AIR QUALITY MONITORING

PHASE 3

Air quality monitoring through the Internet of Things (IoT) is a pioneering advancement that relies on an extensive network of interconnected sensors and cutting-edge technology. The Gas sensor is strategically positioned in various locations to consistently gather data on air pollutants such as particulate matter (PM2.5, PM10), gases (e.g., nitrogen dioxide, sulfur dioxide), and environmental parameters.

Development Stage:

IoT Air Quality Monitoring, featuring the Arduino Uno board and a gas sensor. This project entails a closer look at the hardware components, their connections, and the unique role each device plays in ensuring clean and healthy air.

Components:

We have the Arduino Uno board, which serves as the central hub for data collection and transmission. Additionally, we have the gas sensor, a critical component that measures various air pollutants, including carbon monoxide (CO), carbon dioxide (CO2), and volatile organic compounds (VOCs). Our project's brain is the Arduino Uno board. This microcontroller takes on the responsibility of processing data and transmitting it. To set it up, simply connect the board to your computer via USB for both programming and power.Gas Sensor: Now, the gas sensor—this is the core of our air quality measurement. It's designed to

detect a range of gases, including CO and VOCs. It's also proficient in calculating gas concentrations in parts per million (ppm).

Connections:

The gas sensor is linked to the Arduino Uno via the analog input. This ensures the proper exchange of data. You'll also need to provide the necessary power and ground connections to the sensor. Finally, the sensor's data output is directed to a digital pin on the Arduino.

Gas Sensor Details:

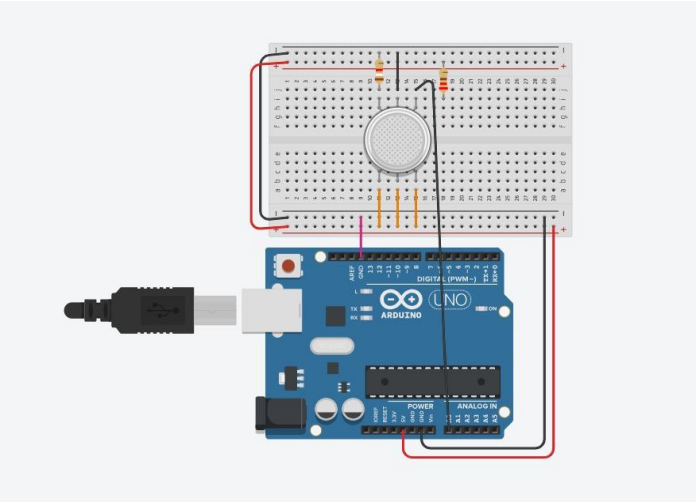
Now, a closer look at the gas sensor. Inside, you'll find two key components: a heating element and a semiconductor sensor. The heating element ensures the sensor's stable operation, while the semiconductor is the part that detects gas concentrations.

Measuring PPM:

One of the most critical functions of the gas sensor is measuring gas concentrations in parts per million (ppm). For example, it can tell you precisely how many parts per million of carbon monoxide are present in the air you're monitoring.

Conclusion:

In conclusion, our IoT Air Quality Monitoring Hardware, powered by the Arduino Uno and a gas sensor, provides a robust solution for measuring and monitoring air quality. Understanding each component and their connections empowers you to create your very own air quality monitoring system, contributing to a healthier and cleaner environment.



***Python Script* :**

Python script to read and process data from the Arduino Uno .

```
    arduino = serial.Serial(arduino_port, 9600, timeout=2)
except Exception as e:
    print(f"Error: {e}")
    exit()

def air_quality_status(air_quality):
    if air_quality == 1:
        return "Good Air Quality"
    elif air_quality == 2:
        return "Moderate Air Quality"
    elif air_quality == 3:
        return "Poor Air Quality"
    elif air_quality == 4:
        return "Very Poor Air Quality"
    else:
```

```
        return "Unknown"

try:
    while True:
        try:
            data = arduino.readline().strip().decode('utf-8')
            if data:
                data = data.split(',')
                temperature, humidity, gas_sensor_value, air_quality =
map(float, data)

                print(f"Temperature: {temperature} °C")
                print(f"Humidity: {humidity} %")
                print(f"Gas Sensor Value: {gas_sensor_value}")
                print(f"Air Quality: {air_quality_status(int(air_quality))}")
                print()

            time.sleep(2)  # Adjust this delay as needed

        except KeyboardInterrupt:
            arduino.close()
            break
        except Exception as e:
            print(f"Error: {e}")
```

This Python script is designed to communicate with Arduino board through serial connection and retrieve important information related to air quality parameters. It starts with making a connection to the Arduino and includes the ability to convert air quality values into descriptive status labels such as "good air quality" or "very poor air quality".

The script runs in an infinite loop, continuously retrieving and displaying data from the Arduino. During each iteration, it receives a set of data sent by the Arduino. This includes temperature, humidity, gas sensor readings and air quality values.

The data is parsed and the script outputs temperature (in Celsius), humidity (in percent), gas sensor values, and air quality status. The latter is determined based on predefined air quality ranges.

To avoid data and read overload, the script introduces a 2 second delay between each read cycle. Additionally, it includes error handling to provide graceful termination in response to user interruption or unexpected errors. When the script is complete, the serial communication with the Arduino is closed.