Engineering Innovations
ISSN: 2813-1002, Vol. 16, pp 75-95
doi:10.4028/p-J80vYP

Submitted: 2025-07-22
Revised: 2025-08-13
Accepted: 2025-08-20
Online: 2026-01-07

METEORO: Low-Cost IoT Weather Station

Douglas Adalberto Aguilar[1,a*], Max Gabriel Sánchez Flores[1,b*], Jhoan Daniel Martínez Marroquín[1,c*] and Oscar Eduardo Díaz Gómez[1,d*]

[1]Universidad Evangélica de El Salvador

[a]douglas.aguilar@uees.edu.sv, [b]maxg.sanchezf@gmail.com, [c]jh3027d@gmail.com, [d*]oskydiaz845@gmail.com

Keywords: IoT, meteorology, climate monitoring, ESP32, microcontrollers, weather station, low-cost, website.

Abstract. The development of low-cost, autonomous scientific instrumentation is crucial for enhancing environmental education and research, particularly in settings with limited resources. This paper presents METEORO v2, a low-cost Internet of Things (IoT) weather station designed to address the need for accessible scientific instrumentation in educational settings. The system overcomes common durability and energy autonomy challenges by integrating a robust physical structure with an optimized power management system. Architected around an ESP32 microcontroller utilizing a deep-sleep strategy and a 6W solar panel, the station provides continuous monitoring of precipitation, wind speed, UV radiation, air quality, temperature, humidity, and pressure. Data is transmitted via Wi-Fi to a Firebase cloud database and visualized through a responsive web interface. This work provides a replicable and affordable model for a sensing platform that serves as both an effective pedagogical tool and a reliable scientific instrument.

Introduction

The METEORO v2 project was conceived as an evolution of an earlier version (METEORO v1) that performed basic climate monitoring using low-cost sensors. The initial system was capable of measuring temperature, humidity, and atmospheric pressure, and served as a functional prototype for environmental data collection. However, it presented several limitations, particularly in terms of energy consumption, environmental durability, and the range of measured variables.

In a context of constant climatic variability, having an autonomous and reasonably accurate monitoring system can support environmental observation efforts within educational settings. This updated version expands the system's capabilities by adding measurements of wind speed, precipitation, UV radiation, and air quality, while also improving energy efficiency and usability for the university community[12].

To address the limitations of the original design, METEORO v2 incorporates a 6W solar panel for improved power management. The ESP32 microcontroller is configured to operate in deep sleep mode between readings, helping to reduce energy consumption. Physical modifications have also been made to the station's structure and sensor enclosures to better withstand weather exposure. The system monitors key climate parameters and includes a communication feature that enables real-time alerts about extreme weather conditions to be sent to students and university staff.

Background

The evolution of meteorological monitoring has transitioned from manual, mechanical instruments to sophisticated electronic systems that offer high accuracy and automatic operation [1]. This progression has been significantly accelerated by the proliferation of the Internet of Things (IoT), a technological paradigm that enables the creation of low-cost, interconnected sensor networks, bridging the gap between complex environmental phenomena and tangible data [2, 3, 13]. This has led to the emergence of low-cost weather stations, which leverage affordable microcontrollers like the ESP32 and Raspberry Pi to create functional monitoring systems for educational and research purposes [4, 5, 7].

In the face of increasing climatic variability and extreme weather events, it is essential to rethink how environmental data is gathered and interpreted, especially within educational contexts. Traditional large-scale meteorological networks often lack the spatial precision needed to detect localized microclimates, an aspect of particular relevance in a university setting where specific research and learning activities may depend on site-specific conditions. Commercially available weather stations, while accurate, often lack the flexibility needed to adapt to the specific requirements of an educational or research setting. To address this problem, specialized weather stations have become indispensable tools for gathering high-resolution, in-situ meteorological data [8].

The application of these IoT-based stations is broad, ranging from microclimate monitoring on university campuses to enhancing smart agriculture and providing data for building energy consumption models [16, 19]. Such systems empower students and researchers to collect, analyze, and visualize real-time data on variables like temperature, humidity, and atmospheric pressure through accessible cloud platforms [10, 15]. The integration of IoT in educational environments, particularly in STEM fields, provides invaluable hands-on experience, allowing students to engage with engineering practices, data analysis, and problem-solving in a real-world context [9, 14]. Furthermore, the use of renewable energy sources, such as solar panels, to power these remote stations is a critical aspect of modern designs, promoting sustainability and ensuring operational autonomy in diverse locations [24, 26, 28]. While the potential of these technologies is evident, their practical implementation reveals significant challenges that must be addressed to ensure their scientific validity and long-term viability.

Problem

Many educational institutions, particularly in developing regions, face significant barriers to implementing such solutions due to the prohibitive cost of professional-grade scientific instrumentation [1]. Within this context, providing students with access to autonomous and reasonably accurate environmental monitoring systems can profoundly support observational efforts and project-based learning. This work details an updated version of a weather station designed specifically for the university community, expanding its measurement capabilities while enhancing its energy efficiency and usability as a pedagogical tool.

Despite the educational potential and widespread development of low-cost weather stations, a review of the literature reveals persistent technical challenges that limit their long-term effectiveness and academic reliability. A primary issue is the lack of durability; many do-it-yourself (DIY) designs utilize materials and enclosures that are not sufficiently robust for prolonged outdoor exposure, leading to sensor degradation and compromised data quality [6]. Furthermore, many reported systems struggle with energy autonomy, relying on inadequate solar power configurations that are insufficient for continuous, unattended operation, a critical requirement for remote monitoring tools [7, 8]. This creates a necessity for a low-cost system that addresses these fundamental engineering challenges of durability and energy independence in a single, replicable framework.

Theoretical Framework

The development of automated meteorological stations has evolved considerably with the incorporation of Internet of Things (IoT) technologies and renewable energy systems. These advances have enabled the creation of devices capable of real-time environmental monitoring with minimal human intervention, facilitating predictive analytics and allowing for rapid responses to climatic variations [26]. The combination of low-cost hardware and modern communication networks has made such systems more accessible for academic, governmental, and industrial applications.

The integration of environmental sensors with IoT platforms allows for the precise measurement of variables such as temperature, humidity, atmospheric pressure, and wind speed. Low-power communication protocols, such as LoRaWAN, optimize data transmission efficiency in remote or hard-to-reach areas [27]. This approach ensures that meteorological stations can operate autonomously while transmitting accurate, timely information to centralized databases for further processing.

Energy autonomy plays a critical role in the operation of meteorological systems, particularly in isolated environments where conventional power supply is unavailable. Solar-powered systems significantly reduce operational costs, lower the environmental impact, and ensure uninterrupted data acquisition [28]. These systems leverage photovoltaic panels and energy storage units to maintain continuous functionality, even under challenging weather conditions.

Data processing in modern weather stations goes beyond simple data logging. Cloud-based storage solutions allow historical data comparison, trend analysis, and the implementation of predictive models [27]. This capability enables decision-makers to anticipate climatic events and adapt agricultural, industrial, or safety measures accordingly. Additionally, secure transmission protocols are essential to protect the integrity and confidentiality of environmental data.

Despite these advances, challenges remain in ensuring sensor calibration accuracy, enhancing hardware durability under extreme conditions, and implementing robust cybersecurity measures [26]. Future developments in meteorological stations should prioritize modular designs, advanced encryption standards for wireless communication, and increased resilience to environmental stress factors, ensuring long-term reliability and scalability of these systems.

Methodological Framework

This study employs a multi-phase applied research and development methodology to design, build, and validate the METEORO v2 weather station. The approach is structured to systematically address the core research objectives of creating a low-cost, energy-efficient, and durable monitoring system suitable for an educational environment.

The conceptual design of METEORO v2 was guided by a framework prioritizing solutions to the documented limitations of existing low-cost weather stations. The primary design requirements were derived from a synthesis of challenges identified in the literature, including power management, data accuracy, and long-term reliability [1, 7]. The following architectural principles were established to guide the development process:

Energy-First Design: The central architectural principle was the maximization of energy autonomy. This principle dictated the selection of a microcontroller with ultra-low-power deep-sleep capabilities, such as the ESP32-WROOM, and the design of a software architecture that minimizes active time. The solar power system was sized not merely for average daily operation but to ensure resilience during extended periods of low solar irradiation.

Modularity for Educational Use: The system was designed with a modular philosophy, both in hardware and software. This ensures that individual sensors can be easily replaced, calibrated, or upgraded without requiring a complete system overhaul. This flexibility is crucial in an educational context, as it allows the station to be used as an extensible platform for various student projects and interdisciplinary research [9,14].

Data Integrity and Accessibility: A data-centric approach was adopted, leveraging a scalable, cloud-based database (Firebase) for data aggregation. This choice aligns with modern IoT architectural patterns and ensures high data availability, integrity, and easy access for the university community through a web-based interface, thereby fulfilling a key objective of making scientific data accessible and engaging [10,15].

System Durability and Reliability: The long-term operational reliability of the system was assessed via a continuous 90-day deployment in an unprotected outdoor environment, exposing it to direct sunlight, precipitation, and wind. System uptime was monitored through the continuous logging of data to the cloud database. After the test period, a qualitative physical inspection of the enclosure, solar radiation shield, and sensor housings was conducted to identify any signs of material degradation, such as water ingress, UV damage, or physical wear.

The selection of a university campus as the primary deployment and validation environment was a deliberate methodological choice. This setting provides an ideal testing ground, offering stable Wi-Fi infrastructure and an engaged user base of students to test the web interface. The design of the user-facing components was guided by the pedagogical goal of making complex environmental data accessible and understandable. The project's modular design is intended to position METEORO v2 as a platform for future student-led research, aligning with modern approaches to Science, Technology, Engineering, and Mathematics (STEM) education [14].

Proposed Structure

The development of an affordable Internet of Things (IoT) weather station can be achieved through diverse methodologies, each with its own distinct objectives and capabilities. The aim of developing the station was to build a functional and useful prototype designed for installation in the university environment in which the project was carried out.

In light of these considerations, the team opted for an agile approach, facilitating its soon implementation within the university. This strategy enabled the progression from a general concept to more precise designs, facilitating the formulation of the desired functionality to be integrated into the university context, as well as the physical design of the prototype and the internal hardware.

As illustrated in Figure 1, this approach was fundamental to guiding the project and achieving results similar to the initial proposal, albeit with slight modifications aimed at maximizing the system's efficiency. From the beginning, the goal was to design an outdoor-use structure made of a durable material that would facilitate installing the weather station on poles. This would allow the station to be placed in an elevated position for accurately measuring different environmental variables.

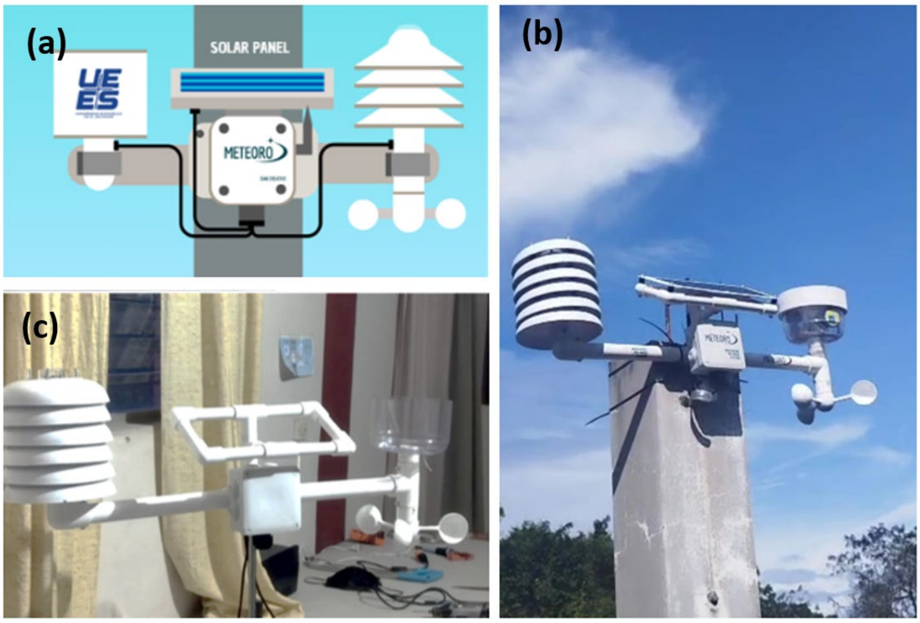

Fig. 1. The project development process begins with the abstraction and conceptualization of the idea. (a) Digital sketching process. (b) Final prototype. (c) PVC structure assembly process.

Initially, we considered basic aspects such as temperature, humidity, wind, and rain, which we will discuss in greater detail later in this document. From that stage on, it was understood that sufficient space was needed to house the system and the components associated with the environmental variables, such as a solar radiation shield for accurate temperature and humidity readings, an anemometer for wind measurement, and a rain gauge for rain measurement. Additionally, the plans included a control box and a solar panel; all these components are essential for the device to function properly as design.

The system adopted a modular design that distributes different functions to specific areas within the structure, clearly organizing the necessary hardware for each sensor. The original design focused on making it strong enough to handle different types of challenges, while keeping costs low by using affordable materials.

Furthermore, one of the main goals of collecting these environmental variables is to send the data to a remote server using a free cloud database and web hosting service. This implementation enables students and academic staff to access daily weather data through a web application, which can be easily accessed via QR codes placed throughout the university. These codes link directly to the site where current and historical data is visualized. If enabled, users could also receive notifications when dangerous or abnormal environmental conditions are detected, enhancing the educational and preventative potential of the system. Throughout the research, the technical and structural details related to this architecture will be explored in greater depth.

Advantages and Disadvantages

While METEORO v2 shares similarities with other low-cost weather stations, such as the use of ESP32 microcontrollers and solar power, its comprehensive sensor suite and energy management system set it apart. For example, the rain gauge's interrupt-based wakeup feature ensured no precipitation events were missed, a functionality not commonly found in comparable systems. Additionally, the station's PVC-based structural design offered superior durability compared to the materials used in [6], which often degrade under prolonged exposure.

The METEORO v2 weather station achieves significant improvements in cost, functionality, and energy efficiency compared to existing low-cost solutions like the 110 USD system by Pourbafrani et al. [1]. While their design measures basic parameters (temperature, humidity, wind speed, and solar irradiance), METEORO v2 expands capabilities with UV radiation, air quality (MQ135), and precipitation sensors-all while maintaining comparable affordability through optimized component selection (e.g., 3D-printed anemometer, PVC enclosure)[23].

Energy management is a key differentiator: METEORO v2's ESP32 deep sleep mode and relay-controlled sensor activation reduce consumption compared to continuous-operation systems [1]. This addresses the high-power limitations of air quality sensors, a challenge overlooked in the Pourbafrani design [1].

Although METEORO v2 demonstrates strong performance, its exclusive reliance on Wi-Fi communication limits deployment in remote or low-connectivity areas and increases dependence on existing network infrastructure. Transitioning to alternative long-range, low-power communication protocols, such as LoRa, could expand its operational range and reduce energy requirements. Additionally, the absence of over-the-air (OTA) firmware update capability means that physical access is currently required for maintenance or software improvements-an area where remote update functionality would enhance flexibility and reduce service interruptions.

From a hardware perspective, the station does not yet integrate the BME680 sensor, which combines temperature, humidity, pressure, and air quality measurements into a single unit. Incorporating this sensor could simplify the design, lower power consumption, and improve calibration consistency. Likewise, the lack of a wind vane limits the ability to record wind direction, and the precision of the rain gauge and anemometer could be further improved through professional calibration and refined measurement algorithms to ensure more accurate readings.

In terms of data processing capabilities, METEORO v2 currently operates without edge AI algorithms for real-time anomaly detection, which could help optimize bandwidth usage and reduce dependence on cloud processing. Implementing such algorithms, along with predictive alert systems based on historical weather data, would allow the station to provide early warnings for extreme weather events. These enhancements, while not yet present, represent valuable directions for future development that could significantly increase the system's autonomy, adaptability, and contribution to proactive environmental monitoring.

Development

A. Microcontroller Selection

The choice of microcontroller is a critical aspect in the design of IoT weather stations, as it determines the energy efficiency, processing capacity, connectivity, and scalability of the system. In this project, we opted to use the ESP32-WROOM, a robust, low-cost, high-performance solution widely adopted in similar research projects due to its integrated Wi-Fi and Bluetooth capabilities, dual-core architecture, and compatibility with energy-saving modes such as deep sleep.

The ESP32 is responsible for collecting data from connected environmental sensors, processing it locally, and sending it via HTTP protocols to a cloud database. Its ability to enter a sleep state between scheduled measurements-or be activated by specific interrupts such as those generated by the rain gauge-allows for energy-efficient operation, a crucial aspect in solar-powered systems.

Other studies of low-cost weather stations have analyzed microcontrollers according to their specific applications. Pauzi [4] highlighted the ESP32 for its ability to integrate sensors, control actuators, and provide real-time visualization in agricultural systems. Iglesias Cabello [11] compared the ESP32 with the Arduino Nano 33 IoT, highlighting the power of the former versus the simplicity of the latter in educational contexts. For their part, Sanchez [3] used a Raspberry Pi 3 for data collection and management, noting its high performance, albeit with limitations in cost and energy consumption.

Table 1. Comparison of microcontrollers considered for the research area.

	ESP32-WROOM	**Arduino Nano 33 IoT**	**Raspberry Pi 3**
Connectivity	Wi-Fi + Bluetooth	only Wi-Fi	Wi-Fi + Ethernet
GPIO pins	30–34	14–20	26+
Energy consumption	Very low (deep sleep)	Low–medium	High
Cost	Low	Medium	High
IoT scalability	High	Medium	High

Based on these criteria, the ESP32-WROOM represents an ideal solution for developing an IoT weather station in educational contexts, as it combines energy efficiency, programming flexibility, broad sensor compatibility, and integrated connectivity, all within a budget that is accessible for academic projects.

B. Selection of environmental variables and sensors

For the construction of a functional weather station, multiple key environmental variables were considered that allow for comprehensive monitoring of environmental conditions. These variables are: precipitation, wind speed, temperature, humidity, atmospheric pressure, ultraviolet radiation, and air quality. The selection of sensors was guided by criteria of low energy consumption, adequate accuracy, ease of integration with the ESP32, and commercial availability.

Rain gauge:

A rain gauge was developed using a rocker mechanism, a Hall sensor, and a neodymium magnet, with the aim of offering a functional, low-cost solution that can be adapted to low-energy monitoring systems such as the ESP32. (Figure 2).

The sensor operates using a seesaw system, which allows the amount of water collected to be counted using a channeling structure (funnel) with an inlet area of 95 cm^2. At one end of the seesaw is a neodymium magnet, while in the central part of the system, attached to the housing, is a Hall sensor, which detects the passage of the magnet each time the seesaw tilts due to the weight of the accumulated water.

Each complete oscillation of the rocker represents 3.2 ml of fallen water, which is equivalent to 0.34 liters per square meter (l/m^2), considering the surface area of the funnel. This count of oscillations is stored internally and converted into an estimate of the accumulated precipitation level. To obtain these values, the system multiplies the number of changes in the rocker arm over a given period by 0.34 l/m^2 and normalizes it per hour. The microcontroller performs these calculations internally, allowing for autonomous and continuous evaluation of precipitation behavior.

A significant advantage of this design is its compatibility with low-power systems. The system is connected to a wakeup pin on the ESP32, which allows any change in the Hall sensor's state (e.g., from 0 to 1 or from 1 to 0) to automatically wake up the microcontroller, even if it is in deep sleep mode. This ensures that no precipitation event goes unnoticed, regardless of the system's energy status.

The system not only detects the onset of rain, but is also able to determine when it has stopped using an algorithm based on the frequency of changes in the rocker. If the frequency of oscillations is high, the system interprets the event as recent and reduces the waiting time before considering the rain to have ended. If the frequency is low, the system waits for a longer interval before closing the event. This logic allows for the precise definition of the start and end of each storm, improving the accuracy of records and optimizing the use of computational and data transmission resources.

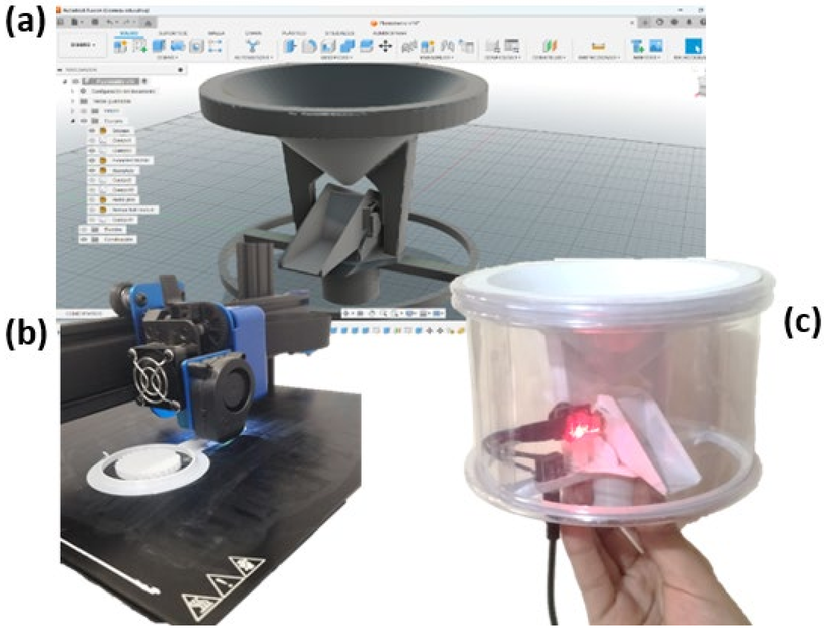

Fig. 2. Process of constructing a 3D-printed rain gauge. (a) 3D model design in Fusion 360 CAD software. (b) 3D printing process with PLA. (c) Assembly of the rain gauge together with the Hall sensor.

The rain gauge was designed and manufactured using 3D printing, with a focus on resistance to the outdoor environment and structural precision. Its main physical characteristics are:

- Sealed walls to prevent interference from wind or external vibrations.
- Upper grid to block the passage of unwanted objects.
- Epoxy coating to protect all parts exposed to water and humidity.

These measures seek to minimize false positives caused by external factors unrelated to rain, as well as to extend the device's useful life in adverse weather conditions.

Although the volume calculations by oscillation have been empirically defined by the development team, they will need to be validated against certified instruments in future research[20]. However, preliminary results indicate reliable and repeatable performance, allowing us to affirm that the sensor is suitable for low-budget environmental monitoring tasks[19]. The rain gauge currently under development is capable of:

- Detecting the start and end of precipitation events.
- Measuring the total amount of rainfall in mm/h.
- Classify rainfall intensity into established categories.
- Calculate the duration of each event.
- Estimate the intensity during the event, useful for activating automatic alerts.

Anemometer:

The anemometer measures wind speed using a structure that combines a static surface and a moving surface, joined by a bearing and a bolt that secures both parts. Its design incorporates a Hall sensor and a neodymium magnet. The blades mounted on the moving surface allow the wind to generate rotation; the magnet, attached to this moving part, passes in front of the Hall sensor, which is fixed to the base of the device and strategically located in the circular path. (Figure 3).

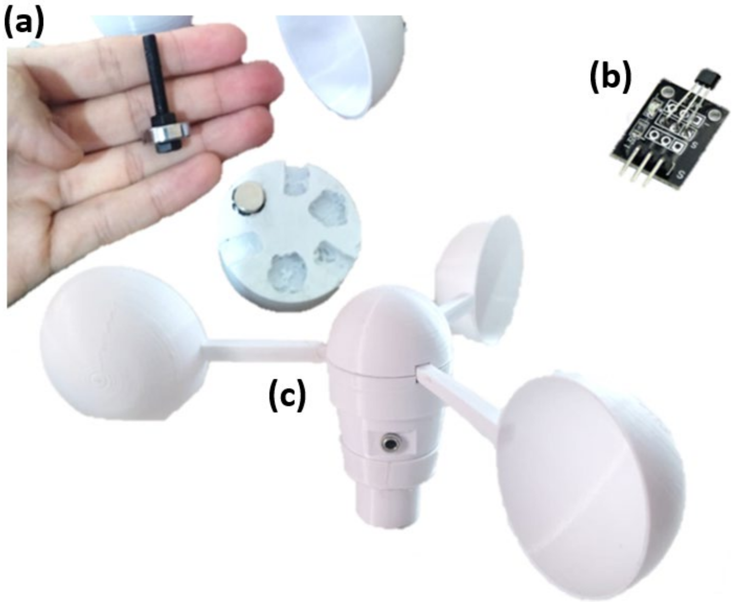

Fig. 3. 3D-printed anemometer. (a) Photograph of rotating mechanism with a bearing and bolt for the anemometer, next to the internal base of the anemometer with a neodymium magnet. (b) Photograph of a Hall sensor used in the project. (c) Final assembly of the anemometer.

The Hall sensor detects each revolution of the magnet, generating an electrical pulse that is captured by the microcontroller. In this way, by counting pulses per unit of time, it is possible to calculate the rotation frequency, which serves as the basis for estimating wind speed. For this purpose, physical principles that relate angular velocity to linear velocity are applied, taking the radius of rotation as a reference.

$$V\left(\frac{m}{s}\right) = w \times r = 2 \times \pi \times r \times f_{rotation\ of\ the\ blades}$$

Equation 1. Relationship between angular velocity and tangential velocity.

In addition, the pulses are counted at a specific time interval, using the millis() function, which returns the number of milliseconds since the ESP32 board started running the current program. This helps avoid interruptions in the system. Through this rotation frequency and an empirically calibrated constant, an estimated value for wind speed is obtained.

$$f_{signal} = \frac{N}{t}$$

Equation 2. Calculation of anemometer rotation frequency.

Finally, wind speed can be obtained from the signal frequency, the number of cycles, and their times, using the following equation:

$$V\left(\frac{m}{s}\right) = \frac{2 \times \pi \times r \times N}{t}$$

Equation 3. Final estimate of wind speed.

Like the rain gauge, this instrument has been designed and manufactured by the project team using 3D printing with PLA material. All calculations and parameters have been determined through internal testing. For future research, it is recommended that a validation process be carried out by comparison with professionally calibrated and certified anemometers in order to determine the actual accuracy of the prototype.

Solar Radiation Shield (DHT11, MQ135, S12SD and BMP180)

The solar radiation shield is an essential component integrated into our system, designed to optimize readings from environmental sensors that require direct exposure to outdoor air to take their measurements, while simultaneously protecting them from weather conditions. To ensure this functionality, we have adapted low-cost plastic plates that protect the DHT11, MQ135, S12SD, and BMP180 sensors [24]. This design promotes constant airflow, which is essential for improving data accuracy, while isolating the devices from the direct effects of solar radiation and preventing rainwater that could damage them. The sensors are integrated into a compact structure that simplifies installation and optimizes the use of available space. Specifically, the DHT11 sensor measures temperature and humidity parameters; the MQ135 evaluates air quality; the S12SD records UV radiation; and the BMP180 monitors pressure, altitude, and temperature. The sensors are carefully arranged inside the solar radiation shield: all are located in the center, except for the UV radiation sensor, which is located in the upper section to ensure direct exposure to sunlight, as shown in Figure 4. The shield structure was made using economical materials such as plastic plates, PVC, nuts, and bolts.

Looking ahead to future versions of the system, there are plans to replace the current set of sensors with a single, more efficient module: the BME680, which integrates temperature, humidity, pressure, and air quality measurements into a single device, thereby reducing energy consumption and space requirements. This decision stems mainly from the high energy demand of the MQ135, which posed a significant challenge in the energy optimization of the system, a topic that will be addressed in a later section. The specific accuracy of each sensor corresponds to the manufacturer's specifications; however, the uniqueness of our station lies in the infrastructure used and the methods applied within the microcontroller programming, which allows measurements to be averaged to achieve greater reliability in the data collected.

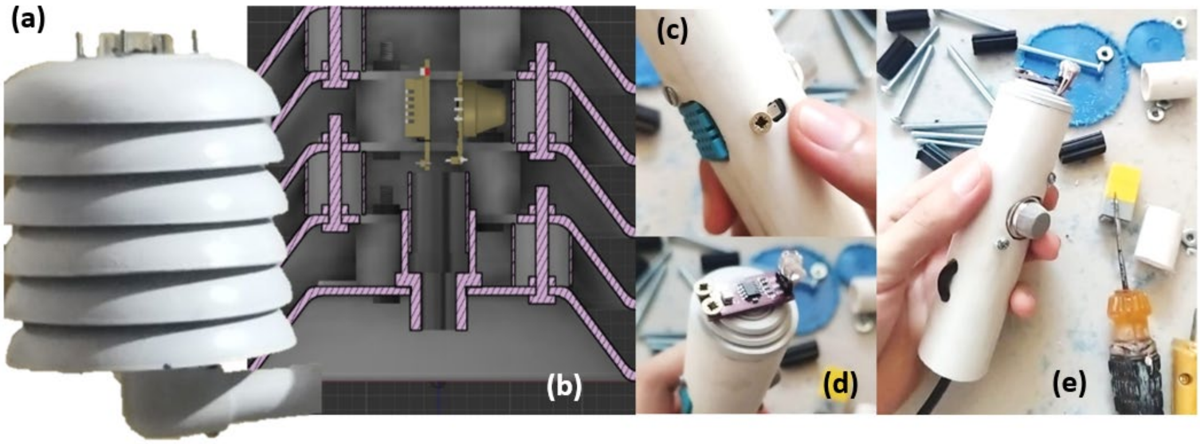

Fig. 4. DIY Solar radiation shield with plastic plates and PVC. (a) Final assembly of the solar radiation shield. (b) Internal analysis in Fusion 360 of the effect of the solar radiation shield on the sensors. (c) Internal tube of the solar radiation shield that integrates various sensors, showing the DHT11, BMP180, and MQ135 sensors. (d) The UV radiation sensor is shown at the top of the internal tube. (e) Complete view of the internal tube of the solar radiation shield with the integration of the sensors.

C. Selecting the power system

With the aim of achieving an autonomous, cable-free, and self-sustaining system, a solar power supply system was integrated. This decision not only promotes the use of clean and renewable energy but is also critical for installing the device in remote or hard-to-reach locations where conventional electrical connections are unavailable or represent a high logistical cost.

To ensure continuous operation, a detailed analysis of the energy consumption of each of the system's components was carried out, considering both the values provided by the manufacturers and the data obtained experimentally. One of the most relevant findings was that the MQ135 sensor, responsible for measuring air quality, is the component with the highest energy consumption in the system. This is because it requires initial and continuous heating to operate properly, with a typical consumption of approximately 150 mA at 5V, according to its technical data sheet. Although this value may seem low in conventional systems, it represents a major challenge for a solar-powered system, where even small energy savings are decisive [16].

To mitigate this problem, an on/off control was implemented using a 5V mechanical relay, ruling out alternatives such as NPN, PNP, MOSFET, or Darlington transistors, which, in empirical tests, did not offer the stability necessary to obtain reliable readings. Thanks to this system, the MQ135 sensor is activated only when measurements are required, including a 5-minute warm-up time, followed by a short sampling window to obtain a representative average. Due to its high power consumption, readings from this sensor are limited to once every 6 hours, thus avoiding compromising the overall energy autonomy of the system.

On the other hand, the low-power sensors integrated into the solar radiation shield (DHT11, S12SD, and BMP180) operate more efficiently and do not require prolonged stabilization periods. These devices take periodic readings every 30 minutes, and their residual consumption has been minimized through the use of an PNP transistor that completely cuts off the power supply when they are not in use. This selective power-on logic allows for greater energy control and contributes significantly to the overall efficiency of the system.

In addition, although the ESP32 microcontroller operates with Wi-Fi connectivity for data transmission, it does not maintain this connection continuously. Instead, a deep sleep function has been implemented, which drastically reduces energy consumption during periods of inactivity. According to the manufacturer, the ESP32 can consume between 160 and 260 mA in active operation with Wi-Fi, but when entering deep sleep mode, consumption can drop to 0.15 mA (with the ULP coprocessor active) and even down to 10 µA, a critical reduction for keeping the system operational 24 hours a day. The microcontroller is programmed to activate in sync with the local time, allowing the system to wake up at specific times to take periodic measurements. In the case of the rain gauge, an interrupt generated by the Hall sensor is used, set on one of the pins of the esp32 with this signal detection functionality enabled, capable of waking up the microcontroller when it detects a change of state, even if it is in sleep mode. Similarly, the anemometer has this capability; however, its activation is limited to once every 30 minutes in order to prevent the microcontroller from continuously waking up due to the unpredictable nature of the wind. This ensures that, during that time interval, there is an opportunity to record the first gust that occurs before the programmed readings are reset.

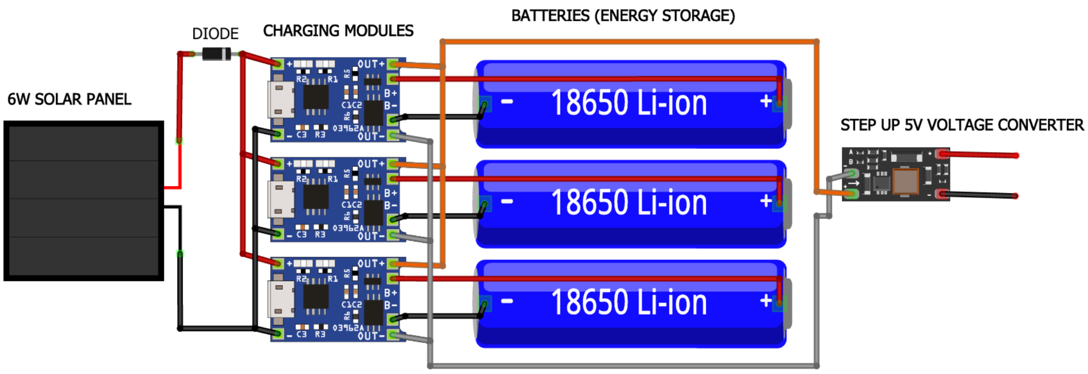

Fig. 5. Solar energy system and energy storage circuit.

After carefully analyzing the daily consumption of all components, an average energy consumption of approximately 540 mAh per day was estimated. To reliably meet this demand, a redundant power supply system was designed with three 18650 batteries, each with a capacity of 8800 mAh, connected in parallel to maximize system autonomy, especially on cloudy days or days with low solar production. Each battery has its own TP4056 charging module, which manages protection against overcharging and overdischarging. (*Figure* 5) A 6W solar panel that can generate 9600 mAh in a day with 8 hours of sunshine, and according to our own theoretical analyses, it is estimated that in low solar production scenarios there is still enough energy to keep the system running for several days. However, the stored energy does not increase much and begins to be limited with more cloudy days.

As mentioned, the charging process is carried out using a 6W solar panel, whose energy is regulated to prevent unwanted current return by means of a Schottky diode. To power the components that require 5V, a Step-Up module is used to raise the voltage of the batteries (3.7V nominal) to a stable 5V. The output of this module is connected directly to the microcontroller via a physical switch, which allows the system to be turned on or off completely if necessary.

The energy design of this weather station not only allows it to operate completely autonomously, but also reflects a series of strategic decisions that balance measurement accuracy with strict energy efficiency. This architecture represents a scalable and replicable model for future implementations in rural, agricultural, or hard-to-reach areas, where sustainability and energy independence are key factors for long-term success.

D. Development of the physical structure

The structural design of Meteoro was conceived with a focus on economy, strength, and ease of replication. To this end, ¾-inch PVC pipes were used, an accessible material that offers the necessary rigidity to withstand adverse weather conditions while allowing for a modular configuration[21]. This modularity facilitated the strategic distribution of key components such as the solar radiation shield, anemometer, and rain gauge, each located in an optimized position to ensure readings and efficient operation. Bolts and nuts were used to assemble the joints, as using PVC glue would have prevented future modifications to the prototype.

The junction box—responsible for protecting the electronic components—was redesigned using a locally sourced waterproof enclosure. This modification significantly improves protection against water, dust, and prolonged exposure to the sun, thus extending the useful life of the devices contained within.

At the top of the structure is the solar panel, positioned to maximize solar radiation capture during the day. In addition to its energy function, it acts as a thermal shade, helping to reduce overheating of the junction box, which is critical especially for preserving the health of lithium batteries, which tend to degrade rapidly when exposed to extreme heat.

The METEORO v2 station is designed to adapt to a wide variety of flat or cylindrical poles using plastic or industrial clamps. These are complemented by an optional metal plate to adapt to cylindrical surfaces, providing greater structural strength and stability against wind or vibrations, ensuring reliable anchoring in outdoor environments. The electrical connections between the sensors and junction box were implemented using UTP cables, which allow both data and power to be transmitted through a single modular channel.

E. Electronic control architecture

As previously illustrated, the system is based on an ESP32 microcontroller, which acts as the central processing and communication unit between the various sensors and actuators. Figure 7 shows the connection diagram detailing the control architecture, including voltage distribution, the use of auxiliary electronic components such as transistors and resistors, and the integration of each sensor into the system.

The ESP32 microcontroller is powered by 5V, which is also required by the MQ135 sensor. The rest of the sensors are powered by the 3.3V output provided by the ESP32 itself. However, because the MQ135 delivers an analog signal with an output range that can reach 5V, a voltage divider composed of 10kΩ and 20kΩ resistors is implemented in order to adapt the signal to the logic level of the ESP32 (3.3V) and prevent damage to the analog pin.

To minimize energy consumption, the MQ135 is activated only at reading times by a relay controlled by an NPN transistor. This transistor acts as a switch that energizes the relay in conjunction with the ESP32 control logic. Since the relay is activated by negative logic, its control can be shared between the signal pin and the transistor's power pin. In order to avoid unwanted activations, especially when the microcontroller enters deep sleep mode, a pulldown resistor is incorporated to ensure a stable low state at the base of the transistor, eliminating possible floating signals.

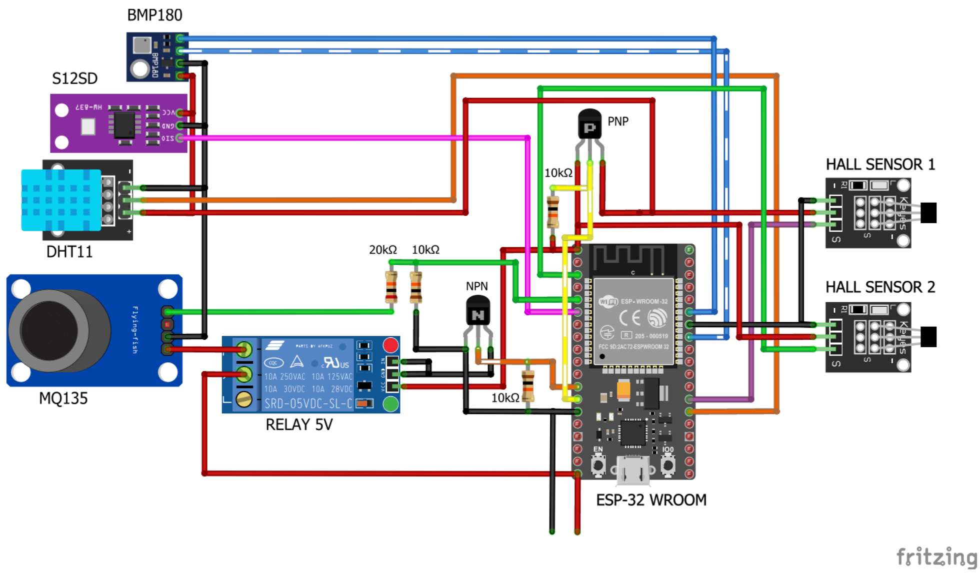

Fig. 6. Control circuit diagram and system sensors.

As for the DHT11 sensor, its communication is digital and requires no additional configuration, operating at 3.3V. Something similar occurs with the S12SD UV radiation sensor, which, despite delivering an analog signal, is compatible with the voltage levels of the ESP32 as it is powered by 3.3V. On the other hand, the BMP180, which measures barometric pressure and altitude, uses the I2C communication protocol, connecting via the SDA and SCL lines, which are fully supported by the microcontroller.

The Hall effect sensors used in the rain gauge and anemometer are digital and operate at 3.3V. These sensors can send signals to the microcontroller both to record events and to wake it up from deep sleep mode using pins configured as wake-up GPIOs, thus optimizing the system's energy use.

Finally, for more efficient energy management, the DHT11, S12SD, and BMP180 sensors are indirectly powered by a PNP transistor. This transistor allows the power supply to these sensors to be cut off when they are not in use. As in the case of the NPN transistor, the correct signal status during low-power mode is guaranteed by a pull-up resistor connected between the transistor base and the 3.3V voltage, ensuring that there are no unwanted fluctuations during periods of microcontroller inactivity.

F. Microcontroller firmware logic

With regard to the microcontroller firmware, METEORO's internal operation is based on a hybrid architecture designed for low energy consumption, in which the ESP32 remains in deep sleep mode most of the time[22]. It is only activated to perform specific measurement or data transmission tasks. When exiting this state, the system automatically selects between three autonomous operating modes: Rain Gauge Status (RG_Status), Anemometer Status (A_Status), and Time Status (T_Status), depending on the event that triggered its activation. These modes are organized and represented in the system control and communication diagram (Figure 7), which outlines both the logical flow and the relationship between physical components.

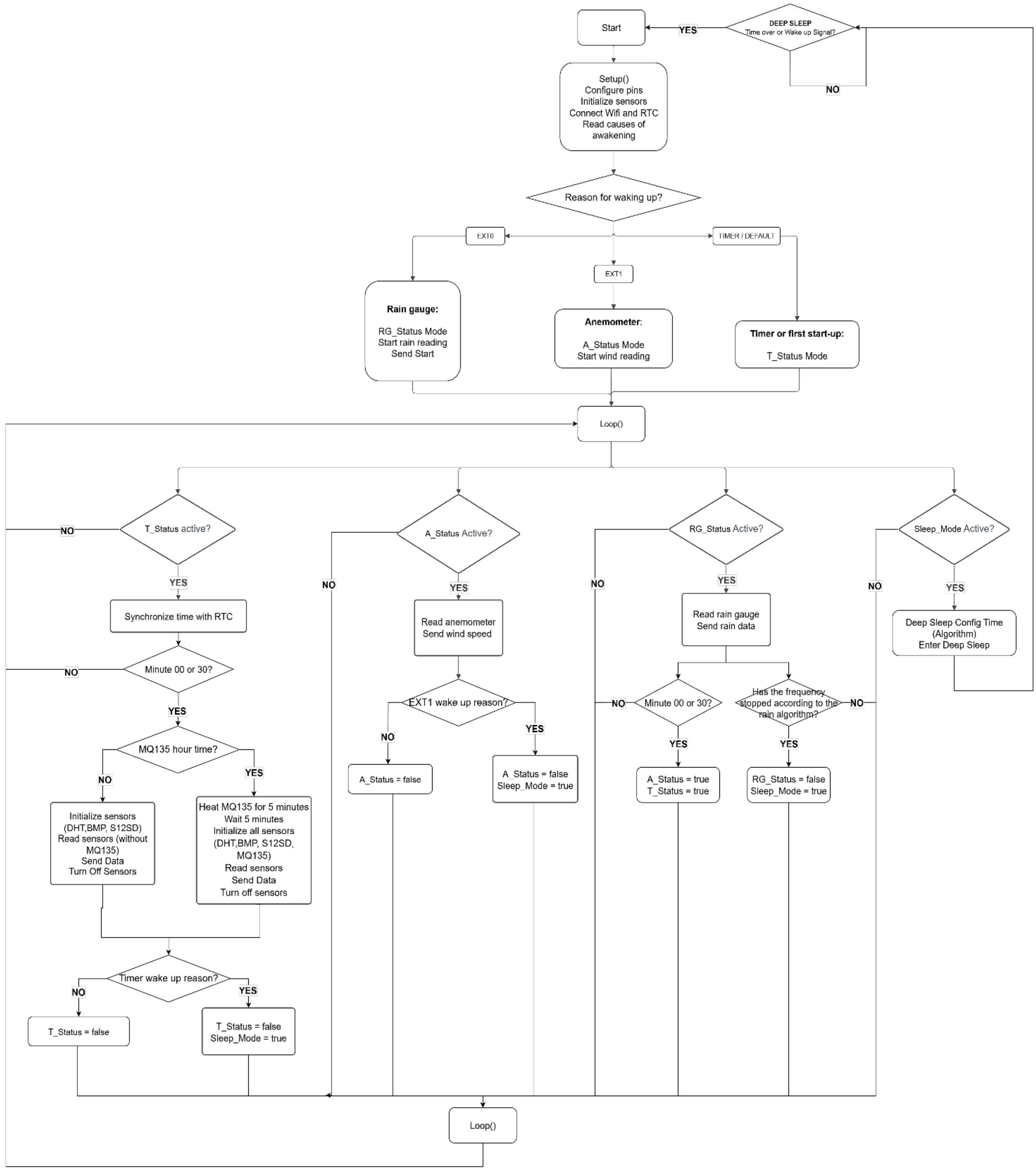

Fig. 7. Microcontroller program logic flowchart.

In basic terms, the ESP32 is programmed to wake up mainly in three ways: by timing, by anemometer signal, or by rain gauge signal. By default, it wakes up automatically every 30 minutes, just before the internal clock strikes --:00 and --:30. This synchronization is achieved using the time.h library, which allows the exact moment when the system should be activated to be calculated, ensuring periodic readings. Once awakened for this reason, the system enters the T_Status mode, in which low-power sensor readings from the Solar Radiation Shield, such as the DHT11, S12SD, and BMP180, are taken. This mode also checks whether a measurement should be taken from the MQ135, a sensor that is only activated every 6 hours due to its high power consumption.

The second activation mode occurs via an external interrupt (EXT1) when the anemometer's Hall sensor detects variations in its magnetic field associated with a gust of wind. This interrupt activates A_Status mode, initiating a wind speed measurement process. During this state, the microcontroller measures the revolutions of the anemometer at a fixed time interval, using the millis() function to avoid interruptions or blockages in execution. Upon completion of the measurement, the data is transmitted and the microcontroller returns to deep sleep until the next scheduled activation. To maintain consistency with the sampling cycles and avoid redundancy, the anemometer can only wake up the system once per interval between timed readings.

The third activation mode is triggered by the EXT0 wake-up pin, linked to the rain gauge's Hall sensor. This sensor detects changes in the status of the tilting mechanism, and if an initial value was recorded when entering deep sleep, the microcontroller can identify a change in status at the time of interruption. This activates RG_Status mode, allowing an immediate response to precipitation events, without waiting for periodic reading cycles. In this state, the microcontroller remains active as long as rain detection continues. Furthermore, if during RG_Status the programmed time for T_Status or A_Status is reached, these processes are executed without conflict, thanks to the non-blocking logic provided by millis(), which allows tasks to be executed in parallel.

The RG_Status termination algorithm is based on the rain gauge's switching frequency: the lower the frequency of changes, the longer the waiting time before the rain status is terminated, and vice versa. This proportional logic improves the accuracy of detecting intermittent rain or prolonged events.

In any of the three scenarios described, once the measurements are complete, the ESP32 briefly activates its WiFi module, connects to the local network, and sends the collected data via HTTP protocol to a cloud database. For this implementation, we chose Google Firebase, a free and accessible service that offers cloud storage (up to 1 GB) and a developer-friendly interface. Since the data packets generated by the weather station are lightweight, Firebase was an ideal solution for our needs.

Finally, once the data transfer is complete, the system closes the WiFi connection and assesses whether there are any pending tasks. If not, it calculates the time remaining until the next scheduled wake-up (for the --:00 or --:30 marks), adjusting the sleep time based on the duration of the operations performed to ensure energy efficiency and timing accuracy.

G. Communication protocol

As for the communication protocols used by the weather station, it was decided to use only the WiFi wireless technology integrated into the ESP32 microcontroller. This decision allows the system to connect directly to the university campus network as another device within the existing LAN infrastructure, without requiring additional hardware to establish communication with the server. This facilitates its implementation within the institutional environment. From the network, the station makes HTTP requests to transmit the collected data to cloud storage services. To improve connection stability and ensure greater coverage in potentially adverse environments—such as thunderstorms or interference caused by the height of its location—an external antenna was integrated into the microcontroller. This significantly improves data reception and transmission, ensuring continuous and reliable operation.

Although WiFi communication has proven functional in this version, it has been considered that, for future iterations, an ultra-low-power protocol such as LoRa could represent a more efficient alternative. LoRa allows the transmission of small packets over long distances with minimal energy consumption, which is ideal for applications in rural areas or areas with limited network infrastructure [18]. However, this approach would require a redesign that includes two nodes: one transmitter (in the field) and one receiver (connected to the WiFi network) [6], which acts as a bridge between the station and the server.

Additionally, the possibility of implementing OTA (Over-The-Air) functionality via WiFi is being considered, allowing for remote updating of the microcontroller's firmware. This feature is particularly useful, given that the station is installed in an elevated position that is difficult to access physically, which complicates manual updates. However, the implementation of OTA must consider future cybersecurity measures to prevent unauthorized access that could compromise the integrity of the system.

H. Backend Cloud services

As part of METEORO's evolution towards a more robust and scalable version, it was also decided to integrate a cloud-based backend based on Google Firebase [15], which allows real-time data collection, storage, and querying without the need for additional physical infrastructure. This environment, in addition to being highly compatible with microcontrollers such as the ESP32, provides features such as user authentication, web hosting, and scalable NoSQL databases, making it an ideal solution for educational and research purposes, especially since it offers free and accessible plans.

The ESP32 microcontroller is responsible for collecting environmental data and sending it securely to the cloud via the HTTPS protocol, using an API Key and the Firebase ESP Client library. The information is structured in JSON format and stored in the Firebase Realtime Database, allowing for efficient, low-latency synchronization. The data is organized under a hierarchical node called UsersData, where each user is identified by a unique Firebase UID. Within each UID, readings are grouped by sensor type (e.g., DHT22 or BMP180) and tagged with timestamps, facilitating both real-time visualization and historical analysis. (Figure 8).

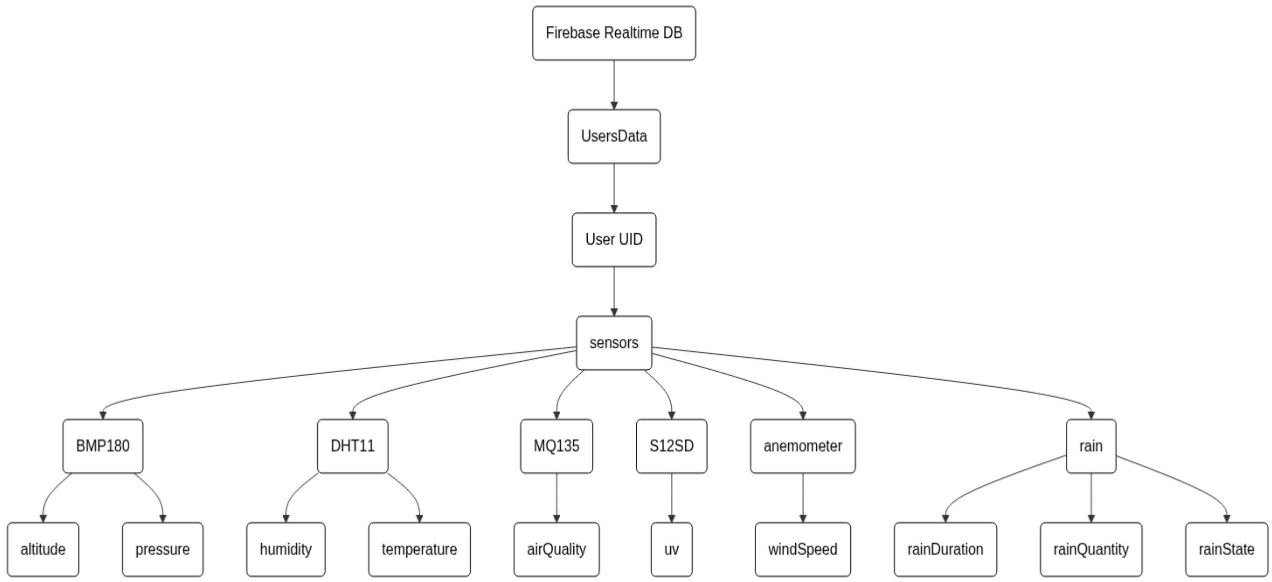

Fig. 8. Firebase database graph.

In this system, Firebase acts as a persistent logger. Although backups must be done manually, the platform allows data to be easily exported in JSON format for future queries or analysis. In addition, alert monitoring is handled from the web interface using real-time listeners that detect critical conditions (such as high temperatures or poor air quality) and trigger visual notifications, reducing the processing load on the ESP32 and maintaining an efficient and decentralized approach.

I. Data visualization method and web interface development

METEORO v2 utilizes Firebase Realtime Database's web hosting services to facilitate data visualization and alert mechanisms. By leveraging Firebase's integrated hosting features, the system ensures reliable performance and uninterrupted availability, aligning with the project's goal of maintaining cost-efficiency.

The web interface was developed using standard front-end technologies, including HTML, CSS, and JavaScript, to create an accessible and intuitive user experience. One of the main architectural features of the interface is its real-time data synchronization. Client-side JavaScript code establishes a persistent, real-time listener to the Firebase database. This event-driven approach ensures that upon the arrival of a new measurement from the weather station, the web application automatically fetches the new data and refreshes the displayed values and graphs without requiring a manual page reload. This functionality provides the university community with a live dashboard of campus weather conditions, fulfilling a primary educational goal of the project. (Figure 9).

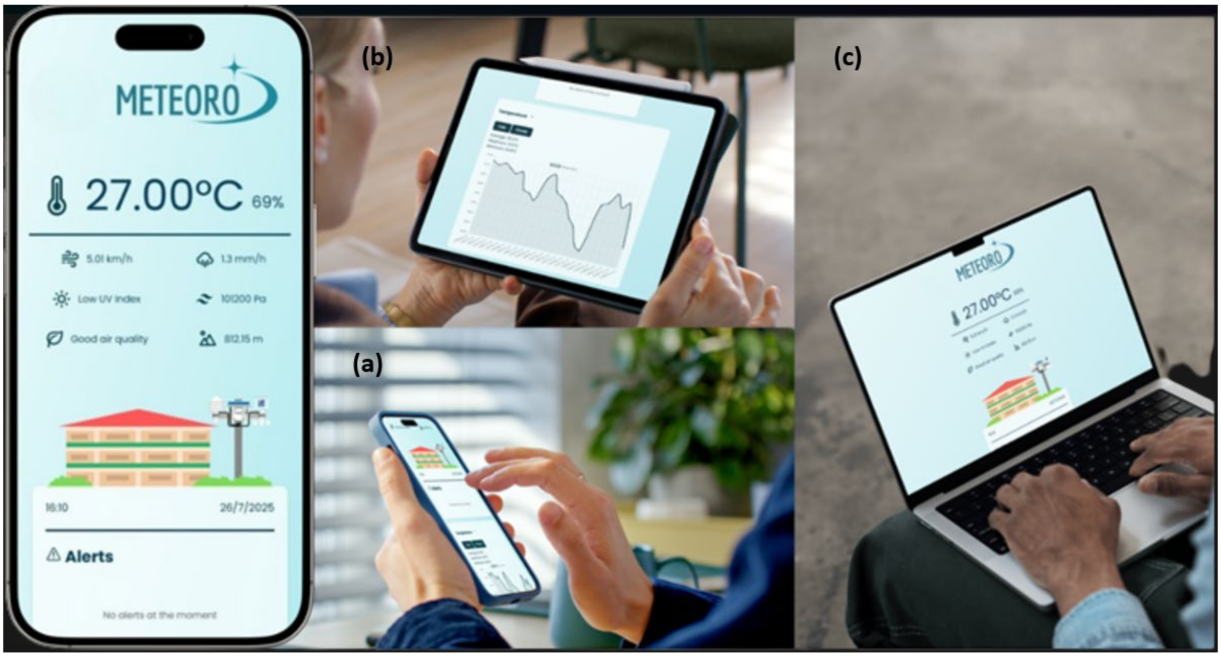

Fig. 9. (a) Viewing the website on a mobile phone (b) Viewing the website on a tablet (c) Viewing the website on a computer.

For data visualization, the open-source Chart.js library was implemented18. This powerful library was chosen for its flexibility in rendering dynamic and interactive time-series graphs directly in the browser[17]. It is configured to display data on both daily and weekly scales, allowing users to easily analyze short-term fluctuations and longer-term weather trends. The entire website was built with a responsive design philosophy, using flexible grid layouts and media queries to ensure full functionality and an optimized user experience across a wide range of devices, from desktop monitors to mobile phones. Figure 11 shows the responsive design of the web interface, illustrating its adaptability to different screen sizes and its commitment to broad accessibility for the entire university community.

Implementation

J. Initial testing in a domestic environment, detection and correction of bugs

Once the METEORO hardware assembly was completed and a preliminary version of the firmware was developed, the testing phase under real conditions began. The system was installed on a pole outside the home of one of the researchers, where it remained in continuous operation for 90 days. During this period, it was exposed to direct sunlight, high temperatures, humidity, and several thunderstorms, which allowed us to validate both its environmental resistance and the efficiency of solar energy collection.

In this initial testing stage, several technical challenges were identified, the most recurrent being event synchronization and stability in sending data to the server. Since the ESP32 had to establish a WiFi connection each time it woke up from low-power mode, the connection time was sometimes insufficient to successfully complete HTTP requests. This required redesigning the firmware's timing logic and adjusting connection times to ensure a more robust link before each data transmission.

In addition, significant improvements were made to the system's energy management, optimizing deep sleep periods and reducing unnecessary component consumption during inactivity. Key sensors such as the UV radiation sensor and the MQ135 (air quality) were calibrated, and algorithms associated with the anemometer and rain gauge data were refined to improve the accuracy of readings and avoid false positives.

A major challenge was that each firmware update required a physical cable connection, which was inconvenient due to the prototype's elevated location. This prompted consideration of incorporating OTA (Over-The-Air) updates or some secure wireless programming method for future versions.

Despite the initial adjustments and difficulties, the system evolved positively and managed to establish itself as a functional and stable prototype within its test context. This current version represents a solid starting point for future more robust and scalable iterations.

K. Installation in the university environment

Once the prototype was completed, the team engaged in discussions with university authorities to identify the most suitable location for the weather station. The selection process was critical, as the station required specific conditions to ensure optimal functionality. Key considerations included:

- Solar Exposure: The chosen site needed uninterrupted sunlight to maximize the efficiency of the 6W solar panel, which is essential for maintaining the system's energy autonomy. A location with minimal shading was prioritized to ensure consistent power generation throughout the day.
- Wi-Fi Connectivity: Proximity to a stable Wi-Fi network was another crucial factor, as the station relies on internet connectivity to transmit real-time atmospheric data to the Firebase cloud database. This enables seamless visualization of the data on the web interface for students and staff.
- Safety and Accessibility: The installation site had to be secure and easily accessible for maintenance, while also being exposed to environmental elements to ensure accurate sensor readings.

After evaluating these factors, an ideal location was identified on the university campus: a lamppost that met all the technical and logistical requirements. The elevated position not only provided ample solar exposure but also facilitated unobstructed wind and rain measurements. Additionally, the site was within range of the university's Wi-Fi network, ensuring reliable data transmission. (Figure 10).

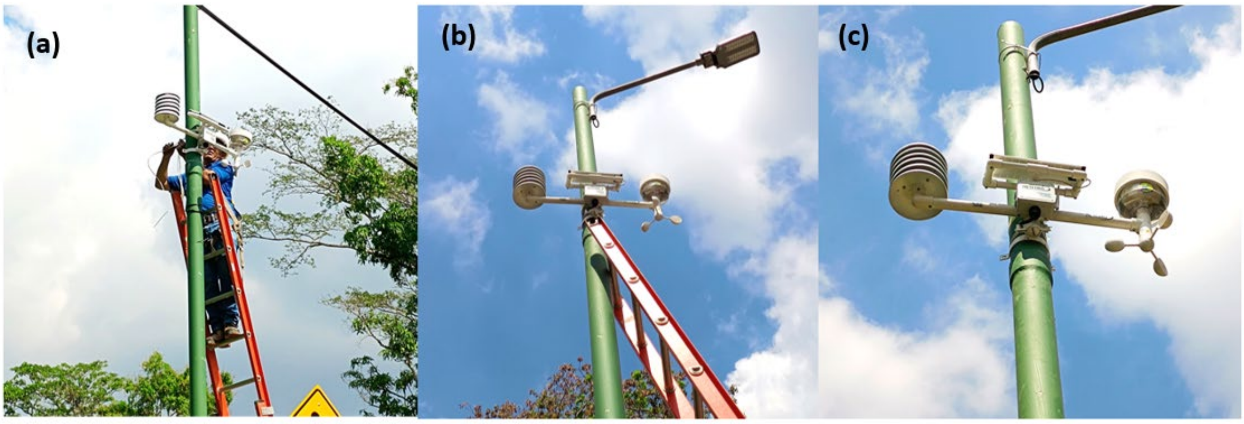

Fig. 10. (a) Installation of the station at the university (b) Correct adjustment of the station for sunlight (c) Final angle of the station.

The installation process involved mounting the PVC structure securely to the lamppost using industrial clamps, positioning the solar panel for optimal sun exposure, and verifying the connectivity of all sensors. This strategic placement ensures the station's long-term functionality and its role as a reliable educational tool for the university community.

L. Awareness and promotion in the educational community

To highlight the significance of the METEORO v2 project for the university, a promotional campaign was implemented to encourage engagement with the weather station's data. Key strategies included:

- QR Code Posters: Informative posters featuring QR codes linked to the METEORO web interface were strategically placed in high-traffic areas across the campus. This allowed students and staff to easily access real-time and historical weather data from their mobile devices or computers.

- Accessibility and Visibility: The placement of posters in central locations-such as corridors, laboratories, and common areas-ensured widespread awareness and facilitated seamless interaction with the system.

Through these efforts, METEORO not only became a functional weather monitoring system but also an educational asset, supporting hands-on learning and interdisciplinary collaboration.

Results and Discussion

The weather station was designed to address the limitations of its predecessor and other low-cost weather monitoring systems by integrating advanced energy management, durable materials, and a comprehensive suite of sensors. The system's performance was evaluated based on its energy efficiency and durability under real-world conditions. The results demonstrate significant improvements over existing solutions, particularly in terms of energy autonomy while maintaining a low-cost framework suitable for educational use.

One of the primary advancements in METEORO v2 is its optimized energy management system. The integration of a 6W solar panel and three 18650 batteries (8800 mAh each) ensured continuous operation, even during periods of low solar irradiation. The ESP32 microcontroller's deep sleep mode reduced energy consumption during inactive periods, a critical feature for long-term autonomy. This aligns with findings from Lefevre et al. [6], who emphasized the importance of energy-efficient designs in low-cost weather stations. However, the MQ135 sensor's high-power consumption (150 mA at 5V) posed a challenge, necessitating a relay-based activation system to limit its usage to once every 6 hours. This strategic power management approach contrasts with the simpler but less efficient designs reported in other studies [1], highlighting METEORO v2's innovative balance between functionality and energy conservation.

The system's data pipeline, leveraging Firebase for cloud storage and a responsive web interface, provided real-time access to weather data for the university community. The use of Chart.js for dynamic visualization enabled users to analyze trends in temperature, humidity, and other variables, fulfilling the project's educational goals. This approach mirrors the data-centric frameworks highlighted by Lefevre et al. [6], who advocated for scalable, cloud-based solutions in environmental monitoring.

Conclusion and Future Work

The METEORO v2 project represents a significant advancement in low-cost, autonomous weather monitoring systems, particularly optimized for educational environments. The system's most notable achievement is its highly efficient energy management system, which combines a 6W solar panel with three parallel-connected 18650 batteries (each 8800mAh) and sophisticated power control logic. The ESP32 microcontroller's deep sleep capability forms the backbone of this efficiency. Strategic decisions like using a mechanical relay to control the high-power MQ135 air quality sensor (limiting its operation to once every 6 hours with a 5-minute warm-up period) demonstrate thoughtful power optimization. The inclusion of wake-up interrupts from the rain gauge ensures no precipitation events are missed while maintaining energy savings.

Equally important is the station's durable physical construction. The team selected affordable yet robust materials including ¾-inch PVC pipes for the main structure, 3D-printed PLA components for sensor housings, and a waterproof electrical enclosure. These choices provide adequate protection against environmental factors while keeping costs low. The modular design allows for easy maintenance and upgrades. The solar radiation shield, crafted from plastic plates and PVC, effectively protects temperature and humidity sensors while permitting proper airflow. For data accessibility, the system provides responsive web-based visualization through Firebase, allowing users to view current and historical measurements.

Future Improvements:

Advanced Communication and Component Reduction

- Replace Wi-Fi with LoRa (Long Range) communication for remote or low-connectivity areas, reducing reliance on network infrastructure.
- Integrate the BME680 sensor (combining temperature, humidity, pressure, and air quality) to simplify hardware, reduce power consumption, and minimize calibration errors.
- Allow the use of OTA or similar technology for remote updating of microcontroller firmware.

Enhanced Sensor Accuracy and Expansion

- Improve the precision of the rain gauge and anemometer through professional calibration and refined measurement algorithms[23].
- Add a wind vane (wind direction sensor) to provide comprehensive data on local wind patterns.

AI and Data Processing

- Implement edge AI algorithms for real-time anomaly detection, optimizing bandwidth and cloud storage usage.
- Develop predictive alert systems based on historical trends to anticipate extreme weather events.

References

[1] M. Pourbafrani, H. Ghadamian, M. Moghadasi, and M. Mardani, Design, Fabrication, and Experimental Study of a Low-cost and Accurate Weather Station Using a Microcontroller System, Journal of Renewable Energy and Environment, *10*(4), (2023) 35–43. https://doi.org/10.30501/jree.2023.383796.1551.

[2] Dwi Ramadhani, A. Taqwa, Ade Silvia Handayani, Wahyu Caesarendra, Nyayu Latifah Husni, and C. R. Sitompul, Multisensor monitoring system for detecting changes in weather conditions and air quality in agricultural environments, Journal of Environment and Sustainability Education, 3(2), (2025) 274–286. https://doi.org/10.62672/joease.v3i2.103.

[3] M. A. Navarrete-Sanchez, Re. Olivera-Reyna, Ro. Olivera-Reyna, R. J. Perez-Chimal, and J. U. Munoz-Minjares, IoT-Based Classroom Temperature Monitoring and Missing Data Prediction Using Raspberry Pi and ESP32, Journal of Robotics and Control (JRC), 6(1), (2025) 234–245. https://doi.org/10.18196/jrc.v6i1.24345.

[4] A. F. Pauzi and M. Z. Hasan, Development of IoT Based Weather Reporting System, IOP Conference Series: Materials Science and Engineering, 917 (2020) 012032. https://doi.org/10.1088/1757-899x/917/1/012032.

[5] J. C. Shovic, Raspberry Pi IoT Projects, Apress, Berkeley, California, 2021. https://doi.org/10.1007/978-1-4842-6911-4.

[6] A. Lefevre, B. Malet-Damour, H. Boyer, and G. Rivière, Advancing Urban Microclimate Monitoring: The Development of an Environmental Data Measurement Station Using a Low-Tech Approach, Sustainability, *16*(7), (2024) 3093–3093. https://doi.org/10.3390/su16073093.

[7] K. Ioannou, D. Karampatzakis, P. Amanatidis, V. Aggelopoulos, and I. Karmiris, Low-Cost Automatic Weather Stations in the Internet of Things, Information, *12*(4), (2021) 146. https://doi.org/10.3390/info12040146.

[8] None Silvia Ganesan, C. Peng, L. Chen, N. Kong, N. Ng, and M. Reyasudin, IoT-enabled Smart Weather Stations: Innovations, Challenges, and Future Directions, Malaysian Journal of Science and Advanced Technology, (2024) 180–190, https://doi.org/10.56532/mjsat.v4i2.293.

[9] N. G. Bowen and Kumar Yelamarthi, Applied Classroom Use of a Mobile Multifunctional Weather Station, Asee.org, (2020). https://doi.org/10.18260/1-2--35726.

[10] Molnár, J., Kiresova, S., Vince, T., Kovac, D., Jacko, P., Beres, M., & Hrabovský, P., Weather Station IoT Educational Model Using Cloud Services, Journal of Universal Computer Science, *26*(11), (2020) 1495-1512. https://link.gale.com/apps/doc/A777671551/AONE.

[11] I. Cabello, Design and Implementation of an IoT Weather Station: Installation at Novia University of Applied Sciences, Finland, Urn.fi, (2025), https://urn.fi/URN:NBN:fi:amk-2025061022243.

[12] Leeming, F. C., Dwyer, W. O., Porter, B. E., & Cobern, M. K. (1993). Outcome Research in Environmental Education: A Critical Review. The Journal of Environmental Education, 24(4), 8–21. https://doi.org/10.1080/00958964.1993.9943504.

[13] Abed, S., Alyahya, N., Altameem, A. (2020). IoT in Education: Its Impacts and Its Future in Saudi Universities and Educational Environments. In: Luhach, A., Kosa, J., Poonia, R., Gao, XZ., Singh, D. (eds) First International Conference on Sustainable Technologies for Computational Intelligence. Advances in Intelligent Systems and Computing, vol 1045. Springer, Singapore. https://doi.org/10.1007/978-981-15-0029-9_5.

[14] A. Arnautov, R. Esin, E. Arnautova (2020) DESIGNING STEM-BASED LEARNING MODULES FOR FRESHMEN ENGINEERING STUDENTS, ICERI2020 Proceedings, pp. 5438-5446.

[15] L. Romero, S. Velarde, y M. Ampuero Rosas, "Smart meeting room scheduling and management system for a university campus using Android tablets with Firebase backend and Headwind MDM," en *Proc. 19th LACCEI Int. Multi-Conf. for Engineering, Education, and Technology – Virtual Edition*, Bogotá DC, Colombia, 19–23 Jul 2021, paper FP383.

[16] Yugank, H.K., Sharma, R. & Gupta, S.H. An approach to analyse energy consumption of an IoT system. Int. j. inf. tecnol. 14, 2549–2558 (2022). https://doi.org/10.1007/s41870-022-00954-5.

[17] Bahadur, T. (2022). Data Visualization: Existing Tools and Techniques. In: Kaur, P., Mallick, P.K., Balas, V.E., Geetha, S. (eds) Advanced Data Mining Tools and Methods for Social Computing. Intelligent Data-Centric Systems, vol 1. Elsevier. https://doi.org/10.1016/B978-0-323-85708-6.00017-5.

[18] Muthukrishnan, H., Jeevanantham, A., Sunita, B., Najeerabanu, S., & Yasuvantha, V. (2021). Performance Analysis of Wi-Fi and LoRa Technology and its Implementation in Farm Monitoring System. In: Proceedings of IVC RAISE 2020. *IOP Conference Series: Materials Science and Engineering*, vol. 1055, issue 1, 012051. IOP Publishing, UK. https://doi.org/10.1088/1757-899X/1055/1/012051.

[19] Hernández-Torres G and Sánchez-DelaCruz E (2025) Challenges and opportunities of ambient intelligence (AmI) in the 21st century: a historical review, Evolutionary Intelligence, 10.1007/s12065-025-01067-1, 18:4, Online publication date: 1-Aug-2025.

[20] Masson, M. H., Canu, S., Grandvalet, Y., & Lynggaard-Jensen, A. (1999). Software sensor design based on empirical data. In: *Measurement*, vol. 30, issue 3, pp. 213–221. Elsevier. https://doi.org/10.1016/S0304-3800(99)00097-6.

[21] Mazlitdinova, D., Suyunova, M., Khaydarova, G., Saibnazarova, M., & Kodirova, M. (2019). Modular Training System as a Factor of Improving Educational Process. In: *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, vol. 9, no. 1, pp. 3160–3166. Blue Eyes Intelligence Engineering & Sciences Publication.https://doi.org/10.35940/ijitee.A9152.119119.

[22] L.M. Broell, C. Hanshans, and D. Kimmerle, 'IoT on an ESP32: Optimization Methods Regarding Battery Life and Write Speed to an SD-Card', Edge Computing - Technology, Management and Integration. IntechOpen, Aug. 02, 2023. doi: 10.5772/intechopen.110752.

[23] G. Jenkins, 'A comparison between two types of widely used weather stations', Weather, vol. 69, no. 4, pp. 105–110, Apr. 2014. doi: 10.1002/wea.2237.

[24] Balamurali, D., Chakankar, S., Sharma, G. et al. A solar-powered, internet of things (IoT)-controlled water irrigation system supported by rainfall forecasts utilizing aerosols: a review. Environ Dev Sustain (2025). https://doi.org/10.1007/s10668-024-05953-z.

[25] B. Tian, K. M. Hou, X. Diao, H. Shi, H. Zhou and W. Wang, "Environment-Adaptive Calibration System for Outdoor Low-Cost Electrochemical Gas Sensors," in *IEEE Access*, vol. 7, pp. 62592-62605, 2019, doi: 10.1109/ACCESS.2019.2916826. keywords: {Gas detectors; Calibration; Monitoring; Air quality; Temperature sensors; Meteorology; Air quality monitoring; environment-adaptive calibration (EAC); IoT cloud platform; low-cost electrochemical gas sensors; non-linear regression}.

[26] H. Abubakar, S. A. Muhammad, M. A. Abubakar, and M. S. Sani, "Design and Development of a Solar Powered IoT-Based Weather Station," *International Journal of Sustainable Science and Engineering (IJSSE)*, vol. 14, no. 1, pp. 251–258, 2024.

[27] D. Franco, M. Costa, L. Rodríguez, and A. López, "IoT-based Smart Meteorological Monitoring System with LoRaWAN Communication," *Frontiers in Communication and Networks*, vol. 2, 2024, Art. no. 1505375.

[28] R. Yadav, A. Kumar, and S. Sharma, "Low-Cost Solar Powered Weather Station for Remote Monitoring Applications," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 2024.I2, no. 009, pp. 1–10, 2024.