void setup() {

pinMode(analogPin, INPUT);

pinMode(pin8, OUTPUT);

lcd.begin(16, 2);

lcd.print("What is the air ");

lcd.print("quality today?");

Serial.begin(9600);

lcd.display();}

void loop() {

delay(1000);

sensorValue = analogRead(analogPin);

Serial.print("Air Quality in PPM = ");

Serial.println(sensorValue);

lcd.clear();

lcd.setCursor(0,0);

lcd.print ("Air Quality: ");

lcd.print (sensorValue);

if (sensorValue<=500) {

Serial.print("Fresh Air ");

Serial.print ("\r\n");

lcd.setCursor(0,1);

lcd.print("Fresh Air");

}

else if( sensorValue>=500 && sensorValue<=650 )

{

Serial.print("Poor Air");

Serial.print ("\r\n");

lcd.setCursor(0,1);

lcd.print("Poor Air");

}

else if (sensorValue>=650 )

{

Serial.print("Very Poor Air");

Serial.print ("\r\n");

lcd.setCursor(0,1);

lcd.print("Very Poor Air");

}

if (sensorValue >650) {

digitalWrite(pin8, HIGH);

}

else

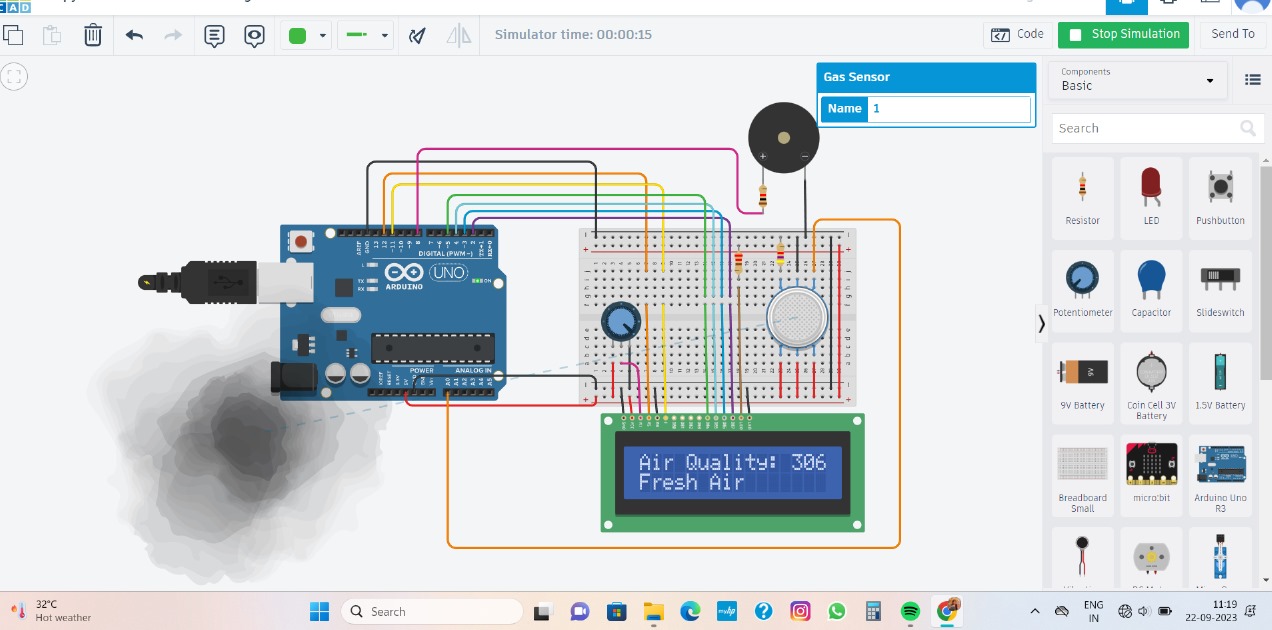
{

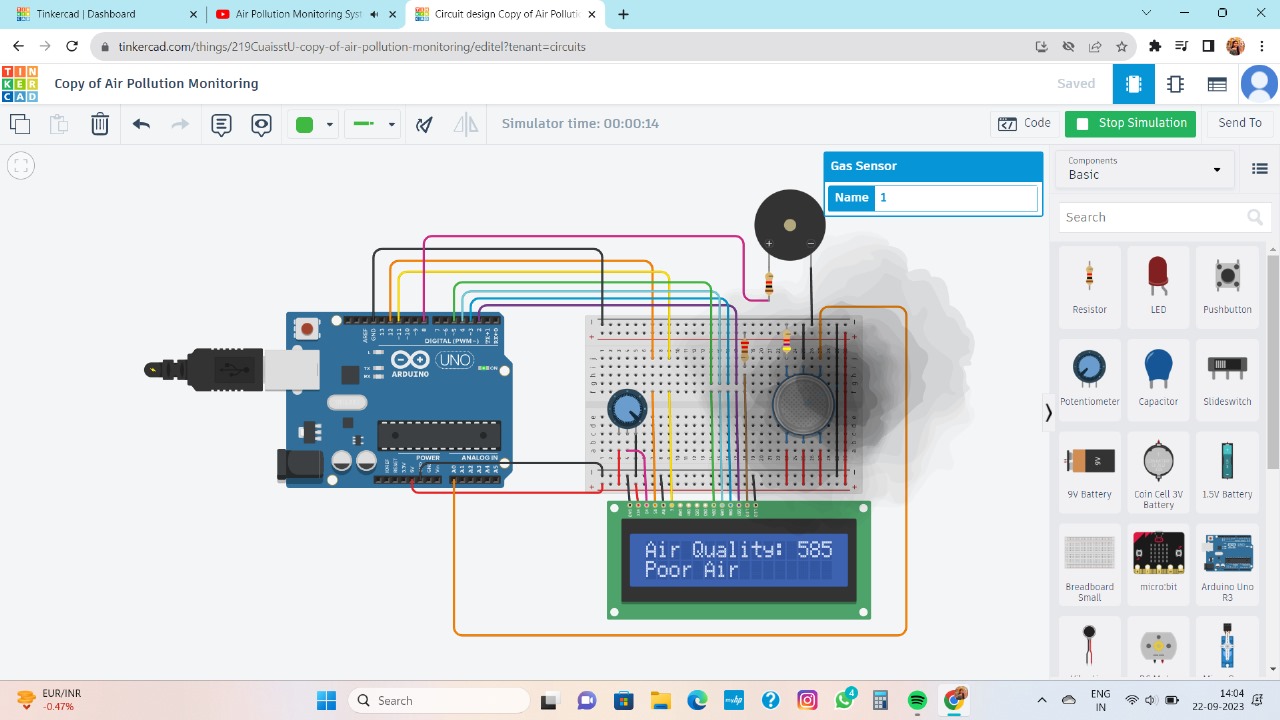
digitalWrite(pin8, LOW);

}

}

**OUTPUT**





**Python Code:**

**Import random**

**Class AirPollutionSensor:**

**Def \_init\_(self, location):**

**Self.location = location**

**Def measure\_pollution(self):**

**# Simulate measuring pollution with a random value for demonstration purposes.**

**Pollution\_level = random.uniform(0, 100)**

**Return pollution\_level**

**# Create an air pollution sensor instance at a specific location.**

**Sensor = AirPollutionSensor(“City Center”)**

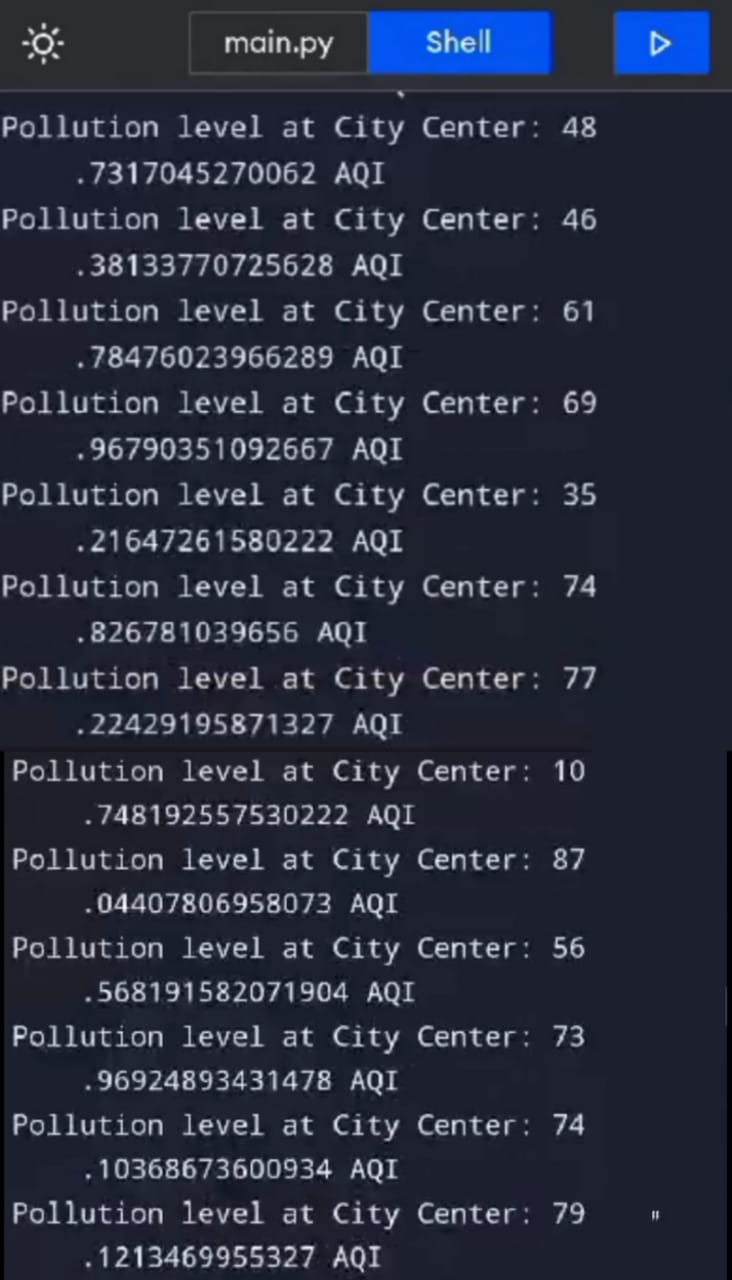
**# Continuously monitor and print pollution levels.**

**While True:**

**Pollution\_level = sensor.measure\_pollution()**

**Print(f”Pollution level at {sensor.location}: {pollution\_level} AQI”)**

**OUTPUT:**

****

**PROBLEM ANALYSIS:**

* We don’t have device which could control the air pollution .
* With the help of this device not only the municipal authorities but even the common people can participate in the process of controlling pollution
* Ensure safe environment.
* We will work on displaying the amount of poisonous gases detected by sensor.

**INNOVATION:**

* In this we found a device which control air pollution.
* This device helps people to participate in the role of controlling air pollution.
* This device also make a noise if the air quality become poor.

**PROCEDURE:**

In this project we are going to make an IoT Based Air Pollution Monitoring System in which we will monitor the Air Quality over a webserver using internet and will trigger a alarm when the air quality goes down beyond a certain level, means when there are sufficient amount of harmful gases are present in the air like CO2, smoke, alcohol, benzene and NH3. It will show the air quality in PPM on the LCD and as well as on webpage so that we can monitor it very easily.

First of all we will connect the ESP8266 with the Arduino. ESP8266 runs on 3.3V and if you will give it 5V from the Arduino then it won’t work properly and it may get damage. Connect the VCC and the CH\_PD to the 3.3V pin of Arduino. The RX pin of ESP8266 works on 3.3V and it will not communicate with the Arduino when we will connect it directly to the Arduino. So, we will have to make a voltage divider for it which will convert the 5V into 3.3V. This can be done by connecting three resistors in series like we did in the circuit. Connect the TX pin of the ESP8266 to the pin 10 of the Arduino and the RX pin of the esp8266 to the pin 9 of Arduino through the resistors.

Then we will connect the MQ135 sensor with the Arduino. Connect the VCC and the ground pin of the sensor to the 5V and ground of the Arduino and the Analog pin of sensor to the A0 of the Arduino.

Connect a buzzer to the pin 8 of the Arduino which will start to beep when the condition becomes true.In last, we will connect LCD with the Arduino. The connections of the LCD are as follows .Connect pin 1 (VEE) to the ground.Connect pin 2 (VDD or VCC) to the 5V.Connect pin 3 (V0) to the middle pin of the 10K potentiometer and connect the other two ends of the potentiometer to the VCC and the GND. The potentiometer is used to control the screen contrast of the LCD. Potentiometer of values other than 10K will work too. Connect pin 4 (RS) to the pin 12 of the Arduino. Connect pin 5 (Read/Write) to the ground of Arduino. This pin is not often used so we will connect it to the ground. Connect pin 6 € to the pin 11 of the Arduino. The RS and E pin are the control pins which are used to send data and characters.

The MQ135 sensor can sense NH3, NOx, alcohol, Benzene, smoke, CO2 and some other gases, so it is perfect gas sensor for our Air Quality Monitoring Project. When we will connect it to Arduino then it will sense the gases, and we will get the Pollution level in PPM (parts per million). MQ135 gas sensor gives the output in form of voltage levels and we need to convert it into PPM. So for converting the output in PPM, here we have used a library for MQ135 sensor, it is explained in detail in “Code Explanation” section below.

Sensor was giving us value of 90 when there was no gas near it and the safe level of air quality is 350 PPM and it should not exceed 1000 PPM. When it exceeds the limit of 1000 PPM, then it starts cause Headaches, sleepiness and stagnant, stale, stuffy air and if exceeds beyond 2000 PPM then it can cause increased heart rate and many other diseases.

When the value will be less than 1000 PPM, then the LCD and webpage will display “Fresh Air”. Whenever the value will increase 1000 PPM, then the buzzer will start beeping and the LCD and webpage will display “Poor Air, Open Windows”. If it will increase 2000 then the buzzer will keep beeping and the LCD and webpage will display “Danger! Move to fresh Air”.

**COMPONENTS:**

* Arduino UNO
* Gas sensor
* LCD display
* Piezo sensor
* Bread board
* Resistor
* Potentiometer

**SOFTWARE:**

**ARDIUNO**

Arduino is an open source electronic platform based on easy to use hardware and software. Arduino board are able to read inputs light on a sensor a finger on a button, or a Twitter messages and turn it into an output activating a sensor , turning on an buzzer, publishing something on LCD.