



IoT-Based Environmental Monitoring System Using ESP8266 NodeMCU and Multi-Sensor Data Acquisition

Adupa Vaibhav¹, Shariff Baig², Kommu Mallesh³

¹Student, EEE, Methodist college of engineering and technology

Abstract

Environmental monitoring is essential in industrial zones, smart cities, homes, laboratories, and other locations where air quality and ambient conditions require continuous supervision. Conventional monitoring practices depend on manual readings or isolated instruments, which limits remote access, automation, and rapid response. This paper presents a low-cost Internet of Things (IoT)-based environmental monitoring prototype developed around an ESP8266 NodeMCU controller. The system integrates a DHT11 temperature-humidity sensor, an LDR light sensor, an MQ-series gas sensor, a 16 x 2 LCD, a buzzer, and cloud connectivity through Firebase or a similar IoT platform. Sensor readings are acquired by the microcontroller, displayed locally, compared with threshold limits, and transmitted through Wi-Fi for remote monitoring. The prototype also generates warning indications when gas concentration, temperature, or humidity exceeds predefined limits. Functional testing confirms that the system can provide real-time sensing, local alerts, and remote data availability in a compact embedded implementation. The proposed design is suitable as a foundation for smart-city pollution monitoring, indoor safety systems, greenhouse supervision, and educational IoT applications.

Keyword: Air quality, DHT11, ESP8266, environmental monitoring, Firebase, Internet of Things, LDR, MQ gas sensor, NodeMCU.

I. INTRODUCTION

Environmental monitoring systems are increasingly required because air quality, temperature, humidity, and light intensity directly affect public health, industrial safety, indoor comfort, agriculture, and urban planning. Manual monitoring with thermometers, hygrometers, and portable gas detectors can provide local readings, but it usually requires human observation and separate data recording. Such systems provide limited automation and cannot easily support real-time alerts or remote supervision.

The Internet of Things (IoT) offers a practical solution by connecting low-cost sensors, embedded controllers, and cloud platforms. In an IoT-based monitoring unit, sensor readings can be collected periodically, processed by a microcontroller, displayed locally, and transmitted to a remote database for observation from a web or mobile interface. This improves response time and makes environmental data more accessible to users, facility managers, and municipal authorities.

This work converts an embedded mini-project into an IEEE-style research paper. The implemented system uses an ESP8266 NodeMCU as the processing and communication unit. The main sensing elements are a DHT11 sensor for temperature and humidity, an LDR for light intensity, and an MQ-series gas sensor for

smoke or air-quality detection. A buzzer is used for local warning, while an LCD displays measured values. The main contribution of this prototype is a simple and low-cost architecture for real-time environmental monitoring with local and cloud-based feedback.

II. RELATED WORK

IoT-based environmental monitoring has been widely studied as an alternative to conventional manual measurement. Prior systems combine embedded controllers, gas sensors, temperature-humidity modules, and wireless communication to provide real-time pollution information and remote data logging [5]-[9]. Cloud-based approaches extend these systems by storing sensor data in online databases, allowing users to observe historical trends and react to abnormal conditions [3], [7].

The ESP8266 platform is frequently selected for low-cost monitoring nodes because it combines microcontroller functions with Wi-Fi connectivity in a compact board [1]. Arduino-based development tools are also common because they reduce programming complexity and provide library support for sensor reading, LCD communication, Wi-Fi communication, and database updates [2]. Compared with higher-cost monitoring stations, low-cost IoT nodes are easier to distribute across multiple locations, although their accuracy depends on sensor calibration, environmental conditions, and reliable network access.

III. PROPOSED SYSTEM

A. System Architecture

The proposed prototype is an IoT-based environmental pollution monitoring system. As shown in Fig. 1, three sensor inputs are connected to the ESP8266 NodeMCU: the DHT11 sensor measures temperature and humidity, the LDR detects light level, and the MQ gas sensor measures air-quality or smoke-related variation. The controller processes the readings, compares them with threshold values, displays the data on a 16 x 2 LCD, activates the buzzer during unsafe conditions, and sends data to Firebase or a similar IoT cloud platform through Wi-Fi.

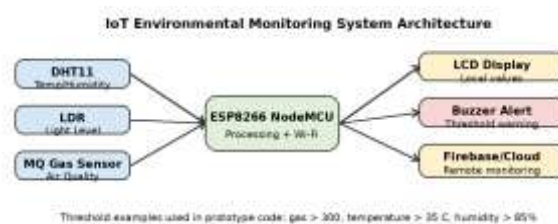


Fig. 1. Architecture of the proposed IoT-based environmental monitoring system.

B. Hardware Components

The hardware design uses widely available embedded modules so that the prototype remains low-cost and easy to reproduce. The ESP8266 NodeMCU reads sensor data and provides Wi-Fi communication. The DHT11 provides digital temperature and humidity readings over a single-wire data line. The LDR is used for light-level sensing, while the MQ-series gas sensor provides an analog output related to smoke or gas concentration. The LCD provides on-site visualization, and the buzzer works as a local warning device. The prototype uses five core modules: (i) ESP8266 NodeMCU for sensing control and Wi-Fi communication, (ii) DHT11 for temperature and humidity, (iii) LDR for light-level detection, (iv) MQ-series gas sensor for smoke or air-quality warning input, and (v) LCD and buzzer for local display and alert indication.



C. Alert and Cloud-Update Logic

The monitoring logic follows a sequential data-acquisition cycle. After Wi-Fi connection, the system reads temperature and humidity from the DHT11, reads gas level from the analog sensor channel, and reads the LDR state. The values are displayed locally and pushed to the database. If any threshold is exceeded, the buzzer is activated and an alert message is displayed. The prototype code uses example limits of gas value greater than 300, temperature greater than 35 C, and humidity greater than 85%. These limits can be changed according to the target environment and calibration results.

IV. IMPLEMENTATION

A. Sensor Interfacing

The implementation uses the ESP8266 NodeMCU board programmed through the Arduino IDE. The DHT11 sensor is connected to GPIO D4. The LDR input is connected to D6, the buzzer is connected to D5, and the smoke or gas sensor output is connected to the analog input A0. The LCD is interfaced through the I2C bus using the common address 0x27. This configuration uses simple digital and analog interfaces, reducing circuit complexity.

B. Software Flow

The embedded program initializes serial communication, LCD display, DHT11 sensor, buzzer output, and Wi-Fi. After connection to the cloud service, the main loop repeatedly acquires sensor data, updates the local display, checks thresholds, controls the buzzer, and updates the cloud fields. The summarized algorithm is shown below.

Algorithm 1: IoT Environmental Monitoring Routine

1. Initialize LCD, DHT11, buzzer, LDR, Wi-Fi, and cloud database
2. while system is powered do
3. Read temperature and humidity from DHT11
4. Read gas sensor value from A0 and light state from LDR
5. Display temperature, humidity, and gas level on LCD
6. if gas > 300 or temperature > 35 C or humidity > 85% then
7. Activate buzzer and update cloud alert flag
8. Display warning message on LCD
9. else deactivate buzzer and clear alert flag
10. Upload latest readings to the IoT database
11. end while

C. Local and Remote User Feedback

The LCD provides immediate local observation of temperature, humidity, and gas level. The buzzer provides a direct acoustic warning when an unsafe condition is detected. The cloud database enables remote monitoring and future data visualization. This dual feedback approach is important because local users can respond immediately while remote users can observe the system status without being physically present near the device.

V. RESULTS AND DISCUSSION

The prototype was evaluated as a functional environmental monitoring system. The ESP8266 successfully connected to the Wi-Fi network and operated as the central controller. The LCD displayed sensor values, and the alert mechanism was triggered when the monitored variables crossed the programmed threshold



limits. Firebase-based data transmission allowed sensor readings and alert states to be made available remotely.

The functional results show three main outcomes. First, the system can acquire multiple environmental variables in real time using low-cost sensors. Second, the local LCD and buzzer make the device useful even without a separate computer screen. Third, cloud connectivity improves accessibility and enables remote monitoring. These features make the system suitable for low-cost smart homes, laboratories, classrooms, small industries, and preliminary smart-city demonstrations.

The prototype also has important limitations. The DHT11 has limited precision and a relatively slow response rate compared with industrial sensors. MQ gas sensors require warm-up time and calibration for quantitative gas concentration measurement. The ESP8266 has only one analog input, so additional analog sensors require a multiplexer or an external analog-to-digital converter. Wi-Fi dependency can also affect data transmission reliability. Therefore, the current design is best interpreted as a proof-of-concept monitoring node rather than a fully calibrated regulatory monitoring station.

VI. APPLICATIONS AND SUSTAINABILITY RELEVANCE

The system can be applied to indoor air-quality monitoring, greenhouse monitoring, laboratory safety, smart-home automation, small industrial units, and campus-level environmental observation. The same architecture can also be extended to include pH, turbidity, dust, CO₂, or particulate-matter sensors depending on the target application.

From a sustainability perspective, the work supports smart-city and climate-action goals by enabling distributed sensing of air quality and environmental conditions. The prototype aligns with sustainable cities through urban air-quality observation, clean-water monitoring if water-quality sensors are added, clean-energy management when integrated with smart buildings, and climate action through greenhouse-gas or disaster-related sensing extensions.

VII. CONCLUSION AND FUTURE WORK

This paper presented an IoT-based environmental monitoring prototype using an ESP8266 NodeMCU, DHT11, LDR, MQ-series gas sensor, LCD, buzzer, and cloud connectivity. The proposed system provides real-time sensing, local display, local warning, and remote data access. The design is low-cost, compact, and suitable for educational and preliminary field-monitoring applications. Functional testing showed that the system can detect abnormal environmental conditions and produce alert outputs based on defined threshold limits.

Future work should focus on higher-accuracy sensors, calibration against reference instruments, secure user authentication, encrypted data transfer, dashboards with trend graphs, and predictive analysis using machine learning. The system can also be connected to ventilation fans, air purifiers, or other actuators so that it not only monitors environmental conditions but also performs automatic corrective action.

ACKNOWLEDGMENT

The authors acknowledge the Department of Electrical and Electronics Engineering, Methodist College of Engineering and Technology, Hyderabad, for academic support and guidance during the development of the mini-project prototype.



REFERENCES

1. Espressif Systems, "ESP8266EX Datasheet," 2023.
2. Arduino, "Arduino IDE Documentation." [Online]. Available: <https://www.arduino.cc>
3. Google Firebase, "Firebase Realtime Database Documentation." [Online]. Available: <https://firebase.google.com/docs>
4. ThingSpeak, "ThingSpeak IoT Platform Documentation." [Online]. Available: <https://thingspeak.com>
5. A. Kumar and S. Patel, "IoT based air quality monitoring system," International Journal of Engineering Research, 2019.
6. R. Singh et al., "Environmental monitoring using IoT and sensors," in Proc. International Conference on Smart Systems, 2021.
7. M. Sharma and P. Gupta, "Cloud-based environmental monitoring using IoT," Journal of IoT Applications, 2020.
8. S. Reddy et al., "Air pollution monitoring system using MQ sensors," International Journal of Advanced Research, 2018.
9. IEEE, "IoT-based smart monitoring systems: A review," IEEE Journals, 2020.
10. D. Brown, Introduction to Embedded Systems and IoT. New York, NY, USA: McGraw Hill, 2019.
11. K. Lee, Smart Environmental Monitoring Systems. Singapore: Springer, 2021.
12. Sensor Datasheets: DHT11, MQ-2, and MQ-135 gas sensors.