SMART SOIL MOISTURE MONITORING FOR IoT-CONTROLLED AUTOMATIC PLANT IRRIGATION

Illia Lazaruk, Student, Halyna Klym, Dr.Sc., Prof.
Lviv Polytechnic National University, Ukraine,
e-mail: halyna.i.klym@lpnu.ua

https://doi.org/10.23939/istcmtm2025.03.

Abstract. The article presents the process of development and research of an IoT-based automated plant irrigation system, which relies on measuring soil moisture levels. As the main module, the ESP32 DevKit v1 microcontroller was used, providing wireless data transmission via Wi-Fi. The device is connected to digital air temperature and humidity sensors DHT11 and DS18B20, as well as a capacitive soil moisture sensor. To implement water supply, a 5 V pump with transistor control is applied.

Data collection and visualization are performed using the ThingsBoard Cloud IoT platform with the MQTT protocol. The software was developed in the PlatformIO environment integrated with Visual Studio Code, ensuring compatibility with the Arduino Framework.

The system was tested under real operating conditions: according to the measurements, the sensor accuracy was within acceptable errors of ±5%, the average initialization time of the device was about 8 seconds, and telemetry was transmitted to the platform at 30-second intervals. The system demonstrated stable performance under varying environmental conditions and has potential for application in smart agricultural solutions.

Key words: automated irrigation, ESP32, ThingsBoard, DHT22, YL-69, MQTT, sensors.

1. Introduction

In recent years, the integration of Internet of Things (IoT) technologies into agriculture has become increasingly prominent due to the growing demand for sustainable and efficient farming practices [1-3]. Among the most transformative applications of IoT is the automation of irrigation systems, which utilize real-time data from environmental sensors to optimize water usage and enhance crop productivity [4,5]. This approach aligns with the principles of precision agriculture, aiming to minimize resource waste while maintaining favorable growth conditions for plants.

As climate variability intensifies and freshwater resources become more strained, the need for intelligent water management solutions in agriculture is more urgent than ever. Traditional irrigation techniques, often based on fixed schedules or manual operation, lack adaptability to changing weather conditions or soil moisture levels, resulting in either excessive or insufficient watering. Such inefficiencies not only waste water but can also negatively affect plant health and yield quality.

To address these challenges, IoT-based systems integrate a network of sensors, microcontrollers, and communication modules to enable autonomous monitoring and control of irrigation processes. Existing research highlights various implementation strategies, from GSM-based irrigation monitoring systems [6] to sensor networks built on Arduino platforms [7], all aiming to improve decision-making and reduce manual intervention. Furthermore, the use of lightweight communication protocols like MQTT and cloud-based dashboards has significantly enhanced system scalability, user accessibility, and data-driven insights [8-10].

These technological developments reflect a broader shift in agriculture toward data-centric, adaptive systems that respond dynamically to real-time environmental conditions. The deployment of such systems not only contributes to water conservation but also supports the resilience and productivity of agricultural ecosystems in the face of environmental and economic pressures.

2. Problem Statement

The agricultural sector faces increasing pressure to optimize resource usage, particularly water, due to growing environmental concerns, climate variability, and the need to ensure food security for a rising global population. Traditional irrigation practices, which are often based on fixed schedules or manual assessment of field conditions, are inherently inefficient and can lead to both over-irrigation and under-irrigation. These inefficiencies not only waste valuable water resources but also negatively impact crop health, soil structure, and overall agricultural productivity [11,12].

Recent advancements in IoT technologies have opened new avenues for addressing these challenges by enabling real-time monitoring and automated control of irrigation processes. IoT-based irrigation systems typically incorporate various environmental sensors and actuators to gather data and respond to changing conditions in the field. However, despite the technological potential, practical implementation of such systems continues to encounter significant obstacles. These include high costs of deployment, inadequate sensor calibration, limited wireless communication range in rural settings, and the lack of standardized platforms for data processing and visualization [13,14].

Moreover, many existing studies focus on theoretical models or lab-scale prototypes without thorough validation under real-world agricultural conditions. As a result, there remains a gap between research-driven innovations and their practical, scalable application in the field, especially in low-resource or smallholder farming contexts [15]. There is also a limited number of studies that provide comprehensive analysis of system performance over time, including aspects such as data transmission reliability, sensor accuracy in fluctuating environments, and overall system resilience.

Therefore, there is a critical need for the development of a robust, cost-effective, and scalable IoT-based irrigation system that integrates reliable sensor networks, efficient communication protocols, and cloud-based platforms for data management. Such a system should not only automate irrigation based on soil moisture and environmental parameters but also ensure consistent operation in real-life agricultural settings. Addressing this need would contribute to advancing precision agriculture and improving water use efficiency, especially in regions vulnerable to drought and limited access to advanced infrastructure.

3. Goal

The mail goal of this study is to design and implement a comprehensive IoT-based system for automated plant irrigation, which utilizes sensor data on soil moisture, ambient temperature, and air humidity to enable precise and timely activation of irrigation, efficient water resource management, and climate control within the growing environment. The system is intended to automatically operate a water pump, a heater, and a cooler based on real-time environmental data, ensuring optimal plant conditions while minimizing manual intervention. Additionally, the system is to be integrated with a cloud-based IoT platform to allow for remote monitoring and control, as well as real-time visualization and adjustment of system parameters via an intuitive user interface. This work seeks to develop a cost-effective and flexible solution that supports smart agriculture applications by combining sensor-driven automation with cloud connectivity for enhanced user access and environmental adaptability.

4. Selection of tools for system implementation

For the implementation of an automated irrigation system capable of controlling environmental parameters and providing remote management, hardware and software components were carefully selected to ensure reliability, scalability, and energy efficiency of the system.

The core of the system is the ESP32 DevKit v1 microcontroller, chosen for its high computing power, built-in Wi-Fi and Bluetooth modules, as well as a sufficient number of digital and analog inputs/outputs. These features allow simultaneous connection of multiple sensors and actuators without the need for additional expansion modules. In addition, ESP32 supports a wide range of communication protocols, which is important for seamless integration with cloud-based IoT platforms.

For collecting soil and air data, sensors were selected with regard to their accuracy, response time, and compatibility with the microcontroller. Soil moisture is measured using the YL-69 sensor, which generates an analog signal corresponding to the soil conductivity. Air temperature and humidity are monitored by the DHT22 sensor, which provides high accuracy (±0.5 °C for temperature and ±2 % for humidity) and transmits data in digital format, simplifying integration and reducing signal noise.

Control of actuators is performed via a 5 V DC water pump controlled by a relay module. The pump is automatically switched on when the soil moisture level drops below the specified threshold, ensuring timely irrigation without human intervention. In the current experimental configuration, heating and cooling functions are simulated by LEDs, which serve as indicators of the respective climate control mechanisms. These indicators are also controlled via relays, simulating the actual switching of thermal regulation systems.

For real-time data visualization and remote access, the ThingsBoard Cloud platform was chosen instead of ThingSpeak. It provides tools for collecting, storing, and graphically displaying data streams received from the microcontroller. Access to the platform is possible through a web interface, allowing users to monitor the system from any internet-connected device. Data transfer from the ESP32 to the cloud is carried out via the MQTT protocol, which is natively supported by the microcontroller and widely used in IoT applications.

The software part of the system was developed using PlatformIO IDE integrated with Visual Studio Code, and the C++ programming language. This development environment supports rapid prototyping, convenient debugging, and seamless firmware uploading to the ESP32 board. The logic of data processing and device control is implemented on the basis of decision-making rules with threshold values, allowing dynamic adjustment of operating parameters depending on conditions.

The choice of such components ensures reliable autonomous operation of the system and flexible adaptation to different types of plants or climatic conditions. In addition, the use of open-source hardware and software facilitates further updates and integration into broader smart agriculture systems.

PlatformIO IDE offers extended functionality that significantly simplifies the development of embedded systems. It supports over a thousand boards, including ESP32, Arduino, STM32, Raspberry Pi Pico, and others, enables simultaneous work with multiple microcontrollers, and provides cross-compilation for different platforms without requiring multiple development environments.

The platform integrates with Visual Studio Code and provides full support for code autocompletion, syntax highlighting, error detection, and debugging. The PlatformIO library manager simplifies the search, installation, and updating of required libraries, while all dependencies are automatically handled during project compilation.

Initialization of a new project is fast with the automatic creation of the platformio.ini configuration file, where board parameters, framework, and libraries are specified. The platform supports various debugging tools, including GDB and OpenOCD, and has a built-in serial monitor for real-time device monitoring. Integration with version control systems such as Git is also supported.

In addition, the platform provides cross-compilation and emulation capabilities, supports various frameworks, including Arduino and ESP-IDF, as well as unit testing. Thanks to its open nature and large user community, the platform is constantly evolving and has extensive documentation that facilitates rapid adoption of the tool.

In the context of implementing the automated irrigation system, this platform allows easy integration of libraries for working with different sensors, efficient use of the MQTT protocol via the PubSubClient library, and automated configuration of the ESP32 microcontroller through the configuration file.

5. Functional Diagram of the Automated Plant Irrigation System

Based on the structural diagram shown in Fig. 1, a functional model has been developed to illustrate the interaction between the hardware and software components of the automated IoT-based plant irrigation system. The structural diagram provides a comprehensive visualization of the system architecture, outlining the interconnections between key modules and enabling an assessment of how signals and data are transmitted within the system.

At the core of the system is the ESP32 (WROOM-32) microcontroller, which performs tasks of data collection, processing, and transmission, as well as controls the actuators. Connected to the microcontroller are the following sensors: DHT22 air temperature and humidity sensor and the YL-69 soil moisture sensor. Each of these sensors is responsible for monitoring microclimatic parameters that are critically important for making decisions regarding irrigation, heating, or cooling.

The values collected by the sensors are sent to the ESP32, where they are analyzed according to pre-defined threshold values. When the soil moisture level drops below the set threshold, a relay module is activated, which in turn switches on a mini submersible water pump to start irrigation. In cases of sharp temperature fluctuations, the microcontroller sends signals to the heating or cooling modules, maintaining optimal conditions for plant growth.

In addition to local control, the system transmits data to the ThingSpeak cloud IoT platform, which provides a convenient interface for data collection, visualization, and analysis. Through this platform, users can track historical changes in parameters, adjust threshold values, and remotely monitor the system's operation.

The core of the device consists of sensors that collect information on soil moisture and air parameters. The actuator responsible for irrigation is the pump. From a technical standpoint, the operation of the heater and cooler was simulated using LEDs.

The following components were used to assemble the device: ESP32-WROOM-32 microcontroller, YL-69 soil moisture sensor, DHT22 air temperature and humidity sensor, two KY-016 RGB LEDs, mini submersible DC 3–6V water pump, relay module, breadboard, and a set of jumpers. The external appearance of the assembled device is shown in Fig. 2.

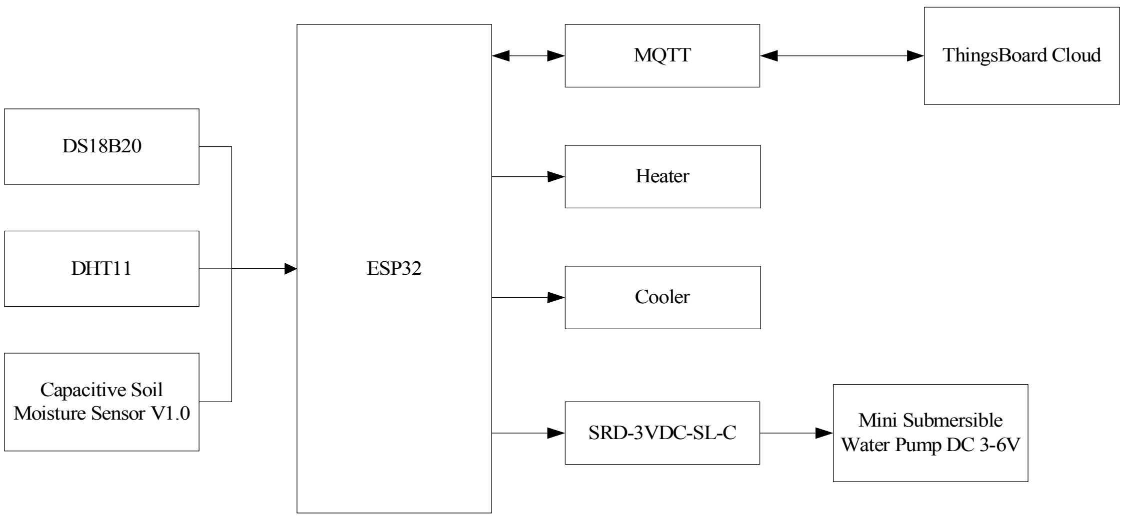

Fig. 1. Структурна схема IoT системи поливу

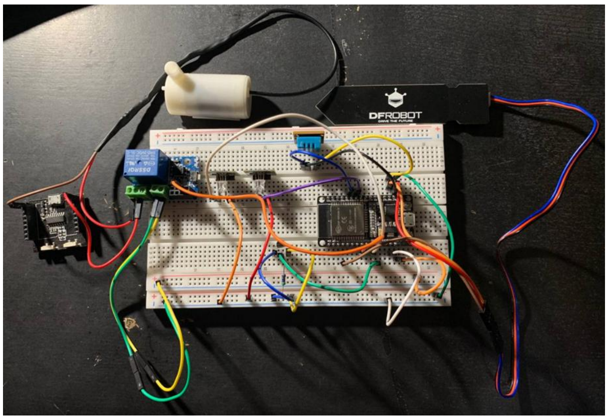

Fig. 2. Irrigation system design

For project setup, the PlatformIO IDE integrated with Visual Studio Code was chosen. This solution provides convenient library management, support for various microcontrollers, and easy integration with version control systems. Unlike Arduino IDE, PlatformIO allows working with larger-scale projects and automates the build and testing processes, making it more suitable for the system under study.

6. System operation algorithm

The overall algorithm of the automatic plant irrigation system, which monitors soil moisture levels, is designed so that the device can autonomously manage the irrigation process, regulate temperature and humidity, and transmit data to the ThingSpeak IoT platform for monitoring and analysis.

Upon powering on, the device connects to the Wi-Fi network using the specified parameters (SSID and password). Next, the YL-69 and DHT22 sensors, as well as the actuators, are initialized. After this, data exchange with ThingSpeak is established, and measurements are transmitted in real-time.

During operation, the system reads data from the soil moisture sensor and the temperature and humidity sensor. The collected data are compared with threshold values set by the user in ThingSpeak settings.

Regarding the operation of the pump: if the soil moisture level drops below the threshold, the device activates the pump. If the moisture reaches the set level before the scheduled pump runtime ends, the pump is switched off early.

The heating and cooling functions are simulated by LED indicators, which turn on according to temperature readings: the "heater" activates when the temperature falls below the minimum threshold, and the "cooler" activates when it exceeds the maximum threshold.

The device regularly transmits all measured parameters (temperature, air humidity, soil moisture) to Thing Speak, allowing remote monitoring. Additionally, data on pump runtime and the volume of water used are sent.

After completing all actions, the system returns to data collection and operates in a cyclical mode, continuously monitoring environmental conditions and adjusting the pump and heating/cooling indicators according to the obtained values.

Analyzing the algorithm, the main characteristics of the system can be highlighted: automation of all processes without user intervention, flexibility in settings through remote control of threshold values on ThingSpeak, and the ability to monitor device operation in real-time via the platform's web interface.

7. Software solution

Software implementation of the device is a crucial stage in creating an IoT-based automatic plant irrigation system that ensures interaction between hardware components, network, and cloud platform. At this stage, algorithms for data collection and processing from sensors, control of actuators, and transmission of information to the ThingsBoard Cloud platform are integrated.

During the software development process, necessary libraries were selected and analyzed, and the main functions and system logic were described. Special attention was given to organizing interaction between sensors, the microcontroller, and the cloud platform via Wi-Fi and MQTT protocols. The project contains two main files — the configuration file platformio.ini and the program file main.cpp. The platformio.ini file defines platform settings and board type (ESP32 with Arduino framework), lists required libraries, and sets serial monitor parameters, enabling efficient project management, automatic library installation, and development environment configuration.

The main.cpp file implements the device logic: initialization of sensors, modules, and actuators, initial settings in the setup() function, and cyclic execution of the main logic in the loop() function, which includes reading sensor data, processing received data, sending telemetry to the IoT platform, and responding to commands.

Several libraries are used to implement the system's functionality, each responsible for specific tasks. The WiFi.h library manages ESP32 connection to the Wi-Fi network, a prerequisite for data exchange with the platform. PubSubClient handles the MQTT protocol used for telemetry transmission and configuration parameter reception. For working with temperature and humidity sensors, DHT.h, OneWire, and DallasTemperature libraries are used, ensuring correct reading of respective values. The ArduinoJson library is employed for creating and processing JSON messages transmitted via MQTT, enabling structured data exchange between the device and the platform.

The software implementation includes selection and detailed analysis of libraries providing Wi-Fi, MQTT, sensor, and JSON data processing capabilities. Based on these libraries, software was developed to realize the device's core functions: connecting to Wi-Fi, reading sensor data, controlling the pump, heater, and cooler, and transmitting telemetry to the ThingsBoard platform.

The implemented IoT system represents a fully integrated automated platform for plant irrigation, taking into account soil moisture levels and climatic conditions (see Fig. 3). The system meets modern IoT solution requirements such as reliability, interactivity, automation, and ease of use.

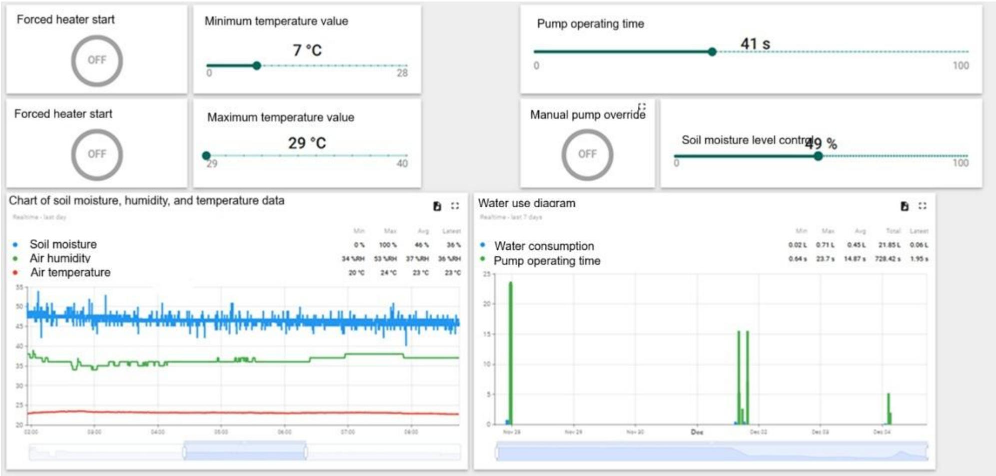

Fig. 3. View of the IoT platform dashboard for the implemented system

As noted above, the automatic plant irrigation system based on IoT is implemented using software developed with the Arduino Framework and PlatformIO IDE. Its primary task is to monitor soil and environmental conditions and control actuators to maintain optimal plant growth conditions.

At the initial stage, the device connects to the Wi-Fi network using the connectToWiFi() function, which initiates the connection, checks its status, and retries upon failure until the connection is established. After a successful connection, the device connects to the ThingsBoard cloud platform via the MQTT protocol using the connectToThingsBoard() function. This function initializes the MQTT client, subscribes to the necessary topics, and requests device attribute settings such as moisture or temperature thresholds.

The callback() function handles incoming messages from the platform; it receives JSON-formatted messages, parses them, and updates local parameters that define thresholds and operating modes. This enables real-time configuration changes without the need for reprogramming.

Main telemetry data, including soil moisture, temperature, and air humidity readings, are collected and sent to the server via the sendTelemetry() function. Data is transmitted in JSON format over MQTT, ensuring stable and structured communication with the cloud platform. Additionally, the sendWaterTelemetry() function sends information about the amount of water used and the pump operating time, helping to monitor resource consumption.

Raw analog soil moisture values are converted into percentage readings using the calibrateSoilMoisture() function, which improves accuracy in soil condition assessment and simplifies further analysis.

Actuator control is implemented through corresponding functions. Pump control is based on soil moisture thresholds and additional conditions, such as forced activation or predefined operating duration. The pump activates if the moisture level drops below the set threshold and turns off when the optimal level is reached or the set operating time expires. The system records and sends data about water usage.

Temperature regulation occurs by switching the heater or cooler on or off depending on air temperature readings and set thresholds. The heater activates if the temperature falls below the minimum threshold; the cooler activates if it exceeds the maximum threshold. Under normal conditions, both devices remain off. Forced activation of these devices is also possible via specific parameters.

The setup() function initializes all device components: it establishes serial communication for monitoring, connects to Wi-Fi and MQTT, initializes moisture and temperature sensors, and configures input/output pins for actuators. Following this, the main logic runs continuously within the loop() function, which cyclically maintains the MQTT server connection, reads sensor data, makes decisions based on received values, controls the pump, heater, and cooler, and sends telemetry data to the ThingsBoard platform.

8. Testing of the developed IoT system

Within the scope of the study, a detailed analysis of the functioning of the developed IoT-based automatic plant irrigation system was conducted. The main focus was on verifying the accuracy of the sensors, the reliability of telemetry data transmission to the ThingsBoard platform, system stability in case of network connection failures, and overall device efficiency under real operating conditions.

At the initial stage, the soil moisture sensor was evaluated. Since the sensor outputs only analog values, preliminary calibration was performed: a completely dry environment corresponded to a value of 2930, and full immersion in water corresponded to 1375. Based on these parameters, an algorithm was implemented to convert analog values to a percentage scale. During tests, the system recorded values of 0–1% in a dry environment, about 66–68% in partially moist soil, and stable 100% in wet soil. This behavior confirms the correctness of sensor calibration and that the results correspond to expected conditions.

The temperature sensor was tested by comparing its readings with those of a household mercury thermometer. Measurements were taken both indoors and outdoors. Indoors, the device recorded 23.67 °C compared to the reference value of 23 °C, achieving over 97% accuracy. In outdoor conditions, accuracy was approximately 93%, indicating high reliability of the readings.

Although precise verification of the air humidity sensor was complicated due to the lack of specialized equipment, the average recorded value was 36% RH. This corresponds to acceptable indoor humidity levels during the winter period, which typically range from 30% to 50%. Thus, the obtained results can be considered reasonable and reflective of actual conditions.

To assess data transmission efficiency, the average device initialization time—from power-on to sending the first message to the platform was measured. The result was 3.653 seconds, confirming rapid connection establishment. A network disconnection scenario was simulated by temporarily turning off the Wi-Fi router. After it was switched back on, data transmission resumed automatically without manual intervention, demonstrating system resilience to failures.

During 24 hours of real-world testing, the device continuously sent up-to-date telemetry data to the ThingsBoard platform, responded to changing readings, and ensured proper operation of actuators. Importantly, even after power outages or Wi-Fi disconnections, the device automatically restored operation without data loss.

After completing all testing stages, the system's performance was analyzed over two days of continuous operation. All components functioned stably; no data transmission failures or actuator errors were recorded. The transmitted telemetry data (Fig. 4) demonstrate stable and uninterrupted device operation, reflecting both fluctuations in readings and correct system responses to deviations from normal conditions.

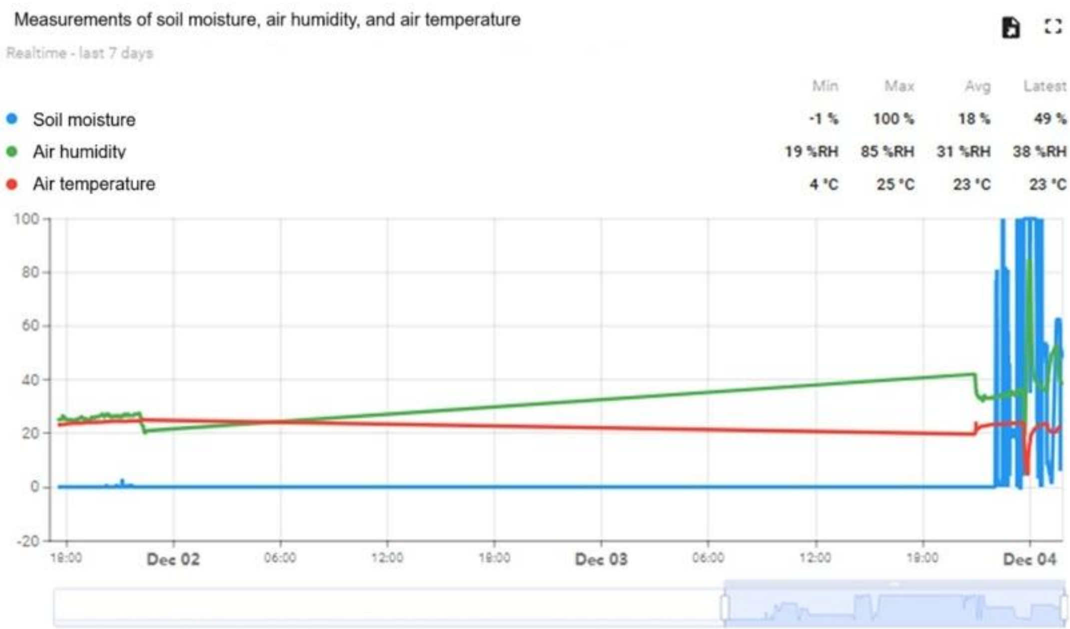

Fig. 4. Results of the IoT device performance validation over a specific period of time

9. Conclusions

In this work, an IoT-based automatic plant irrigation system was developed and investigated, which relies on measuring soil moisture and subsequently automating water supply control. The developed system integrates the YL-69 soil moisture sensor and the DHT22 temperature and humidity sensor with an ESP32 microcontroller, which transmits data to the ThingSpeak platform via Wi-Fi and manages actuators according to predefined settings. During implementation, particular attention was paid to selecting reliable hardware, using

PlatformIO IDE for software development, and integrating with the cloud platform for monitoring and remote control.

Experimental results demonstrated that the system operates stably and efficiently under real-world conditions. The sensors showed acceptable measurement accuracy, confirmed through calibration and comparison with reference devices. The system successfully adapted to environmental changes, activating irrigation on time and turning on heating or cooling indicators according to sensor readings. Data transmission to ThingSpeak occurred without significant delays, and after power interruptions or connection failures, the device automatically resumed operation, maintaining full functionality.

At the same time, the study revealed that connection stability and initialization time may vary depending on Wi-Fi network quality, which should be considered when deploying the system in environments with weak signals. Nevertheless, the overall performance and reliability of the system remain high, confirming its suitability for use in the agricultural sector, greenhouse farming, or home applications.

Thus, the developed IoT-based automatic irrigation system meets modern requirements for smart environmental monitoring systems, combining implementation simplicity with reliability and functional flexibility. Future development prospects include expanding the number of sensors, connecting additional actuators, and improving the software logic to enhance the accuracy of decision-making in dynamic conditions.

Gratitude

The authors are grateful for the support from the Ministry of Education and Science of Ukraine (Project No 0125U001883).

Conflict of Interest

The authors state that there are no financial other potential conflicts regarding this work.

References

[1] A. Rehman, T. Saba, M. Kashif, S.M. Fati, S. A. Bahaj, H. Chaudhry, "A Revisit of Internet of Things Technologies for Monitoring and Control Strategies in smart agriculture'', Agronomy, 12(1), p. 127, 2022.

[2] A.L. Duguma, X. Bai, "How the internet of things technology improves agricultural efficiency", Artificial Intelligence Review, 58(2), p. 63, 2024.

[3] A.L. Duguma, X. Bai, "Contribution of Internet of Things (IoT) in improving agricultural systems", International Journal of Environmental Science and Technology, 21(2), p. 2195–2208, 2024.

[4] A. Hafian, M. Benbrahim, M.N. Kabbaj, "IoT-based smart irrigation management system using real-time data", International Journal of Electrical & Computer Engineering, 13(6), 2023.

[5] N.C. Gaitan, B.I. Batinas, C. Ursu, F.N. Crainiciuc, "Integrating Artificial Intelligence into an Automated Irrigation System", *Sensors*, 25(4), p. 1199, 2025.

[6] A. Nawaz, M. Sadiq, Z. Ullah, "GSM Based Canal Gate and Flood Monitoring and Control System", Journal of Asian Development Studies, 13(3), p. 1432–1442, 2024.

[7] N.T. Tsebesebe, K. Mpofu, S. Sivarasu, P. Mthunzi-Kufa, "Arduino-based devices in healthcare and environmental monitoring", Discover Internet of Things, 5(1), p. 1–31, 2025.

[8] S.A. Fathima, "IoT and Big Data Ecosystems: A Comprehensive Review of Technologies, Use Cases, and Research Trends", International Journal of AI, BigData, Computational and Management Studies, 1(1), p. 11–23, 2025.

[9] S. A. Fathima, "IoT and Big Data Ecosystems: A Comprehensive Review of Technologies, Use Cases, and Research Trends", International Journal of AI, BigData, Computational and Management Studies, 1(1), p. 11-23, 2025.

[10] E. Radlbauer, T. Moser, M. Wagner, "Designing a System Architecture for Dynamic Data Collection as a Foundation for Knowledge Modeling in Industry", *Applied Sciences*, 15(9), p. 5081, 2025.

[11] J.K. Ndegwa, B.M. Gichimu, J.N. Mugwe, M. Mucheru-Muna, D.M. Njiru, "Integrated soil fertility and water management practices for enhanced agricultural productivity", International Journal of Agronomy, 2023(1), p. 8890794, 2023.

[12] C. Ingrao, R. Strippoli, G. Lagioia, D. Huisingh, "Water scarcity in agriculture: An overview of causes, impacts and approaches for reducing the risks", Heliyon, 9(8), 2023.

[13] T. Mutunga, S. Sinanovic, C.S. Harrison, "Integrating wireless remote sensing and sensors for monitoring pesticide pollution in surface and groundwater", Sensors, 24(10), p. 3191, 2024.

[14] C.M. Rosca, A. Stancu, "Integration of AI in Self-Powered IoT Sensor Systems", Applied Sciences, 15, p. 7008, 2025.

[15] G. Sinha, M. Banerjee, "Water Quality Management", Aquaculture: Trends and Techniques, International Journal of Environmental Sciences, p. 1911–1927, 2025.

[16] Priya N. Gupta, "Harvesting Tomorrow: Exploring Real World Applications of AI in Agriculture", Emerging Smart Agricultural Practices Using Artificial Intelligence, p. 163–188, 2025.