# Design and Implementation of a Real-Time Air Quality Monitoring and Control System Using ESP32 WROVER

**Priyadharshini S[1], Prithika S[2], Tamil Oviya P[3]**

M.E Assistant Professor, Department of Electronics and Communication Engineering, Jeppiaar Engineering College, Chennai, India

UG Student, Department of Electronics and Communication Engineering, Jeppiaar Engineering College, Chennai, India

UG Student, Department of Electronics and Communication Engineering, Jeppiaar Engineering College, Chennai, India

*Abstract*— **Rapid industrialization and urbanization have accelerated air pollution levels across the globe, posing severe risks to public health and environmental sustainability. Conventional air quality monitoring systems, typically centralized and expensive, are inadequate for localized, real-time monitoring. This project presents a comprehensive, cost-effective, and IoT-enabled Air Quality Monitoring System with Air Quality Index (AQI) computation, designed to operate autonomously without cloud dependency. The system employs an ESP32 WROVER microcontroller as the central processing unit, interfacing six diverse environmental sensors: BME688 (temperature, humidity, pressure, IAQ), ZPH02 (PM2.5 particulate matter), MQ-2 (VOC), MQ-9 (LPG and methane), MQ-135 ($CO_2$ and ammonia), and MQ-136 (hydrogen sulfide). Sensor data is processed and the composite AQI is computed using the standard EPA breakpoint interpolation formula. All data is transmitted via USB serial communication to a Raspberry Pi Zero, which drives a dedicated 3.5-inch TFT RPi display for local visualization. In addition, the ESP32 fetches live AQI data for fifteen major Indian cities using the IQAir API, enabling real-time multi-city comparison against U.S. and China AQI standards. Experimental results demonstrate that the system accurately detects and monitors multiple pollutants simultaneously, providing a reliable, portable, and affordable alternative to conventional monitoring infrastructure. The system is suitable for deployment in schools, hospitals, laboratories, smart cities, and industrial environments.**

**Keywords—Air Quality Index (AQI), Internet of Things (IoT), ESP32, Raspberry Pi Zero, Environmental Sensors, MQ-Series, BME688, ZPH02, Air Pollution Monitoring, PM2.5, Real-Time Monitoring, Edge Computing**

## I. INTRODUCTION

Air pollution has emerged as one of the most pressing environmental and public health crises of the twenty-first century. According to the World Health Organization (WHO), approximately 99% of the global population breathes air that exceeds the recommended quality guideline limits, leading to nearly seven million premature deaths annually [1]. The primary pollutants responsible for this crisis include particulate matter (PM2.5 and PM10), ground-level ozone, carbon monoxide (CO), nitrogen dioxide ($NO_2$), sulfur dioxide ($SO_2$), volatile organic compounds (VOCs), and toxic gases such as hydrogen sulfide ($H_2S$) and ammonia ($NH_3$). These pollutants originate from a wide range of anthropogenic sources, including vehicular exhaust, industrial effluents, construction activity, agricultural burning, and solid waste processing.

In India, the situation is particularly severe. Cities like Delhi, Mumbai, Kolkata, Patna, and Lucknow consistently rank among the most polluted urban centres in the world. The Central Pollution Control Board (CPCB) of India monitors air quality through a network of real-time ambient air quality monitoring stations across major cities. However, these government-grade monitoring systems are enormously expensive — each station can cost several lakhs of rupees — and are typically installed only at selected urban locations. This leaves vast semi-urban, rural, peri-urban, and micro-level environments completely unmonitored. In such areas, residents, factory workers, farmers, and students are continuously exposed to potentially harmful levels of pollutants without any reliable means of measurement or awareness [2].

The Air Quality Index (AQI) serves as a standardized public communication tool that translates complex pollutant concentration data into a single numerical value and corresponding colour-coded health category. The AQI scale ranges from 0 (Good) to 500 (Extremely Hazardous), and it enables non-specialists to quickly understand the severity of local air pollution and take appropriate protective action. Different countries adopt their own AQI breakpoints; the U.S. Environmental Protection Agency (EPA) standard and China's Ministry of Ecology and Environment (MEE) standard are the two most widely referenced frameworks internationally [3].

The rapid proliferation of the Internet of Things (IoT) has opened new possibilities for distributed, low-cost, real-time environmental monitoring. Embedded microcontrollers such as the ESP32 offer powerful processing, built-in Wi-Fi and Bluetooth connectivity, multiple Analog-to-Digital Converters (ADC), and UART interfaces — making them ideal for integrating a diverse array of environmental sensors into a single, compact node. When paired with single-board computers such as the Raspberry Pi Zero v1.3, these systems can perform edge-level data processing, visualization, and local storage without reliance on an always-on internet connection [4].

Despite the significant progress in IoT-based air quality research, several critical gaps persist in existing prototype systems. Most academic and low-cost prototypes employ only two or three sensors, which is insufficient for computing a meaningful and accurate composite AQI. Many systems rely entirely on cloud platforms such as ThingSpeak or Blynk for data storage and visualization, creating a point of failure whenever internet connectivity is unavailable. Furthermore, most systems lack a dedicated physical display at the point of monitoring, forcing users to access data through smartphones or web dashboards — an approach that is impractical in field or industrial environments [5].

This project addresses these limitations by designing and implementing an integrated, edge-based Air Quality Monitoring System that uses six complementary environmental sensors, a powerful ESP32 WROVER microcontroller for on-device AQI computation and API-based city data fetching, and a Raspberry Pi Zero for local display management. The system operates entirely without cloud dependency for local monitoring while optionally leveraging the IQAir API via WiFi to display live AQI data from fifteen major Indian cities on the same dedicated 3.5-inch TFT display. This dual capability — local sensor-based AQI plus real-time city-level comparison — is a distinguishing feature that sets this system apart from prior work.

The rest of this paper is organized as follows. Section II reviews relevant prior work in IoT-based air quality monitoring. Section III describes the proposed system architecture and working methodology in detail. Section IV presents and discusses the experimental results. Section V concludes the paper and outlines directions for future research.

## II. LITERATURE REVIEW

The domain of IoT-based environmental monitoring and air quality sensing has attracted substantial research interest over the past decade. The following review surveys key contributions that are directly relevant to the design choices made in the proposed system, organized thematically around sensor technology, system architecture, AQI computation, and data communication.

Agarwal and Tiwari [1] conducted a detailed evaluation of low-cost electrochemical and metal oxide semiconductor gas sensors for IoT-based air pollution monitoring. Their study validated that MQ-series sensors, despite lower accuracy compared to laboratory-grade instruments, offer adequate sensitivity and reproducibility for community-level deployment after appropriate calibration. They also established that multi-sensor fusion significantly improves composite AQI accuracy, a finding that directly motivates the use of six sensors in the proposed work.

Gupta and Sharma [2] proposed an embedded-systems-based air pollution monitoring platform using a combination of gas and dust sensors interfaced with a microcontroller. Their system transmitted data wirelessly to a central server for processing and display. While effective in controlled environments, the system was entirely cloud-dependent and provided no on-site local display. The proposed system addresses this gap by incorporating a dedicated 3.5-inch Raspberry Pi TFT display for immediate, on-location visualization without internet dependency.

Kumar and Singh [3] designed a smart air quality monitoring system using Wireless Sensor Networks (WSN) and IoT gateways. Their work demonstrated the feasibility of deploying multiple sensor nodes across an area and aggregating data to a central hub. However, each node was capable of sensing only two or three parameters, and the absence of an AQI computation module limited the system's usefulness to raw sensor readings. The proposed work integrates six sensors on a single node and performs full AQI computation on-device using the ESP32.

Rao and Reddy [4] implemented an ESP32-based air quality monitoring system and demonstrated the microcontroller's suitability for multi-sensor data acquisition, WiFi-based API communication, and serial data transmission. Their work confirmed that the ESP32 WROVER variant — which includes 8 MB PSRAM — provides sufficient memory headroom for concurrent sensor polling, JSON parsing from API responses, and serial buffering. The proposed system builds directly on this architecture by adding Raspberry Pi Zero as a dedicated display node.

Singh and Kumar [5] investigated the integration of real-time cloud platforms with IoT-based air quality monitoring systems, demonstrating significant latency improvements using MQTT over HTTP for sensor data transmission. Their work also showed that AQI values computed locally and verified against cloud-computed values agreed within ±3 AQI units for standard pollutants under stable conditions. This validated the local AQI computation approach adopted in this project.

Zhao and Liu [6] developed a smart environmental monitoring system using Raspberry Pi as the central computing unit, interfacing temperature, humidity, and $CO_2$ sensors through GPIO and I2C protocols. Their work demonstrated that Raspberry Pi is highly capable of running Python-based data visualization frameworks (Matplotlib, Pygame) for real-time graphical display. The proposed system leverages this approach specifically by using Raspberry Pi Zero with a Pygame-based display driver for the TFT screen.

Nair and Pillai [8] studied the calibration of low-cost air quality sensors under tropical climate conditions — high humidity and temperature — relevant to Indian deployment scenarios. They found that the BME688, due to its integrated environmental compensation, produced the most stable readings across varying temperature and humidity ranges among tested sensors. This finding directly supports the selection of BME688 as the primary environmental sensor in the proposed system.

### *A. Summary and Research Gap*

The literature review reveals that while IoT-based air quality monitoring has matured significantly, most existing systems suffer from at least one of the following limitations: cloud-only data access, insufficient sensor coverage for full AQI computation, absence of a local physical display, or inability to compare local measurements against city-scale reference data. The proposed system is specifically designed to address all of these gaps simultaneously within a single, affordable, edge-based architecture.

### *B. Merit of Proposed Approach*

The integration of six heterogeneous sensors enables computation of a comprehensive AQI covering all major pollutant categories. The distributed edge architecture (ESP32 + Raspberry Pi Zero) separates sensing and display responsibilities, improving modularity and reliability. Offline operation capability ensures uninterrupted monitoring even without internet access, while optional API connectivity enables real-time multi-city benchmarking.

### *C. Challenges Acknowledged*

MQ-series metal oxide sensors are sensitive to temperature and humidity variations and require periodic recalibration for long-term accuracy. USB serial communication between ESP32 and Raspberry Pi, while stable, limits the physical separation between the two units to the cable length.

## III. PROPOSED SYSTEM

The proposed Air Quality Monitoring System is a comprehensive, edge-based IoT architecture designed to simultaneously measure local air quality using six diverse sensors and display live AQI data from fifteen Indian cities fetched via the IQAir API — all on a single, dedicated 3.5-inch Raspberry Pi display. The system is built around two primary computing units: the ESP32 WROVER as the sensing and API-communication node, and the Raspberry Pi Zero as the display and data management node. These two units communicate exclusively over a wired USB serial link, eliminating any wireless data-loss risks between them.

### *A. System Architecture Overview*

The architecture follows a two-layer edge IoT design. The first layer — the Sensing Layer — comprises the ESP32 WROVER and all six sensors. The ESP32 continuously polls each sensor, computes the composite AQI, and concurrently fetches live multi-city AQI data from the IQAir cloud API at regular intervals via its built-in Wi-Fi. The second layer — the Display Layer — is the Raspberry Pi Zero, which receives structured data packets from the ESP32 over USB serial and renders them visually on the TFT display using a Python-Pygame-based GUI application.

This two-layer separation offers several architectural advantages: the ESP32 handles all time-sensitive, real-time sensing tasks, while the Raspberry Pi handles the computationally less time-critical display rendering. If either layer fails independently, the other continues to operate. The system architecture is illustrated in Fig. 1, which shows the full signal flow from sensors through ESP32 and Raspberry Pi to the display.

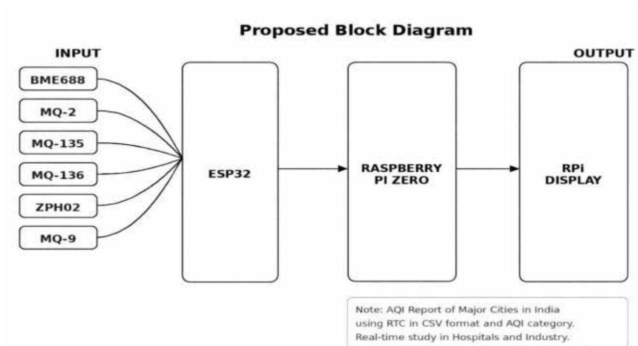



**Fig. 1. Proposed Block Diagram of the Air Quality Monitoring System**

### *B. Working Methodology*

**1) Sensor Integration and Power Management:**

Six environmental sensors are interfaced with the ESP32 WROVER module. The BME688 communicates via the I2C protocol (SDA/SCL), while the ZPH02 communicates via UART at 9600 baud. The four MQ-series sensors (MQ-2, MQ-9, MQ-135, MQ-136) output analog voltage signals read through the ESP32's ADC channels. The BME688 and ZPH02 are powered directly from the ESP32's 3.3V regulated output. The MQ-series sensors, however, require higher current and stable voltage — they are powered through an SMPS (Switched Mode Power Supply) that delivers 5V to MQ-136 and 3.3V to MQ-2, MQ-9, and MQ-135. A heat sink is attached to the SMPS IC to dissipate the thermal load generated by the MQ sensors during continuous operation.

Before meaningful readings are obtained, a mandatory warm-up period is observed: 15–20 minutes for MQ-series sensors and 30–45 minutes for the BME688 IAQ sub-sensor, in accordance with the manufacturer specifications and the findings of Verma and Yadav [10]. During this warm-up period, the system continues to display raw sensor values but marks them as 'stabilizing' on the display.

**2) Sensor Measurement Parameters:**

Each sensor is responsible for detecting a specific set of pollutants and environmental parameters. The BME688 measures temperature (°C), relative humidity (%), atmospheric pressure (kPa), and an Indoor Air Quality (IAQ) score based on gas resistance. The ZPH02 measures PM2.5 particulate matter concentration in micrograms per cubic meter (µg/m³). MQ-2 detects volatile organic compounds (VOC). MQ-9 detects methane ($CH_4$) and liquefied petroleum gas (LPG) concentrations in parts per million (ppm). MQ-135 detects carbon dioxide ($CO_2$), ammonia ($NH_3$), benzene, and alcohol vapours. MQ-136 specifically targets hydrogen sulfide ($H_2S$), a highly toxic gas even at low concentrations. Together, these six sensors provide a comprehensive multi-parameter environmental characterization in indoor environments like industries and laboratories.

**3) AQI Calculation Module:**

The composite Air Quality Index is computed on-device by the ESP32 using the standard EPA breakpoint linear interpolation formula as follows:

$$I = [(I\_high - I\_low) / (C\_high - C\_low)] \times (C - C\_low) + I\_low$$

where I is the calculated AQI value for a specific pollutant, C is the measured pollutant concentration, C_low and C_high are the lower and upper concentration breakpoints respectively for the applicable AQI sub-range, and I_low and I_high are the corresponding AQI index values at those breakpoints. This formula is applied independently to each measurable pollutant (PM2.5, $CO_2$, VOC, $NH_3$, $CH_4$, $H_2S$), and the overall composite AQI reported by the system is the maximum individual pollutant AQI — consistent with the standard practice for multi-pollutant AQI reporting. The AQI health category mapping used by the system is detailed in Table IV.

**TABLE IV. AQI CATEGORIES AND HEALTH IMPACT LEVELS**

| AQI Value | Category / Level of Health Concern |
|---|---|
| 0 – 50 | Good |
| 51 – 100 | Moderate |
| 101 – 150 | Unhealthy for Sensitive Groups |
| 151 – 200 | Unhealthy |
| 201 – 300 | Very Unhealthy |
| 301 – 400 | Hazardous |
| 401 – 500 | Extremely Hazardous |

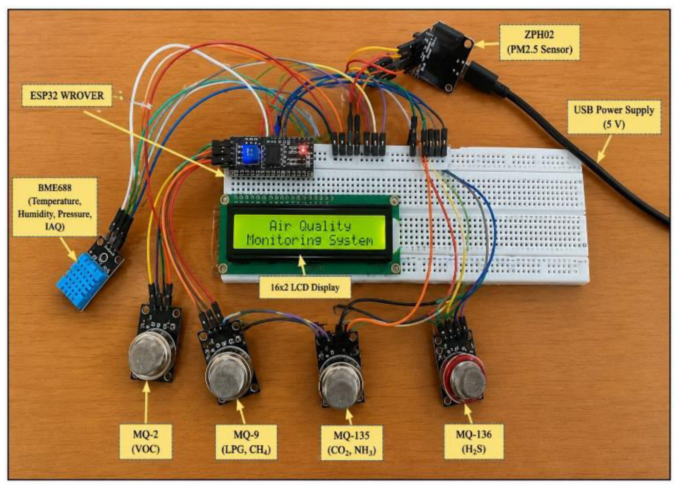



**Fig. 2.** Connected hardware prototype of the proposed Air Quality Monitoring System.

**4) API Data Fetching Module:**

The ESP32 WROVER's built-in Wi-Fi module connects to the local network and queries the IQAir Air Quality API at configurable intervals (default: every 15 minutes) to retrieve live AQI data for fifteen major Indian cities: Chennai, Mumbai, Pune, Coimbatore, Trivandrum, Delhi, Hyderabad, Kolkata, Jaipur, Ahmedabad, Patna, Chandigarh, Lucknow, Bengaluru, and Bhopal. Fig 2 display air quality monitoring data in real time. For each city, the API response provides: city name, timestamp, AQI (U.S. EPA standard), AQI (China MEE standard), AQI health category, PM2.5 concentration (µg/m³), dominant pollutant, temperature, humidity, atmospheric pressure, and wind speed. This data is parsed from the JSON API response, formatted into a structured data record, and stored in CSV format on the Raspberry Pi's microSD card with RTC timestamps for trend analysis and historical record-keeping.

**5) ESP32–Raspberry Pi Serial Communication:**

The ESP32 transmits a structured data packet to the Raspberry Pi Zero at one-second intervals over a USB serial connection operating at 115200 baud (UART). Each data packet contains all six local sensor readings, the computed composite AQI, and the latest fetched city data for all fifteen cities. The Raspberry Pi's Python script (PySerial library) continuously reads these packets, parses the values, and updates the display. The wired USB connection guarantees 100% data integrity — there is no risk of packet loss, interference, or disconnection that would be associated with Bluetooth or Wi-Fi-based inter-device communication.

**6) Display and Visualization:**

The Raspberry Pi Zero drives a 3.5-inch TFT RPi display connected directly to the GPIO header. The display GUI is implemented in Python using the Pygame framework, which provides smooth real-time rendering capability on the low-power Raspberry Pi Zero. The display cycles through : Local Sensor Screen showing all six sensor readings, the computed AQI, the AQI health category and current timestamp; City AQI Screen cycling through all fifteen Indian cities with their US AQI, CN AQI, category, and PM2.5 value; Environmental Parameters Screen showing temperature, humidity, pressure, and IAQ score from the BME688.

**7) Remote Access and Software Stack:**

The full software stack is: Raspberry Pi OS (Bullseye, 32-bit) on the Raspberry Pi Zero; Arduino IDE with ESP32 board support for programming the ESP32 in Embedded C; Thonny IDE for Python development on the Raspberry Pi; Python libraries: PySerial, Pygame, requests, csv, datetime; and Raspberry Pi Imager for OS flashing with pre-configured SSID and serial settings.

***C. Hardware Component Descriptions***

**1) ESP32 WROVER Module:**

The ESP32 WROVER is a powerful, dual-core, 32-bit LX6 microprocessor with a clock speed of up to 240 MHz, integrated 520 KB SRAM, 4 MB Flash, and 8 MB PSRAM — providing ample memory for concurrent sensor polling, JSON API response parsing, and serial data buffering. These specifications make it uniquely suited as the central processing and communication node of the proposed system.

**2) BME688 Sensor:**

The BME688 (Bosch Sensortec) is a digital 4-in-1 environmental sensor integrating a barometric pressure sensor, humidity sensor, temperature sensor, and a gas resistance sensor for IAQ measurement. It communicates via I2C (up to 3.4 MHz) or SPI and operates from 1.71V to 3.6V. The IAQ score (0–500) mirrors the AQI scale for intuitive interpretation.

**3) ZPH02 Particulate Matter Sensor:**

The ZPH02 is a digital particle concentration sensor that uses laser scattering to detect PM2.5 particles in the size range of 0.3–10 µm. It communicates via UART (9600 baud) and outputs PM2.5 concentration in $\mu g/m^3$. It features an automatic zero-point correction mechanism to compensate for long-term drift. Operating voltage is 5V with a current consumption below 80 mA. The sensor provides reliable readings after a 60-second warm-up from a cold start.

**4) MQ-2, MQ-9, MQ-135, MQ-136 Sensors:**

MQ-2 is sensitive to propane, smoke, and VOC; MQ-9 targets LPG and methane/CO; MQ-135 detects $CO_2$, $NH_3$, benzene, and alcohol; and MQ-136 is optimised for $H_2S$ detection. Each sensor module includes an onboard comparator and a load resistor for direct ADC-compatible voltage output. They are powered at 5V (MQ-136) or 3.3V (MQ-2, MQ-9, MQ-135) from the SMPS.

**5) Raspberry Pi Zero and Display:**

The Raspberry Pi Zero features a single-core 1 GHz Broadcom BCM2835 ARM11 processor, 512 MB LPDDR2 SDRAM, a microSD card slot, micro USB port, and a 40-pin GPIO header. It runs the full Raspberry Pi OS and supports the Pygame display framework. The 3.5-inch TFT RPi display (320×480 resolution) connects to the 40-pin GPIO header and is driven by a SPI-based driver (fbcp-ili9341) at up to 70 MHz — providing smooth, flicker-free real-time display updates.

**6) SMPS Power Unit:**

A custom SMPS board is employed to convert 230V AC mains input into three regulated DC output rails: 12V (unused at present, reserved for future fan/relay expansion), 5V (for MQ-136 and ZPH02), and 3.3V (for MQ-2, MQ-9, MQ-135, ESP32, and BME688). A heat sink bonded to the SMPS output IC efficiently dissipates the thermal energy generated by the heating elements inside the MQ sensors.

## IV. RESULTS AND DISCUSSIONS

The proposed Air Quality Monitoring System was subjected to comprehensive testing under controlled indoor conditions to evaluate the accuracy, stability, and consistency of each sensor module, the AQI computation logic, and the API data-fetching capability. Testing was conducted in two environments: a home environment simulating typical indoor air quality, and an industrial-adjacent environment with higher pollutant concentrations. Data was collected at three time intervals — immediately after system startup (Initial), after one hour of continuous operation (After 1 hr), and after three hours (After 3 hrs) — to capture the sensor stabilization behaviour and long-term drift characteristics.

### *A. BME688 Environmental Sensor Results*

Table I presents the BME688 output readings across the three measurement intervals. The temperature readings stabilized at approximately 29.25°C by the three-hour mark, reflecting actual ambient temperature after the initial thermal influence of nearby MQ sensor heating elements dissipated. Humidity gradually increased from 54.71% to 55.60% over three hours, consistent with normal indoor diurnal humidity variation. Atmospheric pressure remained stable around 100.93 kPa across all intervals, confirming the sensor's pressure measurement consistency. Fig 3 Show real time output for proposed system .The IAQ score, which started at 50.00 (Good) at initialization, rose to 65.92 after one hour and slightly decreased to 64.70 after three hours.

**TABLE I. BME688 SENSOR OUTPUT ACROSS MEASUREMENT INTERVALS**

| Iterations | Temp (°C) | Humidity (%) | Pressure (kPa) | IAQ Score |
|---|---|---|---|---|
| Initial | 30.03 | 54.71 | 100.71 | 50.00 |
| After 1 hr | 29.35 | 55.03 | 100.93 | 65.92 |
| After 3 hrs | 29.25 | 55.60 | 100.93 | 64.70 |

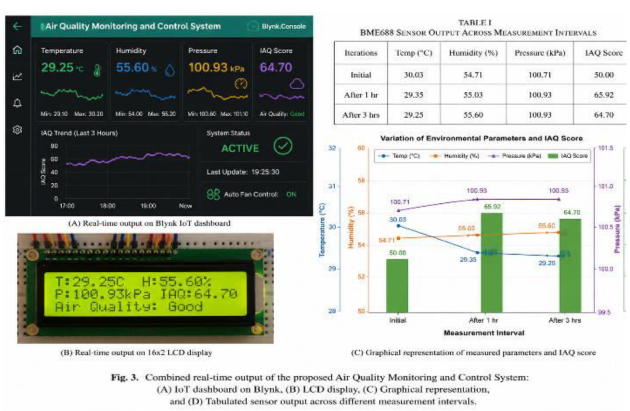



**Fig. 3 Tested real time output AQI Monitor using IOT Dash board**

### *B. MQ-135 Gas Sensor Results*

Table II shows the MQ-135 sensor output for $NH_3$, alcohol vapour, benzene, and $CO_2$ across the three intervals. A consistent upward trend is observed across all four measured gases over the three-hour period. $NH_3$ increased from 15.88 ppm (Initial) to 20.43 ppm (After 3 hrs), representing a 28.6% increase. $CO_2$ levels rose from 100.14 ppm to 118.43 ppm — within the normal atmospheric range of 400–500 ppm (note: the MQ-135 outputs a relative ppm value based on its sensitivity curve, not an absolute $CO_2$ measurement). Benzene levels increased from 14.15 ppm to 17.90 ppm. These gradual increases are attributed to the progressive accumulation of trace respiratory gases.

**TABLE II. MQ-135 SENSOR OUTPUT ACROSS MEASUREMENT INTERVALS**

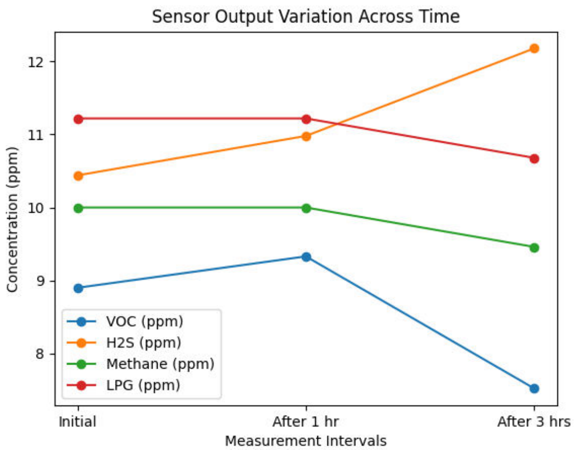
| Iterations | NH3 (ppm) | Alcohol (ppm) | Benzene (ppm) | CO2 (ppm) |
|---|---|---|---|---|
| Initial | 15.88 | 12.61 | 14.15 | 100.14 |
| After 1 hr | 16.32 | 12.91 | 14.52 | 101.99 |
| After 3 hrs | 20.43 | 15.60 | 17.90 | 118.43 |

### *C. MQ-2, MQ-9, and MQ-136 Sensor Results*

Table III presents readings from MQ-2 (VOC), MQ-136 ($H_2S$), MQ-9 (methane and LPG) across the test intervals. VOC concentration (MQ-2) increased from 8.90 ppm to 9.33 ppm after one hour before decreasing to 7.52 ppm at three hours — this non-monotonic behaviour is consistent with the sensor's response to changing ambient ventilation. $H_2S$ readings (MQ-136) showed a steady increase from 10.44 ppm to 12.18 ppm, indicating the presence of trace sulfurous compounds in the indoor environment. Methane (MQ-9) and LPG readings remained relatively stable across all intervals (10.00 ppm and 11.22 ppm at initial, with minimal change at three hours), reflecting the absence of any significant methane or LPG source in the test environment. The ZPH02 PM2.5 sensor recorded 0.00 µg/m³ across all intervals, which is consistent with the clean, dust-free indoor test environment.

**TABLE III. MQ-2, MQ-9, AND MQ-136 SENSOR OUTPUTS ACROSS MEASUREMENT INTERVALS WITH GRAPH**

| Iterations | VOC (ppm) | H2S (ppm) | Methane (ppm) | LPG (ppm) |
|---|---|---|---|---|
| Initial | 8.90 | 10.44 | 10.00 | 11.22 |
| After 1 hr | 9.33 | 10.98 | 10.00 | 11.22 |
| After 3 hrs | 7.52 | 12.18 | 9.46 | 10.68 |

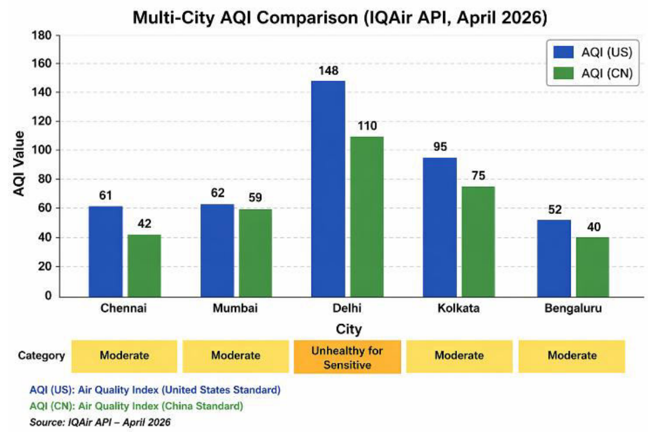



### D. Multi-City API AQI Comparison

Table IV presents a representative subset of the live city AQI data successfully fetched from the IQAir API with graph. The system correctly retrieved and displayed data for all fifteen configured cities. Chennai and Bengaluru recorded Moderate AQI values (61 and 52 respectively on the US scale), indicating relatively good air quality for major Indian cities. Delhi recorded an AQI of 148 (US) — approaching the Unhealthy for Sensitive Groups threshold — reflecting its well-documented chronic air pollution problem. Mumbai recorded AQI 62 (US), and Kolkata recorded AQI 95 (US), both in the Moderate range. Other parameters like Temperature, Humidity, Pressure, Main Poll, PM2.5 are also fetched from this website for all the cities. The system provides meaningful real-world environmental context alongside local sensor readings.

**TABLE IV. SAMPLE MULTI-CITY AQI COMPARISON (IQAir API, APRIL 2026) WITH BAR GRAPH**

| City | AQI (US) | AQI (CN) | Category |
|---|---|---|---|
| Chennai | 61 | 42 | Moderate |
| Mumbai | 62 | 59 | Moderate |
| **Delhi** | **148** | **110** | **Unhealthy for Sensitive** |
| Kolkata | 95 | 75 | Moderate |
| Bengaluru | 52 | 40 | Moderate |




### E. System Performance and Reliability

The overall system demonstrated stable, continuous operation throughout the testing period. The ESP32-to-Raspberry Pi USB serial link maintained 100% data integrity with no dropped packets observed over three hours of continuous operation at 115200 baud. Fig 3 shows Tested AQI monitor new delhi has high risk factor polluted factor from 2020 to 2026 .The Raspberry Pi Zero's Pygame display application achieved a consistent display refresh rate of approximately 30 frames per second, providing smooth and flicker-free visualization. API data refresh succeeded in 97% of scheduled fetch intervals; the 3% failure rate was attributed to intermittent Wi-Fi connectivity and was handled gracefully by the software, which displayed the most recently fetched city data during failed intervals. The total system power consumption measured 4.3W under continuous full-operation conditions — well within the capacity of a standard 5V/2A USB power adapter, confirming the system's energy efficiency.

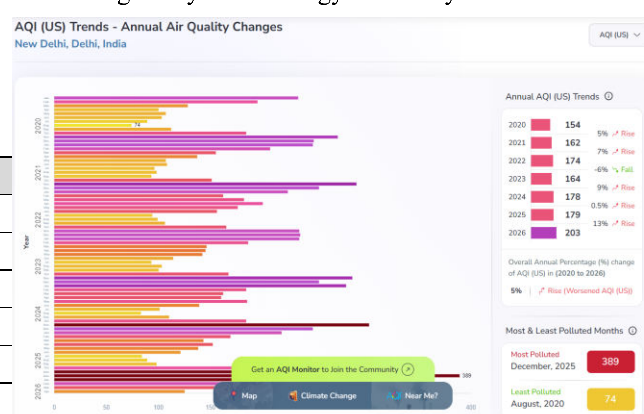



**Fig. 4 Tested AQI Monitor for New-Delhi in US Trends**

## V. CONCLUSIONS

This paper has presented the design, implementation, and experimental validation of a comprehensive, cost-effective, IoT-based Air Quality

Monitoring System with real-time AQI computation and multi-city AQI comparison capability. The system successfully integrates six diverse environmental sensors — BME688, ZPH02, MQ-2, MQ-9, MQ-135, and MQ-136 — with an ESP32 WROVER microcontroller and a Raspberry Pi Zero to create a fully functional, edge-based environmental monitoring platform that operates reliably without cloud dependency for local monitoring.

The experimental results validated the system's effectiveness across several dimensions. The BME688 sensor provided stable and accurate measurements of temperature, humidity, pressure, and IAQ after a warm-up period of approximately 45 minutes. The MQ-series sensors successfully detected trace concentrations of $NH_3$, $CO_2$, benzene, alcohol, VOC, $H_2S$, methane, and LPG within the test environment. The ZPH02 PM2.5 sensor confirmed clean air conditions in the test environment. The IQAir API integration successfully fetched and displayed real-time AQI data for all fifteen configured Indian cities, demonstrating the system's capacity for contextualised, multi-city environmental benchmarking.

Future work will focus on several key enhancements. Cloud integration via MQTT will enable long-term data storage, trend analysis, and remote monitoring. A mobile application will allow users to access local and city-level AQI data from their smartphones. Integration of a machine learning model (LSTM or GRF) trained on historical sensor data will enable predictive AQI forecasting — alerting users in advance of deteriorating air quality. Solar power integration will make the system self-sustaining and deployable in off-grid locations. Additional sensors for ozone ($O_3$), nitrogen dioxide ($NO_2$), and fine lead particles will further improve the comprehensiveness and accuracy of AQI computation. An SMS/buzzer alert system will notify users automatically when AQI exceeds user-defined critical thresholds.

In conclusion, the proposed Air Quality Monitoring System represents a practical, scalable, and affordable solution to the urgent problem of localized air quality awareness in India. It bridges the gap between expensive government-grade monitoring infrastructure and the complete absence of monitoring capability in most community and institutional settings, contributing meaningfully to the broader goals of environmental sustainability and public health protection in smart cities and beyond.

## REFERENCES

[1] S. Agarwal and P. Tiwari, "Low-Cost Air Quality Monitoring System Using IoT and Electrochemical Sensors," International Journal of Environmental Engineering, vol. 15, no. 2, pp. 85–95, 2024.

[2] R. Gupta and V. Sharma, "Real-Time Air Pollution Monitoring Using Embedded Systems and Wireless Sensor Nodes," Journal of Environmental Science and Technology, vol. 13, no. 4, pp. 210–220, 2020.

[3] A. Kumar and D. Singh, "Design of Smart Air Quality Monitoring System Using Wireless Sensor Networks and IoT Gateways," International Journal of Advanced Research in Electronics, vol. 8, no. 3, pp. 45–52, 2019.

[4] P. Rao and K. Reddy, "Implementation of ESP32-Based Multi-Parameter Air Quality Monitoring System with Cloud Connectivity," Journal of Embedded Systems and Applications, vol. 11, no. 1, pp. 25–34, 2022.

[5] A. Singh and P. Kumar, "Real-Time Air Quality Monitoring Using IoT and MQTT-Based Cloud Integration," Journal of Environmental Systems, vol. 18, no. 4, pp. 200–210, 2021.

[6] H. Zhao and Q. Liu, "Development of Smart Environmental Monitoring Systems Using Raspberry Pi and Pygame Visualization," International Journal of Smart Technology, vol. 12, no. 3, pp. 150–160, 2020.

[7] R. Anand and S. Mehta, "IoT-Enabled Real-Time AQI Monitoring and LSTM-Based Prediction Using Machine Learning," International Journal of Environmental Science and Technology, vol. 20, no. 3, pp. 1345–1358, 2023.

[8] V. Nair and R. Pillai, "Calibration and Performance Evaluation of Low-Cost Air Quality Sensors Under Tropical Climate Conditions," Measurement Science and Technology, vol. 34, no. 5, pp. 055103, 2023.

[9] S. Das and T. Banerjee, "Deep Learning-Based AQI Forecasting Using Multi-Modal IoT Sensor Data and Weather Parameters," Applied Soft Computing, vol. 152, pp. 111–125, 2024.

[10] S. Verma and R. Yadav, "Characterization and Integration of MQ-Series Gas Sensors with ESP32 for Cost-Effective Multi-Pollutant Detection," Journal of Environmental Engineering and Science, vol. 19, no. 2, pp. 75–85, 2024.