IJARSCT

International Journal of Advanced Research in Science, Communication and Technology

International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal

ISSN: 2581-9429 Volume 5, Issue 1, May 2025 Impact Factor: 7.67

Design and Implementation of an IoT-Based Air Quality Monitoring System using ESP32

Mrs. Surabhi Pratick Relekar[1] and Dr. A. L. Renke[2]

Student, E&TC, KIT's College of Engineering, Kolhapur, India[1]

Associate Professor E&TC, KIT's College of Engineering, Kolhapur, India[2]

Abstract: *This paper presents a low-cost, portable air quality monitoring system designed using the ESP32 microcontroller. It integrates a DHT11 sensor for temperature and humidity measurements and an MQ-135 sensor to detect harmful gases. An I2C-based LCD display shows real-time data, enabling users to monitor environmental conditions effectively. The system is intended for smart home and indoor air quality applications, providing a scalable solution for real-time pollution tracking and awareness.*

Keywords: ESP32, MQ-135 Gas Sensor, DHT11 Sensor, I2C 16x2 LCD Display, Power Supply (5V, 2A), Cloud platform (BlynkIoT)

I. INTRODUCTION

Air pollution has become a significant environmental concern, especially in urban areas, due to rapid industrialization and increasing vehicle emissions. Monitoring air quality in real-time is essential to take preventive measures for public health and environmental protection. Traditional air quality monitoring systems are often expensive, stationary, and limited in coverage. This project proposes a low-cost, portable, and efficient solution using the ESP32 microcontroller. By integrating sensors such as the MQ-135 for gas detection and DHT11 for temperature and humidity, the system collects environmental data and displays it on an LCD. Additionally, the ESP32's Wi-Fi capability enables real-time data transmission to cloud platforms for remote monitoring. This approach makes the system suitable for smart city applications and personal use in homes or workplaces.

II. RELATED WORK

Several researchers and developers have explored IoT-based air quality monitoring systems in recent years. These systems typically integrate microcontrollers with gas and environmental sensors to collect and transmit data in real time.

In [1], an Arduino-based air monitoring system was developed using the MQ-135 sensor, but the lack of Wi-Fi support limited its remote accessibility. Another study [2] employed a Raspberry Pi with advanced sensors for multi-gas detection; however, the high cost and power requirements made it less suitable for widespread deployment.

Recent developments have shifted toward using the ESP32 microcontroller due to its built-in Wi-Fi and Bluetooth capabilities. For example, [3] implemented an ESP32-based air quality monitor using the BME680 sensor and ThingSpeak for cloud data logging, demonstrating the effectiveness of IoT platforms in environmental monitoring.

Compared to previous work, the current project focuses on a compact, cost-effective design with essential sensor integration and real-time display. It emphasizes simplicity, affordability, and practical deployment, especially in low-resource settings.

III. PROPOSED SYSTEM DESIGN

The proposed system is an IoT-based air quality monitoring solution designed to measure and report environmental parameters such as air quality, temperature, and humidity in real time. The core of the system is the ESP32 microcontroller, selected for its dual-core processor, integrated Wi-Fi, and low power consumption—making it ideal for IoT applications.

Project Overview

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 1, May 2025
Impact Factor: 7.67

This project focuses on designing and implementing a smart, low-cost air quality monitoring system using the ESP32 microcontroller. The primary goal is to detect and report environmental conditions such as air quality (gas concentration), temperature, and humidity in real time. The system integrates the MQ-135 gas sensor and the DHT11 temperature and humidity sensor, with data output displayed locally on a 16x2 I2C LCD and transmitted to the cloud using Wi-Fi for remote monitoring.

The compact design and low power consumption of the ESP32 make it suitable for continuous operation in homes, classrooms, offices, and outdoor environments. This project demonstrates a practical approach to environmental sensing and serves as a prototype for broader IoT-based smart city applications.

Block diagram

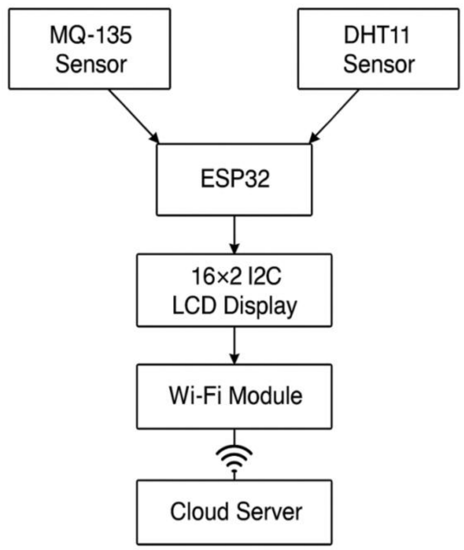


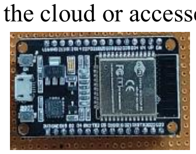
Fig -1: System Architecture

IV. HARDWARE DESCRIPTION

The proposed IoT-based air quality monitoring system is built using essential electronic components selected for their reliability, low power consumption, and ease of integration. Below is a detailed description of each hardware module used:

ESP32 Microcontroller

The **ESP32** is a low-cost, low-power system on a chip (SoC) with integrated Wi-Fi and Bluetooth. It features dual-core processing, multiple GPIO pins, ADCs, UART, I2C, and SPI interfaces. Its built-in Wi-Fi module makes it ideal for IoT applications where data needs to be transmitted to the cloud or accessed remotely.



4.2 MQ-135 Gas Sensor

The **MQ-135** is a versatile air quality sensor capable of detecting a range of gases including ammonia (NH_3), nitrogen oxides (NO_x), alcohol, benzene, smoke, and carbon dioxide (CO_2). It provides an analog output that varies with the concentration of pollutants in the air.

Operating Voltage: 5V
Output: Analog (AO), Digital (DO)
Sensitivity: Detects multiple harmful gases

DHT11 Temperature and Humidity Sensor

The **DHT11** sensor measures temperature and relative humidity. It has a single-wire digital interface, making it simple to connect to microcontrollers like the ESP32.

Temperature Range: 0–50°C ±2°C
Humidity Range: 20–90% RH ±5%
Power Supply: 3.3V–5V

16x2 I2C LCD Display

The **16x2 LCD with I2C interface** is used to display real-time readings of air quality, temperature, and humidity. The I2C interface simplifies wiring, requiring only two data pins (SDA and SCL), which saves ESP32 GPIOs.

Display: 2 lines × 16 characters
Interface: I2C (uses only 2 pins)
Voltage: 5V

Power Supply

A **5V/2A DC adapter** is used to power the ESP32 and connected modules. Consistent power is crucial for stable sensor readings and Wi-Fi transmission.

V. SOFTWARE DESCRIPTION

The software component of this system is built around the **BlynkIoT platform**, which provides a user-friendly interface for creating real-time dashboards and controlling IoT devices remotely. The **ESP32 microcontroller** is programmed using the **Arduino IDE**, with the Blynk library integrated to enable communication between the device and the Blynk cloud server.

Key components of the software setup include:
Arduino IDE: Used to write, compile, and upload code to the ESP32.
Blynk Library: Allows ESP32 to send sensor data and receive commands via the internet.

Blynk Mobile App / Web Dashboard:
Displays real-time data from sensors such as PM2.5, CO2, temperature, and humidity.
Enables customizable widgets like gauges, graphs, and LED indicators for better visualization.

Wi-Fi Communication: ESP32 connects to the internet via Wi-Fi and pushes data to Blynk Cloud.

VI. METHODOLOGY

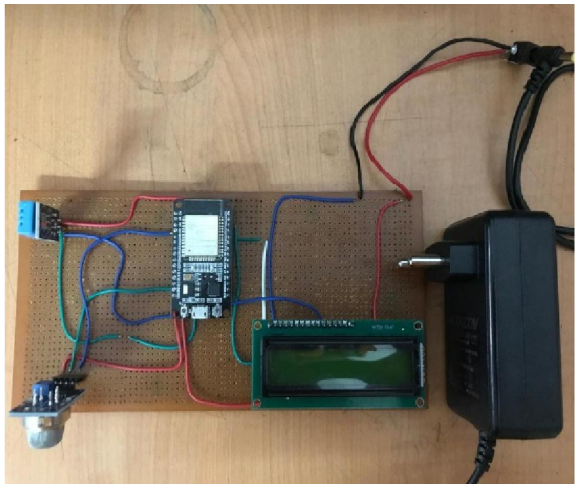
The development of the IoT-based air quality monitoring system follows a modular and systematic approach consisting of the following phases:
System Design
The system is designed to collect real-time environmental data and send it to the cloud for monitoring. Key components include:

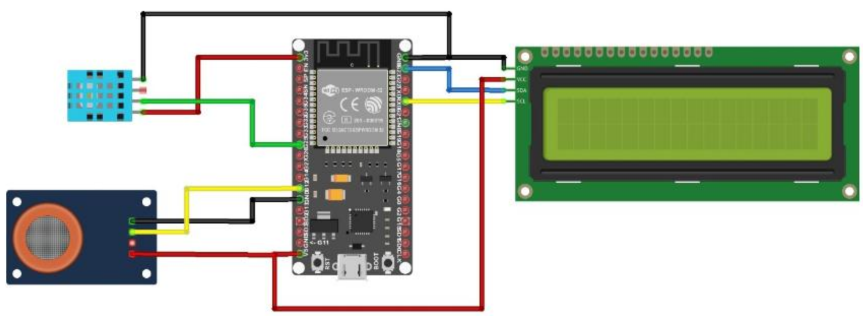
- **ESP32 microcontroller** for data processing and Wi-Fi communication.
- **Sensors** such as:
- **MQ135** for gas detection (CO_2, NH_3, benzene, etc.).
- **PMS5003** for measuring PM2.5 and PM10 particulate matter.
- **DHT11/DHT22** for temperature and humidity.

Circuit Integration
All sensors are connected to the ESP32 through appropriate GPIO pins. Voltage level considerations (3.3V for ESP32) and noise filtering are managed to ensure accurate readings. The entire system is powered via USB or a 5V portable battery.
Firmware Development
The Arduino IDE is used to program the ESP32:
Sensor data is collected using standard sensor libraries.
Blynk library is included to enable connectivity with the Blynk cloud.
Wi-Fi credentials and BlynkAuth Token are configured in the code.
Data is sent to the Blynk dashboard using virtual pins.
Cloud Integration with Blynk
The **Blynk mobile app or web dashboard** is configured with widgets (gauges, graphs, LEDs).
ESP32 pushes real-time data to Blynk Cloud over Wi-Fi.
Users can monitor values such as PM2.5, CO2, temperature, and humidity remotely.
Testing and Validation
The system is tested in various environments (indoor and outdoor) to:
Compare sensor readings with commercial air quality monitors.
Validate data accuracy and transmission stability.
Calibrate sensors if required for improved reliability.

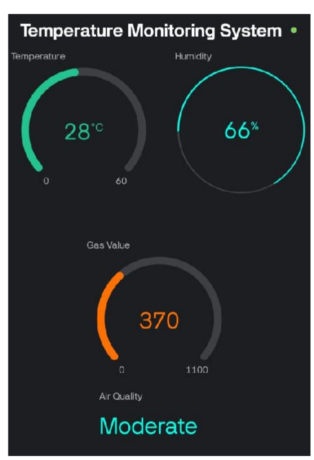
IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 1, May 2025
Impact Factor: 7.67

VII. EXPERIMENTAL SETUP

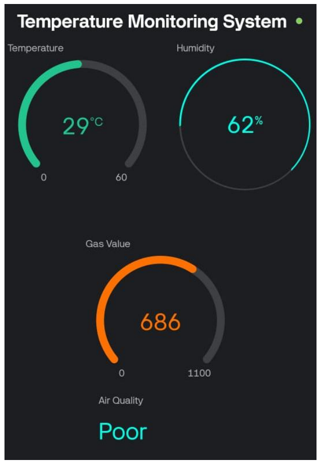


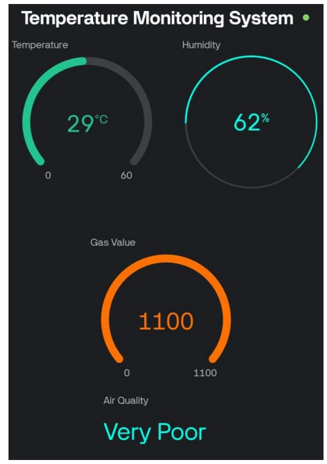


VIII. RESULTS



DOI: 10.48175/IJARSCT-26145





IX. CONCLUSION

This paper demonstrates a working prototype of an IoT-based air quality monitoring system using ESP32. It offers a low-cost, scalable, and wireless solution for urban and rural environmental monitoring. Future work includes battery power integration, mobile app development, and AI-based pollutant prediction.

X. ACKNOWLEDGMENT

I would like to express my sincere gratitude to my project guide, Dr. A. L. Renke, for their valuable guidance and support throughout the development of this project. I am also thankful to the Department of Electronics and Telecommunication Engineering, KIT's College of Engineering, Kolhapurfor providing the resources and infrastructure needed to complete this work.

Special thanks to my family and friends for their constant encouragement and motivation. Finally, I acknowledge the use of open-source platforms and online communities that greatly assisted in resolving technical challenges during this project.

REFERENCES

[1]. MQ-135 Datasheet
[2]. DHT11 Sensor Datasheet
[3]. ESP32 Technical Reference Manual
[4]. Arduino IDE Documentation
[5]. IoT Platform APIs (ThingSpeak/Blynk)
[6]. Kumar, R. Singh and M. Singh, "IoT-Based Air Pollution Monitoring System using Arduino," International Journal of Scientific Research in Computer Science, Engineering and Information Technology, vol. 3, no. 6, pp. 57-62, 2018.
[7]. M. Z. H. Noor, M. Z. A. A. Kadir, and A. F. A. Hanan, "Real-Time Monitoring of Air Quality Using Internet of Things (IoT)," International Journal of Advanced Computer Science and Applications (IJACSA), vol. 10, no. 5, pp. 419–423, 2019.
[8]. S. Dey, A. Das, and T. Chakraborty, "Air Quality Monitoring System Using IoT and Machine Learning," 2020 IEEE International Conference on Computing, Power and Communication Technologies (GUCON), pp. 398-403, 2020.
[9]. Espressif Systems, "ESP32 Technical Reference Manual," [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf

ISSN: 2581-9429

Volume 5, Issue 1, May 2025

Impact Factor: 7.67

[10]. M. R. Islam, M. S. Alam, and M. Hasan, "Design and Implementation of a Low-Cost Air Quality Monitoring System Using IoT and Cloud," 2021 5th International Conference on Electrical Engineering and Information & Communication Technology (ICEEICT), pp. 1-6, 2021.

DOI: 10.48175/IJARSCT-26145