CHEMICAL ENGINEERING TRANSACTIONS
VOL. 127, 2026
A publication of AIDIC
The Italian Association of Chemical Engineering
Online at www.cetjournal.it
Guest Editors: Laura Capelli, Marzio Invernizzi, Selena Sironi

ISBN 979-12-81206-28-1; **ISSN** 2283-9216
DOI: 10.3303/CET26127014

Development of a Fully Configurable IoT Platform for a Low-Cost Air Quality Monitoring Network

Andy Blanco-Rodriguez[a,*], Derick Mazelli Barbosa[b], Lean Lopes[b], Rodolfo Valiente Romero[c], Débora Souza Alvim[d]

[a]Department of Materials Engineering (DEMAR), University of São Paulo (USP), Lorena School of Engineering (EEL), 12612-550 Lorena-SP, Brazil.
[b]Undergraduate program of Physics Engineering, University of São Paulo (USP), Lorena School of Engineering (EEL), Lorena-SP, Brazil.
[c]University of Central Florida, Department of Electrical and Computer Engineering, Orlando, FL, 32816, USA.
[d]Department of Basic Sciences and Environmental Engineering (DEBAS), University of São Paulo (USP), Lorena School of Engineering (EEL), 12602-810, Lorena-SP, Brazil.
andy.blanco@usp.br

This paper presents a dedicated IoT platform for a low-cost air quality monitoring network. The entire application framework was developed specifically for this purpose and built using open-source software. The system was named Argus, with the initial version v 0.0.1, released in this study. It provides real-time monitoring, processing, storage and visualisation of data for multiple measurement nodes. This can include air quality monitors, electronic noses, or any sensor integrated into the network that complies with established protocols. The network was designed for the operation of several digital subsystems, running on a web server. For this structure, a hybrid architecture was developed, combining a Raspberry Pi 3 model B and a centralised virtual server. This methodology allows the use of the institution's existing network infrastructure, providing security, stability, and reliability. At the same time, this approach employs a low-cost network board as a web server, offering flexibility, operating with open-source software, and easy access. On the platform, three distinct measurement devices were evaluated: a low-cost air quality monitor, a DHT sensor, and a computer application that emulates a sensor. As a result, a fully configurable, upgradeable, flexible and expandable IoT digital platform was developed in terms of services, hardware and software applications.

1. Introduction

Atmospheric pollution and its negative impacts have been manifesting for decades worldwide. Air quality degradation leads to reduced human health, environmental deterioration, and economic losses (World Health Organization, 2021). In fact, almost 99% of the world's population breathes air that exceeds the limits established by the WHO (World Health Organization, 2023). In this scenario, air quality monitoring coverage is still low and insufficient (Campo et al., 2023). This constitutes a significant lack, since pollutant concentrations usually disperse with high spatio-temporal variability. To tackle this issue, ubiquitous sensing devices and low-cost monitors have emerged as an engaging option. Therefore, to enable these applications to work together in an integrated, responsive, decentralised, and mass-scale framework, the use of an IoT platform is required.
Low-cost sensor-based networks are a feasible solution to increase the spatio-temporal resolution of air quality monitoring systems (Buelvas et al.,2023). This type of instrument enables the expansion of monitoring networks, mainly due to its low cost, small size, and ease of connection with other devices. These systems are increasingly being used to measure pollution levels and are attractive to the air quality community. Low-cost IoT platforms with gas sensors enable the expansion and popularisation of air pollutant monitoring, mainly due to their versatility, ease of access, and low cost (Arroyo et al., 2016; Campo et al., 2023; González et al., 2024; Prudenza et al., 2024). These features represent an improvement compared to conventional air quality stations, commercial monitors, and satellite monitoring. Despite providing metrological precision via advanced instruments like Gas Chromatography, conventional reference stations incur substantial costs and demand

Paper Received: 7 April 2026; Revised: 24 May 2026; Accepted: 6 June 2026
Please cite this article as: Blanco-Rodríguez A., Mazelli Barbosa D., Lopes L., Valiente Romero R., Souza Alvim D., 2026, Development of a Fully Configurable IoT Platform for a Low-Cost Air Quality Monitoring Network, Chemical Engineering Transactions, 127, 79-84 DOI:10.3303/CET26127014

specialized maintenance, restricting their spatial density (Concas et al., 2020; Kang et al., 2022, Da Costa et al., 2024). Many commercial measuring instruments have low availability of raw measurement data, resulting in "black box" information and limited user enhancement capabilities (De Vito et al., 2020). Remote sensing is useful for estimating the global concentration of pollutants, but satellite measurements have limitations in spatial and temporal resolution (Follette-Cook et al., 2020).

Regarding IoT platforms, there are some open-source communities (e.g. Adafruit IO, EMQX) that allow the use of basic dashboard widgets, enabling proof-of-concept testing. However, they are quite limited, restricted to the free-tier constraints and with low expansion capabilities. In addition, selecting the ideal IoT cloud platform is already a crucial decision, as it presents a significant challenge due to the heterogeneity of features, pricing models, and architectural details (Panagou et al., 2025). IoT networks particularly designed for air quality monitoring have been developed, but their use remains limited, with low standardisation and maintenance gaps. This study presents a complete and fully configurable IoT platform for a low-cost air quality monitoring network. The entire software framework was developed specifically for this application, and it is based on open-source software. The system includes instant data processing, device management, security communication, diverse connectivity protocols, visualisation tools and scalability. The main goal of this approach was the development of a network like Argus, with a hybrid server architecture designed for the integrated, responsive, and real-time processing of multiple input data. While selecting an ideal IoT cloud platform is already a crucial challenge due to heterogeneous features and architectural details, conventional open-source platforms often experience database bottlenecks during high-volume data ingestion. Argus addresses this issue by balancing write speed and analytical query performance. Real-time ingestion of JSON payloads via MQTT occurs directly into a primary table (Table Real-time data). Simultaneously, a batch processing routine calculates historical metrics in 10-minute windows, storing them in a Table called Historical-time data. This structural separation allows the creation of complex, real-time dashboards through hybrid queries without compromising performance.

2. Materials and methods

This section presents the setup features of the proposed IoT platform as well as the digital services, connections among components, and user categories. It also describes network elements in terms of software and application architecture.

2.1 Structure of the IoT platform for real-time air quality monitoring

The schematic diagrams of IoT platforms for connecting physical devices to the cloud are shown in Figure 1. With this structure, it is possible to monitor data in real time, enabling instant data ingestion, multi-protocol connectivity (MQTT, HTTP), security (encryption, authentication) and device management. This network comprises various digital subsystems running on a server, IoT client devices, and communication among the platform elements. Figures 1a and 1b represent, synthetically, a generic IoT platform and the proposed IoT platform for monitoring air quality, respectively.

In Figure 1a, a Web server includes the Database, Middleware, Broker, and Dashboard services. Regarding IoT clients, several devices can be connected to the network, such as sensors, actuators, smartphones, computers and other equipment with internet connectivity. For the connection between the server and devices, it is common to employ wireless communication standards such as IEEE 802.11 (Wi-Fi) or IEEE 802.15.4 (Zigbee, ISA100.11a, and 6LowPAN). In Figure 1b, the proposed air quality monitoring network details the software used to run the services on the Web server: MySQL, Node.js, EMQX and NGINX as a Reverse proxy.

As physical infrastructure for an IoT server, various classes of computers can be used, from common desktops and notebooks to supercomputers and dedicated servers. Other alternatives with sufficient performance and lower cost are the Raspberry Pi and BeagleBone boards. The Web server can even be installed in the cloud as a Virtual Private Server (VPS) using providers such as Amazon or Google. In this study, a hybrid architecture was developed, combining a Raspberry Pi 3 model B and a centralised virtual server of the university.

To ensure the system functions as a fully open-source and reproducible platform, the EMQX MQTT Broker (version 5.1.0) was employed. The backend was developed in Node.js, with Express, node-cron, and mqtt.js libraries, interacting with a MySQL database (version 8.0). MySQL was specifically selected due to its native operational support for spatial data (POINT) and JSON payload structures. In addition, to provide modular scalability and prevent dependency conflicts, the infrastructure was entirely isolated in containers using Docker and Docker Compose, securely configured within a shared network named argus-shared-net.

A wide range of applications and programs can be installed on a server to manage IoT system services (Figure 1a). The database is a crucial element, as it stores data generated by IoT clients. For example, time series from sensor outputs, user identification and access control, and possible relationships between device data. Usually, information in the database is organised into tables, and software such as MariaDB, MySQL, MongoDB, and

Microsoft SQL Server can be executed. In this proposal, MySQL was used, which is one of the most popular open-source relational databases (Figure 1b).
Another common component in servers is Middleware, useful for detecting measurement limits, generating alarms, sending warnings, etc. Through this subsystem, it is possible to establish comparisons between the information in the database and, consequently, issue responses via the broker. Furthermore, IoT middleware serves as a mediator to enable data processing, device management, and interoperability across different platforms. It acts as a bridge that hides hardware complexity and ensures effective communication (MQTT, HTTP, etc). One of the most widely used software programs for establishing rules is Node.js, which is a cross-platform runtime environment that allows the execution of JavaScript code outside of a web browser (Figure 1b). In this approach, two Node.js applications (microservices) are running to manage the demands of: EMQX (from administrator clients, “Admin clients”) and NGINX (from “User clients”).
The Dashboard is also an essential service on web servers, because it acts like a visual interface between users and connected physical devices. These applications allow both the visualisation of data and graphs, as well as the execution of control commands. Web application development involves working with both front-end and back-end. The first one is responsible for visual presentation, organisation, graphic design, and user experience. There are some programming tools for this purpose, like HTML, CSS, PHP and JavaScript. The back-end, however, is related to the logic of interaction between the web application's presentation and the database, also handling responses to user requests. JavaScript is a common language for implementing these functions due to its versatility and robustness. In this survey, the JavaScript code programmed to run the dashboard is represented within the Node.js application block (Figure 1b).

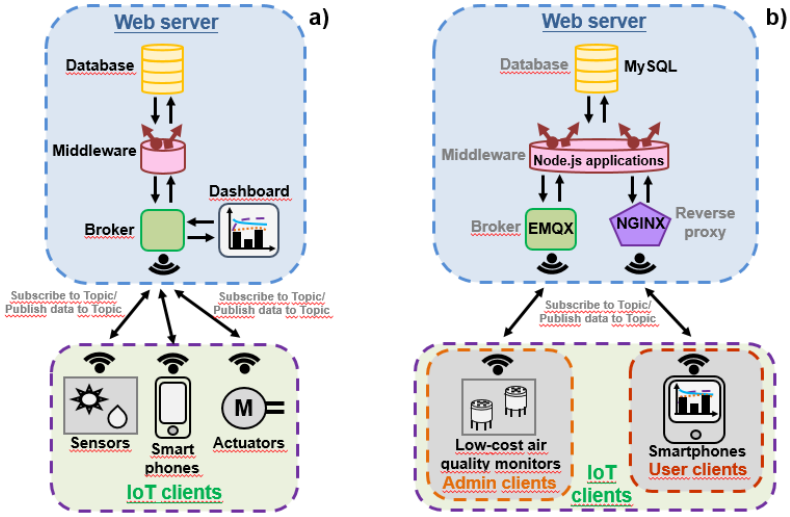

Figure 1: IoT platforms for real-time data monitoring, processing, storage and visualisation: a) Generic diagram of digital services and clients; b) Simplified diagram of the proposed air quality monitoring network

The Broker is an indispensable component of the IoT platform because it manages the flow of data among subsystems. The Broker acts as an intermediary, efficiently controlling and enabling the transmission/reception of information across the elements of the system. For this service, communication via MQTT (Message Queue Telemetry Transport) is recommended. It is a lightweight, asynchronous data transmission protocol based on the publish/subscribe paradigm, originally developed by IBM for telemetry applications. MQTT is message-oriented, publishing data to a specific topic and address. This way, clients subscribed to one or more topics will receive updates from other clients publishing to the same topics. For example, by creating a topic called “temperature” in the Broker, a client (a device with a temperature sensor) can publish its sensor readings to that topic. And asynchronously, another client (a device accessing the dashboard) will be able to view this information in real time. Three of the brokers available for developing IoT systems are EMQX, Mosca and Mosquitto. EMQX was selected for this application, Figure 1b, because it is a high-performance, designed for massive IoT connectivity, offering high scalability, low-latency processing, and robust security features.
In Figure 1b, a Reverse proxy with NGINX was also illustrated. This subsystem allows the configuration of desired digital inputs/outputs and helps with network security. In addition, it enhances the performance and scalability of web applications by managing client requests before forwarding them to the back-end, and protecting the server’s IP addresses. In Figure 1b, IoT clients were segregated to classify them according to

their database access permissions. Basically, Admin clients are the devices that can create/write database tables, such as low-cost air quality monitors. User clients are any devices that access the proposed network and have read-only permissions in the database, however.
In this version of Argus, some platform indicators were considered, such as: total latency (the time it takes from capturing sensor data to displaying it on the dashboard); backend write time (the time between data arrival via MQTT and its writing to the database); frontend read time (the time it takes for the frontend to read data from the database and display it on the screen); number of connected devices (the number of connected devices, differentiating active from inactive); and data volume (the amount of data flowing through the platform).

2.2 Sending data from a measuring device to the Argus platform.

Figure 2 shows the flow chart for the Admin client, assuming that a measuring device is connected to the server. An air quality measuring monitor and three digital services are arranged from left to right. Also, each of these network subsystems, named in that order as “Measuring device 1”, “MQTT broker”, “Node.js middleware”, and “Database”, is represented by different colours in Figure 2.

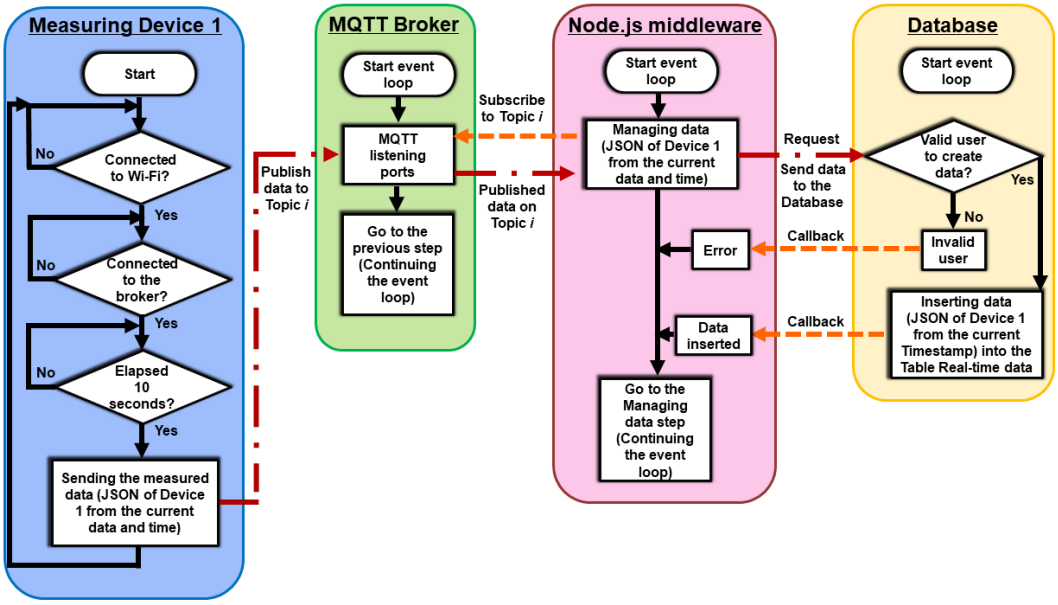

Figure 2: Flow chart for inserting data acquired by a measuring device into the IoT platform database

Regarding the low-cost monitor, its firmware runs a continuous loop, in which the Wi-Fi and broker connections are checked sequentially. Additionally, a microcontroller hardware interrupt has been programmed to acquire data. If all of these three conditions are satisfied, the instrument communicates via Wi-Fi with the broker, which runs on the web server. This wireless connection is based on the MQTT protocol, and the device sends the output data from the sensors in a single JSON package (referred to as “JSON of Device 1 from the current data and time”). This dataset contains each of the measured variables paired with data and time. Then, these values are published to the corresponding Topics. The MQTT broker handles the message among all clients who have previously subscribed to the created topics. In this case, Figure 2 shows that the measuring device is located on the left side of the broker and the Node.js middleware on the right side. The broker was designed to operate in an event loop, checking and managing the flow of information asynchronously. Next, messages using the MQTT protocol are also sent and received between the MQTT broker and the Node.js middleware. This service, which also runs as an event loop, is the entity that controls the creation of new data in the Database, among other functions. This process can only be executed by authorised users, i.e., Admin clients. For this reason, user verification was implemented between Node.js and the Database (Request signal). Within this service, user validation actually takes place. If approved, the data initially generated (and sent to the cloud) by the air quality monitor will be inserted into the Database, particularly into the table named “Table Real-time data”. In this data organisation, the time and date of time series variables will be assigned according to Brasília time, as obtained by the server via the internet (Timestamp). In addition, the Database returns to Node.js that the measured data was successfully stored (Callback signal). Conversely, if user validation fails, the Database notifies Node.js of this event (also via Callback signal), and this service flags it (Error). Therefore, regardless of

the Database response, middleware will continuously execute the event loop. In summary, this process goes from the device's reading to inserting that information into the database.

2.3 Prototypes of measuring devices connected to the IoT platform

To fully assess the performance of all Argus platform services, three distinct measuring devices were connected to the network. Table 1 summarises these instruments, which are a low-cost air quality monitor, a DHT22 sensor, and a data simulator programmed in Python that runs on a computer.

Table 1: Prototype devices for assessing the platform's functionality and performance.

Measuring device	Sensors
Low-cost air quality monitor	MOS-based gas sensors: MQ2, MQ3, MQ9 and M135
DHT22	Temperature and humidity
Simulated device	Data generated by simulation, such as temperature and humidity variables

3. Results and discussion

The results achieved with this approach are mainly shown in terms of: total latency, backend write time, frontend read time, number of connected devices and data volume of network services. In practice, it was observed, through the continuous execution of each network subsystem and the network as a whole, which the aforementioned indicators tend to be adequate. That is, low values for temporal metrics (ms) and high capacity to operate with high data volumes (8 devices sending data simultaneously every 1 ms).

3.1 Data visualisation on the web application dashboard

The data acquired by the measuring instruments can be accessed through the web application's dashboard from any device with a browser connected to the internet (Figure 3).

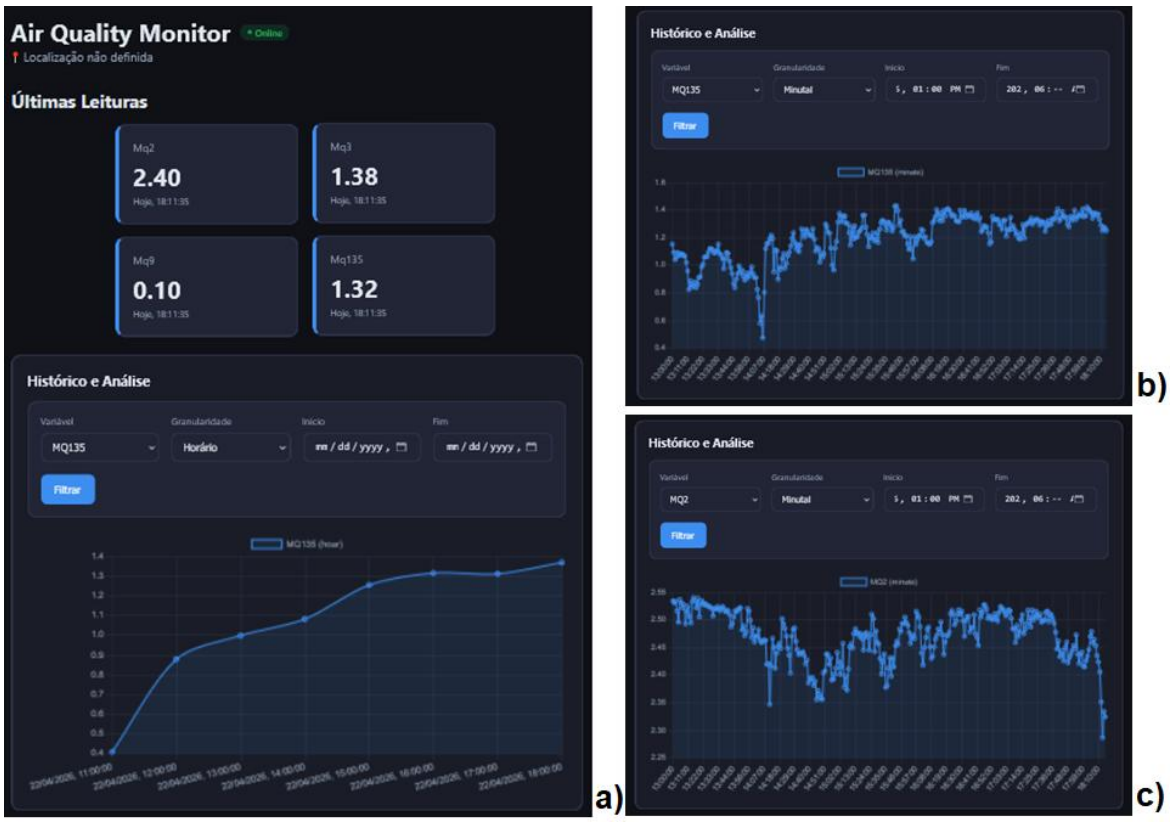

Figure 3: Example of data visualisation on the dashboard for a "Air Quality Monitor" device: a) Sections for current data acquired ("Últimas Leituras") and the time series from the MQ135 sensor ("Histórico e Análise") with hourly granularity ("Horário"); b) Detailed view of the section of the time series from the MQ135 sensor with minute granularity ("Minutal"); and c) Same as b) but for the MQ2 sensor

An example of an “Air Quality Monitor” device was shown in Figure 3. Next, the most recent measurements are displayed directly, as these are the four variables acquired by the measuring device. In the example in Figure 3a, the numerical values observed in the upper indicators are: MQ2 = 2.40, MQ3 = 1.38, MQ9 = 0.10 and MQ135 = 1.32 volts. At the bottom of the same dashboard screen (Section “Histórico e Análise”), the time series for MQ135 is also displayed. To access this data, the user must request the following configuration from the database: “Variável” -> “MQ135” and “Granularidade” -> “Horário”. Figures 3b and 3c specifically show the graphical results for the same variable with a different granularity (“Minutal”) and for the MQ2 sensor with “Minutal” granularity, respectively.
The graphical results in Figure 3 practically illustrate the adequate and consistent performance of the Argus platforms. This demonstrates the coordinated and integrated performance of all network components. For example, the device continuously published JSON telemetry payloads to evaluate the ingestion capacity of the Node.js backend. While the Raspberry Pi hardware operated naturally as a server, the system successfully managed the data flow without service interruption.

4. Conclusions

In the current study, the Argus v 0.0.1 platform for application in air quality monitoring was presented and analysed. All network services, configurations, communications between subsystems, and device operation were fully designed, implemented and assessed. So, the structural viability of Argus was successfully established during its initial development phase by two hardware structures for the web server (hybrid architecture). The platform performs instantaneous data processing, operating with real-time requests and responses. The deployment of a Dockerized infrastructure, coupled with EMQX and Node.js, enabled stable, scalable and continuous data processing. Argus also supports various connectivity protocols, such as MQTT and HTTP, and multiple programming languages like JavaScript and Python, which communicate coherently throughout the network. This project provides an interactive dashboard for visualising the information. In future work, detailed network performance indicators will also be computed and recorded.

References

Arroyo P., Lozano J., Suárez J.I., Herrero J.L., Carmona P., 2016, Wireless Sensor Network for Air Quality Monitoring and Control, Chemical Engineering Transactions, 54, 217-222.

Buelvas J.H., Múnera D., Gaviria N., 2023, DQ-MAN: A tool for multi-dimensional data quality analysis in IoT-based air quality monitoring systems, Internet of Things, 22.

Campo F., Franco D., Campos Santos F., Blanco-Rodriguez A., Garcia-Ramirez A.R., Ratão G., Hoinaski L., 2023, CLEAN - Collaborative low-cost environmental and air-quality network, Environmental Modelling and Software, 163.

Concas F., Mineraud J., Lagerspetz E., Varjonen S., Liu X., Puolamäki K., Murmi P., Tarkoma S., 2021, Low-cost outdoor air quality monitoring and sensor calibration: A survey and critical analysis, ACM Transactions on Sensor Networks (TOSN), 17(2), 1-44.

Da Costa, A. Z., Aniceto, J. P. S., & Lopes, M., 2024. Low-Cost Sensor Network for Air Quality Assessment in Cabo Verde Islands. Sensors, 24(23), 7656.

De Vito, S., Esposito E., Castell N., Schneider P., Bartonova A., 2020, On the robustness of field calibration for smart air quality monitors, Sensors and Actuators B: Chemical, 310.

Follette-Cook M., PRADOS A., GUPTA P., 2020, Measuring Nitrogen Dioxide from Space. An Inside Look at how NASA Measures Air Pollution.

González V., Godoy J., Meléndez F., Arroyo P., Díaz F., López Á., Suárez J.I., Santos J.P., Lozano J., 2024, Sniffing Smartwatch for Online Monitoring of Selected Odorants, Chemical Engineering Transactions, 112, 97-102.

Kang Y., Aye L., Ngo T.D., Zhou J., 2022, Performance evaluation of low-cost air quality sensors: A review, Science of The Total Environment, 818.

Panagou I.C., Katsoulis S., Nannos E., Zantalis F., Koulouras G., 2023, A comprehensive evaluation of IoT cloud platforms: A feature-driven review with a decision-making tool, Sensors (Basel, Switzerland), 25, 1-58.

Prudenza S., Bax C., Capelli L., 2024, Real-time Monitoring of Odour Concentration at the Inlet of an Abatement System by IOMS, Chemical Engineering Transactions, 112, 139-144.

World Health Organization, 2021. WHO Global Air Quality Guidelines: Particulate Matter (PM2.5 and PM10), Ozone, Nitrogen Dioxide, Sulfur Dioxide and Carbon Monoxide. https://www.who.int/publications/i/item/9789240034228. (Accessed on March 19, 2026).

World Health Organization, 2023. Air Pollution. https://www.who.int/health -topics/air-pollution#tab=tab_1. (Accessed on March 11, 2026).