**Journal of Computer Science and Electrical Engineering**
**Print ISSN: 2663-1938**
**Online ISSN: 2663-1946**
**DOI: https://doi.org/10.61784/jcsee3139**

# DESIGN AND REALIZATION OF MONITORING AND CONTROL SYSTEM FOR SMART FARMS USING ESP32-P4 AND LVGL

YuanJu Zhou
*School of Artificial Intelligence and Electronic Engineering, Sichuan Technology and Business University, Chengdu 611745, Sichuan, China.*

**Abstract:** Aiming at the problems of poor real-time performance, weak local interaction and limited data processing capacity of traditional agricultural monitoring systems, this paper presents an intelligent farm monitoring and control system based on ESP32-P4 and a 7-inch IPS touch screen. Taking the dual-core RISC-V ESP32-P4 with a main frequency of 400 MHz as the main controller, the system drives the display screen via the MIPI-DSI interface and adopts the LVGL graphics library to realize a high-frame-rate local human-machine interaction interface. It integrates five types of sensors for temperature, humidity, soil moisture, light intensity and carbon dioxide concentration, and supports Wi-Fi cloud data transmission as well as closed-loop control functions including automatic irrigation and environmental regulation. The test results show that the interface refresh rate is no less than 28 FPS, and the control response time is less than 100 ms. The system can satisfy the requirements of real-time monitoring and precise control for small and medium-sized intelligent farms.

**Keywords:** Smart agriculture; ESP32-P4; LVGL; Embedded GUI

## 1 INTRODUCTION

With the growth of the global population and the intensification of climate change, traditional agriculture is facing challenges such as low resource utilization efficiency and rising labor costs [1]. Smart agriculture, which enables precise perception and automated control of production environments through sensor networks, communication systems, and intelligent decision-making algorithms, has become an important direction in the development of modern agriculture [2].

The Internet of Things (IoT) provides a key technical foundation for smart agriculture. The IoT agricultural monitoring system gathers environmental data in real time to support precision irrigation, fertilization and plant disease control [3-5]. Intelligent decision and automated control serve as the core orientation [6-7]. In recent years, embedded intelligent terminals have been increasingly applied in smart agriculture.

However, prominent drawbacks remain in existing systems: Most systems rely on mobile APP or Web terminals for remote monitoring with poor local interaction capability; General-purpose MCUs feature limited graphics processing performance and cannot smoothly drive high-resolution touch screens.

To address the above limitations, this paper presents a smart farm monitoring and control system based on ESP32-P4 and LVGL. The contributions of this work are summarized as follows. First, ESP32-P4 is introduced into a smart agricultural terminal, where its dual-core RISC-V architecture and MIPI-DSI display interface are used to enhance human–machine interaction. Second, an LVGL-based local GUI is developed to support on-site monitoring and control without relying on a mobile application. Third, a hybrid control strategy integrating threshold-triggered control with PID regulation is proposed to enable coordinated manual and automatic operation. The real-time performance and reliability of the system are validated through experimental testing.

## 2 SYSTEM ARCHITECTURE AND DESIGN

### 2.1 System Architecture

As shown in Figure 1, the system is designed with a four-layer architecture comprising the perception layer, control layer, execution layer, and cloud layer. The perception layer collects environmental parameters [8], while the control layer, centered on the ESP32-P4, is responsible for data processing, decision logic, GUI rendering, and communication management. The execution layer implements control operations such as irrigation, supplementary lighting, and ventilation, and the cloud layer supports remote monitoring and data storage.

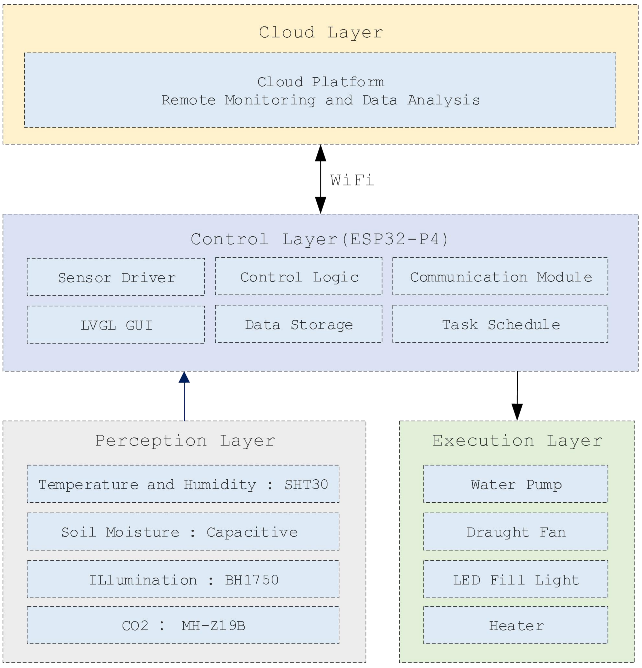


**Figure 1** Overall System Architecture

### 2.2 Functional Requirements

The system is intended for small- and medium-scale scenarios such as greenhouses and vertical farms. Its main functions include: environmental sensing and local touchscreen display; threshold-triggered device control and automatic operation; and cloud-based data uploading, remote access, and abnormal condition alerts.

## 3 HARDWARE DESIGN

### 3.1 Main Controller

ESP32-P4 is a high-performance MCU developed by Espressif for edge computing and human–machine interaction applications. It boasts rich on-chip resources and extensive peripheral compatibility, including MIPI DSI/CSI-2, USB 2.0, Ethernet, SD/MMC, UART, I$^2$C, and I$^2$S. In this system, the ESP32-P4 is responsible for LVGL-based GUI rendering, sensor data acquisition, and multitask scheduling. Its built-in MIPI DSI interface is directly connected to the display, eliminating the need for an external display controller and effectively simplifying the hardware structure.

### 3.2 Display Module

A 7-inch IPS capacitive touchscreen with a resolution of 1024 × 600 is used in the system and is connected to the main controller through the MIPI-DSI interface. The touch module adopts the GT911 controller and communicates via the I$^2$C bus. It supports five-point touch input. The wide viewing angle of the IPS panel also improves readability under outdoor lighting conditions.

### 3.3 Sensor Module

The system integrates five types of sensors. The sensor selection takes into account measurement accuracy and interface type. The main parameters are listed in Table 1.

**Table 1** Sensor Selection

| Sensor | Model | Measurement Parameters | Interface | Accuracy |
|---|---|---|---|---|
| Temperature and Humidity Sensor | SHT30 | Temperature and Humidity | I$^2$C | ±0.3°C, ±2%RH |
| Soil Moisture Sensor | Capacitive | Soil Volumetric Water Content | ADC | ±3% |
| Light Intensity Sensor | BH1750 | Light Intensity（lux） | I$^2$C | ±20% |
| CO$_2$ Sensor | MH-Z19B | CO$_2$ concentration（ppm） | UART | ±50 ppm |
| Soil Temperature Probe | DS18B20 | Soil Temperature | 1-wire | ±0.5°C |

### 3.4 Actuator Control Module

The actuator control module is mainly responsible for controlling devices such as irrigation equipment, supplementary lighting, ventilation units, and heating modules. For switching loads such as water pumps and solenoid valves, relays are used to implement start–stop control. For DC loads that require continuous adjustment, such as DC fans, supplementary lighting lamps, and heating modules, MOSFET-based PWM power regulation can be adopted. This design meets the requirements for both on–off control and regulation control of equipment in agricultural environmental control applications.

## 4 SOFTWARE DESIGN

### 4.1 Overall Software Architecture

The system software is developed based on FreeRTOS and the ESP-IDF framework, and adopts a layered modular architecture, as shown in Figure 2. The application layer includes the LVGL GUI task, control algorithm task, and cloud service task. The middleware layer encapsulates the LVGL v8.3 library, Wi-Fi/BLE protocol stack, and MQTT client. The hardware abstraction layer provides unified management of the display, touchscreen, and peripheral drivers.

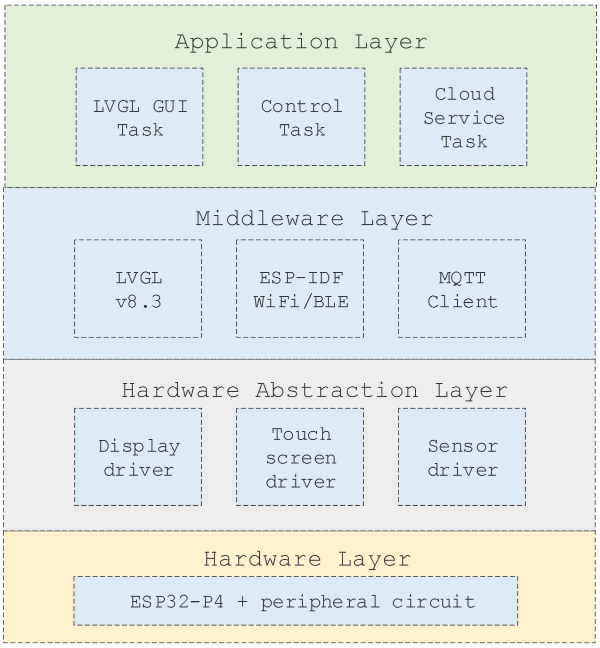


**Figure 2** Software Architecture Diagram

### 4.2 LVGL-Based Graphical User Interface Design

The touchscreen interface of the system is developed based on LVGL v8.3 and consists of four functional pages. The functions of each page are described as follows.

#### *4.2.1 Real-time data monitoring page*

As shown in Figure 3, this page presents five types of environmental parameters, including temperature, humidity, light intensity, soil moisture, and $CO_2$ concentration, in the form of icons and numerical values. The displayed data are updated through a unified data refresh interface, which decouples the UI from the data processing logic.

Environment (Realtime)

Temperature
Unit: °C
24.7 °C

Humidity
Unit: %
66.2 %

Light
Unit: lux
36143 lux

CO2
Unit: ppm
1040 ppm

Soil (Realtime)

Soil Moisture
Unit: %
18.0 %

Soil Temperature
Unit: °C
73.4 °C

29 FPS
55% CPU

**Figure 3** Real-Time Data Monitoring Page

### *4.2.2 Historical trend page*

As shown in Figure 4, this page uses line charts to display the historical trends of environmental parameters, including temperature, humidity, light intensity, and $CO_2$ concentration, as well as the soil parameter, namely soil moisture. The charts are implemented based on the lv_chart component. A sliding-window mechanism is used to retain recent data, and different parameters are distinguished by color. Dynamic refreshing is also supported.

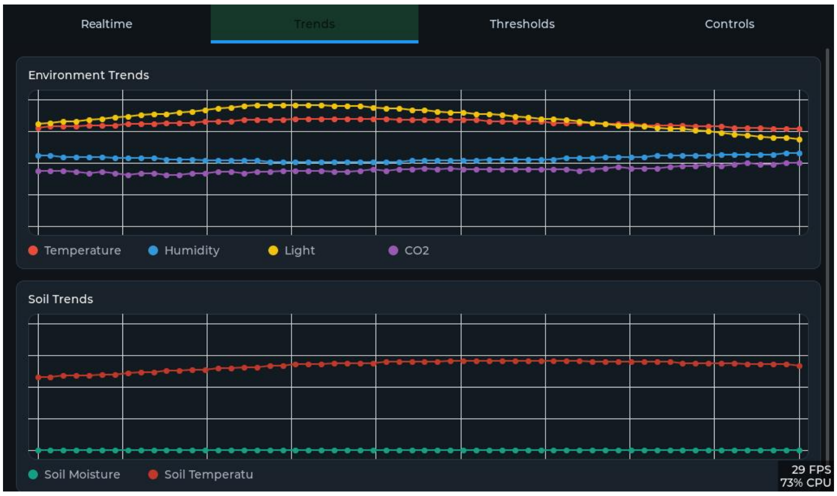
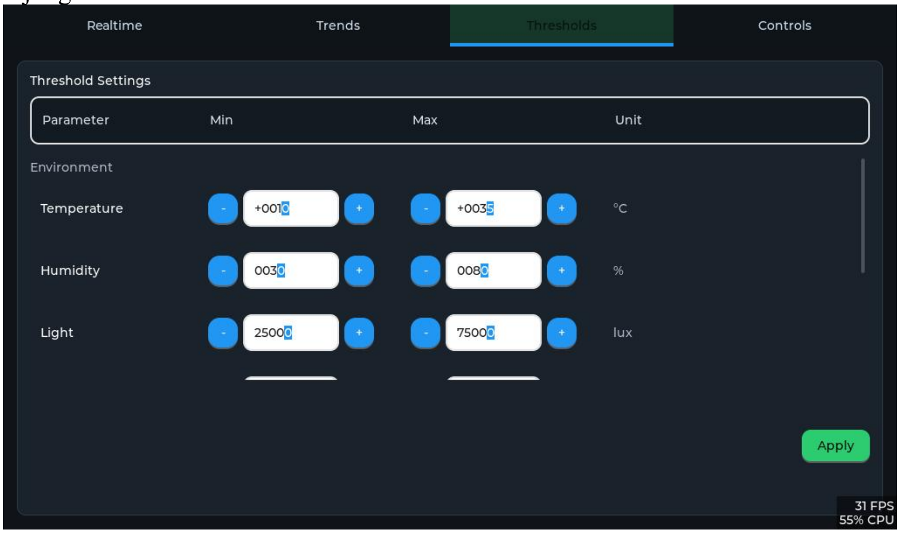


**Figure 4** Historical Trend Page

### *4.2.3 Threshold configuration page*

As shown in Figure 5, this page presents the safe range settings of each parameter in a scrollable form, including the parameter name, upper and lower limits, and unit. Users can adjust the values through numerical adjustment controls, while the adjustable range of each parameter is constrained by its actual measurement range. The configured values are shared by the alarm judgment and control execution modules.



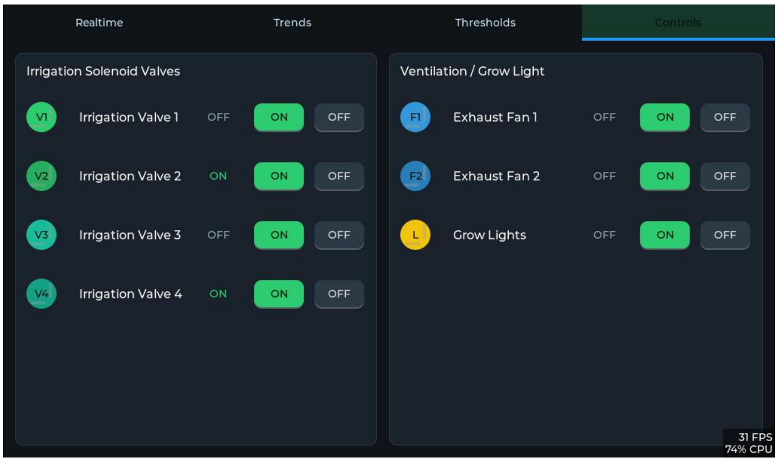
Threshold Settings

| Parameter | Min | Max | Unit |
|---|---|---|---|
| Environment | | | |
| Temperature | +0010 | +0035 | °C |
| Humidity | 0030 | 0080 | % |
| Light | 25000 | 75000 | lux |

**Figure 5** Threshold Configuration Page

### *4.2.4 Device control page*

As shown in Figure 6, this page adopts a partitioned layout. The left side is used to control four irrigation solenoid valves, while the right side controls environmental regulation devices such as the exhaust fan and grow light. Each item includes the device identifier, name, operating status, and switch button, with different colors used to indicate the on and off states. After a button is clicked, the UI is updated immediately, and a control command is sent to the underlying module through a callback function. Meanwhile, a status synchronization interface is provided so that automatic control logic and the remote terminal can write back the actual device status, ensuring consistency among multiple control sources.



Irrigation Valve 1 OFF
Irrigation Valve 2 ON
Irrigation Valve 3 OFF
Irrigation Valve 4 ON
Exhaust Fan 1 OFF
Exhaust Fan 2 OFF
Grow Lights OFF

**Figure 6** Device Control Page

### 4.3 Sensor Data Acquisition and Processing

The sensor acquisition function is encapsulated as an independent FreeRTOS task with a default sampling period of 5s. After system startup, the sensor drivers and communication interfaces, including $I^2C$, UART, ADC, and 1-Wire, are initialized in sequence. The system then periodically reads five types of environmental parameters.

The raw data acquired from the sensors are filtered and then written into a global shared data structure. A mutex mechanism is used to ensure safe access among multiple tasks. Specifically, temperature and humidity data are processed using a moving average filter with a window size of 5, while $CO_2$ data are filtered using a first-order low-pass filter with $\alpha = 0.2$, so as to reduce the influence of transient noise on the measurement results. When parameters such as temperature, humidity, and $CO_2$ concentration continuously exceed the preset safety thresholds, the system triggers a local audible and visual alarm. Meanwhile, the abnormal information is uploaded to the cloud platform through MQTT and then pushed to the user terminal.

### 4.4 Control Logic Design

The system adopts a hybrid control strategy that combines threshold-triggered control with PID regulation. For switching loads such as irrigation and supplementary lighting devices, threshold control with hysteresis is applied. Corresponding equipment is actuated once parameters exceed preset thresholds and stops after values return outside the hysteresis band, which reduces frequent start-stop operations induced by sensor fluctuations.

For continuously regulated devices such as heating and ventilation units, an incremental PID algorithm is used. Taking temperature control as an example, the controller calculates the control increment $\Delta u(k)$ according to the deviation between the target value and the measured value. The current PWM output duty cycle, ranging from 0% to 100%, is then obtained by accumulating the control increment. The incremental PID equation is expressed as follows:

$$\Delta u(k) = Kp[e(k)-e(k-1)] + Ki\cdot e(k) + Kd[e(k)-2e(k-1)+e(k-2)] \tag{1}$$

In the equation, $\Delta u(k)$ denotes the control increment at the kth sampling instant, $e(k)$ denotes the current error, and Kp, Ki, and Kd represent the proportional, integral, and derivative gains, respectively. The initial PID parameters are determined using the Ziegler–Nichols method and then refined through experiments. Compared with the positional PID algorithm, the incremental PID algorithm calculates the change in control output between adjacent sampling periods. It reduces error accumulation, helps prevent integral windup, and limits the impact of occasional misoperation. The control task is executed with a period of 1s.

### 4.5 Communication and Cloud Services

The system connects to the Internet through Wi-Fi and communicates with the cloud platform using the MQTT protocol. The uploaded data are formatted in JSON. The cloud platform can be selected from Aliyun IoT, Tencent Cloud IoT, EMQX, or other similar platforms, and can be switched through the configuration file.

## 5 SYSTEM TESTING AND PERFORMANCE EVALUATION

### 5.1 Test Environment

The tests are conducted in a laboratory environment equipped with artificial light sources, a humidifier, and ventilation facilities to simulate typical greenhouse operating conditions. The system runs continuously for 72 h, covering various scenarios such as strong daytime illumination, low nighttime temperature, and short-term high-temperature fluctuations. The test items include: GUI refresh performance, sensor acquisition accuracy, control command response time, and system stability.

## 5.2 GUI Performance Test

The GUI performance is tested using the built-in performance monitoring tool of LVGL, and the results are shown in Table 2.

**Table 2** GUI Performance Test Results

| Test Item | Test Result | Evaluation |
|---|---|---|
| Frame Rate During Page Switching | 28–32 FPS | Smooth |
| Touch Response Latency | <20 ms | Excellent |
| CPU Usage | 50–70% | Reasonable |

The test results show that, under the test conditions of this system, the ESP32-P4 platform can effectively meet the operating requirements of the LVGL-based graphical interface. By assigning interface refresh, data acquisition, and control logic to separate tasks, the system reduces the impact of data acquisition and control processes on interface refreshing, thereby maintaining smooth GUI operation. These results indicate that the proposed design is well suited for graphical interface display and multitask coordination.

## 5.3 Sensor Accuracy Test

The sensor readings were compared with those measured by standard instruments, including a Fluke 971 temperature and humidity meter and a DLX-LSK2304 illuminance meter. The results are shown in Table 3.

**Table 3** Comparison of Sensor Data Accuracy

| Parameter | System Reading | Standard Instrument Reading | Error | Relative Error |
|---|---|---|---|---|
| Temperature (°C) | 25.4 | 25.2 | +0.2 | 0.8% |
| Humidity (%RH) | 67.5 | 68.1 | −0.6 | 0.9% |
| Light Intensity (lux) | 32500 | 31800 | +700 | 2.2% |
| Soil Moisture (%) | 41.2 | 42.0 | −0.8 | 1.9% |

The relative errors of all sensors were within 2.2%. Among them, the temperature and humidity measurements showed the highest accuracy, mainly due to the factory calibration of the SHT30 sensor. The light intensity measurement had a slightly larger error of 2.2%, but it still met the requirements of agricultural monitoring. The $CO_2$ sensor was not included in this comparison because of the relatively low $CO_2$ concentration in the test environment. However, after calibration under high-concentration greenhouse conditions, its measurement error was controlled within ±50 ppm. Overall, the sensing accuracy meets the application requirements.

## 5.4 Control Performance Test

The control performance test results are shown in Table 4.

**Table 4** Control Performance Test Results

| Test Item | Test Result |
|---|---|
| Control command response time, local | <100 ms |
| Control command response time, remote MQTT | <500 ms, depending on network latency |
| Irrigation triggering accuracy | 98.5% in 100 tests |
| Steady-state error of PID temperature control | ±0.5°C |

The test results show that the system provides fast local control response and relatively low latency for remote MQTT-based control. The two control modes can be used in a complementary manner: local control ensures real-time performance, while remote MQTT control improves management convenience. The irrigation triggering accuracy reached 98.5%, indicating reliable control performance. The steady-state error of PID temperature control was ±0.5°C, showing good stability and control accuracy. Overall, the system meets the requirements for real-time control and stable operation in smart agricultural applications.

## 5.5 Stability Test

The 72-hour continuous operation test showed that: (1) no system crash or abnormal restart occurred during the test, and all 51,840 collected data records were retained without loss; (2) automatic irrigation was triggered 12 times and supplementary lighting was triggered 8 times, and all control actions were executed correctly; (3) the LVGL interface remained smooth throughout the test, with no lagging or display corruption; (4) the FreeRTOS heap memory usage remained below 65%, with no sign of memory leakage. These results indicate that the system can meet the requirement for continuous operation.

# 6 DISCUSSION

### 6.1 Comparison with Existing Systems

Table 5 compares the proposed system with other smart agricultural systems reported in recent studies.

**Table 5** Comparison Between the Proposed System and Existing Systems

| Comparison Item | Proposed System | Abouelmehdi et al.[5] | Indira et al.[3] | Li et al.[9] |
|---|---|---|---|---|
| Main Controller | ESP32-P4 (400 MHz) | ESP32 (240 MHz) | NodeMCU ESP8266 | ESP32 |
| Local Display | 7-inch IPS touchscreen (LVGL) | None | None | None |
| Number of Sensors | 5 types | 4 types | 4 types | 3 types |
| Control Function | Automatic + manual control | Monitoring only | Monitoring only | Monitoring only |
| Interface Interaction | Local GUI | Remote APP | Remote APP | Remote APP |

As shown in Table 5, existing systems generally lack local interaction capability. The systems proposed by Abouelmehdi et al. [5], Indira et al. [3], and Li et al. rely on mobile applications [9], which may lead to poor user experience in scenarios where network access is unavailable or the use of mobile terminals is inconvenient. In contrast, the proposed system implements a 7-inch touchscreen-based local GUI using ESP32-P4 and LVGL, allowing users to complete all operations without external devices. In addition, unlike existing systems that mainly focus on monitoring, this system integrates threshold-triggered control and PID-based automatic regulation, extending its functionality from environmental sensing to active regulation.

### 6.2 Innovations

(1) Hardware platform: ESP32-P4 is introduced into a smart agricultural terminal, helping overcome the graphics processing limitations of conventional MCUs.
(2) Interaction design: An LVGL-based local GUI independent of mobile applications is developed, addressing the limitation of remote monitoring schemes in on-site operation.
(3) Control strategy: A hybrid control strategy combining threshold-triggered control and PID regulation is proposed, taking into account both fast response for switching devices and precise regulation for continuously controlled devices.
6.3 Limitations and Future Work
(1) Limited communication mode: The current system supports only Wi-Fi communication, with a coverage range of approximately 100 m. In future work, LoRa or NB-IoT communication will be introduced to support deployment in large-scale farms.
(2) Dependence on external power supply: The system currently relies on an external power source. A hybrid power supply scheme based on solar energy and lithium batteries will be considered in future work to enable off-grid operation.
(3) Underutilized edge AI capability: The ESP32-P4 provides computing resources with a dual-core 400 MHz processor and 32 MB PSRAM. In the future, TinyML models can be integrated to support intelligent functions such as disease recognition and yield prediction.
(4) Basic cloud platform functions: At present, the cloud platform mainly supports data uploading and remote viewing. Future extensions may include crop growth models and planting decision-support modules.

## 7 CONCLUSION

This paper designs and implements a smart farm monitoring and control system based on ESP32-P4 and LVGL. With the ESP32-P4 as the core controller, the system achieves smooth local GUI interaction, integrates five types of sensors, and supports automatic control and MQTT-based remote communication. Experimental results show that the GUI refresh rate reaches no less than 28 FPS, the sensor error is within 2.2%, the control response time is less than 100 ms. Compared with existing systems, the proposed solution shows clear advantages in local interaction capability and control completeness, making it suitable for intelligent upgrading of small- and medium-sized greenhouses and family farms. Future work will focus on the introduction of solar power supply, LoRa communication, and edge AI functions.

## COMPETING INTERESTS

The authors have no relevant financial or non-financial interests to disclose.

## REFERENCE

[1] Mansoor S, Iqbal S, Popescu S M, et al. Integration of smart sensors and IOT in precision agriculture: trends, challenges and future prospectives. Frontiers in Plant Science, 2025, 16(000). DOI: 10.3389/fpls.2025.1587869.
[2] Thilakarathne N N, Bakar M S A, Abas P E, et al.Internet of things enabled smart agriculture: Current status, latest advancements, challenges and countermeasures. Heliyon, 2025, 11(3). DOI: 10.1016/j.heliyon.2025.e42136.
[3] Indira P, Arafat IS, Karthikeyan R, et al. Fabrication and investigation of agricultural monitoring system with IoT & AI. SN Applied Sciences, 2023, 5: 322. DOI: 10.1007/S42452-023-05526-1.

[4] Lin J, Yu W, Zhang N, et al. A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications. IEEE Internet of Things Journal, 2017, 4(5): 1125-1142. DOI: 10.1109/JIOT.2017.2683200.

[5] Abouelmehdi K, Elhattab K, Moutaouakkil E A. Smart Agriculture Monitoring System using Clean Energy. International Journal of Advanced Computer Science and Applications (IJACSA), 2022, 13(5). DOI: 10.14569/IJACSA.2022.0130544.

[6] Araby A A, Elhameed M M A, Magdy N M, et al. Smart IoT Monitoring System for Agriculture with Predictive Analysis. 2019. DOI: 10.1109/MOCAST.2019.8741794.

[7] Saiz-Rubio V, Rovira-Más F. From Smart Farming towards Agriculture 5.0: A Review on Crop Data Management. Agronomy, 2020, 10(2): 207. DOI: 10.3390/agronomy10020207.

[8] Khoa T A, Man M M, Nguyen T Y, et al. Smart Agriculture Using IoT Multi-Sensors: A Novel Watering Management System. Journal of Sensor and Actuator Networks, 2019(3). DOI: 10.3390/jsan8030045.

[9] Li P, Deng X. An intelligent plant growth monitoring system based on ESP32 and IoT resource explorer platform. Journal of Electronics and Information Science, 2023, 8(6). DOI: 10.23977/JEIS.2023.080604.