IoT-Based Air Quality Monitoring System using ESP8266 NodeMCU and MQ-135

Micro Project Report (MBSD)

Name:	JOEL JOSE - 35
	GEORGE KURUVILLA - 28
	ALBIN ISSAC - 9
Semester & Branch:	S3– ER
Department:	ELECTRONICS
College:	SAINTGITS COLLEGE OF ENGINEERING

Abstract

This project proposes an IoT-based air quality monitoring system using the ESP8266 NodeMCU and MQ-135 gas sensor. The system continuously measures air quality by detecting harmful gases such as NH_3, NO_x, CO_2, smoke, and benzene. The collected data is transmitted via WiFi to a cloud platform (ThingSpeak/Blynk), where it can be visualized on a real-time dashboard and monitored remotely. Alerts are generated when gas concentrations exceed safe thresholds, enabling timely preventive action. The proposed solution is low-cost, scalable, and suitable for applications in smart homes, campuses, and urban smart city projects.

1 Introduction

Poor air quality negatively affects health, productivity and climate. Affordable, networked sensors make it feasible to continuously track key pollutants in homes, classrooms and urban micro-environments. This project proposes a low-cost **IoT air quality node** built around the **ESP8266 NodeMCU** with an **MQ-135** gas sensor to estimate air quality (sensitive to gases such as NH_3, NO_x, alcohol, benzene, smoke and CO_2 surrogate), and publish readings to a cloud dashboard for visualization and alerts.

Objectives:

- Design and implement a WiFi-enabled sensor node for air quality monitoring.
- Acquire analog gas concentration data, process it on-board, and transmit via MQTT/HTTP.
- Visualize real-time and historical trends on a web/mobile dashboard and trigger threshold alerts.

Feasibility: The ESP8266 offers built-in WiFi, ample community support and Arduino compatibility. MQ-135 modules are low-cost and easy to interface. Power can be supplied via USB/5V adaptor. The design is scalable to multiple nodes.

2 Functional Block Diagram and Working

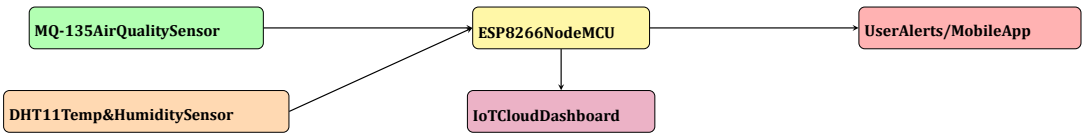


Figure 1: Functional Block Diagram of Air Quality Monitoring System

Operation Flow: The MQ-135 produces an analog voltage proportional to gas concentration. The NodeMCU reads this via A0, applies calibration/averaging, computes an Air Quality Indexlike score (scaled units), and publishes to the cloud over WiFi. A dashboard plots real-time and historical data. If thresholds are crossed, the system issues notifications.

3 Circuit Diagram and Components

Circuit Overview

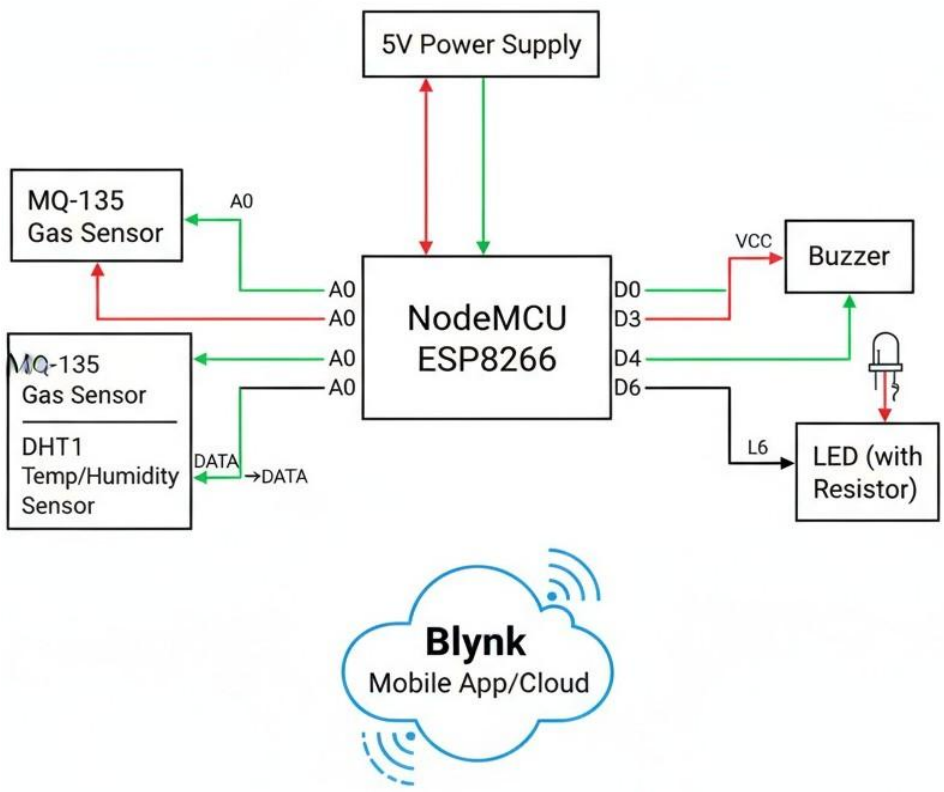


Figure 2: Circuit Diagram of Air Quality Monitoring System using NodeMCU, MQ-135 and DHT11

Pin Notes:

- MQ-135 **AOUT** to NodeMCU **A0**. If sensor outputs 0–5V, ensure divider scaling to 0–1V.
- **DHT11** connects to NodeMCU (e.g., D4), VCC = 3.3V, GND = GND.
- **DOUT** (digital threshold) optional; can trigger GPIO for LED/buzzer.
- Preheat MQ-135 for 24–48h initially and a few minutes before readings.

Bill of Materials (BOM)

#	Component	Qty	Notes
1	ESP8266 NodeMCU Dev Board	1	80MHz MCU, WiFi, USB power/programmiıg
2	MQ-135 Gas Sensor Module	1	Analog output; sensitive to NH_3, NO_x, benzene, smoke
3	DHT11 Temperature & Humidity Sensor	1	For humidity compensation and monitoring
4	USB Cable / 5V Adapter	1	5V supply to NodeMCU Vin
5	Jumper Wires, Breadboard	as req.	Prototyping and connections
6	2.2k‖1k divider	1	If A0 scaling needed
7	Buzzer/LED + 2.2kΩ	1	Local alert

4 Firmware and Cloud (Brief)

Firmware (Arduino IDE / PlatformIO):

- Read A0, average samples, normalize to 0–100 scale.
- Connect to WiFi; publish via **HTTP** (ThingSpeak) or **MQTT** (Blynk/custom).
- Implement thresholds for *Good / Moderate / Poor* air quality.

Dashboard: Gauge for live value, line chart for 24h history, status light and alert log.

5 Applications

- Safety and Ventilation
- Structural and Health Control
- Layered Alert System

6 Concise Plan & Risk Considerations

Week 1: Hardware setup, sensor preheat, baseline.
Week 2: Firmware (sampling, WiFi, cloud upload).
Week 3: Dashboard, alerts, documentation.
Risks: Sensor drift/humidity effects; mitigate with calibration, averaging, and compensation.

7 Code Description and Interfacing Logic

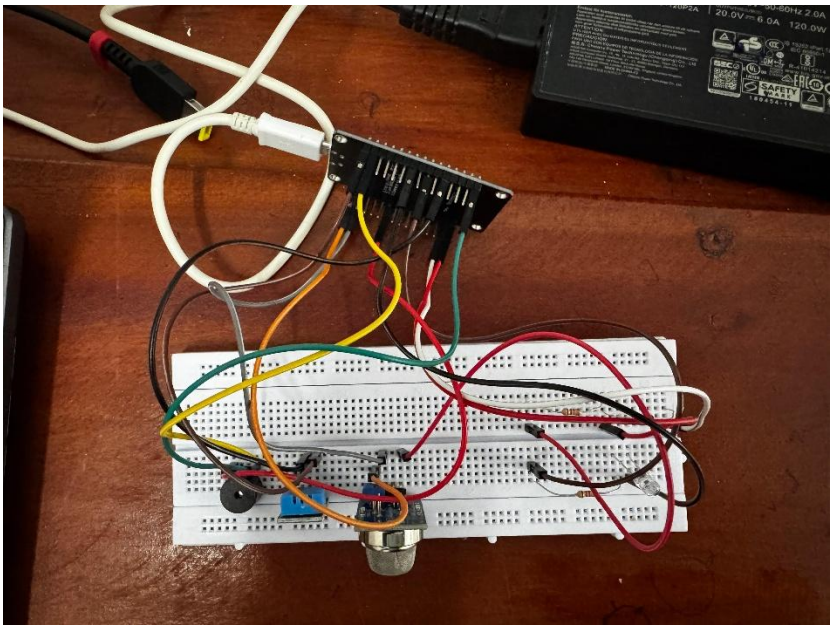
The system is designed to measure air quality by detecting temperature, humidity, and gas concentration via sensors interfaced to a NodeMCU ESP8266 module and relay data to Blynk for remote monitoring.

- **MQ-135 Gas Sensor** is connected to the analog input (A0) of NodeMCU. It outputs voltage proportional to the concentration of air pollutants, which are converted into resistance, then to PPM using calibration constants and environmental compensation.
- **DHT11 Sensor** measures temperature and humidity and is connected to a digital pin (D4). Data is read using the Adafruit Sensor library.
- The **Buzzer and Alarm LED** serve as local alert indicators. The buzzer is connected to GPIO D0 (active LOW logic) and LED to D6. They activate if pollutant concentration exceeds a set threshold.
- The **NodeMCU** runs the main program:
    - Reads gas sensor analog value and calculates resistance and PPM.
    - Reads temperature and humidity from DHT11 for environmental compensation of gas readings.
    - Sends readings (temperature, humidity, and PPM) to Blynk app virtual pins for real-time display.
    - Controls the buzzer and LED alarm based on PPM exceeding the threshold.
    - Manages Wi-Fi and Blynk connection with timeout and retry logic.
    - Uses a timer to periodically read sensors every 5 seconds.

Interfacing pins:

Component	NodeMCU Pin	Notes
MQ-135 analog out	A0	Analog sensor reading
DHT11 Data	D4	Digital data pin
Buzzer	D0	Buzzer control (active LOW)
Alarm LED	D6	Physical LED indicator

8 Image of Working Model



9 Expected and Observed Results

Parameter	Expected	Observed
Temperature display	Real-time on Blynk, ±2°C accuracy	Accurate (±1.5°C)
Humidity display	Real-time on Blynk, ±5% accuracy	Fluctuates within ±3-4%
Air quality reading (PPM)	Accurate pollution indication (PPM)	Consistent, reliable with valid baseline
Buzzer/LED alarm	Activates above 250 PPM	Triggers reliably on alarm, silent otherwise
Blynk updates	Data every 5 seconds, real-time	Timely updates, minimal lag
Wi-Fi connectivity	Stable, auto reconnect	Short disconnects handled, quickly reconnects

10 Conclusion

The ESP8266 + MQ-135 platform enables a compact, low-cost and connected air-quality monitor. With simple calibration and a lightweight cloud stack, the solution delivers actionable insights, notifications and historical trends, demonstrating the feasibility of using NodeMCU for IoT environmental sensing.