**Air quality monitoring**

**Replication of project:**

As our project is based on IOT, let us throw some light on the topic of IOT itself. The Internet of Things (IOT) is a system of interrelated computing devices, mechanical and digital machines, objects, animals, or people that are provided with unique identifiers (UIDs) and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction. 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 over a 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.

A growing portion of IOT devices are created for consumer use, including connected vehicles, home automation, wearable technology, connected health, and appliances with remote monitoring capabilities. In the consumer market, IOT technology is most synonymous with products pertaining to the concept of the “smart home”. Therefore, using IOT, the aim of this project is to monitor the air quality level in the surrounding of our device using Arduino and MQ135 gas sensor and show the results in (Parts Per Million) PPM units.

The aim of this paper is to present an Internet of Things or IOT-based system which can monitor the air quality and send notiﬁcations in case of emergency. The main goal of the whole system is to provide a way to prevent emergencies inside the compartments containing this system. This system monitors the following Properties of air: temperature, humidity, the level of carbon dioxide, gas and alcohol. After calibration, it will be able to provide notiﬁcations via SMS if the values of the variables mentioned above exceed a certain threshold. The main protocols are described below…..

The availability of real-time air quality data could make drivers better educated about driving patterns and how it impacts the environment and increases pollution. Better driving habits will lead to reduced pollution. Also, more health-conscious citizens may choose alternate “healthy” routes based on pollution information. It will benefit them as well as others by reducing pollution concentration in peak roadways so everybody breathes cleaner air.

**IOT Device set-up:**

*Required components:*

* MQ135 Gas sensor.
* Arduino Uno.
* Wi-Fi module ESP8266
* 16X2 LCD.
* Breadboard.
* 10K potentiometer.
* 1K ohm resistors.
* 220 ohm resistor.
* (Piezo)Buzzer.

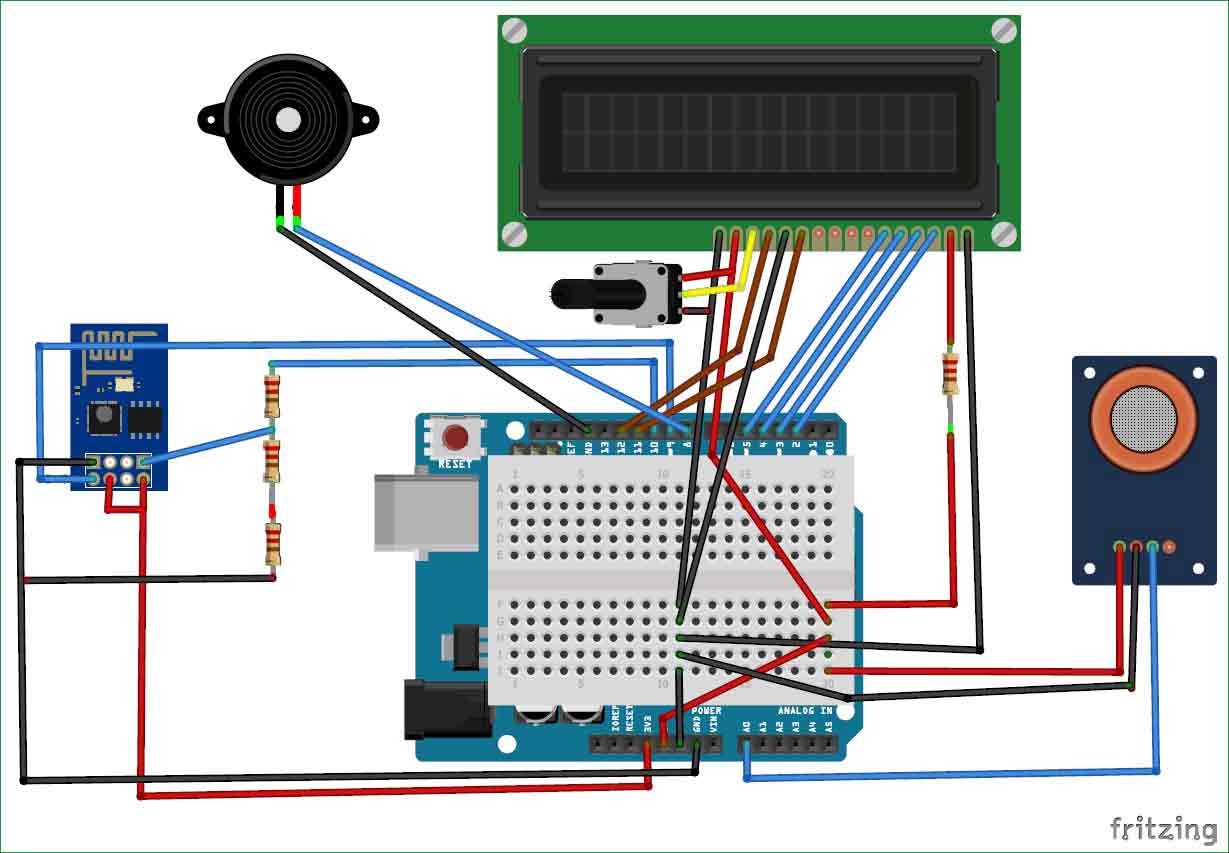
*Circuit Diagram Explanation:*

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.

ESP8266 Wi-Fi module gives your projects access to Wi-Fi or internet. It is a very cheap device and make your projects very powerful. It can communicate with any microcontroller and it is the most leading devices in the Iot Project.

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 to 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 following four pins are data pins which are used to communicate with the Arduino.
* Connect pin 11 (D4) to pin 5 of Arduino.
* Connect pin 12 (D5) to pin 4 of Arduino.
* Connect pin 13 (D6) to pin 3 of Arduino.
* Connect pin 14 (D7) to pin 2 of Arduino.
* Connect pin 15 to the VCC through the 220 ohm resistor. The resistor will be used to set the back light brightness. Larger values will make the back light much more darker.
* Connect pin 16 to the Ground.



**Working explanation:**

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 500 PPM, then the LCD and webpage will display “Fresh Air”. Whenever the value will increase 500 PPM, then the buzzer will start beeping and the LCD and webpage will display “Very Poor Air”. If it will increase 1000 then the buzzer will keep beeping and the LCD and webpage will display “Danger! Move to fresh Air”.

**Program:**

Import numpy as np

From sklearn.model\_selection import train\_test\_split

From sklearn.linear\_model import LinearRegression

From sklearn.metrics import mean\_absolute\_error, mean\_squared\_error, r2\_score

# Generate synthetic air quality data

Np.random.seed(0)

Num\_samples = 1000

Temperature = np.random.uniform(10, 30, num\_samples)

Humidity = np.random.uniform(30, 70, num\_samples)

Time\_of\_day = np.random.uniform(0, 24, num\_samples)

Previous\_pm25 = np.random.uniform(0, 50, num\_samples)

Previous\_pm10 = np.random.uniform(0, 60, num\_samples)

Pm25 = 20 + 1.5 \* temperature – 0.5 \* humidity + 0.1 \* time\_of\_day + 0.2 \* previous\_pm25

Pm10 = 15 + 1.0 \* temperature – 0.3 \* humidity + 0.05 \* time\_of\_day + 0.15 \* previous\_pm10

# Step 3: Data Splitting

X = np.column\_stack((temperature, humidity, time\_of\_day, previous\_pm25, previous\_pm10))

Y = pm25 # Predicting PM2.5, you can change this to predict other metrics

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.15, random\_state=42)

# Step 5: Model Selection

Model = LinearRegression()

# Step 6: Model Training

Model.fit(X\_train, y\_train)

# Step 7: Model Evaluation

Y\_pred = model.predict(X\_test)

Mae = mean\_absolute\_error(y\_test, y\_pred)

Mse = mean\_squared\_error(y\_test, y\_pred)

R2 = r2\_score(y\_test, y\_pred)

Print(“Model Evaluation:”)

Print(f”Mean Absolute Error: {mae}”)

Print(f”Mean Squared Error: {mse}”)

Print(f”R-squared (R²): {r2}")

**Output:**

Model Evaluation:

Mean Absolute Error: 3.434289889507151e

Mean Absolute Error:15

Mean Squared Error: 2.553805703835489e

Mean Squared Error:25

R-squared (R²): 1.0

By using this entire python coding will integrate in the Arduino board, the presentation will takes place.

**Project design:**

<https://www.tinkercad.com/things/9qJI4XguMtx>

First of all, this project is describes how to interface Arduino with gas sensor. So, login to your Tinker cad account and go to designs, then circuits click on new circuits, after that the black page will appear in that page to select the components which is going to takes place. Here, select Arduino UNO, then bread board, LED, gas sensor, Resistor, potentiometer, piezo and LCD 16×2.

The arduino board which is a microcontroller that controls everything basically and the bread board through which all the connections are made for external which is basically used for external connections and the LCD, which displays the amount of smoke and here, the LED light which describes the commander of sensor and there is a gas sensor that sense the amount of gas and the potentiometer which is used to make contrast to the LED. And as we can see we have made the particular connections.

This design represents the simulation of Air detecting sensor. When the gas is sensed through the gas sensor with the help of Arduino the piezo indicates through it’s buzzer alarm And it intimates to LCD board as “Harmful gas is detected”.

The design of simulation represents the air quality that simulates and describes the quality using the gas sensor with the help of Arduino Uno In the LCD board as “FRESH AIR” When the air quality is below 500.

The design of simulation represents the air quality that simulates and describes the quality using the gas sensor with the help of Arduino uno In the LCD board as “VERY POOR AIR ” When the air quality is Above 500.