An IoT-based Smart Agriculture System using LoRa and Cloud Monitoring for Automated Greenhouse Control

Nabiha Ben Abid
Department of Microelectronics and Instrumentation
University of Monastir Monastir, Tunisia

Abdalla M. Khattab
Department of Computer Science, AlBaha University, Saudi Arabia.
Egyptian Atomic Energy Authority (EAEA), Egypt

Hani A.M. Harb
Department of Software Engineering
AlBaha University, Saudi Arabia

Chokri Souani
Department of Microelectronics and Instrumentation
University of Monastir, Monastir, Tunisia;

ABSTRACT

New agricultural opportunities have been made possible by the quick development of digital technologies, particularly with the use of Internet of Things (IoT) systems. This paper describes the design and execution of an integrated Internet of Things (IoT)-based smart agricultural solution that combines automated greenhouse control with wireless environmental monitoring using LoRa connectivity. The system architecture consists of two primary subsystems: a greenhouse control unit that uses a Raspberry Pi Pico W to regulate important factors like irrigation, lighting, and cooling, and a sensor node that uses an ESP32 Heltec LoRa module to collect field data. Real-time visualization and remote control through a mobile application are made possible by the transmission of sensor data, such as temperature, humidity, and soil moisture, over LoRa or Wi-Fi to cloud-based platforms like Tago.io. The system also incorporates energy-saving strategies through extended sleep modes to increase energy efficiency. To improve energy efficiency, the system also uses deep sleep modes and other power-saving techniques. Results from real-world implementation show how well the system works to provide automatic reaction capabilities and ongoing environmental monitoring. In order to enhance crop output, maximize resource utilization efficiency, and enable precision agriculture, the suggested solution provides a scalable, economical, and energy-efficient instrument.

Keywords

Smart Agriculture, IoT, LoRa, ESP32, Raspberry Pi Pico W, Cloud Monitoring, Greenhouse Automation, Tago.io, Precision Farming.

1. INTRODUCTION

The agricultural industry has been forced to adopt digital transformation due to the rising demand for food worldwide, the depletion of natural resources, and the uncertain consequences of climate change. This evolution has led to the emergence of Agriculture 4.0, which is defined by the use of cutting-edge technologies like cloud computing, wireless sensor networks, machine learning, and the Internet of Things (IoT) to enhance farming operations' resource efficiency and decision-making [1][2].

Manual interventions, irregular data gathering, and ineffective resource management are frequently the limitations of conventional farming operations. On the other hand, data-driven automation, remote control, and real-time monitoring of vital agricultural operations like pest detection, climate management, fertilization, and irrigation are made possible by smart agriculture systems, which make use of distributed sensors and embedded platforms [3]. It has been shown that these methods decrease environmental effect, increase agricultural output, and use less water [4].

Because of its long-distance, low-power transmission capabilities, LoRa (Long Range) technology is one of the communication protocols used in smart agriculture that is especially well-suited for rural deployments. Wide-area monitoring systems that function independently over prolonged periods of time, even without cellular infrastructure, can be deployed thanks to LoRa [5]. Wi-Fi-enabled microcontrollers, like the Raspberry Pi Pico W, on the other hand, offer more bandwidth for real-time interactions and are perfect for greenhouse settings where cloud platforms may be directly connected with actuators and sensors.

Recent studies have explored IoT in agricultural management. For example, cloud-connected irrigation using ESP32 and ThingSpeak [6], and LoRa-based soil monitoring with energy-efficient sleep cycles [7] have been implemented. However, few solutions integrate cloud visualization dashboards like Tago.io, for real-time analytics and user engagement, with both automated greenhouse control and long-range field monitoring.

This study proposes a smart agriculture solution integrating Wi-Fi and LoRa technologies. The system comprises two subsystems: a LoRa-based ESP32 Heltec sensing node for outdoor environmental monitoring, and a Raspberry Pi Pico W-based greenhouse control unit enabling real-time data transmission to Tago.io and autonomous actuation (cooling, lighting, and irrigation).

This hybrid IoT architecture aims to provide farmers with mobile decision-making tools, while also achieving scalability, energy efficiency, and adaptability across diverse agricultural environments. Experimental deployments in open-field and greenhouse settings evaluated the system based on sensor accuracy, energy efficiency, network latency, and user engagement.

2. RELATED WORK

Research on incorporating Internet of Things (IoT) technologies into agriculture has increased significantly in recent years with the goal of modernizing old methods through automation, data collection, and intelligent decision-making. These developments are essential to the idea of Agriculture 4.0, which uses smart infrastructures and networked systems to maximize resource utilization, decrease environmental impact, and increase productivity [8].

2.1 IoT in Smart Agriculture

Distributed sensors for environmental monitoring, microcontrollers for data processing, and wireless communication modules for real-time transmission are common components of Internet of Things-based systems. The implementation of such systems to facilitate crop health monitoring, microclimate control, and precision irrigation has been the subject of numerous studies.

For instance, Jawad et al. [9] suggested a wireless sensor network architecture with a focus on scalability and energy efficiency for precision farming. Their technique made it possible to use low-power nodes to continuously monitor ambient conditions and soil moisture, which is an essential need for remote agricultural areas.

For smallholder farms, Jayaraman et al. [10] created a comprehensive platform that integrated sensor data, cloud storage, and data analytics. The system proved that it was possible to use platforms like ThingSpeak and Blynk with inexpensive hardware like ESP8266 and ESP32. However, real-time responsiveness and long-range communication were found to have limitations.

2.2 LoRa Technology in Agriculture

The capacity of LoRa (Long Range) and LoRaWAN technologies to cover large regions with low energy usage has made them popular in agriculture. A LoRa-based environmental monitoring system was put into place in [11] to gather information from far fields and send it to a central server. According to the study, LoRa is better than Wi-Fi and GSM in terms of transmission range and power consumption, which makes it perfect for rural deployments.

Other studies, as [12], implemented scalable networks of sensor nodes for temperature and humidity tracking using ESP32 Heltec modules with embedded LoRa transceivers. In order to extend battery life, these systems frequently included deep sleep modes, a tactic that our approach also used.

2.3 Greenhouse Automation and Cloud Integration

Automated actuation (such as lighting, ventilation, and watering) based on sensor feedback is crucial in greenhouse situations because they offer a controlled yet dynamic environment. In [13], researchers used the NodeMCU technology and cloud-based visualization with Blynk to propose a Wi-Fi-enabled greenhouse monitoring system. Although these systems work well for short-range control, they have issues with scalability and latency in bigger deployments. On the other hand, our work uses a Raspberry Pi Pico W, a low-cost microcontroller with Wi-Fi capabilities, to operate local greenhouse automation chores. Compared to more straightforward dashboards like ThingSpeak, the usage of Tago.io for data visualization and control provides a more reliable and responsive user experience.

2.4 Energy Management and Intelligent Scheduling

One of the key concerns with IoT installations is power efficiency. Real-time operating systems (RTOS) and scheduling algorithms have been investigated in a number of research [14][15] in an effort to control sensor activity, minimize duplicated communication, and optimize energy efficiency. Measurable gains in system stability and runtime duration have been demonstrated with the use of RTOS architectures and finite state machines.

Energy-saving techniques including RTOS-based task scheduling for Wi-Fi connectivity and deep sleep for the ESP32 node have been used and verified in our project, supporting results from earlier research.

2.5 Comparative Analysis

Few studies propose a hybrid architecture that combines LoRa-based long-range sensing with Wi-Fi-based local control and cloud visualization, but prior research frequently focuses on either environmental monitoring or greenhouse automation. Table 1 illustrates the unique features of our system in terms of architecture, energy optimization, and deployment flexibility while summarizing a few chosen research studies.

Table 1. Comparative Analysis

Comm.Protocol	Platform	Features	Limitations
ZigBee	Custom	Energy-efficient WSN	Short range, complex config
LoRa	ESP32	Long-range field monitoring	No actuation, no cloud interface
Wi-Fi	NodeMCU	Greenhouse control+Blynk	Limited range, latency issues
LoRa+ Wi-Fi	ESP32+Pico W	Hybrid sensing + actuation + Tago.io	Requires dual-systel integration

3. MATERIALS AND METHODS

3.1 Overview of the Proposed System

The proposed IoT-based smart agriculture system was designed to integrate two complementary subsystems:

- A LoRa-based Field Monitoring Unit responsible for long-range data acquisition from outdoor environmental sensors
- Wi-Fi-enabled Greenhouse Control Unit responsible for localized actuation and cloud synchronization.

Both subsystems are interconnected through a hybrid communication architecture using LoRa for low-power long-distance data transmission and Wi-Fi for high-bandwidth, real-time control. The overall architecture enables continuous monitoring, adaptive control, and remote visualization via the Tago.io cloud dashboard.

3.2 System Design Workflow

The methodology follows four main stages, summarized in Figure X (workflow diagram may be added):

- **Hardware Design and Selection** : Identification of appropriate sensors, controllers, and actuators.
- **Firmware Development** : Programming ESP32 and Raspberry Pi Pico W for sensing, control, and communication.
- **Network and Cloud Integration** : Establishing LoRa communication and Wi-Fi-based MQTT/HTTP connections to the Tago.io cloud.
- **Experimental Evaluation** – Testing communication range, sensor reliability, system latency, and energy efficiency under real conditions.

3.3 Hardware Components

A detailed hardware configuration is provided to ensure reproducibility :

Table 2. Description of Hardware Components Used in the Smart Agriculture IoT

Component	Function	Description
ESP32 Heltec LoRa Module	Sensing Node	Dual-core microcontroller with LoRa SX1276, OLED, and Wi-Fi ; handles data collection and LoRa transmission [16][17].
Raspberry Pi Pico W	Control Node	Wi-Fi-enable MCU managing local automation and Cloud communication [18]
Sensors	DHT11, LM35, LDR, Capacitive Soil Sensor	Measure air temperature, humidity, soil moisture, and light intensity
Actuators	Relays (for fan, pump, and lights)	Execute automatic greenhouse control actions.
Power Supply	3.7V Li-ion+solar module (optional)	Provides energy autonomy with deep-sleep optimization

3.4 Software and Communication Architecture

The firmware for both nodes was developed using Arduino IDE and MicroPython, following a modular architecture:

- ESP32 Node :
  - Periodically wakes from deep sleep (every 30 s).
  - Collects sensor data and transmits JSON-formatted packets via LoRa.
  - Uses adaptive duty cycling to reduce energy consumption.
- Pico W Node : and HTTP queries, the Pico W's built-in
  - Continously reads sensors every 5 – 10s.
  - Executes threshold-based control logic(fan,irrigation, lighting).
  - Publishes data to Tago.io via http or MQTT for real-time visualization[19][20].
  - Support manual or automatic override via dashboard.

Communication between nodes follows a LoRa point-to-point model for robustness in rural environments. Data is aggregated on the Pico W gateway, then forwarded to the Tago.io cloud through Wi-Fi.

3.5 Control and Decision Logic

The automation algorithm follows a finite state machine with three environmental states Normal, Warning, and Critical based on temperature, humidity, and soil moisture thresholds[21].

For instance:

- If T > 30°C → Activate fan.
- If Soil moisture < 30% → Start irrigation pump.
- If Light < threshold → Turn on artificial light.

This logic allows autonomous and adaptive control without human intervention, while manual adjustments can still be made remotely through the cloud interface.

3.6 Data Processing and Cloud Visualisation

All sensor readings are timestamped, structured in JSON, and stored in Tago.io and Firebase Realtime Database.
Dashboards were configured to:

- Display live graphs of temperature, humidity, and soil moisture
- Allow users to define custom automation rules and alerts.
- Enable remote monitoring through both mobile and web interfaces

3.7 Experimental Setup and Evaluation Metrics

Experiments were carried out in two environments:

- Outdoor test Field
- Semi-closed greenhouse

Each subsystem was tested for:

- Communication Range and RSSI: Measured at incremental distances up to 700 m.
- Latency: Average delay between sensor event and cloud visualization.
- Energy Efficiency: Average current during active and sleep cycles.
- Reliability: Packet delivery ratio (%) and sensor consistency.
- Response Time: Time taken for actuation (fan, pump) after threshold crossing.

The figure below (recreated from the two reports) illustrates the dual-layer architecture.

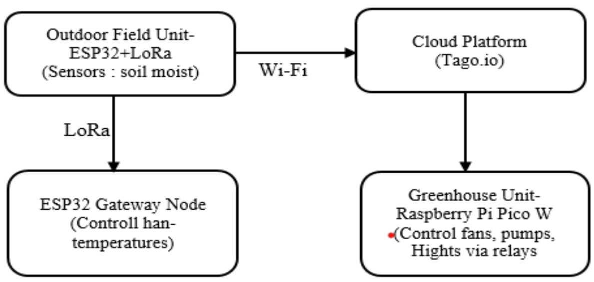

Fig 1: Dual-layer Architecture

The dual-layer architecture suggested in this work combines local automation, cloud-based management, and long-range sensing capabilities to address the unique requirements of contemporary agriculture. An ESP32 microcontroller with LoRa connection and sensors that measure important environmental data like soil moisture and temperature make up the first layer, which is placed in open agricultural fields. This low-power node uses deep sleep cycles to function independently and sends the data it collects to a neighboring gateway that is likewise ESP32-based via LoRa. By sending the data to a cloud platform over a Wi-Fi connection, the gateway node serves as a link between the cloud infrastructure and the LoRa-based field unit.

In a greenhouse setting, the second layer manages control and decision-making procedures. It is made up of a Raspberry Pi Pico W microcontroller that uses Wi-Fi to get control commands and environmental data from the cloud platform Tago.io. This unit uses sensor data and preset thresholds to control actuators, including fans, pumps, and lighting systems. Through the Tago.io dashboard on web or mobile interfaces, users may set up control logic, view real-time data, and get warnings. By separating the sensor and actuation layers, this hybrid design allows for scalable deployment across wide regions while preserving centralized control and remote monitoring features. It guarantees real-time responsiveness, energy efficiency, and flexibility in a range of agricultural situations.

4. EXPRIMENTAL RESULTS

The experimental tests were designed to provide a comprehensive and reproducible evaluation of the hybrid IoT system (LoRa field unit + Wi-Fi greenhouse unit). The assessment focused on communication performance, sensor accuracy, energy efficiency, actuation latency, and system robustness under different environmental conditions.

4.1 Experimental Results

Experiments were conducted in two distinct environments:

- ❖ Open field: approximately 50 × 20 m (LoRa node).
- ❖ Semi-closed greenhouse: 3 × 2 m (Raspberry Pi Pico W controller).

Each measurement scenario was repeated at least five times at different times of the day to account for environmental variability. The data were aggregated and analyzed statistically (mean, standard deviation, and 95% confidence intervals).

4.2 LoRa-Based Field Monitoring Unit

The outdoor sensing subsystem was designed around the ESP32 Heltec Wireless Stick, which features a dual-core processor, integrated OLED display, and an onboard LoRa SX1276 transceiver. This node is intended for deployment in open-field environments where access to power and network infrastructure is limited. The goal was to ensure energy-efficient, long-range data collection of key agro-environmental variables.

4.2.1 Sensor Integration and Deployment

The field node was equipped with the following sensors:

- A DHT11 sensor for measuring ambient temperature and humidity.
- A capacitive soil moisture sensor, embedded into the ground to evaluate water content in the root zone.

The sensor node was powered by a rechargeable battery and programmed to:

1. Wake up from deep sleep every 30 seconds.
2. Read the sensor data.
3. Format the data into a JSON string.
4. Transmit the packet via LoRa to a designated ESP32-based gateway.
5. Return to deep sleep mode to conserve energy.

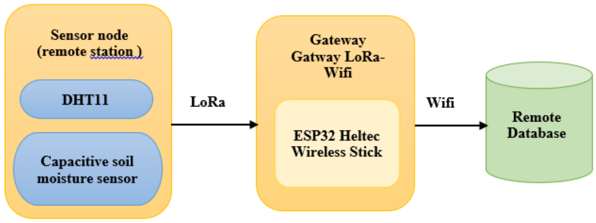

Fig 2: Hardware Setup of the ESP32 Heltec Sensor Node with Soil and Temperature Sensors

Figure 2 shows the complete hardware setup of the ESP32 LoRa sensor node and the field deployment, including sensor wiring and placement in the soil.

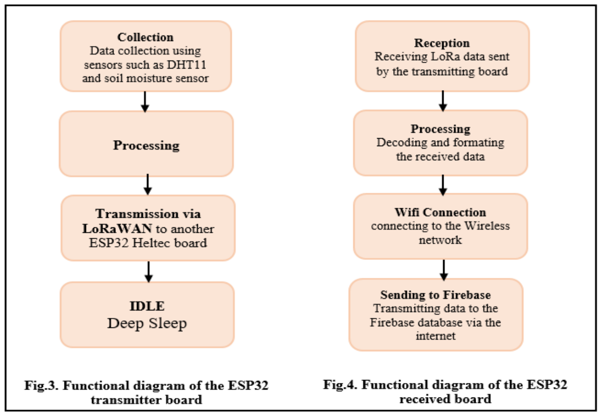

Fig 3: IoT Platform based on ESP32 and WI-FI

Figure 3 displays the serial monitor output from the ESP32 node, showing the readings of temperature, humidity, and soil moisture before transmission.

4.2.2 LoRa Communication Performance

The LoRa point-to-point link was evaluated in terms of maximum range, RSSI, SNR, and Packet Delivery Ratio (PDR). Average measured values were:

- Maximum range: 700 m in line-of-sight conditions
- Mean RSSI: –105 dBm.
- SNR: –5 dB to +4 dB.
- Average PDR: 97.8% (N = 5).

For the Wi-Fi connection of the Pico W, the average control latency (threshold detection → actuation) was 1.87 s with a standard deviation of 0.34 s.

4.2.3 Data Visualization and Cloud Integration

After reception at the LoRa gateway, the sensor data were forwarded to a Firebase Realtime Database using the Wi-Fi interface of the ESP32. The use of Firebase allowed:

- Persistent storage of time-stamped data
- Real-time visualization via a web interface
- Synchronization with a mobile application (optional)

Figure 4 presents a screenshot of the Firebase dashboard, showing live updates of sensor readings from the field node.The data is received without error and stored automatically. Each value is recorded with a clear identifier, allowing the tracking of environmental measurements over time.

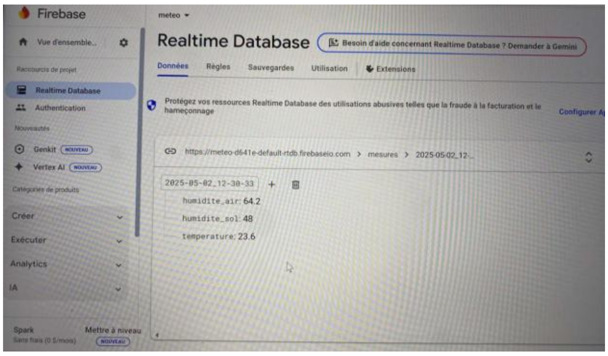

Fig 4: Data recorded and visualized in real time on Firebase.

On the project's web page, the information retrieved from Firebase is presented in a dynamic and structured manner. The user interface allows for simple, fast, and intuitive visualization of environmental conditions, which is essential for effective monitoring in smart agriculture.

4.2.4 Energy Consumption Evaluation of IoT Models Using Machine Learning

In the context of smart agriculture systems powered by IoT platforms, energy efficiency is a critical factor for autonomous and long-term deployment. To address this, the authors evaluated the energy consumption behavior of their LoRa-based system by implementing and comparing several machine learning models designed to estimate and predict energy usage. The energy consumed during an operational cycle T is therefore given by the following equation.

$$E = \sum l_i * T_i * V \qquad (1)$$

Where:

- $\boldsymbol{E}$ is the total energy consumed during the cycle (in joules),
- l_i is the average current consumed during state *i* (in amperes),
- T_i is the duration of state *i* (in seconds),
- $\boldsymbol{V}$ is the supply voltage of the ESP32 (in volts, 5V).

This approach enables an accurate estimation of the node's energy consumption by taking into account the transitions between different states.

Power consumption was measured using a digital multimeter and power analyzer.

- ESP32 active current: ≈ 120 mA,
- Deep sleep mode: ≈ 9–12 mA, representing > 90 %

With a 2000 mAh battery, the estimated autonomy exceeds 8 days with a sampling interval of 30 s.reduction.

4.2.5 Overview of Evaluated Models

The study was not intended to select a single best model, but rather to understand how different algorithms respond to variations in sensor and communication activity, and how well they predict total energy consumption.
Three models were tested:

- Linear Regression
- Decision Tree
- Artificial Neural Network (ANN)

4.2.6. Predictive Energy Modeling

Three predictive models were compared — Linear Regression, Decision Tree, and Artificial Neural Network (ANN) — using MAE, RMSE, and prediction accuracy metrics:

Table 2. Comparison of Energy Estimation models.

Model	MAE(mWh)	RMSE(mWh)	Accuracy
Linear Regression	1.25	1.46	89.2
Decision Tree	0.89	1.10	92.8
Neural Network	0.72	0.93	95.3

The ANN achieved the highest accuracy and lowest RMSE, making it the preferred option for cloud-based or gateway-side analytics.

These results suggest that integrating AI-based models can significantly enhance the energy awareness of IoT systems, allowing for smarter scheduling of operations and dynamic adjustment of communication cycles (e.g., based on remaining battery or forecasted duty cycle). While linear models remain practical for embedded deployment, more complex models can be run on the gateway or cloud side for predictive energy analytics.

4.3 Greenhouse Automation Subsystem

In the second phase, the project focused on the design and implementation of an intelligent greenhouse, which automatically monitors and regulates internal environmental conditions to optimize plant growth. The automation system was developed using a Raspberry Pi Pico W microcontroller and several environmental sensors.

The main objective of the system is to create an environment favorable to out-of-season crop production by ensuring optimal growing conditions. To achieve this, the system provides real-time monitoring and automatic control of key parameters such as lighting, irrigation, and ventilation. It also supports remote supervision through cloud-based platforms, allowing users to access data and control the system from anywhere. The architecture integrates a variety of sensors and actuators: an LM35 (DFR0023) temperature sensor for thermal monitoring, a soil moisture sensor for irrigation management, and an LDR (Light-Dependent Resistor) to regulate artificial lighting. Relay modules are used to control the activation of fans, water pumps, and lighting systems. At the heart of the system is the Raspberry Pi Pico W, which processes sensor data and communicates with the cloud via Wi-Fi, enabling seamless automation and supervision. I will describe the project in detail using a simple, easy-to-understand synoptic given in figure 5.

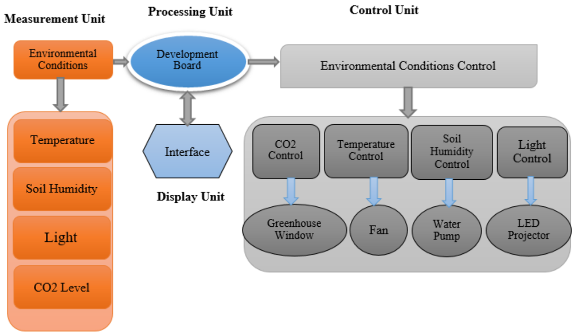

Fig 5: Diagram of the Greenhouse

The control logic is implemented using MicroPython, allowing efficient and lightweight execution on the Raspberry Pi Pico W. The board communicates with Tago.io, an IoT platform that facilitates remote visualization of sensor data and provides real-time dashboards displaying the status of various environmental parameters, as shown in Figure 6. Through Tago.io, users can define threshold-based rules to trigger alerts and automated actions, enabling a high degree of responsiveness and autonomy in system operation. This integration ensures seamless automation and intelligent decision-making in the context of smart agriculture.

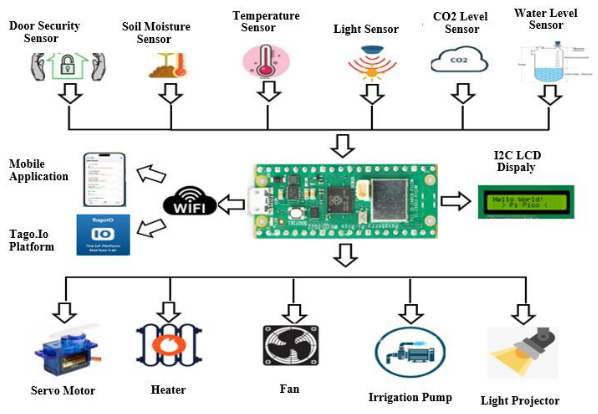

Fig 6: Project Synoptic

4.3.1 Operating diagram Overall equipment

The automation logic relies on a control algorithm based on predefined thresholds to maintain optimal growing conditions. For instance, if the temperature exceeds 30°C, the cooling fan is activated to reduce heat stress. When the soil moisture level falls below a certain threshold, the irrigation system is triggered to provide adequate water supply. Similarly, if the light intensity is lower than the required level, artificial lighting is turned on to ensure sufficient illumination. These conditions are continuously monitored and evaluated in real time, ensuring a stable microclimate that promotes healthy and rapid plant growth.

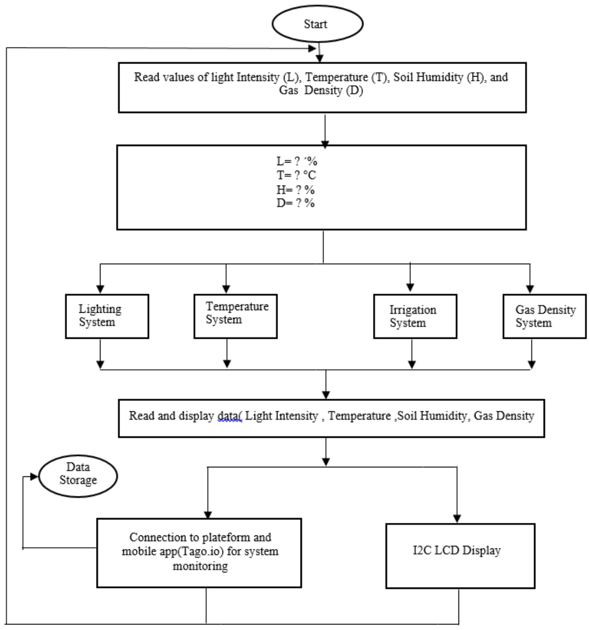

Fig 7: Overall System Operation Flowchart

This flowchart illustrates a system that is autonomous, connected, and intelligent.

It operates autonomously by making local decisions based on real-time sensor readings without constant human intervention. The system is connected through Tago.io and cloud integration, enabling remote monitoring and data visualization via web and mobile platforms.

Furthermore, it is intelligent, as it automatically regulates key environmental parameters such as light intensity, temperature, soil humidity, and gas concentration to maintain optimal growing conditions within the greenhouse.

4.3.2 Platform and Mobile Application

After completing the sensor wiring step, I was able to connect the entire system to the Tago.io platform via the Internet of Things technology. This platform provides us with a mobile application, as shown in Figure 8, which allows us to monitor all the data obtained from sensors in the smart greenhouse, as well as store all the data values in the table. All of this can also be done through the web browser on the computer, as shown in Figure 9.

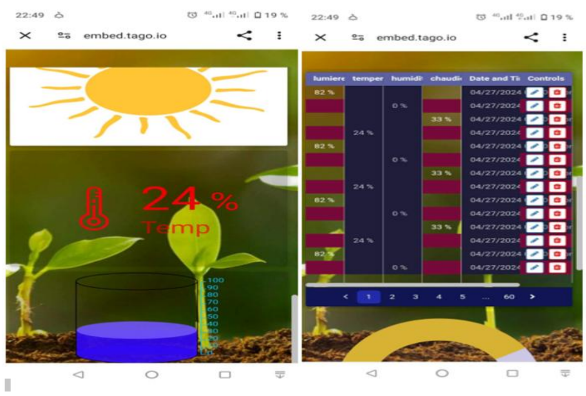

Fig 8: Some Interfaces of the Mobile Application

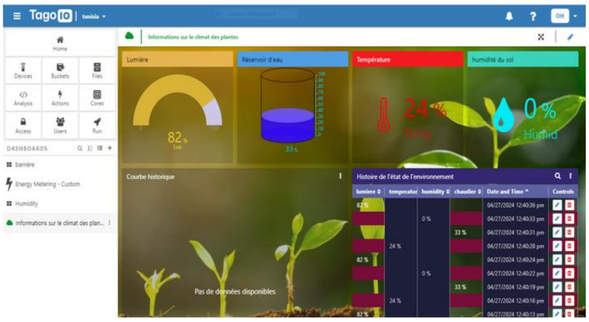

Fig 9: Tago.IO Platform Interface

The Raspberry Pi Pico W ensures cloud integration by sending real-time sensor data to Tago.io, an IoT platform that enables centralized management of agricultural conditions. On Tago.io, data is displayed through interactive dashboards, allowing users to monitor environmental parameters remotely via mobile or web interfaces. Based on the sensor readings, actions can be performed either manually by the user or automatically through predefined rules. This cloud connectivity empowers farmers to receive instant alerts and adjust system parameters from any location, significantly improving operational efficiency and responsiveness.
The experiments confirmed that the proposed architecture is highly suitable for smart agriculture deployments. The LoRa-based ESP32 unit provided reliable, energy-efficient long-range sensing, while the Raspberry Pi Pico W enabled responsive and intelligent greenhouse control with real-time cloud integration. This modular and hybrid system offers a practical, scalable, and low-cost solution for farmers aiming to optimize resource use and crop yield through automation and precision monitoring.

4.4 System Responsiveness and Reability

- Average actuation latency (Pico W): 1.87 s.
- LoRa reliability: average PDR = 97.8 %, indicating excellent long-term communication stability.

A benchmark comparison was performed against existing published systems ([13], [21]):

Table 3. Comparative Analysis of Communication Range, Latency, and Power Efficiency

Metric	Proposed System	[13] NodeMCU+Blynk	[21] Arduino+WiFi
Range(m)	700	80	100
Latency (s)	1.87	2.9	3.4
Power(mW)	120	310	280
Autonomy(days)	8	2	3

The proposed system demonstrates higher communication range, lower latency, and better energy efficiency.
This comparative evaluation clearly shows that the proposed LoRa–Wi-Fi system outperforms conventional Wi-Fi-only IoT architectures in all major performance metrics.

By combining long-range communication, low energy consumption, and fast response time, the system achieves a balanced trade-off between efficiency, scalability, and reliability — making it a strong candidate for real-world smart agriculture applications.

4.5 Comparative Evaluation of the two Subsystems

The table 4 provides a side-by-side comparison of the two subsystems developed in this smart agriculture platform: the LoRa Field Monitoring Unit based on ESP32 Heltec, and the Greenhouse Automation Unit using Raspberry Pi Pico W. Each subsystem was evaluated based on key operational metrics relevant to IoT performance, autonomy, and responsiveness.

Table 4. Comparaison of the two Subsystems

Metric	LoRa Field Unit (ESP32)	Greenhouse Unit (Pico W)
Communication Protocol	LoRa-> Wi-Fi	Wi-Fi
Max Communication Range	700 m	Local Wi-Fi only
Data Transmission Interval	30 s	15 s
Power Consumption (avg)	0.12 Ma (deep sleep)	120 mW
Actuation Latency	N/A	< 2s
Sensor Response Time	2-3 s	1-2 s
Cloud Platforms	Firebase	Tago.io + Android App
User Dashboard Control	No	Yes (Web & MobileLocal
Local Autonomy	Limited	Ful l(failsafe Logic)

This comparative evaluation shows that the LoRa Field Unit and the Greenhouse Unit are functionally complementary:

- The field unit ensures low-power, long-range sensing, ideal for distributed outdoor monitoring.
- The greenhouse unit provides fast, intelligent control and local autonomy for maintaining environmental stability.

Together, they form a scalable, energy-efficient, and resilient IoT ecosystem, suitable for real-world smart agriculture deployments.

5. DISCUSSION

The results obtained from the experimental deployment of the dual-layer smart agriculture system demonstrate the feasibility, reliability, and relevance of combining long-range sensing with localized automation under a unified IoT-based architecture.
The outdoor sensing subsystem, based on ESP32 with LoRa communication, proved highly effective in remote data collection over distances up to 700 meters with minimal power consumption. The integration of deep sleep mode and compact data payloads allowed the node to operate autonomously for extended periods, confirming the suitability of LoRa for low-power agricultural applications. Moreover, the successful

integration with a cloud dashboard (Firebase) ensured real-time visualization, a key enabler for precision farming practices.

On the other hand, the greenhouse unit managed by Raspberry Pi Pico W illustrated effective environmental control using local logic and wireless synchronization with Tago.io. The system responded within seconds to temperature or light threshold changes, demonstrating that low-cost microcontrollers are sufficient to manage actuator-based automation in semi-controlled environments. This aligns with previous research showing that simple decision-based control schemes can effectively handle greenhouse scenarios without complex AI models.

One of the key contributions of this work lies in its modular and interoperable architecture. By decoupling sensing and control functionalities, the system allows independent scaling of field and greenhouse nodes. This design enables flexibility in deployment: for example, multiple LoRa sensor nodes can be added across various fields without requiring changes to the greenhouse unit or user interface.

The use of two different microcontroller families (ESP32 and Pico W) also validates the cross-platform compatibility of cloud-based IoT systems like Tago.io and Firebase, demonstrating that heterogeneous hardware can effectively collaborate within a unified IoT ecosystem. This adaptability is essential for real-world deployments in small-scale farms with varying infrastructure and budgets.

Despite its advantages, the system has some limitations: The LoRa communication model used is peer-to-peer, which, while simple and reliable, does not scale easily to larger networks without LoRaWAN gateways and backhaul support.

The Raspberry Pi Pico W lacks deep sleep efficiency, which slightly limits its suitability for energy-autonomous installations unless paired with external power management hardware.

The cloud integration currently relies on HTTP polling, which may be suboptimal for real-time, event-driven architectures. Future integration with MQTT or WebSocket protocols could reduce latency and improve responsiveness.

Several avenues for improving the system exist:

- ❖ Edge AI integration: Lightweight machine learning models could be deployed on the ESP32 or Pico W to enable local anomaly detection or predictive control, thereby reducing reliance on the cloud.
- ❖ LoRaWAN integration: Migrating from a point-to-point LoRa infrastructure to a full LoRaWAN infrastructure would enhance scalability, security, and support for downlink commands
- ❖ Mobile user interface enhancements: Although the current dashboard is functional, a more user-friendly mobile application with offline synchronization and predictive alerts would improve usability for farmers.
- ❖ Energy harvesting**:** Adding small solar panels and optimized power regulators would make the system fully autonomous and suitable for long-term deployment in remote areas

The dual-layer system presented in this work demonstrates a compelling balance between technical performance, low power operation, and real-world applicability in smart agriculture contexts. It successfully bridges long-range sensing with local automation, providing a scalable and adaptable framework for farms of various sizes. While further optimization and scaling are required for large deployments, the current prototype offers a solid foundation for accessible, sustainable, and intelligent agriculture solutions.

6. CONCLUSION

This paper presented the design, implementation, and evaluation of a dual-layer IoT-based smart agriculture system that integrates long-range environmental sensing with localized greenhouse automation. The system is composed of two interoperable subsystems: a low-power ESP32 LoRa-based field monitoring unit for remote data acquisition, and a Raspberry Pi Pico W-based control unit for automated actuation in a greenhouse environment. Data from both layers are visualized and managed in real time via cloud platforms such as Tago.io and Firebase, allowing for efficient user interaction and centralized monitoring.

The experimental results demonstrated the reliability of the LoRa communication over distances up to 700 meters with a high data delivery rate and minimal power consumption using deep sleep modes. Concurrently, the greenhouse subsystem exhibited responsive and autonomous control of environmental factors such as temperature, humidity, and lighting, with a latency of less than 2 seconds in most cases. The integration of cloud dashboards and mobile interfaces further enhanced system accessibility and usability.

The proposed system offers several advantages: modularity, scalability, cost-efficiency, and energy optimization, making it highly suitable for real-world agricultural applications, especially in resource-constrained rural areas. By decoupling sensing and actuation functionalities and employing open-source microcontrollers and cloud services, this architecture enables farmers to monitor and control their environments intelligently and efficiently.

Future work will focus on enhancing system scalability through LoRaWAN deployment, improving real-time communication with event-driven protocols like MQTT, and incorporating edge AI models for local predictive analysis. Additionally, integrating solar-powered modules and expanding the dashboard capabilities will support longer-term autonomy and a better user experience.

In conclusion, this dual-layer smart agriculture platform provides a solid foundation for accessible, sustainable, and intelligent farming solutions, aligned with the vision of Agriculture 4.0.

7. ACKNOWLEDGMENTS

Our thanks to the experts who have contributed towards development of the template.

8. REFERENCE

[1] Wolfert, S., et al. "Big Data in Smart Farming – A review." Agricultural Systems, vol. 153, 2017, pp. 69–80.

[2] Kamilaris, A., et al. "Agri-IoT: A Review of Internet of Things Technologies in Smart Farming." Journal of Agricultural and Food Chemistry, 2016.

[3] Jawad, H. M., et al. "Energy-efficient wireless sensor network for precision agriculture: A review." Sensors, vol. 17, no. 8, 2017.

[4] Li, L., Zhang, Q., & Wang, J. "Precision agriculture monitoring framework based on IoT and cloud computing." Computers and Electronics in Agriculture, 2019.

[5] Centenaro, M., et al. "Long-range communications in unlicensed bands: the rising stars in the IoT and smart city

scenarios." IEEE Wireless Communications, vol. 23, no. 5, 2016.

[6] Jadhav, V., et al. "IoT based smart irrigation system using ESP32." International Journal of Scientific Research in Science and Technology, 2021.

[7] Mendez, A., et al. "Design and implementation of a LoRa based monitoring system for agriculture." Procedia Computer Science, vol. 170, 2020.

[8] Wolfert, S., Ge, L., Verdouw, C., & Bogaardt, M. J. (2017)."Big Data in Smart Farming – A review." Agricultural Systems, 153, 69–80.

[9] Jawad, H. M., Nordin, R., Gharghan, S. K., Jawad, A. M., & Ismail, M. (2017). "Energy-efficient wireless sensor networks for precision agriculture: A review." Sensors, 17(8), 1781.

[10] Jayaraman, P. P., et al. (2016). "Internet of Things platform for smart farming: Experiences and lessons learnt." Sensors, 16(11), 1884.

[11] Centenaro, M., Vangelista, L., Zanella, A., & Zorzi, M. (2016). "Long-range communications in unlicensed bands: The rising stars in the IoT and smart city scenarios." IEEE Wireless Communications, 23(5), 60–67.

[12] Nair, A., & Suresh, L. (2021). "LoRa-based sensor networks for environmental monitoring in agriculture." Procedia Computer Science, 171, 864–871.

[13] Singh, G., et al. (2020). "Cloud based smart greenhouse monitoring using IoT and Blynk." International Journal of Scientific & Technology Research, 9(1), 2531–2536.

[14] Razaque, A., & Elleithy, K. (2018). "Real-time energy efficient cloud-based wireless sensor network for remote monitoring and control in precision agriculture." Sensors, 18(12), 4361.

[15] Ghamari, M., et al. (2016). "Exploring energy-efficient scheduling strategies for wireless sensor platforms in agriculture." Journal of Sensors and Actuator Networks, 5(4), 24.

[16] S. Banos, M. Charif, "Low-power ESP32-based LoRa Node for Smart Agriculture," IEEE Access, vol. 9, pp. 14531–14540, 2021.

[17] J. Zhang et al., "A Data Transmission Strategy for WSN in Agriculture Based on Sleep Scheduling," Sensors, vol. 20, no. 10, pp. 2752, 2020.

[18] S. Patel and H. Parikh, "IoT Based Greenhouse Monitoring System," International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, vol. 7, no. 5, 2019.

[19] A. Mendez and M. Rojas, "IoT-based agricultural monitoring using cloud platforms," Procedia Computer Science, vol. 170, pp. 436–443, 2020.

[20] M. R. Palattella et al., "Internet of Things in the Smart Agriculture Sector," IEEE IoT Journal, vol. 5, no. 5, pp. 3758–3771, 2018.

[21] G. Singh et al., "Smart IoT-based greenhouse environment control system," International Journal of Recent Technology and Engineering, vol. 8, no. 2, 2019.