SLIIT International Conference on Engineering and Technology
Malabe, Sri Lanka, 9th to 12th September 2025

Smart Irrigation System Using Arduino and LabVIEW for Real-Time Monitoring and Efficient Water Management

Siranjeevi Vinayagamoorthy [a]*

[a]*Department of Electrical and Electronic Engineering, Faculty of Engineering, Sri Lanka Institute of Information Technology, Malabe 10115, Sri Lanka*

[a]*Email: vinayagamoorthysiranjeevi@gmail.com*

Abstract

Efficient water management is a critical challenge in agriculture, particularly in regions experiencing water scarcity and climate variability. Conventional irrigation systems, typically based on fixed schedules or manual practices, often result in over- irrigation or under-irrigation, leading to resource wastage and reduced crop performance. This paper presents the design and implementation of a Smart Irrigation System that integrates Arduino-based embedded hardware with a LabVIEW graphical interface to achieve real-time monitoring, automation, and user-driven control of irrigation. The system continuously measures soil moisture, temperature, and humidity, enabling dynamic threshold-based irrigation supported by a manual override option. Experimental validation confirmed that the system operates reliably, responds accurately to changing soil conditions, and prevents unnecessary irrigation by activating the water pump only when required. Unlike existing solutions, the proposed system provides dual operating modes, interactive threshold adjustment, and intuitive real-time visualization through LabVIEW, making it both scalable and cost-effective for small- to medium-scale farming applications. This work contributes to sustainable agricultural practices by combining affordability, adaptability, and improved water-use efficiency.

Keywords: Arduino Uno; automation; DHT22; LabVIEW; real time monitoring; smart irrigation; soil moisture sensor; water conservation

https://doi.org/10.54389/PZSM8967

1. Introduction

Agriculture consumes nearly 70% of the world's freshwater resources, and with increasing population growth and climate variability, efficient water management has become an urgent necessity. Irrigation, being one of the most critical processes in crop cultivation, often suffers from inefficiencies when carried out using traditional time- based or manual methods. Such approaches fail to respond to real-time soil and environmental conditions, frequently leading to water wastage through over-irrigation or yield losses caused by under-irrigation. This has created a strong demand for automated and intelligent irrigation solutions capable of adapting to dynamic agricultural environments.

In recent years, researchers have explored the integration of sensors, microcontrollers, and software platforms to enhance irrigation efficiency. Aswale et al. [1] demonstrated an Arduino–LabVIEW interfaced irrigation controller, but the system was primarily restricted to basic soil moisture feedback without comprehensive environmental monitoring. Barkunan et al. [2] proposed a drip irrigation system for paddy cultivation using smart sensors, yet its effectiveness across different crop types and field conditions was not fully examined. Gogoi [4] introduced a LabVIEW-monitored irrigation framework, but it lacked scalability and did not provide adaptive control strategies. More recently, Montesano et al. [5] highlighted the potential of IoT-based irrigation monitoring with cloud integration, although the higher cost and complexity of such systems limit their feasibility for small and medium-scale farms. These studies illustrate significant progress in the field, but they also reveal persistent limitations in terms of adaptability, user interaction, and affordability.

The present work aims to address these gaps by developing a cost-effective and scalable smart irrigation system that integrates real-time monitoring of multiple parameters with an intuitive decision-support interface. The proposed system is designed to continuously measure soil moisture, temperature, and humidity using low-cost sensors interfaced with an Arduino microcontroller, while a LabVIEW-based graphical user interface provides real-time data visualization, threshold configuration, and manual override features. Unlike conventional systems that operate solely on preset schedules, this solution enables automatic irrigation based on dynamic soil conditions while still allowing user control when necessary.

Through this combination of embedded hardware and virtual instrumentation, the system offers an affordable and user-friendly approach to precision irrigation. Its ability to adapt irrigation decisions to actual field conditions improves water efficiency, reduces resource wastage, and enhances crop health. Moreover, by validating the system under controlled conditions, the study demonstrates the potential of this approach to serve as a practical and sustainable solution for small and medium-scale agricultural operations.

2. System architecture

The proposed smart irrigation system is structured into three major functional components: environmental data acquisition, control and decision-making, and irrigation execution with user feedback. These components work together in a closed-loop cycle that begins with environmental sensing, proceeds through processing and decision- making, and culminates in actuation and feedback to the user. Figure [1] illustrates the overall flow of operations, showing how soil and climate parameters are continuously monitored, evaluated against thresholds, and translated into irrigation actions.

2.1 Environmental Data Acquisition

The first stage of the architecture focuses on capturing real-time environmental parameters that govern irrigation requirements. A soil moisture sensor forms the core of this layer by measuring volumetric water content in the soil. Its analog output is calibrated between two extreme conditions: completely dry soil and fully saturated soil. The calibrated output is then mapped to a percentage scale between 0% and 100%, ensuring accuracy in determining the irrigation need [3]. Since soil moisture directly influences plant growth, this parameter is prioritized in the decision-making process [2].

In addition to soil conditions, ambient temperature and relative humidity are continuously recorded using a DHT22 sensor. With a resolution of 0.1 °C and ±0.5 °C accuracy for temperature, and ±2–5% for humidity, this sensor provides reliable information on environmental dynamics that affect evapotranspiration and crop water demand [7]. Incorporating temperature and humidity ensures that irrigation control is not solely dependent on soil moisture but also considers broader climatic influences.

All sensor data are acquired by the Arduino Uno microcontroller, which acts as the central collection point. The Arduino performs analog-to-digital conversion (ADC) for the soil moisture sensor and aggregates the readings into structured packets. These packets are transmitted to the LabVIEW interface via serial communication at regular intervals, enabling continuous monitoring while balancing communication efficiency [1]. The combination of multi-parameter sensing and reliable transmission provides a strong foundation for intelligent irrigation decisions.

2.2 Control and Decision-Making Unit

At the core of the system lies the integration of the Arduino Uno with a LabVIEW-based graphical user interface, forming the control and decision-making unit. The Arduino receives and preprocesses data from the sensing layer before transmitting it to LabVIEW. Within LabVIEW, a custom Virtual Instrument (VI) interprets the incoming data stream, displaying soil moisture, temperature, and humidity in both numerical and graphical formats for real- time monitoring [4].

The system allows users to set a soil moisture threshold through a slider interface in LabVIEW. This threshold is critical for determining irrigation cycles. When the soil moisture value falls below the threshold, LabVIEW recognizes the condition as “dry” and transmits a binary signal (‘1’) back to the Arduino. The Arduino then activates the relay module to switch on the water pump. Conversely, if the soil moisture value is equal to or above the threshold, LabVIEW sends a ‘0’, ensuring that the pump remains inactive [5]. This simple yet effective binary communication scheme reduces latency, conserves bandwidth, and ensures rapid responsiveness, which is vital for real-time agricultural applications.

A significant advantage of this architecture is the inclusion of a manual override function. Farmers or users can bypass the automatic control by directly commanding the pump through the LabVIEW interface. This feature is particularly useful during system calibration, maintenance, or exceptional conditions such as sudden weather changes. The dual-mode operation—automatic and manual—provides both flexibility and reliability, addressing a limitation observed in several prior works that relied solely on automatic scheduling without user intervention [2,6].

2.3 Irrigation Execution and Feedback

The final component of the system is responsible for executing irrigation commands and providing immediate feedback to the user. The Arduino drives a single-channel relay module, which serves as an electronic switch to control the submersible water pump. When the soil is identified as dry, the relay energizes, activating the pump and initiating irrigation. Once the soil moisture level reaches or exceeds the threshold, the pump is automatically switched off, conserving water resources while maintaining optimal soil conditions [2].

To improve usability, the system incorporates visual indicators through red and green LEDs. The red LED illuminates when the soil is dry and irrigation is in progress, while the green LED signals sufficient soil moisture and an inactive pump. This provides farmers with an immediate, low-complexity means of interpreting system behavior, even without the LabVIEW interface.

This stage effectively closes the system’s control loop. Real-time sensing drives decision-making, which in turn governs actuation, while feedback ensures transparency and user trust in the system’s operation. The architecture demonstrates how embedded systems and virtual instrumentation can be integrated into a cost-effective and scalable irrigation solution, supporting the broader goals of sustainable agriculture and precision farming [5,6].

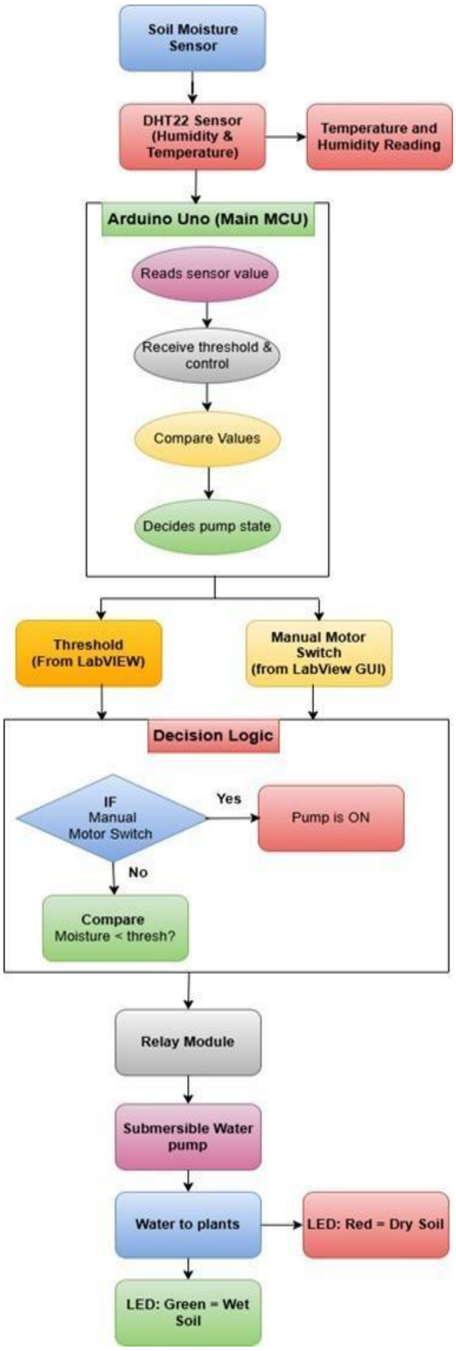

Figure 1: Flow diagram of the smart irrigation system architecture, illustrating the closed-loop process of environmental data acquisition, control and decision-making, and irrigation execution with feedback. Sensor inputs (soil moisture, temperature, and humidity) are processed by the Arduino and LabVIEW interface to determine irrigation needs, which are executed through a relay-controlled water pump with visual feedback to the user.

3. Hardware Components and Experimental Setup

The prototype comprises an Arduino Uno as the central controller, a DHT22 sensor for ambient temperature and relative humidity, an analog soil-moisture probe for substrate water status, a single-channel relay to switch a DC submersible pump, and red/green LEDs for user feedback. The Arduino is powered via USB, which also serves as the serial link to the LabVIEW host for data exchange and user interaction. The relay drives the pump according to the irrigation decision logic, while the LEDs indicate "dry/irrigate" versus "wet/hold" states to enable at-a- glance status checking in the field [1], [4], [7].

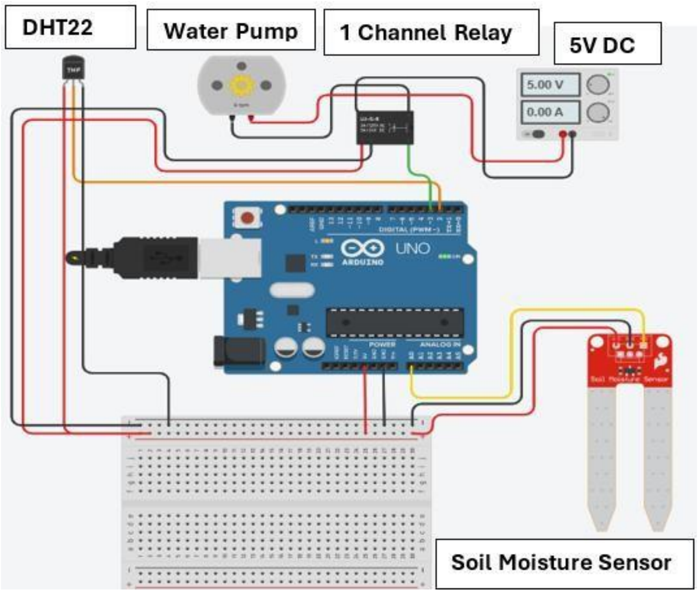

Figure 2: Complete wiring of the Arduino Uno, DHT22, soil-moisture sensor, active-low relay, indicator LEDs, and submersible pump. The schematic reflects the USB power/serial link used in bench testing.

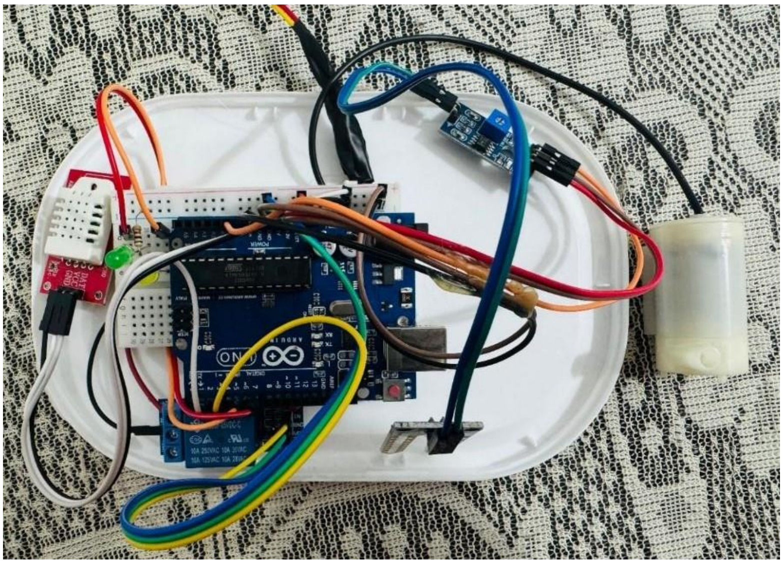

Figure 3: *implementation showing sensor placement, relay–pump connection, and visual indicators used during evaluation.*

Table 1: The materials with roles and rationale for each device used in the prototype.

Component	Description	Justification
Arduino Uno	Microcontroller	Central control unit for data acquisition
DHT22 y sensor	Temperature/Humidit	Provides accurate environmental data for analysis
Soil Moisture Sensor	Analog sensor	Determines soil dryness and irrigation need
Relay Module	1-channel relay	Electrically isolates and switches water
pump Water Pump	Submersible DC pump	Delivers irrigation water to crops
Red/Green LEDs	Visual indicators	Indicate dry or wet soil status
USB Cable communication	Communication link	Connects Arduino to PC for serial

4. Operating Principle and Control Logic

The system supports automatic and manual irrigation modes. In automatic operation, the Arduino continuously samples the soil-moisture sensor and DHT22, computes a moisture percentage from calibrated dry/wet references, and transmits the current humidity, temperature, and moisture values to LabVIEW. The LabVIEW front panel exposes a user-defined moisture threshold; when the real-time moisture drops below this threshold, a command is sent to the Arduino to energize the relay and run the pump until the threshold is met. Manual override allows the operator to force pump state regardless of the measured moisture, which is useful for maintenance and calibration [1], [4], [6], [7].

4.1 Equation

$$M = \frac{V_{raw} - V_{dry}}{V_{wet} - V_{dey}} \times 100 \quad (1)$$

The system determines the soil moisture percentage by processing the current sensor reading (V_{raw}), the dry calibration value (V_{dry}, such as 740), and the wet calibration value (V_{wet}, such as 330). This calculation is mapped so that the output always falls between 0% and 100%, ensuring valid and reliable data for further processing [3]. At the same time, the DHT22 sensor measures the ambient temperature (T) and humidity (H), and all three values humidity, temperature, and soil moisture percentage are transmitted as a comma-separated string from the Arduino to the LabVIEW interface via the serial port. Within LabVIEW, users can view real-time numeric indicators and graphical plots of these sensor readings. A slider control on the interface allows the user to set a desired soil moisture threshold (M_{thresh}). The system continuously compares the current soil moisture (M) to this threshold. If the measured soil moisture is less than the threshold ($M < M_{thresh}$), the soil is considered dry, and LabVIEW sends a command to the Arduino to activate the pump (relay ON). If the soil moisture meets or exceeds the threshold ($M \geq M_{thresh}$), the soil is considered sufficiently moist, and the pump remains off (relay OFF). Additionally, LabVIEW provides a manual override through a toggle button. When this button is activated, it sends a command to the Arduino to force the pump ON or OFF, regardless of the soil moisture reading. This manual feature is particularly useful for maintenance or special irrigation needs. The Arduino receives these control commands from LabVIEW as single-character inputs: '1' signals a dry condition (pump ON), and '0' indicates a wet condition.

(pump OFF). Based on the received command, the Arduino updates the relay state and the LED indicators accordingly. When the soil is dry and the pump is running, the red LED illuminates; when the soil is adequately moist and the pump is off, the green LED is lit. This integrated approach ensures both automated and user- controlled irrigation, providing clear feedback and efficient water management [2], [7].

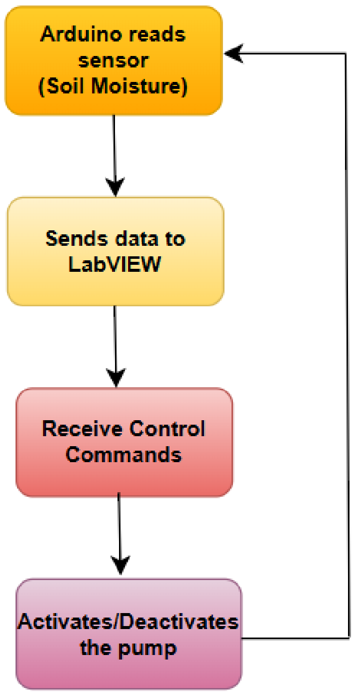

Figure 4: Serial data flow for sensing, threshold comparison, command generation, and relay actuation enabling autonomous irrigation with manual override.

5. Software Implementation

The software stack consists of (i) Arduino firmware for data acquisition, decision execution, and serial I/O, and

(ii) a LabVIEW Virtual Instrument (VI) that provides visualization, thresholding, override control, and bidirectional communication using VISA blocks [1], [4].

5.1 Arduino Firmware

The firmware initializes the DHT library and I/O pins, sets **Serial** at 9600 baud, and defaults the relay to a safe **OFF** state (logic HIGH on an active-low module). In the main loop, it samples humidity and temperature (DHT22) alongside the raw analog moisture value, converts moisture to a percentage using the calibrated endpoints, and streams a comma-separated record ⟨H, T, M⟩ to LabVIEW once per cycle. Concurrently, it listens for the one- byte LabVIEW command and updates the relay and LEDs accordingly [1], [7].

1.1 LabVIEW Virtual Instrument

The LabVIEW 2020 VI displays real-time numeric indicators and trend plots for humidity, temperature, and moisture, and exposes a “Soil Thresh” slider to configure the decision boundary without reflashing firmware. VISA Read parses the CSV stream; Boolean logic compares moisture to the threshold and drives status indicators and the outgoing command via VISA Write. A manual toggle supersedes the automatic branch to force pump state when needed [1], [4], [6].

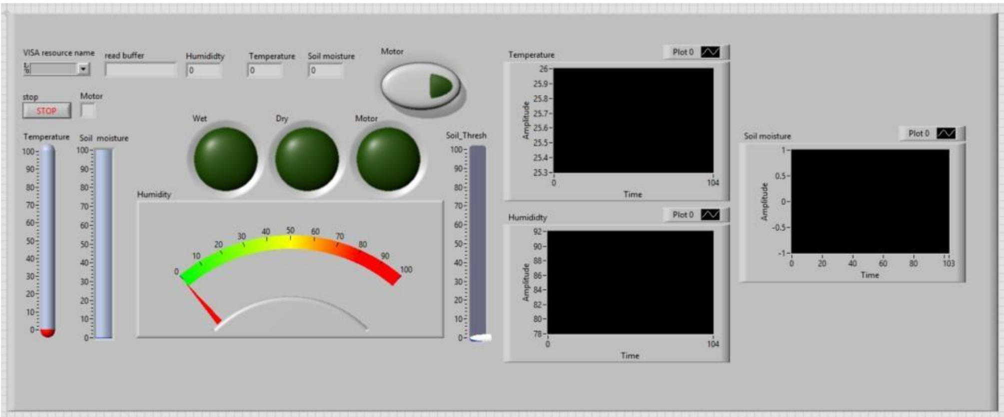

Figure 5: This figure displays the graphical user interface (GUI) of the automated plant watering system, developed using LabVIEW. The interface provides a comprehensive dashboard for monitoring and controlling the system.

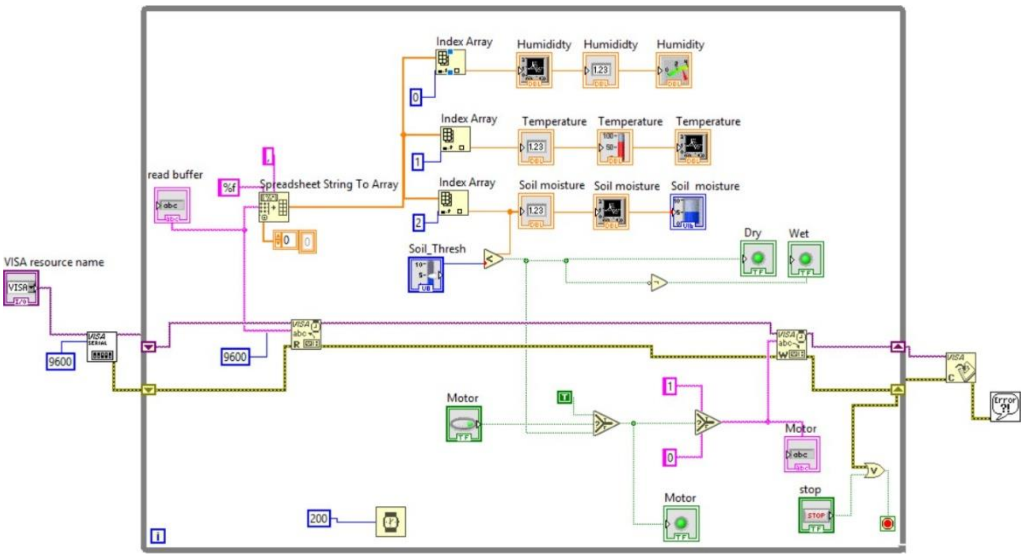

Figure 6: This figure shows the LabVIEW block diagram, which represents the programmatic logic behind the automated plant watering system. The diagram is a visual representation of the code, demonstrating how data flows through the system.

Figure 6 illustrates the LabVIEW control panel (block diagram) that governs the logic and communication flow of the smart irrigation system. This back-end representation outlines how sensor data is received from the Arduino, processed within LabVIEW, and how user inputs influence the control decisions. It also shows how commands are transmitted back to the Arduino to activate or deactivate the irrigation pump. On the left side of the diagram, the VISA resource name and VISA Read blocks initiate and maintain serial communication with the Arduino. Incoming data formatted as comma-separated values representing humidity, temperature, and soil moisture is acquired through the serial buffer. This raw string is converted into an array using the "Spreadsheet String to Array" function, after which individual sensor values are extracted using "Index Array" blocks. These sensor values are routed to both numeric indicators for visualization and logical decision blocks. The "Soil Thresh" slider enables the user to set a desired soil moisture threshold. This threshold is then compared against the real-time soil moisture reading using a comparator. If the current moisture level is below the threshold, the system logic activates the red "Dry" LED and sets the yellow "Motor" indicator ON, triggering irrigation. Conversely, if the moisture level is above the threshold, the green "Wet" LED is illuminated and the motor remains OFF. In addition to automatic control, a manual override switch allows the user to bypass the automated logic and directly control the pump's state. The final command determined either by automatic comparison or manual input is formatted as a single-character string ('1' for pump ON, '0' for pump OFF) and transmitted back to the Arduino via the VISA Write block. The processed sensor readings are also routed to graphical and numeric displays on the LabVIEW front panel, offering real-time environmental monitoring. This block diagram serves as the core engine of the smart irrigation system, enabling closed-loop control that is both autonomous and user-adjustable, thereby enhancing precision and efficiency in agricultural water management.

2. Experimental Results

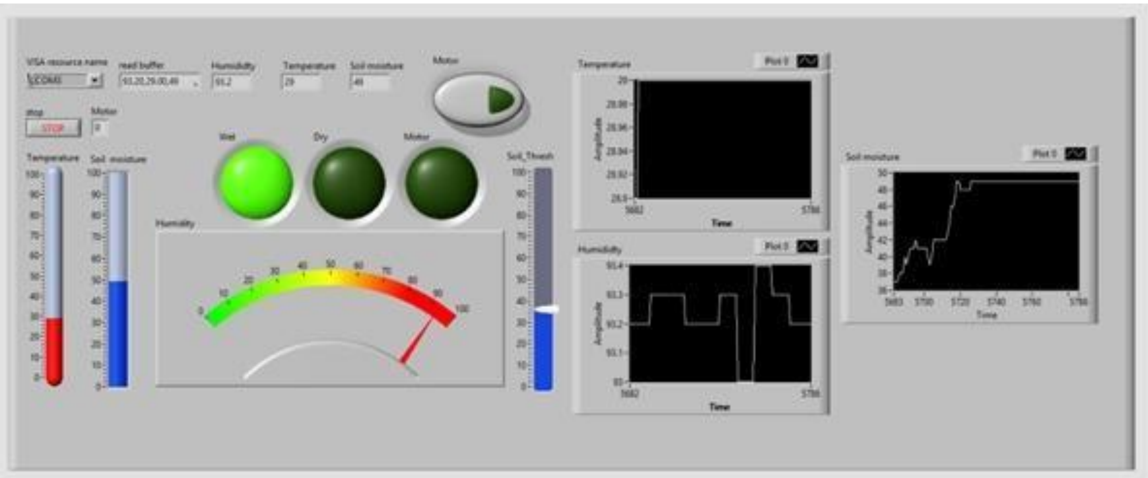

Figure 7: This figure shows the LabVIEW front panel in a live monitoring state, capturing a specific moment during the automated watering process. The interface displays real-time data and system status, demonstrating the functionality of the user interface. (Interface of LabVIEW when the threshold of the moisture is less than the moisture)

In the Figure 5, the current soil moisture percentage displayed both numerically and on the semicircular gauge exceeds the user-defined threshold set using the "Soil Thresh" slider. As a result, the system identifies the soil as adequately moist, and irrigation is not required. The green LED labeled "Wet" is illuminated, while the "Dry" and "Motor" indicators remain off, confirming that the pump is inactive. The interface includes numeric indicators for live temperature, humidity, and soil moisture readings. A semicircular gauge provides a visual representation of the soil moisture level, with the needle positioned in the green (wet) zone. The manual threshold slider allows users to dynamically adjust the moisture threshold. Additionally, real-time plots on the right-hand side of the GUI display trends in temperature, humidity, and soil moisture, supporting data-driven decision-making. The top left section displays VISA communication status, confirming active data exchange between LabVIEW and Arduino.

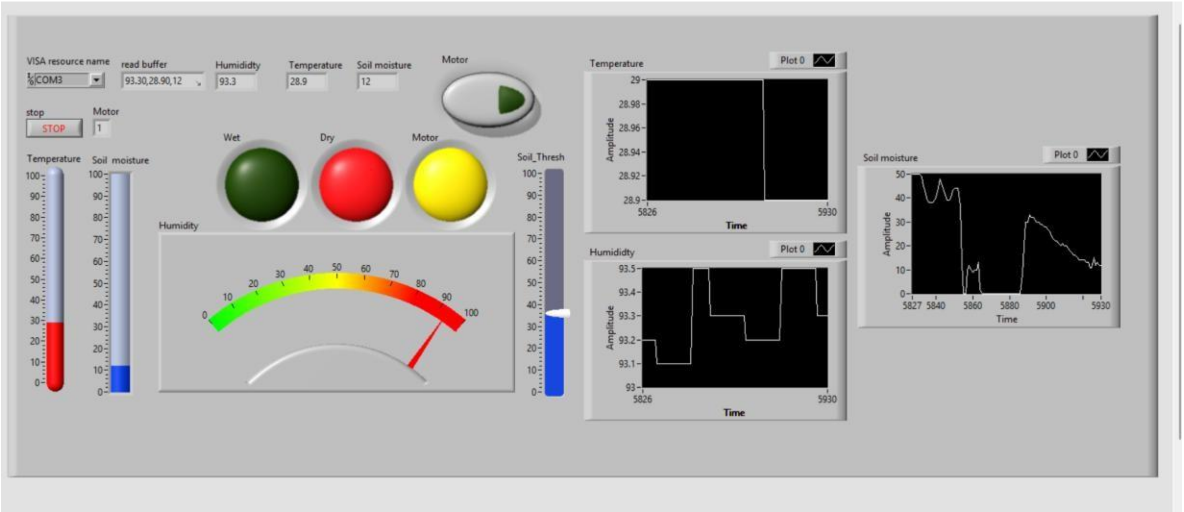

Figure 8: This figure shows a critical moment on the LabVIEW front panel where the automated system has activated the pump.

Figure 8 presents the LabVIEW graphical user interface (GUI) of the smart irrigation system operating in a scenario where the soil moisture level has fallen below the user-defined threshold. In response, the system has automatically identified the soil as dry and activated the irrigation pump to initiate watering. The interface highlights this condition through multiple visual indicators. The red LED labeled "Dry" is illuminated, indicating that the soil moisture is insufficient. Simultaneously, the "Motor" LED (yellow) is on, confirming that the pump has been triggered to irrigate the soil. The green "Wet" LED remains off, signifying that the soil has not yet reached adequate moisture levels. The semicircular soil moisture gauge shows the needle positioned in the red (dry) zone, visually reinforcing the low soil moisture condition. Numeric indicators display the real-time values of temperature, humidity, and soil moisture, with the moisture reading clearly below the threshold set by the "Soil Thresh" slider. The slider's position above the current soil moisture reading confirms the trigger condition for automatic irrigation. To the right, real-time plots of temperature, humidity, and soil moisture allow the user to observe recent environmental trends and system behavior. Additionally, the motor status indicator at the top right confirms that the pump is actively running. This figure effectively demonstrates the system’s automatic control capability: when soil moisture falls below the defined threshold, the system autonomously activates irrigation. The GUI provides clear and immediate feedback, ensuring efficient and responsive water management without the need for manual intervention.

The implemented smart irrigation system was rigorously tested under controlled environmental conditions to evaluate its functionality, reliability, and performance [1], [4]. During the testing phase, the system consistently responded to real-time soil moisture readings. When the soil moisture percentage fell below the user-defined threshold, configured via the LabVIEW interface, the Arduino successfully activated the relay to turn on the water pump [1], [4]. Once the soil moisture reached an adequate level, the system automatically deactivated the pump, demonstrating precise and repeatable control [2], [6].

The manual override feature, accessible through LabVIEW, functioned as intended, allowing the user to bypass automatic control and directly operate the pump [1], [4]. This feature proved valuable during calibration or in response to unexpected environmental changes [6].

Real-time serial communication between the Arduino Uno and the LabVIEW GUI was stable and highly responsive. Sensor data, including temperature, humidity, and soil moisture percentage, were continuously transmitted and displayed on the LabVIEW front panel through numeric indicators and graphical plots [7]. The moisture threshold could be adjusted dynamically via slider control, and the system promptly adapted its irrigation behavior according to the updated settings [1], [4].

Visual feedback through LED indicators provided clear operational status: the red LED illuminated when soil moisture was low and irrigation was active, while the green LED indicated sufficient moisture levels and that the pump was off [1], [4].

Compared to conventional time-based irrigation methods, the smart irrigation system demonstrated a substantial reduction in water usage [2], [5]. By irrigating only when necessary, based on real-time soil conditions, the system maintained optimal moisture levels, promoting healthier plant growth and improving overall resource efficiency [3], [5].

The setup operated reliably throughout testing and shows strong potential for scalability. Future extensions may include multi-zone irrigation, integration with additional environmental sensors, cloud-based data logging, and remote system access [1], [4], [6].

3. Discussion

The implementation of the smart irrigation system in this project clearly demonstrates the benefits of integrating embedded systems with virtual instrumentation to automate and optimize agricultural water management. Using the Arduino Uno as the core processing unit for environmental data acquisition and actuator control ensures real- time responsiveness and straightforward deployment [1], [4]. The LabVIEW-based graphical user interface (GUI) enhances user experience by enabling intuitive interaction, real-time visualization of sensor data, threshold configuration, and manual pump control [1], [4]. This combination of hardware and software provides a flexible,

user-configurable platform that bridges the gap between traditional irrigation practices and modern precision agriculture [2], [6].

A key achievement of the system is its ability to autonomously regulate irrigation based on soil moisture levels, ensuring water is used only when necessary. This prevents over-irrigation, which can lead to nutrient leaching and root diseases, as well as under-irrigation, which can cause plant stress and reduced yields [3], [5]. The LabVIEW interface allows users to dynamically set and adjust soil moisture thresholds, adapting the irrigation response to varying crop types, soil conditions, and climatic variations [1], [4]. This adaptability makes the system suitable for small and medium-scale farms where resource optimization is critical [6].

The system features dual operating modes: automatic mode, which minimizes user intervention while maintaining efficient irrigation, and manual mode, which provides flexibility for field testing, calibration, or specific user preferences [1], [4]. LED indicators offer immediate status feedback, allowing non-technical users to interpret system behavior easily. Additionally, the modular design facilitates future expansion, such as integrating additional sensors or supporting multi-zone irrigation [2], [5].

However, several limitations were observed during practical testing. The current implementation relies on wired serial communication between the Arduino and LabVIEW interface, which restricts deployment in large or remote fields where wireless access is preferable [1], [4]. Moreover, the system lacks data logging and historical analysis capabilities, limiting long-term environmental monitoring and irrigation optimization [5], [7]. There is also no fault notification mechanism for sensor errors or pump failures, potentially affecting reliability [7].

Despite these constraints, the system provides a robust and scalable solution that can serve as a foundation for advanced smart farming applications. Its affordability and ease of implementation make it particularly relevant for developing regions with limited access to high-end commercial systems [6].

Future enhancements can significantly increase system effectiveness, scalability, and real-world applicability. Integration of wireless communication modules such as Wi-Fi (ESP8266/ESP32), GSM (SIM800L), or LoRa would remove physical tethering and enable remote monitoring and control via smartphones or web dashboards [1], [4]. Cloud-based data storage and visualization platforms (e.g., ThingsBoard, Firebase, or Blynk) could archive sensor data, track trends, and support predictive irrigation models using machine learning algorithms [2], [5]. Multi-zone irrigation support, additional environmental sensors (pH, light, rain), and AI-based decision- making would allow more precise, automated water management [3], [6]. Finally, implementing error detection and notification systems, such as SMS alerts or LED fault indicators, would enhance reliability by alerting users to sensor malfunctions, communication failures, or pump issues [7]. These improvements would help evolve the system toward an industry-grade solution suitable for varying environmental conditions.

4. Conclusion

This paper presented the design and development of a smart irrigation system that effectively combines Arduino- based hardware with a LabVIEW-based user interface to automate and optimize irrigation processes. The system monitors real-time soil moisture, temperature, and humidity data to intelligently control a relay-operated water

pump, reducing water wastage while maintaining optimal soil conditions. User-defined thresholds and manual override controls enhance flexibility and usability, empowering farmers to make informed irrigation decisions. Experimental validation confirmed the system's reliability, responsiveness, and positive impact on water conservation and plant health. With its modular and scalable architecture, the proposed solution offers strong potential for adaptation in various agricultural settings. Future enhancements such as wireless connectivity and cloud integration could further evolve the system into a comprehensive smart agriculture platform, contributing to sustainable and precision farming practices.

References

[1] P. S. Aswale, S. Sali, S. Zarole, and P. Patil, "Automatic irrigation control system based on LabVIEW Arduino interfacing," *Int. J. Innovative Res. Technol.*, vol. 4, no. 3, pp. 261–264, 2018. [Online]. Available: https://ijirt.org/publishedpaper/IJIRT143549_PAPER.pdf

[2] S. Barkunan, V. Bhanumathi, and R. Sethuram, "Smart sensor for automatic drip irrigation system for paddy cultivation," *Comput. Electron. Agric.*, vol. 165, p. 104968, 2019. [Online]. Available: https://doi.org/10.1016/j.compag.2019.104968

[3] G. J. Gaskin and J. D. Miller, "Measurement of soil water content using a simplified impedance measuring technique," *J. Agric. Eng. Res.*, vol. 63, no. 2, pp. 153–160, 1996. [Online]. Available: https://doi.org/10.1006/jaer.1996.0016

[4] M. N. Gogoi, "An automated garden irrigation system: Controlled and monitored via Arduino and LabVIEW," *Int. J. Comput. Sci. Eng.*, vol. 6, no. 5, pp. 982–987, 2018. [Online]. Available: https://ijcseonline.isroset.org/pub_paper/195-IJCSE-03599.pdf

[5] M. Montesano, A. Parisi, P. Santamaria, and F. Serio, "Development of a smart irrigation monitoring system employing the Internet of Things," *J. Hydroinform.*, vol. 26, no. 12, pp. 3224–3240, 2024. [Online]. Available: https://doi.org/10.2166/hydro.2024.123

[6] S. Rathika, T. Ramesh, V. Priyadharshini, A. Suriya Prasanth, and P. Shanmugapriya, "Automation in irrigation – A review," *Int. J. Curr. Microbiol. Appl. Sci.*, vol. 9, no. 12, pp. 2974–2982, 2020. [Online]. Available: https://doi.org/10.20546/ijcmas.2020.912.352

[7] W. Whisnumurti Adhiwibowo, A. Firman Daru, and A. Maulana Hirzan, "Temperature and humidity monitoring using DHT22 sensor and Cayenne API," *Transformatika*, vol. 17, no. 2, pp. 209–214, 2020. [Online]. Available: https://journals.usm.ac.id/index.php/transformatika/article/viewFile/1820/1348