

Design and Experimental Evaluation of an IoT-Based Smart Greenhouse System with Closed-Loop PID Control and Cloud Integration

Vo Van An

Abstract— The paper presents the design, implementation, and experimental evaluation of a smart greenhouse system based on the Internet of Things (IoT), integrating real-time environmental monitoring, closed-loop temperature control, and data synchronization on a cloud platform. Unlike conventional systems that primarily rely on threshold control or open-loop control, the proposed system implements a traditional digital PID controller on a low-cost ESP8266 platform. A first-order thermal model based on the principle of energy balance is used to support the controller design. The developed architecture integrates environmental sensors, actuator control, Firebase cloud services, and an Android mobile application to facilitate real-time monitoring, bidirectional data synchronization, and remote supervision. Experiments were conducted on a laboratory-scale greenhouse model under various operating conditions, during which the transient characteristics of the control system were assessed using response time, settling time, and overshoot. Experimental results indicate that the response time ranges from 31 to 53 seconds and the settling time from 44 to 56 seconds. These results demonstrate the feasibility of the closed-loop temperature control method in the developed laboratory-scale greenhouse model; however, the quantitative metrics should be interpreted within the DHT11 sensor's measurement limits. The main contribution of the research lies in the integration and empirical validation of the embedded PID controller, IoT-based monitoring, bidirectional cloud communication, and mobile device supervision within a unified low-cost greenhouse platform.

Index Terms— Smart greenhouse, PID, Closed-loop control, Real-time monitoring, Cloud integration.

I. INTRODUCTION

IN the context of climate change and rising food demand, the application of advanced technologies in agriculture is becoming an inevitable trend to enhance productivity and optimize resource use. In particular, smart greenhouse models allow for the control of microclimate factors such as temperature, humidity, and light, playing a crucial role in ensuring optimal growth conditions for crops. However, current research primarily focuses on individual solutions, where monitoring and control functions are often implemented independently, leading to a lack of an integrated architecture capable of synchronizing the system components.

The rapid development of the Internet of Things (IoT) has

opened many opportunities in the field of smart agriculture. IoT enables the integration of sensors, control devices, and data processing platforms to build automated systems for remote monitoring and control, thereby continuously and accurately optimizing the cultivation environment [1-4]. Research on smart agriculture is quite diverse [5-9]. In study [5], the author proposes a system using an Arduino microcontroller to monitor temperature, humidity, and soil moisture through sensors, while controlling cooling fans, heating lamps, and water pumps as needed. In the event of an anomaly, information will be sent to the user via an Android application for timely feedback. In [6], this study proposes a smart garden system to automate the irrigation process, minimize maintenance, and eliminate the inconvenience of manual watering. The system uses Arduino, sensors, relays, valves, and pumps, powered by a dual-axis solar tracking system. Experimental results show effectiveness in water conservation, demand-based irrigation, ease of installation, and low maintenance costs. In [7], this study proposes a smart irrigation system based on a wireless sensor network (WSN), consisting of heterogeneous sensor nodes, actuators, and a management platform supporting IPv6. The system can adapt to environmental conditions and the specific water needs of each crop type, thereby optimizing home irrigation. Study [8] presents a smart crop-monitoring system that uses an integrated sensor kit and mobile connectivity. The system includes an automatically controlled water pump via an application and sensors for soil moisture, light, pressure, and DHT11, combined with a weather station providing real-time climate data. Study [9] describes a system for continuously monitoring environmental parameters to optimize crop growth and improve quality of life. The automated system replaces the manual system, with an Android application for monitoring and automating the irrigation process. NodeMCU connects sensors to collect soil data and transmits the data to Firebase via its integrated Wi-Fi. Although previous IoT-based greenhouse systems have demonstrated effective environmental monitoring, wireless connectivity, and remote supervision, several important limitations remain. Most existing studies rely on threshold-based actuator control without tightly integrating real-time monitoring and closed-loop control, provide only limited implementation details of feedback control algorithms on low-cost embedded platforms, and lack a comprehensive quantitative evaluation of control

V.V.An is lecturer with Institute of Engineering Technology, Thu Dau Mot University, Ho Chi Minh City, Vietnam (corresponding author; e-mail: anv@tdmu.edu.vn)

performance. Furthermore, the integration of real-time cloud synchronization with mobile applications for seamless remote monitoring and control has not been fully optimized. To address these limitations, the proposed system implements a digital PID controller on an ESP8266 platform. It integrates with Firebase Cloud Services and an Android application to enable real-time monitoring, data synchronization, and remote control. Unlike previous studies, the proposed approach is experimentally validated using a physical greenhouse prototype, where control performance is quantitatively assessed through standard dynamic response metrics, including response time, settling time, and overshoot, thereby providing experimental evidence of the effectiveness of closed-loop temperature regulation. Although our previous work investigated IoT-based environmental monitoring for greenhouse applications [3, 4], the present study substantially extends that work by incorporating a closed-loop digital PID controller, real-time bidirectional cloud communication, and comprehensive experimental validation on a physical prototype. Therefore, this paper represents a significant methodological and experimental extension rather than a simple incremental update.

The paper focuses on the design and implementation of a smart greenhouse system based on IoT, with comprehensive integration of environmental monitoring, PID-based temperature control, and cloud computing platforms. The system is designed to ensure low cost, ease of deployment, and scalability. At the same time, the study emphasizes evaluating the system's performance through characteristic metrics of the control problem, such as response time, settling time, and overshoot under various operating conditions. Unlike previous smart-garden implementations that mainly focused on environmental monitoring and threshold-based control, this work integrates (i) an experimentally validated first-order thermal model, (ii) a closed-loop digital PID controller implemented on ESP8266, (iii) cloud-based bidirectional synchronization through Firebase, and (iv) quantitative experimental evaluation under different operating conditions. The novelty, therefore, lies in the complete integration and experimental validation of the control system rather than in proposing a new PID algorithm. The remainder of this paper is organized as follows. Section II presents the design of the smart greenhouse monitoring and control system, including the mathematical model, digital PID controller, embedded implementation, and cloud communication architecture. Section III describes the experimental prototype and evaluates the system performance. Finally, Section IV summarizes the main findings and outlines future research directions.

II. SYSTEM DESIGN AND ARCHITECTURE

A. Mathematical Modeling and PID Controller Design

Mathematical Model of the Greenhouse Temperature System. The temperature dynamics inside the greenhouse are modeled based on the principle of energy balance. Assuming a lumped-parameter system with uniformly mixed air, the

system's thermal behavior can be approximated as a first-order system, consistent with recent studies on greenhouse climate control [10, 11]. The energy balance equation is written as follows:

$$C \frac{dT(t)}{dt} = Q_{in}(t) - Q_{out}(t) + Q_{dist}(t) \quad (1)$$

Where $T(t)$ is the internal temperature, C is the equivalent heat capacity, $Q_{in}(t)$ is the heating power, $Q_{out}(t)$ is the heat loss, and $Q_{dist}(t)$ represents disturbances from the environment. This energy balance model is widely used in thermal systems and smart greenhouses. The heating power is assumed to be proportional to the control signal:

$$Q_{in}(t) = K_h u(t) \quad (2)$$

Where K_h is the heating gain coefficient and $u(t)$ is the control signal. This linear model is commonly used in PID controlled temperature control systems [11, 12].

The heat loss to the environment is described by:

$$Q_{out}(t) = K_l (T(t) - T_{env}(t)) \quad (3)$$

Where K_l is the heat transfer coefficient and $T_{env}(t)$ is the outside environmental temperature. This expression reflects the mechanisms of convective and conductive heat transfer between the internal and external environments [10], [13].

Substituting (2) and (3) into (1), we obtain the kinetic equation:

$$C \frac{dT(t)}{dt} + K_l T(t) = K_h u(t) + K_l T_{env}(t) + Q_{dist}(t) \quad (4)$$

To build a model convenient for controller design, the following standard assumptions are made:

- The environmental temperature $T_{env}(t)$ varies slowly and can be considered constant over a short period.
- The disturbance component $Q_{dist}(t)$ is neglected in the nominal model and is treated as an external disturbance.

Under these assumptions, the system of equations is simplified to:

$$C \frac{dT(t)}{dt} + K_l T(t) = K_h u(t) \quad (5)$$

Applying the Laplace transform (with initial conditions set to 0), the transfer function of the system is determined:

$$G(s) = \frac{T(s)}{U(s)} = \frac{K_h}{C_s + K_l} \quad (6)$$

To verify that the greenhouse temperature dynamics can be represented by a first-order system, an experimental identification procedure was conducted using the developed greenhouse prototype. Before each experiment, the greenhouse was allowed to reach thermal equilibrium with the ambient environment. A step input corresponding to full heater power (PWM = 255) was then applied while the fan remained inactive. The internal temperature was measured using the DHT11 sensor at a 1 s sampling interval until the temperature reached steady state.

The recorded temperature response was fitted to the standard first-order model

$$G(s) = \frac{K}{\tau s + 1} \quad (7)$$

where $K = \frac{K_h}{K_l}$ is the gain coefficient, and $\tau = \frac{C}{K_l}$ is the

time constant. The parameters were estimated directly from the measured step response by determining the steady-state temperature change and the time required to reach approximately 63.2% of the final temperature rise.

The identified model was subsequently compared with an independent experimental dataset collected under the same operating conditions. The measured and simulated temperature responses showed good agreement throughout both the transient and steady-state phases, confirming that the proposed first-order approximation adequately represents the greenhouse's thermal dynamics and is suitable for PID controller design [11], [14].

The PID controller is used to adjust the temperature in the greenhouse to follow the setpoint. Based on the developed first-order thermal model, the control objective is to minimize the tracking error:

$$e(t) = T_{set}(t) - T(t) \quad (8)$$

Figure 1 describes the structure of the PID controller. The control signal is determined according to the PID control law:

$$u(t) = K_p e(t) + K_I \int_0^t e(\tau) d\tau + K_D \frac{de(t)}{dt} \quad (9)$$

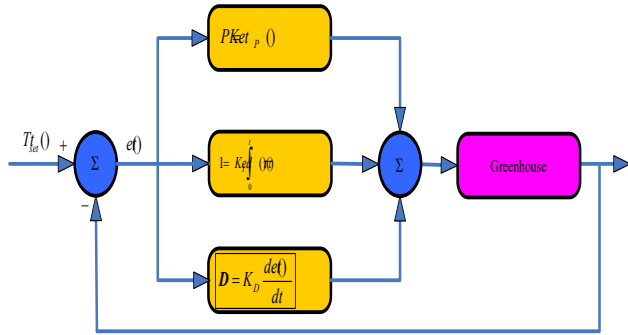


Fig. 1. Closed-loop PID temperature control architecture for the greenhouse system.

Where K_I, K_P, K_D are the proportional, integral, and derivative coefficients, respectively. The PID controller is widely used in greenhouse climate control due to its simplicity and effectiveness for first-order systems [12], [14]. Based on the first-order model obtained in (7), the PID controller is capable of ensuring tracking quality, eliminating steady-state error, and improving the transient response of the system.

The control signal is implemented as a PWM cycle in the range [0,255], and the influence of environmental disturbances may not be mitigated by classical tuning methods such as Ziegler-Nichols. Therefore, the PID parameters are determined through an experimental tuning method based on the system's real-time response, in line with recent studies [12], [13], [15]. The tuning process is carried out iteratively by adjusting the control coefficients and observing the system response. The following principles are applied:

- Increase the proportional coefficient K_p to improve the

response speed.

- Adjust the integral coefficient K_i to eliminate steady-state error.
- Fine-tune the derivative coefficient K_d to reduce overshoot and oscillation.

The integral coefficient is chosen to be relatively large to compensate for the system's high thermal inertia, thereby reducing steady-state error. Meanwhile, the proportional and derivative coefficients are set to average values to ensure stability and limit oscillations caused by disturbances.

B. Digital PID Controller Implementation

Although a first-order model can approximate the greenhouse thermal dynamics, the controller is implemented digitally on the ESP8266 microcontroller. Considering the maximum sampling frequency of the DHT11 sensor (1 Hz), the control algorithm is executed with a fixed sampling interval of

$$T_s = 1s$$

which ensures that each control action is computed immediately after a new temperature measurement becomes available.

The digital PID controller is implemented using the discrete-time control law [16]

$$u(k) = K_p e(k) + K_I T_s \sum_{I=0}^k e(I) + \frac{K_D}{T_s} [e(k) - e(k-1)] \quad (10)$$

where $e(k) = T_{ref} - T(k)$ is the temperature error at the k -th sampling instant, $u(k)$ is the controller output, and K_I, K_P, K_D denote the proportional, integral, and derivative gains, respectively.

To prevent excessive accumulation of the integral term during large temperature deviations or actuator saturation, an integral anti-windup mechanism limits the accumulated integral value within predefined bounds [17, 18].

$$H(k) = \min(H_{max}, \max(H_{min}, H(k))) \quad (11)$$

Where H_{min} and H_{max} represent the lower and upper saturation limits of the integral component.

Because the DHT11 sensor exhibits limited measurement accuracy ($\pm 2^\circ\text{C}$) and a sampling frequency of only 1 Hz, sensor noise is reduced before control computation using a moving-average filter [16].

$$T_f(k) = \frac{1}{N} \sum_{I=0}^{N-1} (T(k-I)) \quad (12)$$

where $N=5$ consecutive samples are averaged, the filtered temperature $T_f(k)$ is subsequently used for error calculation and PID control.

The controller output is converted into an 8-bit pulse-width modulation (PWM) command.

$$PWM(k) = \min(255, \max(0, u(k))) \quad (13)$$

which directly regulates the heating element through the L298N driver. The PWM duty cycle is therefore constrained within the hardware operating range of 0-255.

The actuator logic is implemented using a complementary heating-cooling strategy. When the measured temperature is below the setpoint, the heater is driven by the calculated PWM signal, while the cooling fan remains inactive. Conversely, when the measured temperature exceeds the reference temperature, the heater is disabled, and the ventilation fan is activated to accelerate heat dissipation. This complementary operating mode avoids simultaneous operation of the heater and cooling fan, thereby improving energy efficiency and preventing conflicting control actions.

The PID parameters are determined experimentally through a repetitive tuning process based on the observed closed-loop response of the laboratory-scale greenhouse model. During the tuning process, the proportional gain is adjusted first to achieve an appropriate response speed, followed by simultaneous adjustments of the integral and derivative coefficients along with the proportional gain as a combined parameter set to improve the overall response characteristics of the system. The final values of K_I , K_P and K_D are selected through multiple trials to achieve a suitable balance between response speed, oscillation characteristics, and the ability to control temperature at steady state under the surveyed operating conditions. Since the current study has not conducted PI-PID control experiments or separate analyses of the influence of each control component, no specific performance improvement can be attributed solely to the derivative component, and the reported results reflect the overall performance of the PID controller with the selected parameters.

TABLE I
FINAL PID PARAMETERS USED IN THE EMBEDDED CONTROLLER

Parameter	Value
Sampling interval T_s	1s
K_P	28
K_I	0.6
K_D	8
Integral limit $[H_{\min}, H_{\max}]$	[-200, 200]
PWM range	0-255

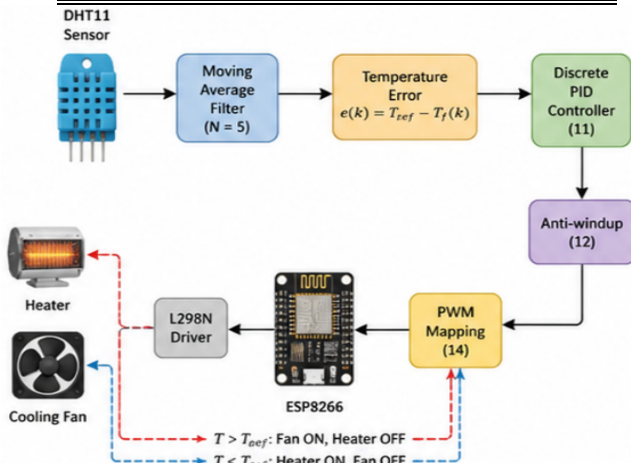


Fig. 2. Embedded digital PID implementation executed on the ESP8266 microcontroller.

Figure 2 illustrates the embedded digital PID implementation on the ESP8266. The filtered DHT11 temperature is used to compute the control error, and the discrete PID output is converted into a PWM command for the heater or cooling fan through the L298N driver. Complementary actuator logic prevents simultaneous heating and cooling, while the ESP8266 synchronizes measurements, control parameters, and actuator states with Firebase for real-time monitoring and remote configuration. This approach enables the controller to adapt to model errors and environmental fluctuations, thereby enhancing control performance under real conditions [13], [15].

C. System Block Diagram

Figure 3 illustrates the block diagram of the smart greenhouse system based on the IoT platform. The central processing unit uses the ESP8266 NodeMCU microcontroller, which plays a key role in collecting sensor data, processing signals, and controlling actuators. Temperature and humidity sensors measure environmental parameters inside the greenhouse and transmit the data to the microcontroller for analysis and processing. Based on the measured values, the controller sends control signals to the L298N module and relays to activate devices such as fans, heaters, and water pumps, thereby adjusting the microclimate within the greenhouse. The fan is used to reduce the temperature when the environment is too hot, while a PID algorithm controls the heater to maintain a stable temperature when the temperature drops. Additionally, the water pump provides moisture to the soil and can be activated automatically or remotely via a mobile application. Environmental parameters and system status are displayed directly on an LCD screen for on-site monitoring. At the same time, sensor data is also transmitted to the Firebase platform via the ESP8266's Wi-Fi connection, allowing users to monitor and control the greenhouse system remotely using a smartphone. This system structure helps maintain optimal environmental conditions for plant growth while enhancing monitoring and automation capabilities in smart agriculture.

In this study, additional sensors and actuators, including humidity monitoring functions, water pumps, and lighting systems, are integrated to verify the functionality and coordination capabilities of the overall IoT architecture. However, the quantitative assessment of control performance in the current research focuses solely on the PID temperature control loop; the remaining functions are used to verify the system's monitoring, communication, data synchronization, and device control capabilities.

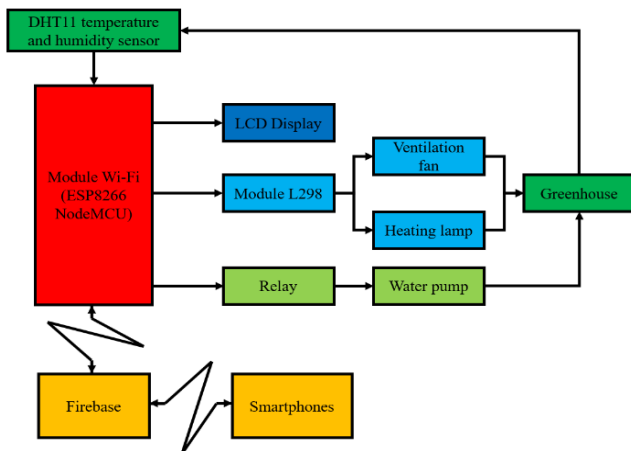


Fig. 3. Overall architecture of the IoT-based smart greenhouse monitoring and control system.

D. Hardware and Sensor Implementation

The system is built on the ESP8266 NodeMCU platform, serving as the central controller for sensor data collection, control signal computation, and data exchange with the cloud via Wi-Fi [19]. The DHT11 sensor is used to measure temperature and relative humidity in the greenhouse model. The sensor operates within a temperature range of 0–50 °C with a nominal error of ± 2 °C and a humidity range of 20–90% RH with an error of approximately $\pm 5\%$ RH [20, 21]. The sensor data is transmitted to the ESP8266 via one-wire digital communication and is used as feedback for the temperature control loop. Due to the limitations in the DHT11's accuracy and update speed, the quantitative results in the study are interpreted within the sensor's measurement capabilities.

The system's actuators include ventilation fans, heating/lighting units, and water pumps, which are controlled via a power circuit coordinated by the ESP8266. In automatic mode, the temperature control signal is determined by the PID controller based on the deviation between the set and measured temperatures. In manual mode, users can control actuator states via the mobile application. This structure enables the integration of local control with monitoring and remote control capabilities via the IoT platform.

The overall operation process of the system is presented in Figure 4. After initializing the hardware, sensors, and network connections, the ESP8266 continuously acquires temperature and humidity, displays the parameters on-site, and checks the Wi-Fi connection status. When a network connection is available, the controller synchronizes the operating mode, set values, and device status with the cloud platform. In Manual mode, the actuators are controlled by user commands; in Auto mode, the PID controller adjusts the temperature based on the deviation between the measured and set values. After each processing cycle, the sensor data and system status are updated to the cloud platform for remote monitoring. This process is continuously repeated throughout the operation.

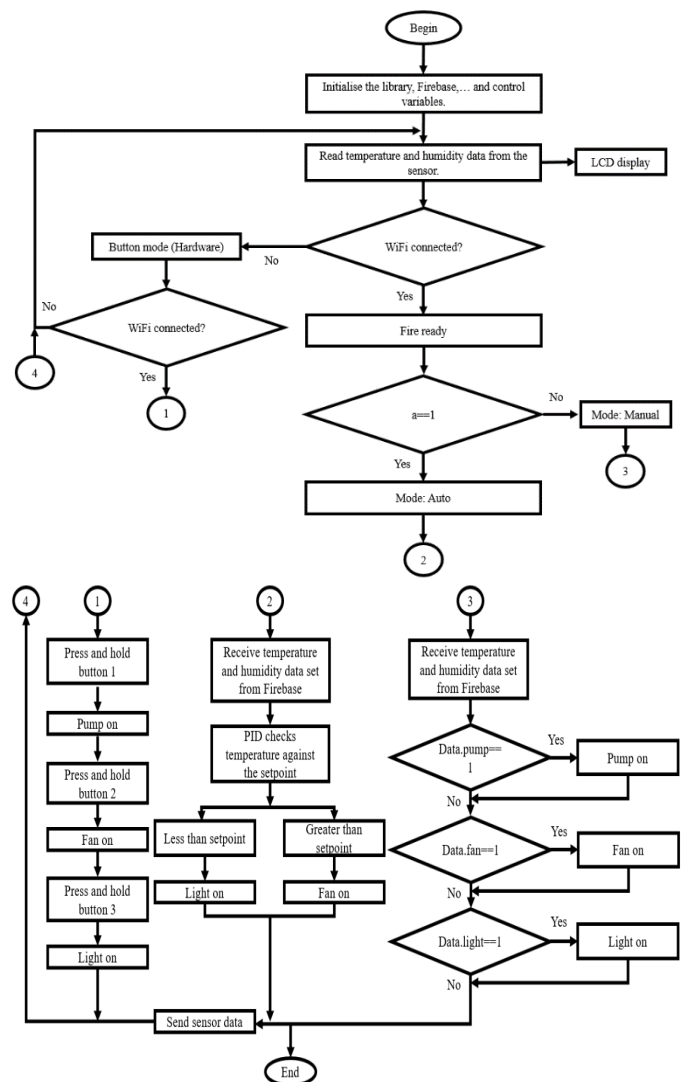


Fig. 4. Flowchart of the embedded greenhouse monitoring and closed-loop control algorithm.

On the user side, the mobile application's algorithm is shown in Figure 5. After initialization and database connection, the application continuously updates the temperature, humidity, operating mode, and actuator states. In Auto mode, the application primarily handles parameter setting, monitoring, and data display, while the PID algorithm runs locally on the ESP8266. In Manual mode, users can directly send control commands for the fan, lights, and water pump. The device statuses are synchronized with the database to maintain consistency between the user interface and the physical system.

In this study, additional sensor and actuator functions, including humidity monitoring, water pump operation, and lighting systems, are integrated to verify the functionality and coordination capabilities of the overall IoT architecture. However, the quantitative assessment of control performance in the current research is limited to the PID temperature control loop; the remaining functions are used to verify the system's monitoring, communication, data synchronization, and device control capabilities.

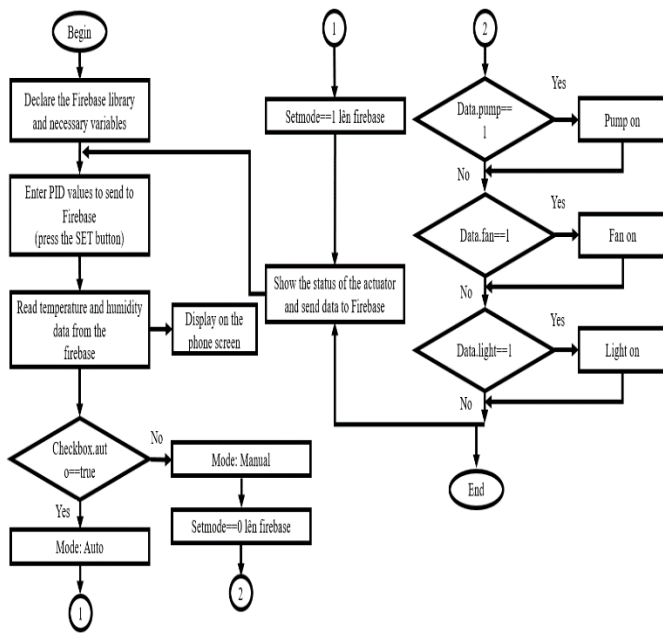


Fig. 5. Flowchart of the Android application for remote monitoring and greenhouse control.

E. Cloud Communication Architecture and Mobile Application

The IoT architecture of the system is designed to support bidirectional communication between the greenhouse model, the cloud platform, and the mobile application. The ESP8266 serves as the central controller, collecting environmental data, executing control algorithms, and transmitting data to the Firebase Real-time Database via Wi-Fi. Conversely, set values and control commands from the mobile application are synchronized through Firebase and sent to the ESP8266 for execution on the actuators. This structure enables the integration of local control with real-time monitoring and remote control capabilities (Figure 6).



Fig. 6. Overall IoT architecture and bidirectional data communication of the proposed greenhouse system.

In the proposed architecture, Firebase Realtime Database serves as the intermediate layer for data synchronization between the embedded controller and the mobile application. The Firebase library is integrated into the ESP8266 firmware to facilitate communication with the cloud database, as illustrated in Figure 7. At the same time, the necessary configuration parameters for establishing the connection are shown in Figure 8. Once the connection is established, sensor data, operating mode, actuator status, set temperature, and control parameters are exchanged between the ESP8266 and Firebase. This mechanism allows the PID controller to continue operating locally on the embedded device, while Firebase handles data synchronization and remote monitoring.



Fig. 7. Firebase software library integrated into the ESP8266 firmware for cloud communication.

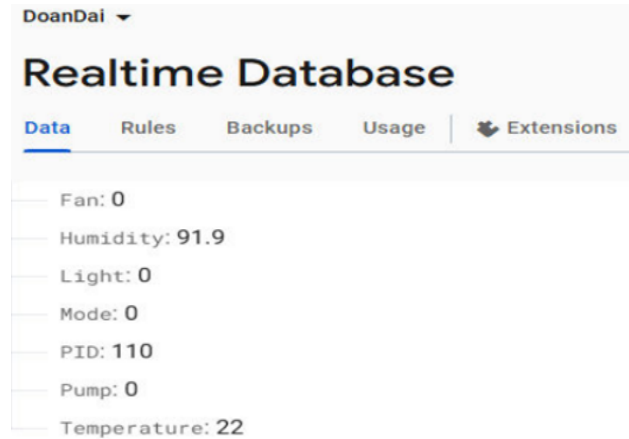


Fig. 8. Firebase configuration parameters for establishing cloud communication.

The mobile application is developed on the Android platform and connects to Firebase Realtime Database. Instead of exchanging data directly with the ESP8266, the application interacts with the system through the cloud database, thereby separating the user interface layer from the embedded control layer. When users change the set temperature or operating mode, or send control commands, the corresponding data is updated in Firebase and then received by the ESP8266 in the next processing cycle. Conversely, the measured values and device statuses updated by the ESP8266 to Firebase are used by the application to refresh the monitoring interface.

The application interface, shown in Figure 9, displays temperature, humidity, operating mode, and actuator status. Users can set the desired temperature and choose between Auto or Manual mode. In Auto mode, the temperature adjustment is handled by the PID controller embedded in the ESP8266; in Manual mode, users can directly control the devices' status through the application interface. Synchronizing system variables via Firebase helps maintain consistency between the data displayed in the application and the greenhouse model's status.

The actual data synchronization operation between the ESP8266, Firebase, and the mobile application is illustrated in Figure 10. During operation, the sensor data collected by the ESP8266 is updated to Firebase. In contrast, the set values and control commands from the application are recorded in the same database and retrieved by the embedded controller for execution. This bidirectional communication structure enables the integration of local closed-loop control with remote monitoring and configuration without requiring a dedicated web server.

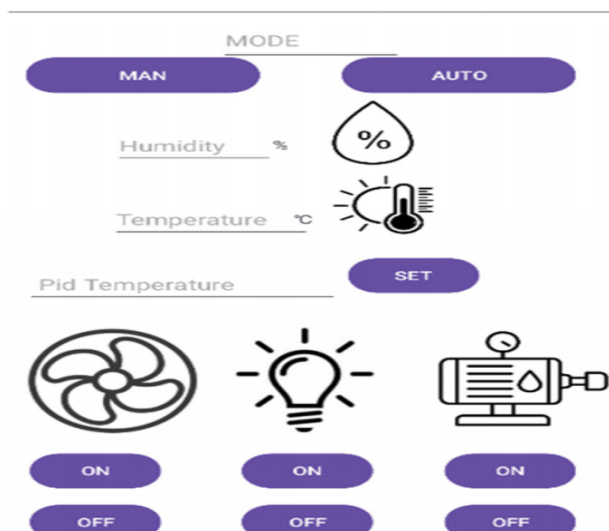


Fig. 9. User interface of the Android application for real-time greenhouse monitoring and control.

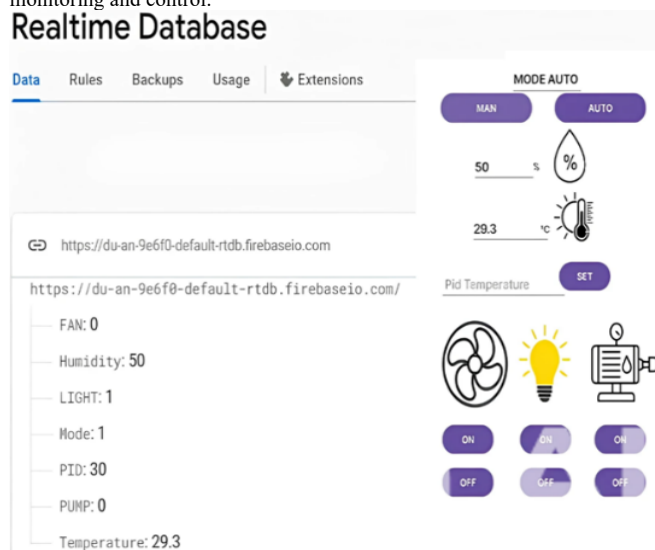


Fig. 10. Real-time data synchronization between the ESP8266 controller and the Firebase cloud database.

III. EXPERIMENTAL SETUP AND PERFORMANCE EVALUATION

A. Finalizing the Greenhouse Model

Figure 11 presents the schematic diagram of the smart greenhouse hardware system, illustrating the connection structure between the main components, including the central microcontroller, environmental sensors, and actuators. The microcontroller serves as the central processing unit, collecting data from sensors such as temperature and humidity, and transmitting it to the cloud platform via Wi-Fi. Based on the collected data and control signals from the system, the microcontroller controls actuators like fans, lights, and pumps to maintain optimal environmental conditions in the greenhouse. This diagram clarifies the flow of data and control signals between the functional blocks in the system.

Figure 12 illustrates the design of the printed circuit board (PCB) and the layout of the hardware components in the smart greenhouse system. The diagram shows the integration of the microcontroller, sensor modules, power circuit, and actuator

control blocks on the same hardware platform. The PCB design is optimized to ensure signal transmission stability, reduce noise, and enhance the reliability of the system. Additionally, the logical arrangement of components facilitates the assembly, maintenance, and expansion of the system.

Figure 13 presents the fully assembled experimental model of the proposed IoT-based smart greenhouse system. The model integrates the ESP8266 NodeMCU as the central embedded controller, a DHT11 sensor for measuring temperature and humidity, an LCD screen for on-site information display, and control interfaces for actuators such as ventilation fans, heating units, lighting systems, and water pumps. The control hardware is connected to the greenhouse model to perform real-time environmental monitoring and closed-loop temperature control. During operation, the ESP8266 collects environmental parameters, executes a digital PID algorithm, generates control signals for the actuators, and synchronizes measurement data and system status with the Firebase cloud platform via Wi-Fi. Thus, the assembled model serves as a physical experimental platform for verifying the integration of sensor functions, control, actuation, and cloud communication in the proposed system.

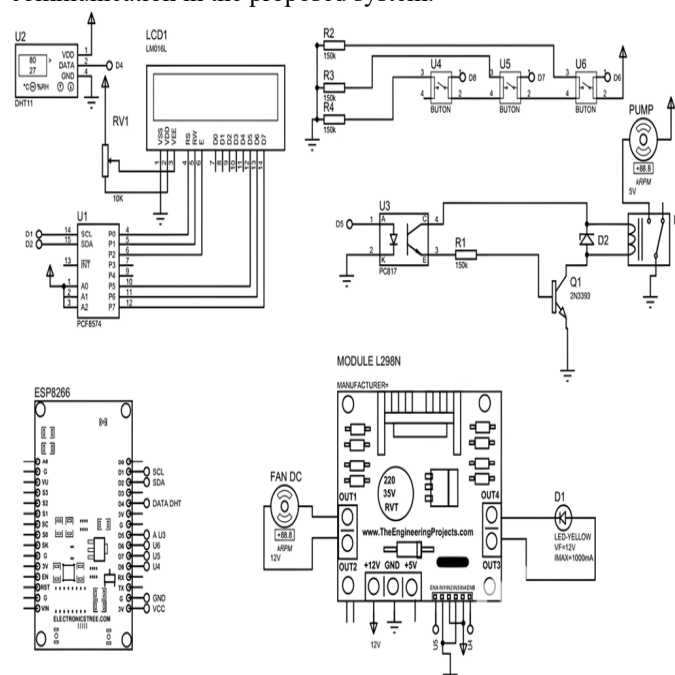


Fig. 11. Hardware schematic of the proposed IoT-based smart greenhouse system.

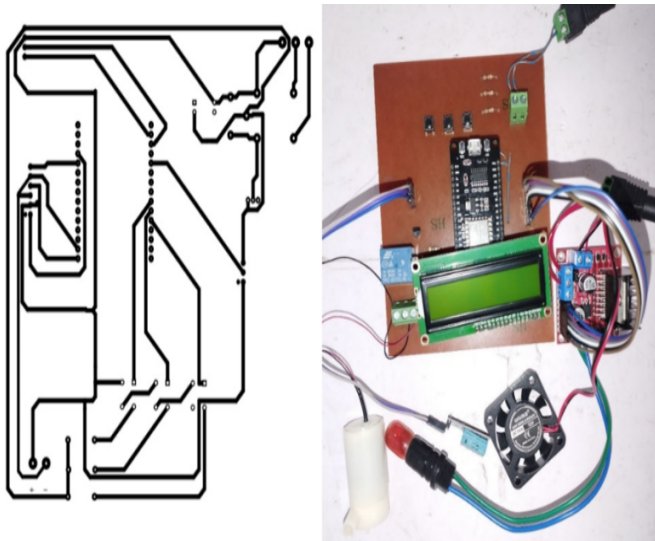


Fig. 12. Printed circuit board (PCB) layout of the greenhouse control hardware.

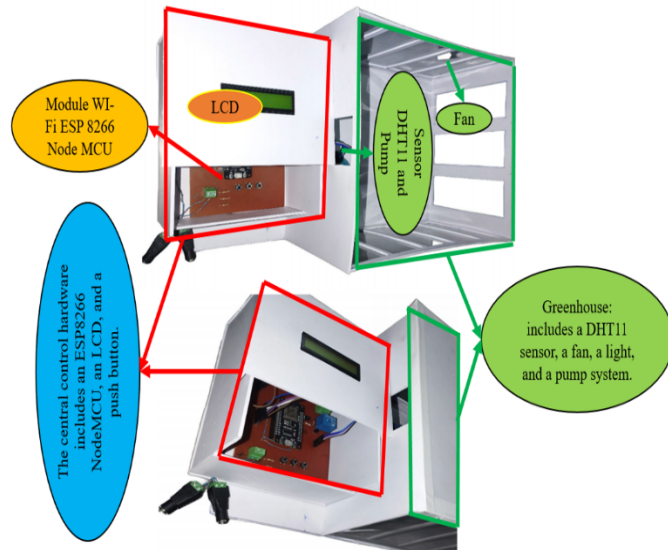


Fig. 13. Assembled prototype of the IoT-based smart greenhouse system.

B. Experimental Results and System Parameter Evaluation

Figure 14 illustrates the greenhouse system after completion, demonstrating its stable and effective operation for continuously monitoring environmental parameters, with the results displayed directly on the LCD screen. The system is designed to operate flexibly in two modes: offline and online. When the system starts up, the LCD screen shows the temperature and humidity values inside the greenhouse, collected directly from the DHT11 sensor. In the absence of a Wi-Fi connection, the screen displays the message "No Wi-Fi," indicating that the system is in offline mode, performing only local monitoring functions.

When the Wi-Fi connection is successfully established, the LCD screen updates the information and displays the status "SET: ...," signaling that the system has switched to online mode. In this mode, sensor data is displayed locally and synchronized with the Firebase database in real time. At the same time, the system is ready to execute automatic control commands via smartphone when the user sends the desired control values from the mobile application.

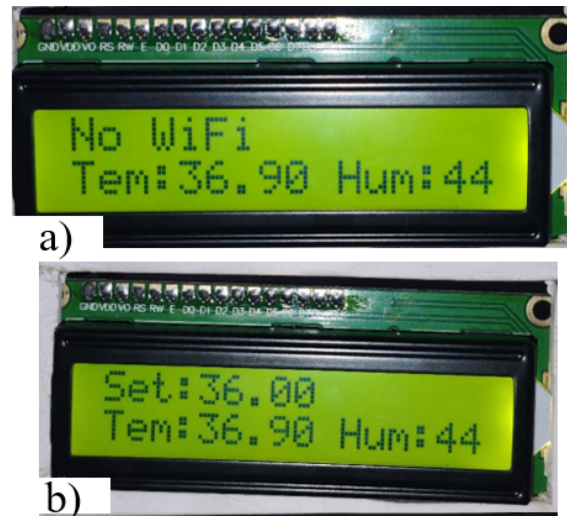


Fig. 14. Experimental operation of the greenhouse system under (a) offline mode and (b) online mode with cloud synchronization.

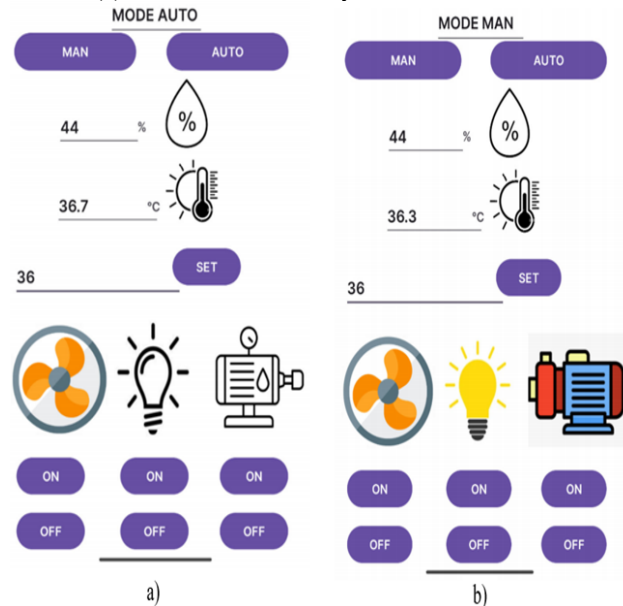


Fig. 15. Experimental demonstration of (a) real-time data synchronization in Firebase and (b) manual control using the Android application.

The greenhouse system is implemented to support two main control modes: automatic mode (MODE AUTO) and manual mode (MODE MAN). It is designed to provide flexibility and automation in controlling the actuators, thereby optimizing the operation of the system (Figure 15). In MODE AUTO, the system automatically activates or deactivates the actuators (fans, lights, pumps) based on the error between the actual temperature and the setpoint. Conversely, in MODE MAN, users can manually control the devices through an application interface with ON/OFF command buttons. The user interface, developed on the Android platform, can monitor sensors and control devices in real time. The state transition mechanism is clearly programmed and directly writes to the Firebase database, ensuring real-time synchronization between software and hardware. It also allows for future scalability in operations due to multi-point access, stable storage, and support for multiple platforms. Data log images and operational status are continuously stored on Firebase (Figure 10), enhancing reliability and capability.

To evaluate the operational performance of the developed smart greenhouse system, experiments were conducted at three time points during the day: 10 AM, 12 PM, and 7 PM. The evaluation parameters were selected based on the characteristic criteria of the control system, including:

- Initial deviation: defined as the difference between the setpoint and the initial temperature of the system.
- Response time (s): the time taken for the system to first reach the setpoint temperature.
- Settling time (s): the time taken for the system to stabilize within the allowable error range around the setpoint.
- Overshoot (%): indicates the extent to which the temperature exceeds the setpoint during the transient process, calculated as a percentage of the setpoint.

These parameters enable a comprehensive assessment of the system's control quality, including both transient and steady-state criteria.

Because the experimental greenhouse prototype uses the DHT11 sensor, which has a nominal temperature accuracy of $\pm 2^\circ\text{C}$ and a sampling period of 1 s, the interpretation and numerical precision of the experimental results are constrained by the measurement system. Temperature readings are reported with a resolution of 0.1°C , consistent with the digital output used in the experimental setup; however, this resolution should not be interpreted as measurement accuracy. Response and settling times are therefore rounded to the nearest second in accordance with the sensor sampling period. This reporting convention avoids implying temporal or numerical precision beyond that supported by the experimental instrumentation and ensures that the results presented in Table II are interpreted within the measurement limitations of the DHT11 sensor.

Table II presents the experimental results of the greenhouse temperature control system at three different times of the day, including 10 AM, 12 PM, and 7 PM, in terms of initial temperature, set temperature, initial deviation, response time, settling time, and overshoot. Across the nine experimental trials, the response time ranged from 31 to 53 s, while the settling time ranged from 44 to 56 s. At 10 AM, the response time ranged from 44 to 48 s, with settling times of 51-56 s and overshoot values of 3.5-4.1%. At 12 PM, the system exhibited shorter response times of 31-40 s, with a settling time of 44-55 s, and an overshoot of 2.0-2.8%. At 7 PM, the response and settling times were 42-53 s and 45-56 s, respectively, with an overshoot of approximately 0.9%. These results indicate that the closed-loop PID controller was able to regulate the greenhouse temperature toward the specified setpoints under the tested operating conditions. Nevertheless, small temperature deviations and overshoot values should be interpreted with caution, given the DHT11 sensor's nominal $\pm 2^\circ\text{C}$ accuracy and 1-s sampling period.

To complement the quantitative metrics reported in Table II, Figure 16 presents the reconstructed temperature-response profiles for the nine experimental conditions at 10 AM, 12 PM, and 7 PM. The profiles were generated using the measured initial temperatures, setpoints, response times, settling times, and overshoot values reported in Table II and

are therefore intended to provide a visual representation of the transient characteristics rather than to reproduce the original raw temperature time series. The reconstructed profiles illustrate the different heating and cooling conditions considered in the experiments and facilitate comparison of the transient behavior across the three measurement periods.

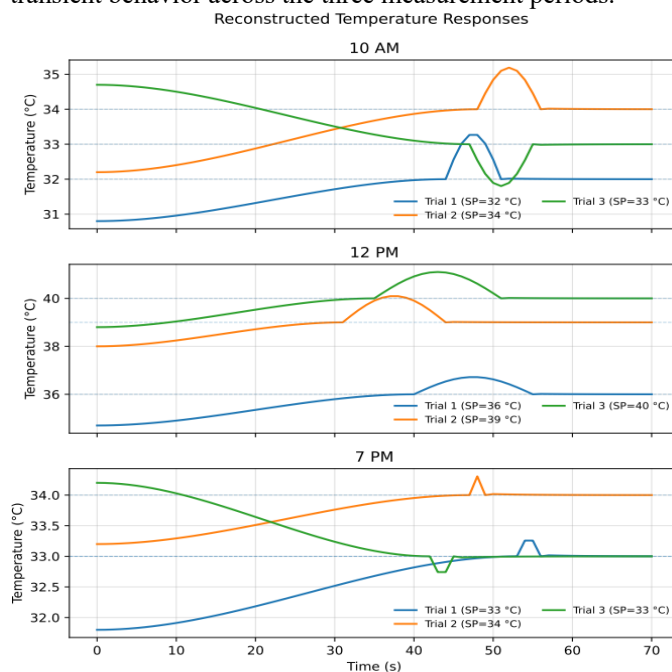


Fig. 16. Reconstructed temperature-response profiles based on the experimentally measured transient metrics at 10 AM, 12 PM, and 7 PM.

As shown in Figure 16, the reconstructed profiles are consistent with the transient metrics summarized in Table II and indicate that the controller drives the temperature toward the corresponding setpoint under different initial thermal conditions. The 12 PM experiments generally exhibit shorter response times than those conducted at 10 AM and 7 PM, whereas the 7 PM experiments show lower reported overshoot values. These differences suggest that the transient behavior of the laboratory-scale greenhouse is influenced by its initial thermal condition and operating period. However, Fig. 20 should be interpreted together with Table II and within the measurement limitations of the DHT11 sensor. The reconstructed profiles are provided to visualize the reported transient metrics and should not be interpreted as raw experimental temperature trajectories.

To systematically evaluate the performance and architectural characteristics of existing IoT-based greenhouse systems, Table III presents a comparative analysis of representative studies, covering criteria such as hardware platform, sensing capabilities, control functions, level of cloud integration, and system limitations. The comparative analysis in Table III shows that most current IoT greenhouse systems remain at the basic monitoring and control level, primarily relying on ON/OFF strategies or threshold-based control, which do not ensure consistent control quality under varying environmental conditions. Although low-cost hardware platforms like ESP8266 and Arduino are widely used, their ability to leverage feedback control and intelligent data processing remains limited. Furthermore, while many systems

have integrated cloud platforms to support real-time data transmission, most offer only display capabilities and lack meaningful analysis and control mechanisms for optimization. Most studies do not provide quantitative assessments of control performance, thereby diminishing scientific value and comparability across systems. In this context, the proposed

system directly addresses these limitations by implementing closed-loop PID control combined with comprehensive quantitative assessment, thereby not only enhancing control quality but also establishing an integrated framework for monitoring, control, and performance evaluation in IoT greenhouse systems.

TABLE II
RESULTS OF THE TEMPERATURE RESPONSE SURVEY OF THE CONTROL SYSTEM AT DIFFERENT TIMES OF THE DAY (10 AM, 12 PM, AND 7 PM)

Measurement No.	Initial Temperature ($^{\circ}\text{C}$)	Set Temperature ($^{\circ}\text{C}$)	Initial Deviation ($^{\circ}\text{C}$)	Settling Time (s)	Response Time (s)	Overshoot (%)
Time 10 AM						
1	30.8	32	-1.2	51	44	4.06
2	32.2	34	-1.8	56	48	3.5
3	34.7	33	-1.7	55	47	3.63
Time 12 PM						
1	34.7	36	+1.3	55	40	2.0
2	38	39	+1.0	44	31	2.82
3	38.8	40	+1.2	51	35	2.75
Time 7 PM						
1	31.8	33	+1.2	56	53	0.9
2	33.2	34	+0.8	49	47	0.9
3	34.2	33	-1.2	45	42	0.9

TABLE III
COMPARATIVE ANALYSIS OF IOT-BASED SMART GREENHOUSE SYSTEMS

Reference	Hardware Platform	Monitored Parameters	Control Strategy	Cloud/IoT Platform	Real-Time Capability	Performance Evaluation	Limitations
Astutik et al., 2019 [22]	NodeMCU ESP8266	Temperature, Humidity, Light	Basic ON/OFF	Android app	Partial	Not reported	Limited control functionality
Naik et al., 2022 [23]	Arduino Nano + Wi-Fi	Temp, Humidity, Rain, Light	Rule-based irrigation	Mobile app	Yes	Not reported	No cloud analytics
Kadam et al., 2024 [24]	Arduino + NodeMCU + LoRa	Temp, Humidity, Soil, Light	Automatic irrigation	LoRa IoT	Yes	Not reported	High deployment cost
Wardani et al., 2025 [25]	NodeMCU ESP8266	Temp, Soil Moisture, Light	Threshold-based control	Cloud IoT	Yes	Not reported	Limited sensing scope
Fathi et al., 2025 [26]	ESP8266 + Firebase	Temp, Humidity, Light	Basic device control	Firebase	Yes	Not reported	Small-scale validation
Angkasa et al., 2025 [27]	ESP32-WROOM	Temp, Soil Moisture, Water Level	Automatic irrigation	Firebase	Yes	Not reported	No data analytics
Méndez-Guzmán et al., 2022 [28]	IoT Gateway + Sensors	Temp, Humidity, Plant images	Irrigation control	Android + Cloud	Yes	Limited	System complexity
Austria et al., 2023 [29]	Arduino IoT	pH, Light, Temp, Humidity	Hydroponic control	ThingSpeak	Yes	Not reported	No intelligent control
Han et al., 2018 [30]	ZigBee + STM32	Temp, Humidity	Environmental control	Desktop + Mobile	Partial	Not reported	Limited scalability
Harsha et al., 2018 [31]	Arduino + Cloud	pH, Water, Light	Irrigation control	Adafruit IO	Yes	Not reported	Limited application scope
Hartawan et al., 2024 [32]	Wemos D1 Mini	Temp, Humidity	Monitoring only	Firebase	Yes	Not applicable	No control capability
Proposed System [This study]	ESP8266 NodeMCU	Temp, Humidity	PID-based control (closed-loop)	Firebase + Mobile App	Full real-time	Quantitative (response, settling, overshoot)	Single-zone laboratory-scale prototype; limited sensor accuracy; no plant-based or long-duration validation

IV. CONCLUSION

This paper presented the design and experimental evaluation of an IoT-based smart greenhouse system integrating environmental monitoring, closed-loop digital PID temperature control, bidirectional cloud communication, and mobile supervision on a low-cost ESP8266 platform. The developed system combines DHT11-based temperature and humidity sensing, embedded PID control, actuator management, Firebase Realtime Database, and an Android application within a unified architecture. Experimental evaluation under different operating conditions demonstrated that the system was capable of regulating the greenhouse temperature toward the specified setpoints, with response times of 31-53 s and settling times of 44-56 s. These results support the feasibility of implementing closed-loop temperature control, realtime IoT monitoring, and cloud-based supervision on a low-cost embedded platform.

Nevertheless, the present study has several limitations. The experiments were conducted using a single-zone laboratory-scale greenhouse prototype, and the quantitative evaluation was limited by the DHT11 sensor's nominal $\pm 2^\circ\text{C}$ accuracy and 1 s sampling period. Therefore, small temperature deviations and transient performance metrics should be interpreted within the measurement capability of the experimental instrumentation. In addition, the system has not yet been validated with actual plants or under long-duration operating conditions. These limitations restrict the generalization of the reported results to full-scale greenhouse environments.

Future work will focus on employing higher-accuracy environmental sensors, conducting long-duration experiments with actual crops, and extending the proposed architecture to multi-zone greenhouse environments. Further studies will also investigate additional environmental variables and more advanced control and data-processing strategies, including adaptive and intelligent control methods and AI-based prediction, to improve robustness, scalability, and practical applicability.

REFERENCES

- [1] B. Et-Taibi, M. R. Abid, E. -M. Boufounas and D. Benhaddou, "Cloud and Edge based Smart Agriculture: A Real-World Deployment," *2024 International Conference on Circuit, Systems and Communication (ICCS)*, pp. 1-6, 2024.
- [2] Y. Ghoul, N. Zitouni and M. B. Amor, "IoT-Based System for Greenhouses Remote Monitoring and Climate Control," *2024 IEEE 4th International Maghreb Meeting of the Conference on Sciences and Techniques of Automatic Control and Computer Engineering (MI-STA)*, pp. 239-244, 2024.
- [3] V. V. An, N. B. Bien, N. T. N. Thao, N. V. Tien and T. P. Nam, "Building a smart garden model using IoT technology," *Scientific research conference 2022 (EIUsrC 2022)*, pp. 138-144, 2022.
- [4] V. V. An, T. L. Mien and B. T. Tam, "Ứng dụng IoT thiết kế và xây dựng mô hình vườn thông minh," *Hội thảo khoa học: Công nghệ mới & ứng dụng trong lĩnh vực điện-dien tu-tu dong hoa*, Đại Học Giao thông Vận Tải, pp. 187-194, 2022. (V. V. An, T. L. Mien, B. T. Tam, "Application of IoT in the design and development of a smart garden model," in *Proceedings of the Scientific Conference on New Technologies and Applications in Electrical, Electronics, and Automation Engineering*, University of Transport and Communications, pp. 187-194, 2022.)
- [5] P. Vinoth Kumar, K. C. Ramya, J. S. Abishek, T. S. Arundhathy, B. Bhavvy, and V. Gayathri, "Smart Garden Monitoring and Control System with Sensor Technology," *2021 3rd Int. Conference on Signal Processing and Communication*, pp. 93-97, 2021.
- [6] A. Sundaram and H. Z. Al Garni, "A Smart Garden System with a Dual-Axis Solar Tracker," *International Journal of Advanced Science and Technology*, Vol. 29, No. 8, pp. 1390-1397, 2020.
- [7] C. M. Angelopoulos, S. Nikolettas, and G. C. Theofanopoulos, "A Smart System for Garden Watering using Wireless Sensor Networks," *Proceedings of the 9th ACM int. symposium on Mobility management and wireless access*, pp. 167-170, 2011.
- [8] A. Karjagi and T. Bagewadi, "IoT Enabled Smart Garden Kit Along With Weather Station," *International Research Journal of Engineering and Technology (IRJET)*, Vol. 7, No. 7, pp. 1535-1538, 2020.
- [9] T. Thamaraimanalan, S. P. Vivek, G. Satheshkumar and P. Saravanan, "Smart Garden Monitoring System Using IOT," *Asian Journal of Applied Science and Technology (AJAST)*, Vol. 2 No. 2, pp. 186-192, 2018.
- [10] Huang, S, Xiang, H, Leng, C, Dai, T and He, G, "Intelligent Regulation of Temperature and Humidity in Vegetable Greenhouses Based on Single Neuron PID Algorithm," *Electronics*, Vol. 13, No. 11, 2083, 2024.
- [11] M. Berenguel, C. Iborra, F. Rodríguez, F. García-Mañas, J. L. Guzmán, and J. A. Sánchez-Molina, "Nonlinear PID-based temperature control techniques in greenhouses using natural ventilation," *IFAC-PapersOnLine*, Vol. 58, No. 7, pp. 454-459, 2024.
- [12] A. H. Hooshmand, "Smart modular greenhouse control via IoT, LabVIEW, and PSO-PID integration," *Computers and Electrical Engineering*, Vol. 128, 110713, 2025.
- [13] X. Wang, C. Wang, Z. Liu, Y. Kang, and Z. Wang, "Electric heating control of agricultural greenhouse in winter using an embedded technology based chaotic particle swarm optimization PID controller," *International Communications in Heat and Mass Transfer*, Vol. 164, 108777, 2025.
- [14] E. B. Edet, M. Chacón-Vásquez, and E. C. Onyecho, "Application of Fractional-order PID controllers in a Greenhouse Climate Control System," *IFAC-PapersOnLine*, Vol. 58, No. 12, pp. 179-184, 2024.
- [15] P. Lin, R. Xu, H. Wang, J. Huang, Z. Guo, X. Sun, I. A. Darwish, Y. Guo, and J. Zhao, "Design Of IoT-Based Greenhouse Monitoring And Control System Using Adaptive Particle Swarm Optimized Fuzzy Pid Controller And Visualization Platform," *INMATEH - Agricultural Engineering*, Vol. 75, pp. 1219-1232, 2025.
- [16] M. Huba, S. Chamraz, P. Bistak, and D. Vrancic, "Making the PI and PID Controller Tuning Inspired by Ziegler and Nichols Precise and Reliable," *Sensors*, Vol. 21, No. 18, pp. 6157, 2021.
- [17] S.-K. Yang, "A new anti-windup strategy for PID controllers with derivative filters," *Asian Journal of Control*, Vol. 14, pp. 564-571, 2012.
- [18] L. R. da Silva, R. C. C. Flesch, and J. E. Normey-Rico, "Analysis of Anti-windup Techniques in PID Control of Processes with Measurement Noise," *IFAC-PapersOnLine*, Vol. 51, No. 4, pp. 948-953, 2018.
- [19] Espressif System, "ESP8266Ex Datasheet (version 7.0), China: Espressif Inc.," 2023.
- [20] T. Kaewwiset and P. Yodkha, "Automatic Temperature and Humidity Control System By Using Fuzzy Logic Algorithm For Mushroom Nursery," *2017 International Conference On Digital Arts, Media and Technology*, Chiang Mai, pp. 396-399, 2017.
- [21] D-Robotics UK, "DHT 11 Humidity & Temperature Sensor," *D-Robotics UK*, 2010.
- [22] Y. Astutik, M. Murad, G. M. D. Putra, and D. A. Setiawati, "Remote monitoring systems in greenhouse based on NodeMCU ESP8266 microcontroller and Android," *AIP Conference Proceedings*, Vol. 2199, 030003, 2019.
- [23] M. R. Naik, D. N. Naik, S. H. Basha, and K. J. Reddy, "Greenhouse Environment Monitoring and Controlling Through IoT," *International Journal for Research in Applied Science & Engineering Technology (IJRASET)*, Vol. 10, pp. 2412-2417, 2022.
- [24] T. Kadam, T. Vaishya, S. Rauf, M. Sayed, and V. Shah, "Wirelessly Empowered Agriculture a Study on Arduino, NodeMCU and LoRaWAN Enabled Greenhouse Automation," *International Journal of Innovative Science and Research Technology (IJISRT)* *IJISRT24APR2167*, Vol. 9, No. 4, pp. 1845-1848, 2024.
- [25] E. P. Wardani, M. A. A. Khusain, S. Kusumastuti, and Sihono, "An IoT-Based Monitoring and Control System for Chrysanthemum Cultivation," *(JAIC) Journal of Applied Information and Communication Technologies*, Vol. 11, No. 2, pp. 34-40, 2025.

- [26] I. Fathi, "An IoT-Based Low-Cost Smart Greenhouse Monitoring System Using ESP8266 and Firebase for Real-Time Environmental Control," *International Journal of Advanced Natural Sciences and Engineering Researches*, Vol. 9, No. 5, pp. 268–286, 2025
- [27] M. I. B. Angkasa, R. T. Adek, and S. F. Anshari, "Full Automation and Control System Based on IoT in a Greenhouse (Case Study: Faculty of Agriculture)," *Malikussaleh University. Proceedings of International Conference on Multidisciplinary Engineering (ICOMDEN)*, Vol. 2, pp. 00077, 2025
- [28] H. A. Méndez-Guzmán, J. A. Padilla-Medina, C. Martínez-Nolasco, J. J. Martínez-Nolasco, A. I. Barranco-Gutiérrez, L. M. Contreras-Medina, and M. Leon-Rodriguez, "IoT-Based Monitoring System Applied to Aeroponics Greenhouse," *Sensors (Basel)*, Vol. 22, No. 15, pp. 5646, 2022.
- [29] A. C. H. Austria, J. S. Fabros, K. R. G. Sumilang, J. Bernardino, and A. C. Doctor, "Development of IoT Smart Greenhouse System for Hydroponic Gardens," *International Journal of Computing Sciences Research*, Vol. 7, pp. 2111-2136, 2023
- [30] Z. Han, Z. Wu, S. Lin, and F. Luan, "An intelligent household greenhouse system design based on Internet of Things," *IOP Conference Series: Materials Science and Engineering*, Volume 399, *3rd China-Romania Science and Technology Seminar (CRSTS 2018)*, pp. 24–27, 2018.
- [31] A. Harsha, K. Deekshith, B. K. Murali Krishna, K. T. Sachin, and N. Sushanth, "Automated Hydroponics Greenhouse Monitoring System using Adafruit.io Controlled by Google Assistant," *International Journal of Engineering Research & Technology 2018*, Vol. 06, No. 15, pp. 1-4.
- [32] L. Hartawan, G. A. Rakhmat, N. Nugraha, N. D. Anggraeni, K. B. Maulana, M. R. Ridjali, M. Iqbal, M. K. Mahardhika, B. S. Alam, J. A. Fani, L. F. Amarullah, M. I. Alfadhli, M. R. Sbastio, and A. A. Purba, "Design of IoT-based Greenhouse Temperature and Humidity Monitoring System," *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, Vol. 12, No. 4, pp. 1051 - 1062, 2024.

Vo Van An, received a master's degree from Hanoi University of Transport, Hanoi, Vietnam. From 2016 to June 2022, he served as vice head of the electrical and electronics department at Binh Duong University of Economics and Technology. Currently, he is a lecturer at Thu Dau Mot University. His research interests include intelligent control, the Internet of Things, Artificial Neural Networks and unmanned aerial vehicles,

