# PROJECT TITLE: AIR QUALITY MONITORING

## Phase 3: Development Part 1

### INTRODUCTION:

The air pollution is a major concern for every individual in the society. As the air pollution increases, human faces many problems and health issues. The air pollution is also harmful to living things. The factors which are majorly responsible for air pollution are use of vehicles and industrialization. Vehicles release lots of gaseous pollutants when we use them. These pollutants increase air pollution and these are harmful for human beings.

Industries leads to produce harmful gases which affects air pollution which is directly affecting the human health. These harmful gaseous pollutants cause allergic reactions to humans, throat and nose infection, allergies leads to skin diseases. As the title suggests, it is a monitoring system which monitors air pollution with the help of **raspberry pi** based on Iot. This project deals with monitoring of air pollution as well as air quality. It helps to every individual in the society to determine the air pollution in their surroundings.

### REQUIRED COMPONENTS:

- Raspberry Pi 3
- Air Quality Sensor
- Mic Sensor
- Wifi Module
- LCD Display
- Resistors
- Capacitors
- Transistors
- Cables and Connectors
- Diodes
- PCB and Breadboards
- LED
- Transformer/Adapter

- Push Buttons
- Switch
- IC
- IC Sockets

## IOT DEVICES:

1) **MQ 2 Sensor**: It is a robust gas sensor. It is used to detect gases like LPG, alcohol, propane, hydrogen, methane and carbon monoxide. It provides analog as well as digital output so it can be used as analog and digital sensor. The working range of this gas sensor is 200 to 10000 PPM.
2) **MQ 7 Sensor** : It is a highly sensitive towards carbon monoxide so it is also known as carbon monoxide sensor. It provides fast response time. It is having low cost. It is stable and long-life sensor. The working range of this gas sensor is 20 to 2000 PPM.
3) **Raspberry Pi3**: It's like a small independent computer. It is one of the smallest computers popular for its size. It can be used for programming for electronic projects, to play HD videos. We can use languages like c, c++, python, java, ruby for programming. It has its own operating system as Raspbian
4) **Arduino UNO**:- Arduino Uno is a microcontroller board based on the ATmega328P It has 14 digital input/output pins 6 analog inputs, a 16 MHz quartz crystal, a USB Connection, power jack, an ICSP header and a reset button
5) **MQ135 sensor**:- The MQ135 sensor can sense NH3, NOx, alcohol, Benzene, smoke, CO2 and some other gases. It gives the output in form of voltage levels. Fig.3 shows the sensor MQ135.

**6) WIFI Module (ESP8266)**:- The ESP8266 is a low-cost Wi-Fi chip with full TCP/IP stack and MCU (microcontroller unit) capability. It runs on 3.3V and gives our system access to Wi-Fi or internet. Fig.4 shows Wi-Fi Module (ESP8266)**.**

**7) Buzzer**:- A Buzzer or beeper is an audio signaling device. Whenever the air pollution goes above the threshold level the Buzzer starts beeping indicating Danger. Fig.5 shows Buzzer.

**8) LCD (Liquid Crystal Display):-** This is a basic (16x2) 16 character by 2 line display. Black text on Green background. It is used to indicate the Air and Humidity in PPM. Fig. 6 shows LCD (16x2).

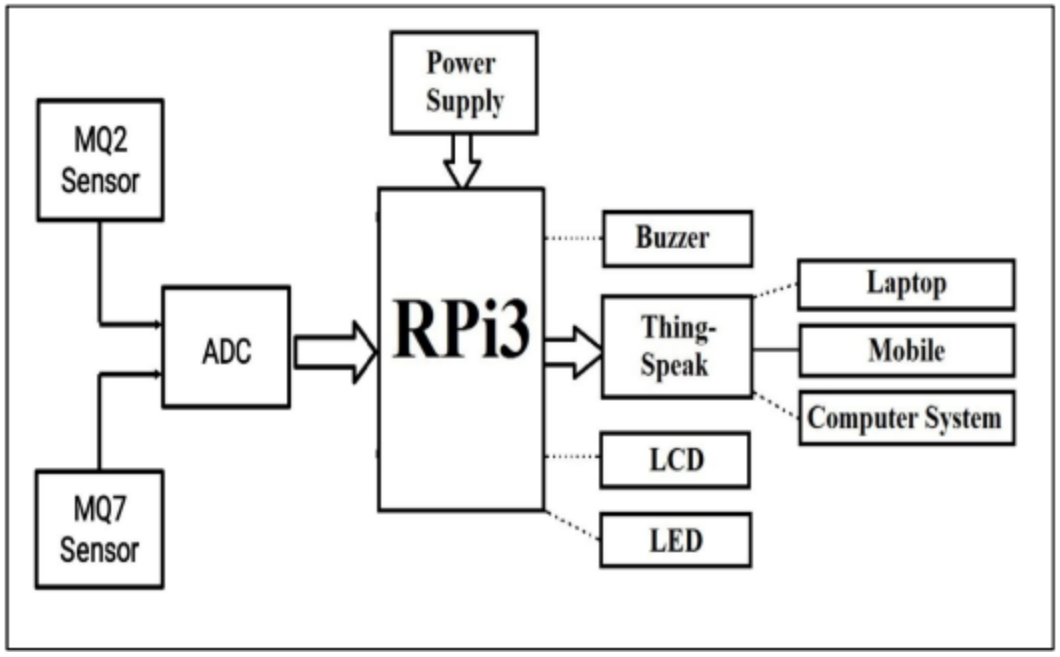
**9) LPG Sensor**:- MQ-6 sensor is a simple-to-use liquefied petroleum gas (LPG) sensor, suitable for sensing LPG (composed of mostly propane and butane) concentrations in the air. The MQ-6 can detect gas concentrations anywhere from 200 to 10000ppm. Fig.7 shows LPG sensor (MQ6).

## BLOCK DIAGRAM AND WORKING:-

**Working**:-

The above block diagram shows the working of IOT Based Air Pollution Monitoring System Using Raspberry Pi. The MQ2 and MQ7 gas sensors are used in this project. These sensors are used as an analog sensor which are connected to adc. The adc is further connected to raspberry pi. Raspberry pi is provided with a power supply. The output of raspberry pi is sent to the thing speak IoT platform for monitoring graphically. We can use thing speak in mobile, laptop as well as in computer systems through an internet application that can operate from anywhere.

Arduino Uno is a low-cost microcontroller board based on ATMEGA-328P which can be easily interfaced with Raspberry pi and has a very effective ADC. Since Raspberry pi 2 model B does not have built in Wi-Fi adapter therefore Wi-Fi adapter is used for providing the internet to the complete system. The light weight protocol MQTT (Message Queuing Telemetry transport). MQTT plays an important role in establishing communication between the sensors and the clients. The client can access the data that is being displayed on the dashboard by using the device id but the client will be not able to do any modification to the data received.




## CODE EXPLANATION:

Before beginning the coding for this project, we need to first Calibrate the MQ7 Gas sensor. There are lots of calculations involved in converting the output of sensor into PPM value, we have done this calculation.

Using this library you can directly get the PPM values, by just using the below two lines:

```
gasSensor = MQ135(A0)
air_quality = gasSensor.getPPM()
```

In the code, first of all we have defined the libraries and the variables for the Gas sensor and the LCD. By using the Software Serial Library, we can make any digital pin as TX and RX pin. In this code, we have made Pin 9 as the RX pin and the pin 10 as the TX pin for the ESP8266. Then we have included the library for the LCD and have defined the pins for the same. We have also defined two more variables: one for the sensor analog pin and other for storing *air_quality* value.

```
import serial
DEBUG = True
esp8266 = serial.Serial(9, 10)
import RPi.GPIO as GPIO
from RPLCD import CharLCD
lcd = CharLCD(cols=16, rows=2, pin_rs=12, pin_e=11, pins_data=[5,
    4, 3, 2])
sensorPin = 0
air_quality = 0
```

Then we will declare the pin 8 as the output pin where we have connected the buzzer. l*cd.begin(16,2)* command will start the LCD to receive data and then we will set the cursor to first line and will print the *'circuitdigest'*. Then we will set the cursor on the second line and will print *'Sensor Warming'*.

Then we will set the baud rate for the serial communication. Different ESP's have different baud rates so write it according to your ESP's baud rate. Then we will lsend the commands to set the ESP to communicate with the Arduino and show the IP address on the serial monitor.

The following code will call a function named *sendData* and will send the data & message strings to the webpage to show.

```
lcd.setCursor(0, 0)
lcd.print("Air Quality is ")
lcd.print(air_quality)
lcd.print(" PPM ")
lcd.setCursor(0, 1)
if air_quality <= 1000:
    lcd.print("Fresh Air")
    digitalWrite(8, LOW)
```

The following code will print the data on the LCD. We have applied various conditions for checking air quality, and LCD will print the messages according to conditions and buzzer will also beep if the pollution goes beyond 1000 PPM

```
def sendData(command, timeout, debug):
    response = ""
    esp8266.print(command)
    time = millis()
    while (time + timeout) > millis():
        while esp8266.available():
            c = esp8266.read()
            response += c
    if debug:
        Serial.print(response)
    return response
```

Monitoring air quality using a Raspberry Pi typically involves the use of sensors to measure various air quality parameters such as particulate matter (PM2.5 and PM10), carbon dioxide (CO2), temperature, humidity, and more. Here's a basic Python code example to get you started with air quality monitoring using a Raspberry Pi and a sensor like the SDS011 for PM2.5 and PM10 measurements:

```
import time
import serial

def read_sensor_data(ser):
    data = ser.read(10)
    if data[0] == 170 and data[1] == 192:
        pm25 = int.from_bytes(data[2:4], byteorder='little') / 10.0
        pm10 = int.from_bytes(data[4:6], byteorder='little') / 10.0
        return pm25, pm10
    else:
        return None, None

def main():
    ser = serial.Serial('/dev/ttyUSB0', baudrate=9600, timeout=2)

    try:
        while True:
            pm25, pm10 = read_sensor_data(ser)
            if pm25 is not None and pm10 is not None:
                print(f'PM2.5: {pm25} µg/m³, PM10: {pm10} µg/m³')
```

```
            else:
                print('Error reading sensor data')

            time.sleep(60) # Read data every 60 seconds
    except KeyboardInterrupt:
        ser.close()

if __name__ == '__main__':
    main()
```

In this example, we use the serial library to communicate with the SDS011 sensor over a USB connection. You might need to adjust the serial port ('/dev/ttyUSB0' in this example) and baud rate to match your setup. Additionally, you may need to install the pyserial library if it's not already installed on your Raspberry Pi.Remember that this code is a simple starting point. Depending on the specific sensors and air quality parameters you want to monitor, you may need to incorporate additional libraries and code for things like CO2, temperature, humidity, and data storage. Also, consider handling error conditions and implementing a more robust data collection and storage system for long-term monitoring.

## CONCLUSION:

The system to monitor the air of environment using Arduino microcontroller, IOT Technology is proposed to improve quality of air. With the use of IOT technology enhances the process of monitoring various aspects of environment such as air quality monitoring issue proposed in this paper. Here, using the MQ135 and MQ6 gas sensor gives the sense of different type of dangerous gas and arduino is the heart of this project.