Integrated Weather Monitoring: Design and Implementation of an Embedded Sensor Array

Peter Sideris (AM: 58341)

Democritus University of Thrace

Department of Electrical and Computer Engineering

Electronic Measurements (TMA: 101)

Instructor: Ioannis Andreadis

Xanthi, June 2024

Abstract

This paper presents the design and implementation of an ESP32-based automatic weather station connected to a sensor array, featuring a DHT11 for humidity sensing, a BMP180 for temperature and pressure measurement, and a MQ135 for air quality monitoring. By combining affordable hardware with advanced cloud services, this integrated weather monitoring system provides a cost-effective, infinitely scalable, and efficient solution for real-time environmental monitoring. The system's IoT-based architecture enables seamless data transmission and analysis, facilitating timely decision-making and improved environmental sustainability. This research demonstrates the potential of IoT in transforming traditional weather stations, enhancing the accuracy and reliability of weather forecasting, and offering a blueprint for future developments in this field. The proposed system's adaptability and cost-effectiveness make it an attractive solution for widespread deployment in various settings, from urban centers to remote areas, ultimately contributing to a better understanding and management of environmental phenomena.

Keywords: Weather Station; Internet of Things; Microcontrollers; Cloud; Edge Computing

Table of Contents

Introduction

In this chapter, we'll explore the history and fundamental principles of weather monitoring systems. We'll take a closer look at Automatic Weather Stations (AWS), explaining their important roles and components. Additionally, we'll discuss what the Internet of Things (IoT) is, why it's revolutionizing embedded systems in general, and how edge computing is transforming the way we monitor meteorological events, particularly in remote and underserved areas. By exploring these key ideas, our goal is to provide a clear understanding of how Modern Weather Stations work, and their significance in advancing weather monitoring capabilities.

A Millennium of Weather Observation

The history of humans observing and recording weather phenomena dates back thousands of years, with early civilizations such as the Babylonians and Egyptians developing methods to monitor and predict weather patterns. Early measurements include the Babylonians (650 BC) using cloud formations and astrological signs to predict the weather [12]. Similarly, the Greek philosopher Aristotle documented weather patterns in his texts around 350 BC [12]. It took humanity another 1500 years to develop the first thermometer, created by Galileo Galilei in Italy in 1593, with a major revolution in weather observation coming in 1837 [12] with the invention of the telegraph, making long-range communication possible and allowing for faster collection and transmission of weather information. In the 1860s, weather maps started gaining traction [12] after they were created using telegraph data, enabling the tracking of storms and prediction of their paths. Modern advancements include the inventions of radar technology in the 1940s and satellite technology in the 1950s, allowing us to monitor various weather phenomena with unprecedented precision. Currently, supercomputers, various machine learning models, and artificial intelligence analyze large amounts of meteorological data and generate predictions from complex models. Recent advancements in the Internet of Things have enabled the deployment of interconnected weather stations across vast geographic areas, which would otherwise be impractical to cover, thus easing real-time data collection and analysis for enhanced forecasting accuracy and early warning systems.

Automatic Weather Station Systems

According to the World Meteorological Organization (WMO), an Automated Weather System (AWS) is a fully configurable integrated system of components that is used to measure, record, and often transmit weather parameters such as temperature, wind speed and direction, solar radiation, and precipitation. Such systems are generally a combination of sensing instruments, interfaces, processing devices, and transmissions units [1], and are mostly operated, maintained and controlled by aviation service providers. In the United States, various types of automated weather stations exist, including the automated weather observing system (AWOS) and the automated surface observing system (ASOS) [9], each with its own area of application. In such systems, where communication is critical even in the toughest environments and conditions, data transmission happens in near real time via the Argos System, LoRa or the Global Telecommunications System (GTS). In the past, such systems were often placed where electricity and communication lines were available. Nowadays, advancements in Renewable energy have made it possible to deploy wireless stations anywhere. Modern AWS systems leverage IoT and edge computing technologies to enhance data accuracy, processing efficiency, and reliability, even in remote and harsh environments. Adhering to standards set by organizations like the World Meteorological Organization, AWS systems ensure high-quality and consistent data collection, vital for scientific and practical applications. With developments being made each and every day, further improvement of such systems is to be expected, increasing the rate of adoption and thus marking a new era of precision and accessibility in environmental data collection and analysis.

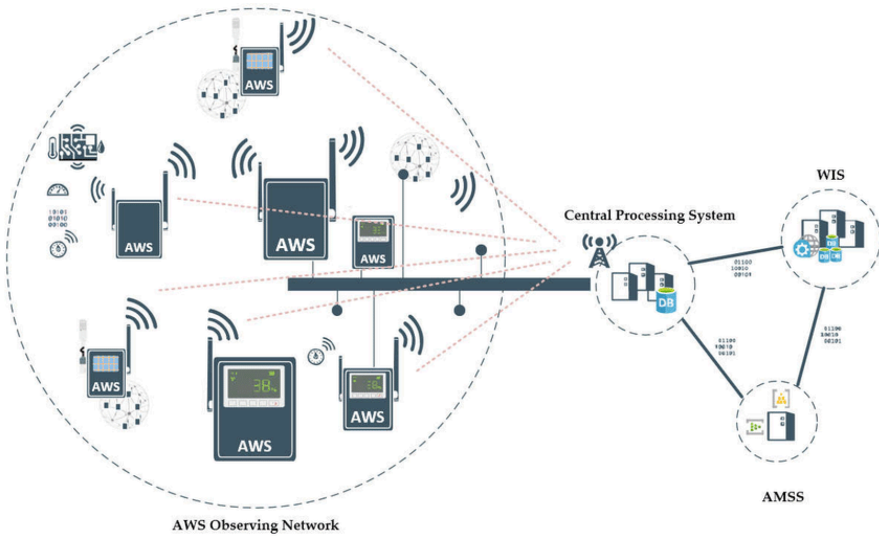

Figure 1. An integrated observing system based on World Meteorological Organization (WMO) system architecture. Source: [1]

The case for Decentralized Weather Stations

Historically, integrated weather stations have been centralized, relying on a single, often remote, location to collect and process data. This approach has had several limitations, including high infrastructure costs, limited coverage in remote areas, and vulnerability to single-point failures. In contrast, decentralized weather monitoring systems enable the deployment and monitoring of individual weather stations with ease, without having to rely on third parties. Recent advancements made on the Internet of Things, the interconnected network of physical devices embedded with sensors, software, and other technologies, enable them to collect and exchange data over the internet, and in particular cloud computing, offering as a result a more deployable and cost-effective solution. By distributing data collection across a local network of devices, decentralized systems can provide more accurate and localized weather insights, especially in areas where traditional centralized systems struggle to reach. Such is the case, in many agricultural applications, where relying on a single, centralized weather station is no longer sufficient, leading to an increasingly rapid adoption of such systems.

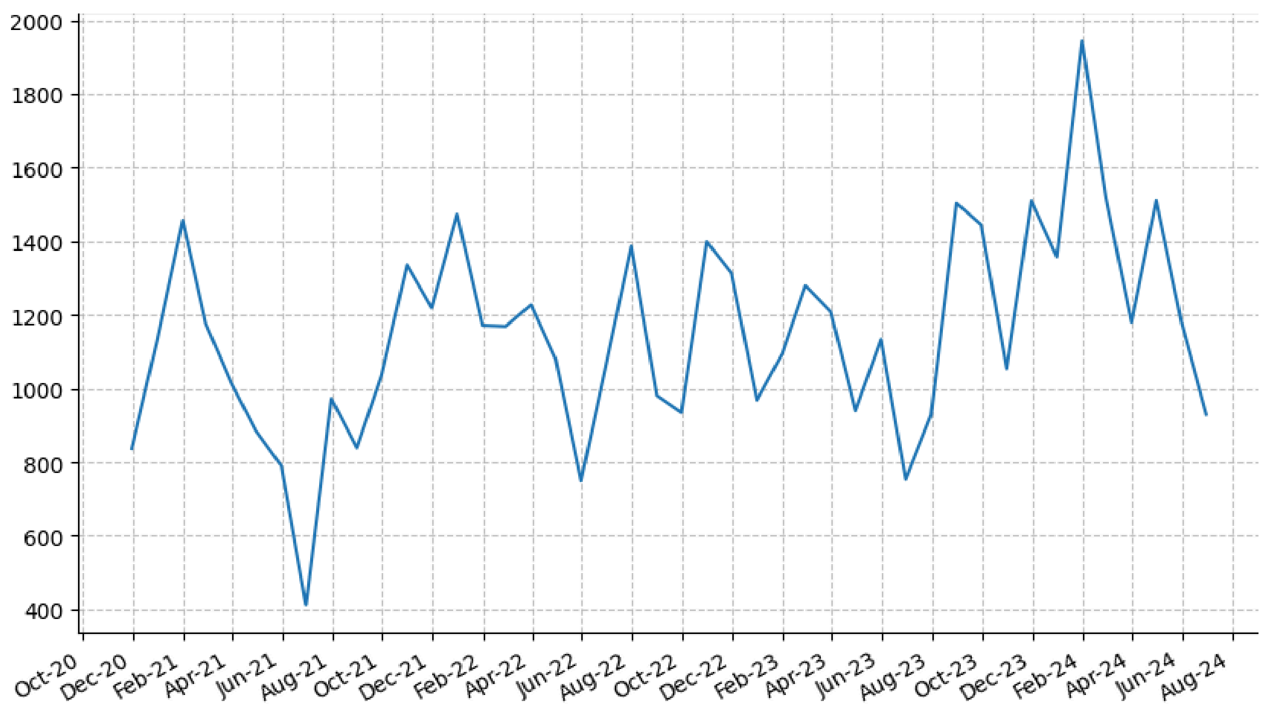

Figure 2: Energy produced from renewable energy sources (solar, wind etc.) in Greece, (MWh). This allows an AWS to operate independently from the electrical grid. Source [16]

Background and Motivation

As mentioned above, recent developments in the area of Internet of Things (IoT) have led to numerous industries adopting this technology. Having our interests piqued by the various real-world problems this technology can address and solve, we decided to explore its potential further. After having researched various areas this technology can be applied to, we became motivated by the need for an accurate, affordable and highly portable weather monitoring system, and thus we embarked on developing an innovative solution that harnesses the power of IoT. This paper presents the design and implementation of an Integrated Automatic Weather Monitoring System using an embedded sensor array. It highlights the system's capabilities for real-time data storage and processing enabling remote accessibility, provides detailed descriptions of the hardware and software components, communication protocols, and data processing methods, and discusses the benefits of leveraging a cloud-based infrastructure.

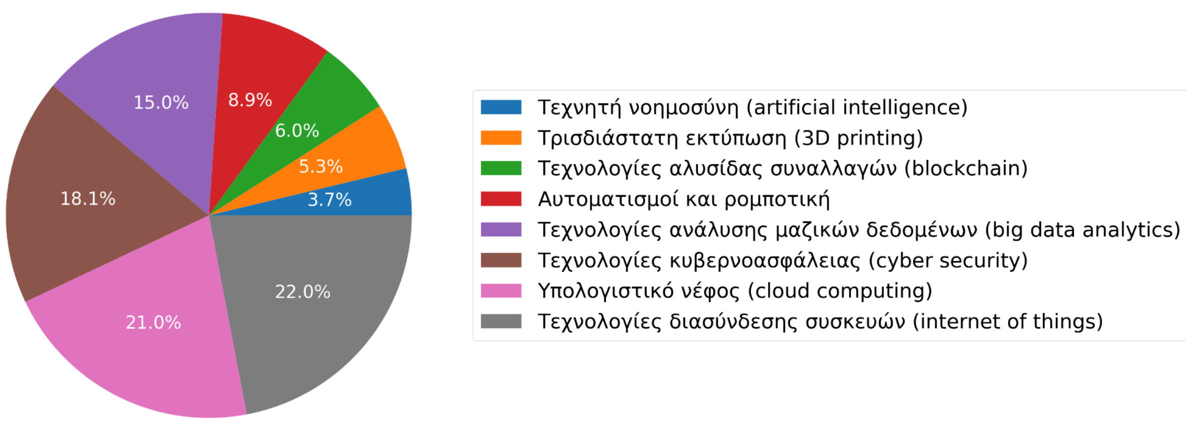

Figure 3: Innovative Businesses and Startups in Greece (2018-2020): Digital Tech used for New/Improved Products & Services. IoT sees the biggest adoption. Source [15]

General System Overview

The system is centered around the FireBeetle 2 ESP32-E microcontroller, based on the ESP-WROOM-32E, chosen for its ultra-low power consumption, on-board charging circuit and compatibility with a wide range of sensors. The sensor array includes a DHT11 for humidity, a BMP180 for temperature and barometric pressure, and an MQ135 for air quality monitoring. Upon sampling the sensor array at a pre configured interval, the microcontroller generates the appropriate JSON payload and sends it via HTTPS/1.1 POST requests to a remote cloud server. The server then processes the stream of data and executes the appropriate queries on the database, allowing for real-time monitoring as well as historical analysis and trend identification across multiple days. The collected data is then visualized and served dynamically on a website.

Figure 4: The Intersection of IoT and Agriculture: Revolutionizing Agricultural Practices in Remote Areas

Theoretical Framework

Overview of General-purpose Weather Monitoring Systems

As previously discussed, weather monitoring has traditionally relied on large, stationary stations equipped with high-precision instruments to measure various atmospheric parameters. These conventional systems, while accurate, are often costly, require significant maintenance and oftentimes impractical for certain scenarios, as they lack the flexibility and scalability needed for deployment in remote or resource-constrained areas. Recent advancements in IoT technologies have paved the way for more versatile and affordable weather monitoring solutions.

IoT in Environmental Monitoring

With Internet of Things (IoT) technologies gaining popularity across various industries due to their ability to deliver real-time data and insights, while maintaining a low-cost and an efficient infrastructure, their integration into weather monitoring systems offers a promising solution for enhancing the accuracy of environmental data. By incorporating IoT into environmental monitoring, sensors can be strategically deployed to gather data on parameters such as air quality, temperature, humidity, and atmospheric pressure in real-time. These sensors communicate data wirelessly to centralized servers or cloud platforms, where advanced analytics and machine learning algorithms can process the information. This integration enables continuous monitoring of environmental conditions over large geographical areas and facilitates early detection of anomalies or trends that could impact ecosystems or human health. Moreover, IoT-based environmental monitoring systems are scalable, allowing for flexible deployment and expansion as needed, thereby supporting sustainable management practices and informed decision-making for environmental conservation and resource allocation.

Core Challenge

The primary problem addressed in this paper is the need for a cost-effective, easily deployable solution for continuous environmental monitoring and data visualization in remote areas. Traditional weather stations are prohibitively expensive and require significant infrastructure, making them unsuitable for such applications. Existing IoT-based solutions, while promising, often fall short in terms of scalability, affordability or ease of deployment and maintainability.

Requirements for an Effective Weather Monitoring System

To address the identified challenges and gaps, an effective weather monitoring system must meet several key requirements:

1. Cost-Effectiveness: The system should utilize affordable hardware solutions to make it economically viable for widespread deployment.
2. Data Accuracy and Precision: The system must maintain consistently high levels of accuracy and precision in the collection and recording of environmental data.
3. Scalability - Maintainability: The system should be scalable to accommodate an increasing number of sensors and data points without significant redesign or cost.
4. Low Power Consumption: The system should be designed to operate efficiently with minimal power requirements, suitable for areas with limited power resources.
5. Reliable Communication - Security: The system should have robust communication protocols to ensure data integrity, security, and reliability at all times..
6. Weather Resistance: The system should be durable and capable of withstanding various weather conditions, ensuring continuous operation.

Figure 5: IGERT trainees deploy a weather station as part of a study on tornadoes, 2009

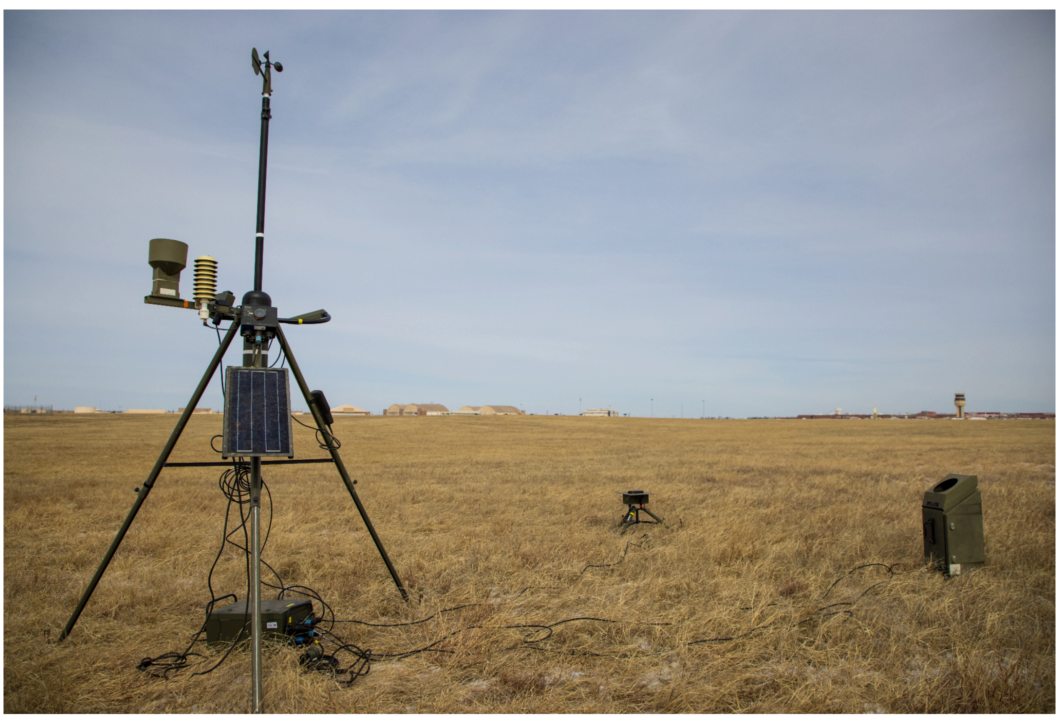

Figure 6: Tinker Weather Flight uses deployment gear, skills to cover scheduled maintenance

System Design and Architecture

The design and implementation of our integrated automatic weather monitoring system is centered around the FireBeetle 2 ESP32-E microcontroller, which acts as the central hub for data collection, monitoring and data transfer. This chapter outlines the hardware and software components, their integration, and the overall system architecture, offering a comprehensive understanding of how the system operates under the hood.

Overview of the Components

At the heart of the system is the ESP32 microcontroller, where it communicates with the DHT11 for humidity measurements, the BMP180 for temperature and barometric pressure readings, and the MQ135 for air quality monitoring. Immediate feedback is provided through a 16x2 LCD display showing real-time data locally, a Piezo Buzzer alerting users to critical sensor readings, and a push button that toggles the LCD's backlight, limiting the power the system draws when it's battery powered, via a high side switch.

Secure Communication and Data Transmission

Utilizing the ESP32's built-in Wi-Fi capabilities (802.11n, up to 150 Mbps) , the system sends data to a remote server, at regular intervals, via HTTPS/1.1 POST requests. This communication, being SSL Encrypted, ensures reliable data transmission, preventing data centric attacks from bad actors, most notably a man-in-the-middle attack. The server then processes incoming data and communicates that with an Azure Database for MySQL instance, via the Azure Virtual Network. This setup ensures that our sensor readings are secure, organized and easily retrievable.

Cloud Integration and Edge Computing

Azure's cloud services significantly enhance the system's capabilities. After writing our main backend microservice in Go, which processes the data stream from the ESP32, visualizes the data and serves the charts dynamically, we used a Docker container to deploy it on Azure App Services. We chose to deploy it this way, since it can be easily scaled horizontally (by adding more instances) and managed across various environments. This setup ensures continuous and efficient server operation while allowing for future modifications and adjustments. Finally, by leveraging Edge computing, a distributed computing model that brings computation and data storage closer to the sources of data, we can deploy Deep learning algorithms for the Internet of Things [14], thus enabling real-time processing, reduced latency, and improved performance in IoT applications, ultimately enhancing the overall efficiency and effectiveness of our weather monitoring system.

Data Visualization

The web-based interface that would host our data visualizations is designed with high performance and intuitiveness in mind. Its main purpose would be to provide real-time updates for informed decision-making, as well as the ability to examine historical data to identify patterns over time. We opted to use industry-standard tools to meet this challenge. Since we were using Go for developing the server/backend that would process, route and serve the required data accordingly, we chose TEMPL, a Go library for generating HTML code on the server using SSR (Server Side Rendering). Along with TEMPL, we utilized HTMX to develop the frontend interface of our web app. Finally, to generate the actual graphs, we used a popular JavaScript library called D3.js for producing dynamic and interactive data visualizations on the client side.

System Architecture

The system's architecture is designed with modularity and scalability in mind, allowing it to seamlessly adapt to diverse environments and requirements. This modular approach enables the logical organization of system components, making it possible to easily interchange or upgrade individual components as needed. This flexibility ensures that the system can be tailored to meet specific needs, whether it's expanding its capabilities by adding more sensors, or integrating new technologies (for example 5G, RF, GPS etc.).

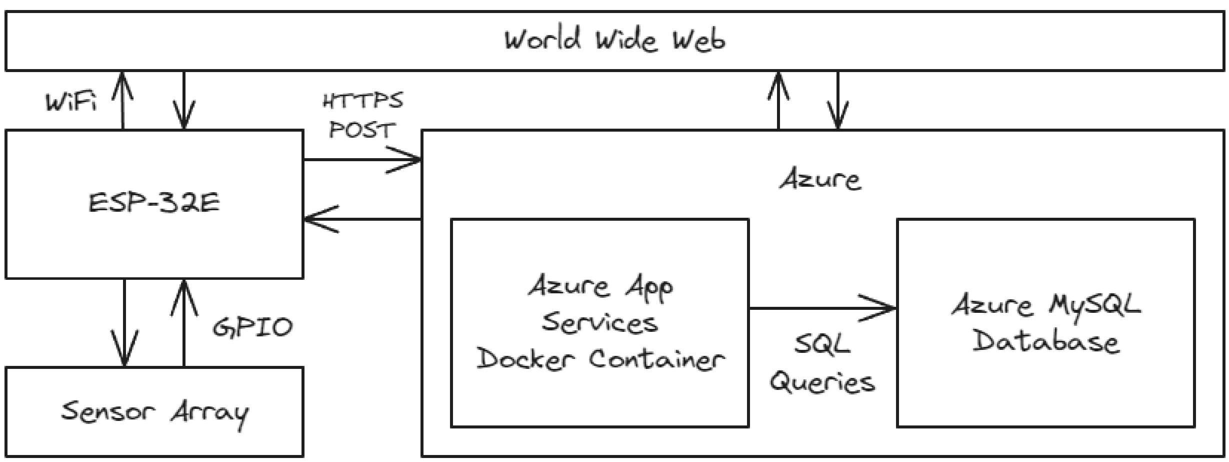

Figure 7: High-Level Overview of the Software Architecture

By combining affordable hardware with advanced cloud services, the system provides a cost-effective, scalable, and efficient solution for real-time environmental monitoring, particularly suitable for remote and resource-constrained areas. This addresses the need for continuous and reliable environmental monitoring and data visualization.

Testing and Calibration

With the implementation of our system's design being finalized, we recognized the critical need for vicious testing. Given that potential points of failure primarily reside within the software components of our design (such as misconfigured JSON payloads, database schema issues, and sensor sampling), our testing efforts were focused accordingly. In this chapter, we'll focus on assessing the reliability and accuracy of the integrated weather monitoring system as a whole. We will outline the testing methodology we employed to ensure that the system functions effectively under various conditions and scenarios. Finally, each sensor's factory calibration data will be explored and compared with our findings to evaluate accuracy and consistency.

The hardware testing began with sensor calibration. Multiple readings were taken under different environmental conditions to assess consistency. The DHT11 sensor's humidity readings were compared against a calibrated digital hygrometer to verify accuracy and were found to be within the margin of error (±5% Relative Humidity at 25℃ [3]). The BMP180 sensor's temperature and barometric pressure readings were validated against a standard digital thermometer and barometer. The sensor was found to be accurate, with an observed absolute pressure accuracy of ±1.0 hPa at 85°C and an absolute temperature accuracy of ±0.5°C [5]. In addition, the MQ135 sensor's air quality readings were compared against known measurements [13] from the Atmospheric Physics and Technology Station, part of the Laboratory of Air Pollution and Air Pollution Control Technology, Department of Environmental Engineering, Democritus University of Thrace, which is integrated into PANACEA, a research infrastructure.

Finally, data integrity checks were performed to ensure that the data transmitted from the ESP32 to the server via HTTPS was accurate and in the right shape, by examining each JSON Payload the ESP32 was generating before being sent via a POST request to the cloud.

Discussion and Future Work

A lot of thought has been put into possible ways of improving the system's functionality and usefulness. In this chapter, we'll discuss some of them, as well as the problems that they introduce. The scalability of the system's architecture allows for much of those improvements to be implemented with relative ease.

Many-to-one communication via LoRa

Given the nature of our system's real world applications, radio communication has always been a consideration. Deploying multiple "sub" weather stations (nodes), would have been the ideal way of implementing an off-grid remote weather station. Each node (slave) would be equipped with a low power microcontroller like the ATtiny84 (whose 12 GPIO pins [8] and very low power consumption would be ideal for our purposes), its own Integrated Sensor Array and an RF module, such that it would communicate its sensor readings back to the ESP32 main microprocessor (master) via RF. Each node would be responsible to sample its sensor array at regular intervals, and communicate that information back to the master via LoRaWAN, the communication protocol that's based on LoRa (from "long range"), a physical proprietary radio communication technique. The low power, low bit rate, and extensive IoT [7] use distinguish LoRaWAN from the currently used WiFi, that's designed to carry more data (JSON encoded payload from the ESP32 to Azure App Services), thus using more power. For reference, the LoRaWAN data rate ranges from 0.3 kbit/s to 50 kbit/s per channel, thus taking a typical JSON payload of 2.8 KB, between approximately 0.5 seconds to 76.46 seconds to be fully transmitted, depending of the course on the data rate, the weather conditions etc. Given the above-mentioned limitations we are working under alongside the bloaty nature of a JSON payload, we'd be using signal modulation techniques (like PWM) to transmit the sensor data from the slave microcontrollers to the master.

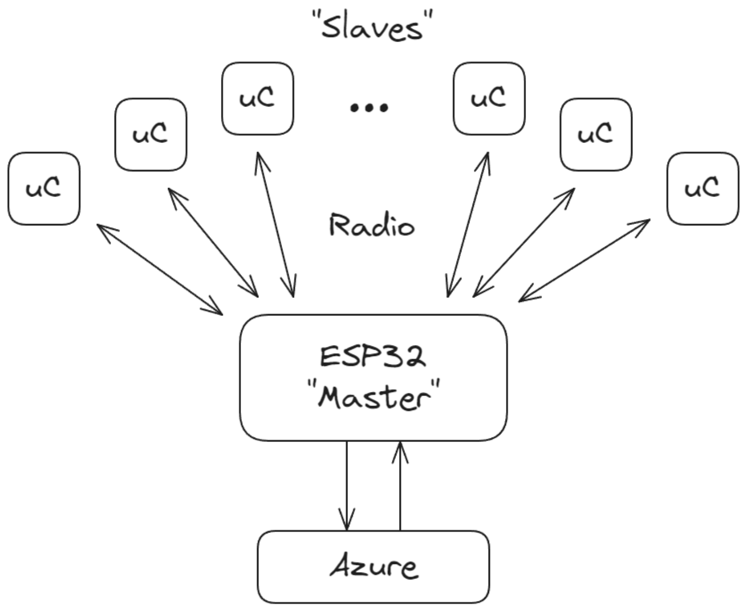

Figure 8: High-Level Overview of the proposed communication via Radio Frequencies.

GPS Integration for Accurate Geolocation and Time Stamping

Integrating GPS functionality into the system's architecture was one of the original ideas we had when designing our implementation. Using a GPS module alongside with the previously mentioned many-to-one communication schema would eliminate the need for WiFi connectivity, while at the same time making the final application more reliable, especially when working in environments where Internet connectivity is not available. The GPS module would provide precise geolocation and time information, enabling accurate timestamping and geographic tagging of weather data. Due to the high price and low availability of reliable GPS modules in the market, we opted not to use them and instead relied on WiFi communication and NTP timekeeping for our final implementation.

Energy Harvesting and Smart Power Management

Finally, in an effort to extend the operational life of our proposed system, integrating energy harvesting technologies such as solar panels and advanced power management algorithms is essential. Implementing efficient sleep modes and optimizing the duty cycle of sensor nodes helps to minimize power consumption, allowing for sustainable, long-term deployments in remote areas, whilst a built-in solar panel would allow for continuous recharging of the system's batteries. Due to the complexity of implementing our own smart BMS, we opted to use our microcontroller's (ESP32) built-in BMS with two Lithium-ion batteries.

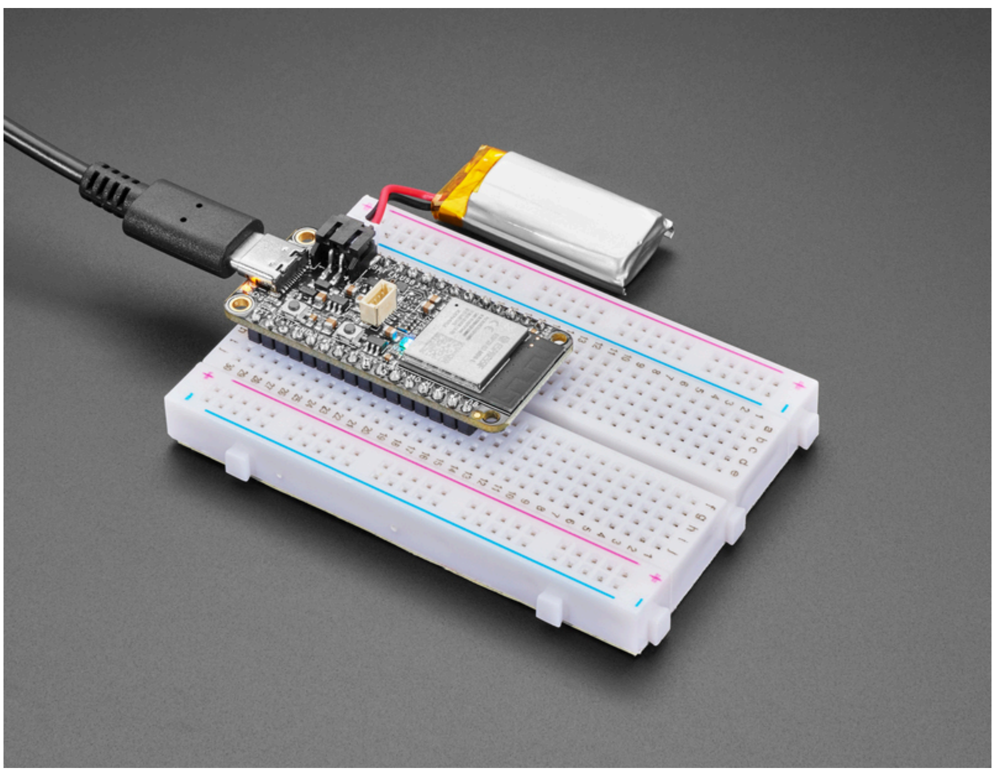

Figure 9: Power Management on an Adafruit ESP32-S2 Feather

Acknowledgments

We decided to make the design and implementation of the aforementioned system free and open source, available under the MIT License. This includes both our circuits schematic, the code for the ESP32 (firmware), as well as the Go backend and Go/Templ frontend services (packaged into one containerized environment). By making these resources openly accessible, we encourage collaboration and innovation within the community, while also allowing others to build upon and improve the system and to adapt the system for their specific use cases or applications.

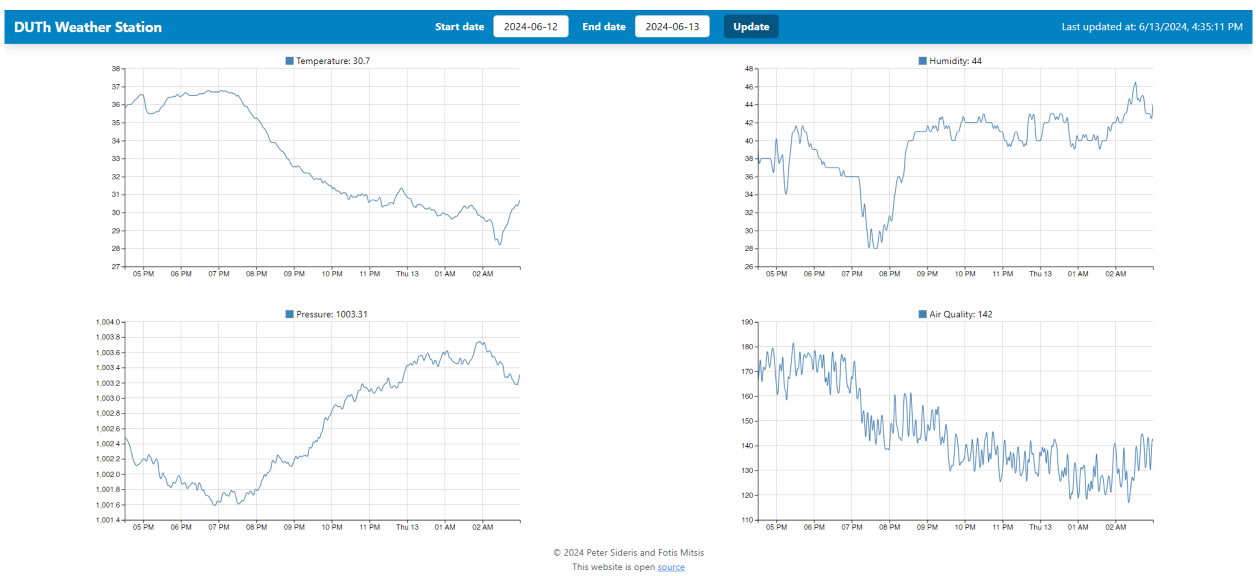

Figure 10: Screenshot of our final implementation in production.

References

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

[2] Bagiorgas, H.S., Assimakopoulos, M.N., Patentalaki, A., Konofaos, N., Matthopoulos, D.P., Mihalakakou, G. The Design, Installation and Operation of a Fully Computerized, Automatic Weather Station for High Quality Meteorological Measurements. Fresenius Environ. Bull. 2007, 16, 948–951. http://users.csa.upatras.gr/~dmatthop/p31.pdf

[3] Aosong. (n.d.). Temperature and humidity module DHT11 Product Manual. Retrieved from https://files.waveshare.com/upload/c/c7/DHT11_datasheet.pdf

[4] Waveshare. (n.d.). MQ-135 Gas Sensor Technical Datasheet. Retrieved from https://files.waveshare.com/upload/7/71/MQ-135.pdf

[5] Bosch Sensortec. (n.d.). BMP180 Digital barometric pressure sensor. Retrieved from https://cdn-shop.adafruit.com/datasheets/BST-BMP180-DS000-09.pdf

[6] Figure 5; IGERT trainees deploy a portable weather station as part of a study on tornadoes, (2009) https://www.nsf.gov/news/mmg/mmg_disp.jsp?med_id=73035&from=

[7] Adelantado, F; Vilajosana, X; Tuset-Peiro, P; Martinez, B; Melia, J; Watteyne, T. Understanding the limits of LoRaWAN. https://doi.org/10.1109/MCOM.2017.1600613

[8] ATtiny84; 8-bit AVR Microcontroller. Summary Datasheet; Page 5. Retrieved from https://ww1.microchip.com/downloads/aemDocuments/documents/OTH/ProductDocuments/DataSheets/8006S.pdf

[9] Observation Networks. Weather Forecast Office. US National Weather Service https://www.weather.gov/lmk/observation_networks#:~:text=ASOS%20is%20the%20backbone%20of,but%20generally%20at%20smaller%20airports.

[10] Figure 6; Tinker Weather Flight uses deployment gear, skills to cover scheduled maintenance; March 8, 2022. Retrieved from

https://www.arnold.af.mil/News/Article-Display/Article/2958829/tinker-weather-flight-uses-deployment-gear-skills-to-cover-scheduled-maintenance/

[11] Figure 9: Battery + USB Power on an Adafruit ESP32-S2 Feather. Retrieved from https://learn.adafruit.com/adafruit-esp32-s2-feather/power-management

[12] Weather Forecasting Through the Ages. Earth Observatory NASA. Retrieved from https://earthobservatory.nasa.gov/features/WxForecasting/wx2.php#:~:text=Around%20650%20B.C.%2C%20the%20Babylonians,a%20different%20type%20of%20weather

[13] Ατμοσφαιρικές μετρήσεις – βαθμονομημένα αποτελέσματα – ενδεικτικά των διακυμάνσεων ρύπανσης; ΠΑΝελλΑδιΚή υποδομή για τη μΕλέτη της ατμοσφαιρικής σύστασης και κλΙματικής Αλλαγής (ΠΑΝΑΚΕΙΑ). Retrieved from: https://panacea-ri.gr/index.php/atmospheric-measurements/

[14] H. Li, K. Ota and M. Dong, "Learning IoT in Edge: Deep Learning for the Internet of Things with Edge Computing," in IEEE Network, vol. 32, no. 1, pp. 96-101, Jan.-Feb. 2018, doi: 10.1109/MNET.2018.1700202. Retrieved from https://ieeexplore.ieee.org/abstract/document/8270639

[15] Χρήση ψηφιακών τεχνολογιών για ανάπτυξη καινοτομιών; Εθνικό Κέντρο Τεκμηρίωσης Retrieved from https://data.gov.gr/datasets/ekt-digital-tech-use/

[16] Ανανεώσιμες πηγές ενέργειας; Ανεξάρτητος Διαχειριστής Μεταφοράς Ηλεκτρικής Ενέργειας. Retrieved from https://data.gov.gr/datasets/admie_realtimescadares/