Raspberry Pi-based AQI Monitoring System Project

iTechBuilders

■ Introduction

Air pollution is one of the most pressing global issues today. An Air Quality Index (AQI) monitoring system provides real-time data on pollutants, helping communities and individuals take necessary precautions. This project uses a Raspberry Pi as the central controller and an Arduino Nano for sensor interfacing, making it ideal for IoT-based smart city and educational applications.

■■ System Overview

The system consists of an Arduino Nano connected to air quality and environmental sensors (MQ135, DHT11, PMS5003). The Arduino collects sensor readings and sends data via serial communication to the Raspberry Pi. The Raspberry Pi processes the data, stores it, and uploads it to a cloud platform such as ThingSpeak or Blynk.

■ Components Required

Component	Description
Raspberry Pi 4	Central controller with Wi-Fi for cloud upload
Arduino Nano	Microcontroller for sensor interfacing
MQ135 Sensor	Air quality (CO2, NH3, NOx) detection
DHT11 Sensor	Temperature & humidity monitoring
PMS5003	Particulate Matter (PM2.5, PM10) sensor
Jumper Wires & Resistors	Connections
Breadboard	Prototyping

■ Pin Connections

Sensor	Arduino Nano Pin
MQ135	A0
DHT11	D2
PMS5003	TX → D3, RX → D4

■ Arduino Nano Code Example

```
#include <DHT.h>
#define DHTPIN 2
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);
```

```
int mq135 = A0;

void setup() {
  Serial.begin(9600);
  dht.begin();
}

void loop() {
  float t = dht.readTemperature();
  float h = dht.readHumidity();
  int airQuality = analogRead(mq135);

  Serial.print("Temperature: "); Serial.print(t);
  Serial.print(" C | Humidity: "); Serial.print(h);
  Serial.print(" % | Air Quality: "); Serial.println(airQuality);

  delay(2000);
}
```

■ Raspberry Pi Integration (Python)

```
import serial, time
import requests

ser = serial.Serial('/dev/ttyUSB0', 9600)
API_KEY = "YOUR_THINGSPEAK_KEY"
URL = "https://api.thingspeak.com/update"

while True:
    data = ser.readline().decode().strip()
    print("Data:", data)
    try:
        params = {"api_key": API_KEY, "field1": data}
        requests.get(URL, params=params)
    except:
        print("Upload failed")
    time.sleep(15)
```

■ Learning Outcomes

- Learn interfacing of air quality and environmental sensors - Understand Arduino Nano and Raspberry Pi integration - Experience serial communication between microcontrollers - Upload and visualize IoT data on cloud platforms - Develop an end-to-end IoT environmental monitoring project

■ Applications & Future Improvements

- Home and school air quality monitoring - Smart city and environmental projects - Integration with mobile apps for alerts - Adding machine learning for pollution prediction - Expanding with GPS for location-based AQI mapping

■ Project Guide by iTechBuilders