# Intelligent Irrigation System Based on Raspberry Pi and Computer Vision

Mingyu Yu[1], Xiaoxia Zheng[1*]

[1]JiNing Normal University; China

**Abstract.** In today's fast-paced life, people are often too busy with work to water their houseplants in a timely manner, which not only affects the growth and development of plants but also reduces their survival rate. Aiming at the shortcomings of traditional watering methods and existing automatic watering devices, as well as people's demand for further improving the intelligence and automation of watering equipment, this paper designs an automatic irrigation system by combining soil moisture sensors and OpenCV visual recognition. The system uses a Raspberry Pi 4B single-chip microcomputer as the control core, which can monitor the soil moisture of flowers in real-time. The monitored data is transmitted to the CPU module. When the moisture is lower than the set threshold, the single-chip microcomputer starts the water pump to pump water, realizing the function of automatic flower watering. For different water requirements of various plants, the starting threshold can be manually adjusted, making the system flexible and reliable in operation and having certain popularization value.

**CCS Concepts**: Computing methodologies → Artificial intelligence → Application of intelligent systems in horticultural automation

*Keywords: Intelligent flower-watering system; Artificial intelligence; Raspberry Pi 4B; Soil moisture sensor; Automatic irrigation decision-making; Threshold adaptive adjustment*

## 1. Introduction

### 1.1 Research Background and Significance

With the improvement of living standards, home green plant cultivation has become an important part of a quality life. However, according to surveys, more than 62% of households experience plant death due to water shortage because the owners are on long - term business trips or are too busy with work[1-3]. Traditional automatic watering devices mostly adopt timed

* Corresponding Author: Xiaoxia Zheng (27288473@qq.com)

irrigation or siphon principles and have two core defects: First, they cannot be dynamically adjusted according to the actual water demand status of plants. For example, succulent plants have low water requirements, and traditional devices are likely to cause root rot[4, 5]. Second, they lack environmental adaptability. For instance, in summer, when the soil evaporation increases at high temperatures, the fixed irrigation mode cannot meet the demand[6].

To solve the problems existing in the current watering systems, it is urgent to design a system that can accurately adjust multi - channel temperature and humidity values through sensors and realize automatic watering. Single - chip microcomputers are widely used in various fields and can be used to realize the automatic detection and control of the watering process[7, 8]. The optimal time for potted plant watering and the amount of water to be applied need to consider the influences of various factors such as the type of flowers, ambient temperature, and soil humidity[9]. Therefore, the automatic watering system needs to detect and track the changes of relevant influencing factor parameters in real - time to realize the continuous control of the watering process.

The technical significance of this research lies in the following aspects: It constructs a closed - loop control system of "perception - decision - execution - feedback", breaking through the limitation of the single - threshold control of traditional devices[10]. It realizes the autonomous positioning of the mobile irrigation platform through the OpenCV visual recognition technology, providing a technical reference for small - scale intelligent agricultural equipment[11]. In terms of hardware selection and design, the system has carried out efficient cost optimization. The overall construction cost is only one - fifth of that of commercial intelligent irrigation equipment. With a highly competitive investment, it provides an economically feasible solution for the intellectualization of home plant maintenance, and has significant promotion potential[12, 13].

## 1.2 Overseas Research Progress and Limitations

In the field of intelligent plant irrigation, foreign - based intelligent planting systems represented by AeroGarden, Rachio, and others have achieved relatively refined irrigation control by virtue of multi - sensor fusion technology (integrating multi - dimensional sensing modules such as soil moisture, light intensity, and ambient temperature and humidity)[14]. For example, the AeroGarden system can dynamically adjust the irrigation frequency and water volume based on a preset plant growth model[15, 16]. However, its price of more than $200 creates a significant market barrier. Moreover, the product positioning focuses on indoor standardized planting scenarios (such as small - scale planting cabins on home balconies and modular planting areas in commercial greenhouses), and it has extremely low adaptability to complex outdoor environments (such as scenarios of multi - variety mixed planting in courtyards, direct sunlight and uneven ventilation on terraces) and large - scale horticultural spaces[17].

Some European and American scientific research teams have attempted to break through scenario limitations. For instance, the “plant - machine symbiotic irrigation” research carried out by the MIT Media Lab uses plant electrical signal feedback (collecting bioelectrical signals through microelectrodes implanted in plant stems) to trigger irrigation actions[18]. Although “plant - autonomous demand response” has been achieved in a laboratory environment, there are problems such as plant damage caused by electrode implantation and large fluctuations in signals due to environmental interference. There is still a 5 - to 8 - year technological iteration cycle before commercial application. In addition, foreign high - end systems generally rely on cloud - platform data interaction, and have inherent defects in terms of network latency (such as poor network coverage in remote areas) and data privacy (information security concerns caused by the uploading of users’ planting data). It is difficult to meet the needs of home gardening users for “localized and low - dependency” control[19, 20].

In the research on existing irrigation systems, low-cost solutions often suffer from insufficient adaptability to dynamic environments. The Raspberry Pi irrigation system designed by Nawandar and Satpute only supports the static deployment of a single potted plant and cannot accurately identify water - needing objects when the plant's position changes. The low-cost agricultural irrigation solution by Anagha et al[21, 22]. relies on a fixed sensor network and has difficulty adapting to changes in the water - needing area caused by plant growth. In contrast, this research takes Raspberry Pi + OpenCV computer vision as the core. While keeping the hardware cost relatively low, it realizes the dynamic recognition and multi - target adaptation of plants through OpenCV. Even if the plant's position moves or the light changes, it can still accurately locate the water - needing objects. Combined with the adaptive adjustment of the soil moisture sensor threshold, it can adapt to the different water needs of plants such as succulents and epipremnums, filling the gap in the "dynamic environmental adaptability of low-cost irrigation systems".

## 2. System Design

The system architecture of this project is divided into two parts: hardware and software. In terms of hardware, it consists of a Raspberry Pi 4B single - chip microcomputer as the core control unit, a soil moisture sensor module, OpenCV visual recognition module, and a water pump.

## 2.1 Raspberry Pi 4B Single - Chip Microcomputer

As the core control of this system, this single - chip microcomputer is equipped with a high - performance processor[23]. It features a quad - core Cortex - A72 CPU with a main frequency of up to 1.5 GHz, which can quickly process various data and instructions. It has multiple USB interfaces, HDMI interfaces, GPIO interfaces, etc., enabling convenient connection of various external devices. It can run multiple operating systems such as Linux and Windows IoT, providing developers with abundant choices. In this system, it is responsible for receiving the humidity data transmitted from the soil moisture sensor module and determining whether watering is required according to the preset threshold.

## 2.2 Soil Moisture Sensor Module

The soil moisture sensor module on the trolley is used to collect soil humidity data, which is then transmitted to the transmitting end of the JDY - 40 2.4G wireless transmission module[24]. The signal is processed by the receiving end of the JDY - 40 2.4G wireless transmission module to monitor the humidity of the flower soil in real - time and transmit the humidity data to the Raspberry Pi 4B single - chip microcomputer[25]. When the soil humidity is lower than the preset threshold, the system will automatically trigger the water pump drive circuit to water the plants, and vice versa. The automatic watering function is realized by driving a relay. The circuit design of the soil moisture sensor is shown in Figure 1.

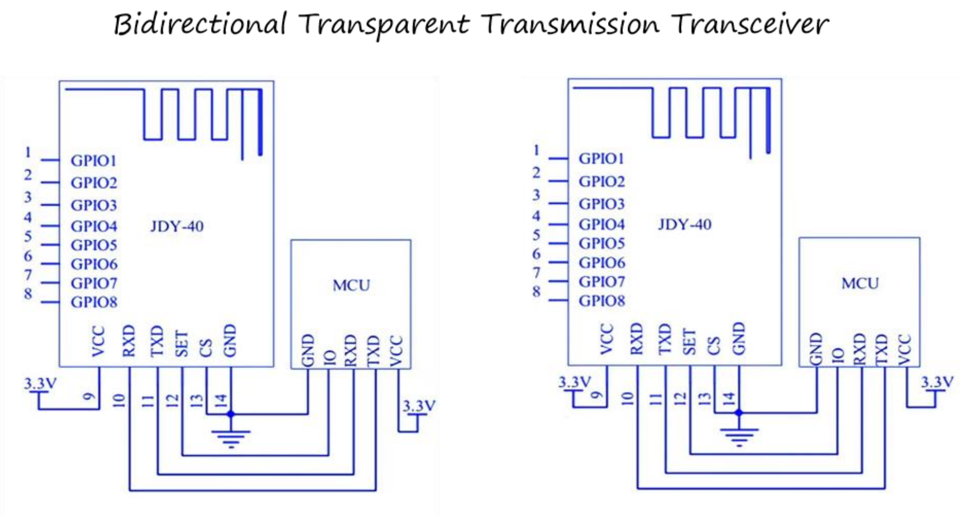


Figure 1. The circuit design of soil moisture sensor is shown in the figure.

## 2.3 Water Pump Drive Circuit

A 5V DC water pump is used as the water inlet and outlet device. The transistor model adopted is 8550, with its emitter connected to the positive pole of the power supply and its collector connected to the coil pin of the relay to drive the relay to close[26]. The base is controlled by the main control circuit of the single - chip microcomputer through a resistor to indirectly control the start and stop of the water inlet pump. The design of the water pump drive circuit is shown in Figure 2.

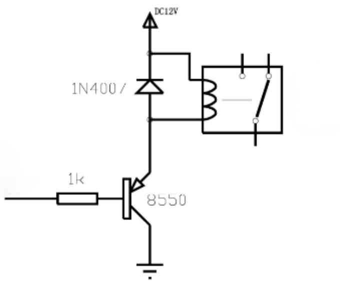


Figure 2. Water Pump Drive Circuit Design.

# 3. Software Design

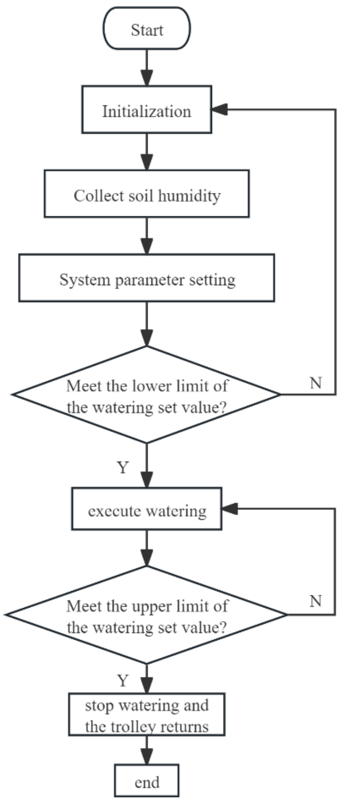


Figure 3. Software design flow chart.

The software is written in Python. After the device is powered on, the system first initializes. Then, the system reads the stored humidity value and saves it in a variable for comparison with the measured soil humidity. If the measured value is lower than the preset soil humidity threshold, the water pump will be activated to water the plants. If the measured value meets the set value for watering, the watering will stop and the trolley will return along the original path. The design flow of the software program is shown in Figure 3.

# 4. Comprehensive Debugging

Comprehensive debugging is a crucial step to ensure the performance and stability of the intelligent flower - watering system, covering multiple key parts of hardware and software. After completing the system design, debugging of hardware and software is required.

## 4.1 Hardware Debugging

### 4.1.1 Soil Moisture Sensor Debugging

The soil moisture sensor, as a key component for determining the soil moisture condition, its accuracy directly affects the watering decision of the flower - watering system. First, conduct a visual inspection to ensure the sensor has no obvious physical damage, the pin connections are firm, and there is no short - circuit or open - circuit. Use standard moisture samples to perform initial calibration on the sensor. Prepare standard soil samples with different moisture levels, insert the sensor into the samples, measure and record the electrical signal values output by the sensor. Establish a calibration curve by comparing the relationship between the actual moisture and the sensor output signal. Then conduct tests in the actual soil environment. Insert the sensor into the flower pot soil to be monitored, measure the soil moisture at different time points, and observe whether the sensor output is stable and conforms to the actual soil moisture change. Analyze and record factors that may affect the accuracy of the sensor, such as soil texture, salt content, etc. If there is a deviation between the sensor output and the actual moisture, further adjust the calibration parameters until the sensor can accurately reflect the soil moisture.

### 4.1.2 OpenCV Visual Recognition Debugging

The OpenCV visual recognition module is used to locate the position of flowers. Its debugging process needs to focus on the accuracy of image acquisition and processing. First, ensure the camera is correctly connected to the single - chip microcomputer, the power supply is stable, and the data transmission line has no faults. Then start the system, observe whether the images collected by the camera are clear, complete, and free of obvious distortion or noise. Check whether parameters such as the resolution and frame rate of the images meet the system design requirements. Test the operation of the visual recognition algorithm under different lighting conditions and flower placement positions, and observe whether the system can accurately identify the approximate position of the flowers. Conduct a detailed analysis of the recognition errors. Possible reasons include uneven lighting leading to unclear image features, mismatch between the flower shape or color and the preset model of the algorithm, etc. For these problems, adjust the parameters of the visual recognition algorithm, such as thresholds, feature extraction methods, etc., to improve the recognition accuracy.

#### 4.1.3 Signal Transmission Debugging

The stability of signal transmission is crucial for the coordinated work between various modules of the system. For wireless signal transmission (JDY - 40 2.4G wireless transmission module), correctly connect the wireless transmission module to relevant hardware devices, ensure the connection is firm, and the antenna is installed correctly. Simulate sending various data at the transmitting end, and check whether the data received at the receiving end is complete, accurate, or has error phenomena. Conduct a detailed investigation of the transmission errors that occur. Possible reasons include interference sources, improper transmission frequency settings, etc. Take corresponding measures to adjust, such as changing the frequency band, eliminating interference sources, etc.

#### 4.1.4 Water Pump Drive Circuit Debugging

The water pump drive circuit is responsible for controlling the start and stop of the water pump. Its debugging focus is on ensuring the normal operation of the circuit and effective control of the water pump. Carefully check whether the connections between the water pump drive circuit and the Raspberry Pi 4B single - chip microcomputer, power supply, and water pump are correct, ensure there is no short - circuit or open - circuit in the lines, and the solder joints are firm. Use tools such as a multimeter to measure the voltage and current output by the drive circuit, and ensure it can provide sufficient power to drive the water pump to work normally. Send control instructions through the single - chip microcomputer, check whether the water pump can accurately start and stop according to the instruction requirements, realize the automation of flower watering, and make the water pump drive circuit can accurately respond to the control instructions of the single - chip microcomputer.

#### 4.1.5 Raspberry Pi 4B Single - Chip Microcomputer Debugging

The Raspberry Pi 4B single - chip microcomputer, as the core control unit of the system, its debugging involves multiple aspects, including hardware connection and software operating environment. During the test process, it is necessary to check whether the connections between the single - chip microcomputer and various external devices (such as sensors, cameras, water pump drive circuits, etc.) are correct, ensure all pin connections are firm, and the data transmission lines are normal. Check the stability of the power supply of the single - chip microcomputer, and the voltage meets the requirements. Use a voltmeter to measure the power input voltage, check whether there is voltage fluctuation or too low or too high voltage. If there is a problem, adjust the power supply in a timely manner. Then power on the single - chip microcomputer, observe whether its startup process is normal. Check whether the startup indicator light is on and whether it can normally load the operating system. If the startup fails, it is necessary to re - check the hardware connection and power supply, and troubleshoot possible software problems, such as boot program errors.

## 4.2 Software Debugging

In the software debugging phase, to ensure the accuracy and stability of software functions, a modular debugging strategy was strictly adopted. Each critical part of the system was thoroughly and meticulously debugged. The detailed process is as follows.

### 4.2.1 Debugging of the Cart Function Module

First, the control codes for the cart's basic motions, including forward movement, backward movement, and turning, were comprehensively inspected. Elaborate test cases were carefully designed within the codes to simulate various potential instruction inputs. The actual responses of the cart were closely observed to verify their consistency with the expected results. Simultaneously, the logical relationships within the control codes were deeply analyzed to identify possible errors. For instance, the calculation process of the steering angle was repeatedly verified for correctness, and the rationality of the forward - backward speed settings was assessed to ensure that the speed parameters met the operational requirements in real - world scenarios. Moreover, the key variables and parameters in the codes were meticulously examined one by one to ensure that their initial values and variation patterns fully conformed to the original design requirements. Subsequently, multiple combinations of start and end positions were set for the cart, and simulated environments with complex obstacle arrangements and distributions were built. This was to thoroughly check whether the cart could quickly and accurately calculate a reasonable driving path based on real - time environmental information. During the cart's movement, continuous observation was carried out to ensure that it strictly followed the planned path and effectively prevented route deviation. When encountering obstacles, the cart was tested for its capability to promptly and keenly detect the obstacles and rapidly implement effective obstacle - avoidance measures. By simulating obstacles of different shapes and sizes, the response speed of the cart and the scientific rationality of its obstacle -

avoidance strategies (such as deceleration, turning, or stopping) were observed multiple times. It was verified whether the cart could execute appropriate operations at the right moments to successfully avoid collisions, thus ensuring a reliable autonomous driving ability in complex environments.

#### 4.2.2 Debugging of the Soil Moisture Reading Module

During the debugging of the soil moisture reading module, the communication interface between this module and the hardware sensors was first thoroughly inspected. Measures were taken to ensure the accurate and stable transmission of data between the sensors and the software module, preventing problems like data loss or transmission errors. The communication protocol was carefully verified to ensure its correct implementation as per the design, and the data format was checked to ensure full compliance with the specified standards. To further validate the accuracy of the data collected by the soil moisture reading module, extensive tests were conducted in soil environments with known moisture levels. The data collected by the software was compared with the actual moisture values one by one. Through repeated testing, the error range was comprehensively statistically analyzed. This ensured that the data collected by the module could truly and accurately reflect the actual soil moisture situation, providing reliable data support for subsequent work based on the soil moisture data.

### 4.3 System - level Debugging

#### 4.3.1 Function Completeness Test

Conduct comprehensive tests on the system in real - world flower - watering scenarios to ensure that all designed functions are fully realized. These include soil moisture monitoring, flower position recognition, water pump - driven watering, and the system's automatic control and reset functions. Evaluate the system's performance under different plant species and flower pot layouts. Since different plants have varying water requirements, the system should be able to adjust the watering strategy flexibly based on preset thresholds.

#### 4.3.2 Performance Test

Measure the response time of the system from detecting soil moisture below the threshold to initiating the watering program, as well as the time from detecting soil moisture above the threshold to stopping the watering program. Ensure that the system has a sufficiently fast response speed to meet the timely water needs of plant growth. Run the system continuously for an extended period to observe potential failures or anomalies. For example, check for data drift in sensors, errors in visual recognition, or abnormal starts/stops of the water pump. Ensure the system maintains sufficient stability during continuous operation. Conduct a comprehensive evaluation of the accuracy of soil moisture monitoring, flower position recognition, and watering control. Compare the system's results with standard measurement tools and manual observations to ensure the system can provide accurate services that meet the needs of plant growth.

#### 4.3.3 Reliability Test

Test the system's reliability under various environmental conditions, including the impacts of temperature variations, humidity changes, and electromagnetic interference on system performance. Ensure that the system can still operate normally under harsh environmental conditions and provide stable watering services for plants.

## 5. Conclusion

This design successfully constructs an automated and intelligent automatic watering system based on the Raspberry Pi 4B single - chip microcomputer, integrating technologies such as soil moisture sensors, OpenCV visual recognition, and Bluetooth modules. Hardware - wise, the system uses the Raspberry Pi 4B as the core control unit, equipped with a soil moisture sensor module to collect soil humidity data in real - time. It realizes data interaction through the JDY - 40 wireless transmission module, accurately locates the position of flowers with the OpenCV visual recognition technology, and completes the automatic watering action via the water pump drive circuit. In terms of software, the system adopts Python language to write programs, achieving the full - process automation from system initialization, humidity data reading and comparison, to water pump start - stop control and trolley path planning. Through comprehensive debugging, in terms of hardware, after calibration, the soil moisture sensor can accurately reflect the actual soil humidity. The OpenCV visual recognition can locate flowers relatively accurately under different lighting conditions. The signal transmission is stable without obvious bit errors. The water pump drive circuit responds sensitively, and the Raspberry Pi 4B single - chip microcomputer is normally connected to each module and runs stably. For software, the trolley's basic motion control is accurate, the path planning and obstacle avoidance strategies are reasonable, and the data collected by the soil humidity reading module has an error within an acceptable range. The system as a whole has complete functions, can automatically adjust the watering strategy according to the water demand thresholds of different plants, has a fast response speed, high continuous operation stability, and can work reliably under different environmental conditions. This automatic watering system effectively solves the problems of traditional watering methods and existing automatic watering equipment, such as the inability to dynamically adapt to the water demand of plants and poor environmental adaptability, meeting people's needs for the intelligence and automation of watering equipment. With low cost, simple operation, and high reliability, it has certain promotion value and provides a convenient and scientific solution for plant maintenance in families and offices.

## References

[1] X. Zhang and H. Khachatryan, "Investigating homeowners' preferences for smart irrigation technology features," *Water,* vol. 11, no. 10, p. 1996, 2019.

[2] B. L. R. Stojkoska and K. V. Trivodaliev, "A review of Internet of Things for smart home: Challenges and solutions," *Journal of cleaner production,* vol. 140, pp. 1454-1464, 2017.

[3] A. D. Gupta, P. Pandey, A. Feijóo, Z. M. Yaseen, and N. D. Bokde, "Smart water technology for efficient water resource management: A review," *Energies,* vol. 13, no. 23, p. 6268, 2020.

[4] E. A. Abioye *et al.*, "A review on monitoring and advanced control strategies for precision irrigation," *Computers and Electronics in Agriculture,* vol. 173, p. 105441, 2020.

[5] O. Bodunde, U. Adie, O. Ikumapayi, J. Akinyoola, and A. Aderoba, "Architectural design and performance evaluation of a ZigBee technology based adaptive sprinkler irrigation robot," *Computers and Electronics in Agriculture,* vol. 160, pp. 168-178, 2019.

[6] F. De Lorenzi *et al.*, "Adaptability to future climate of irrigated crops: The interplay of water management and cultivars responses. A case study on tomato," *Biosystems engineering,* vol. 157, pp. 45-62, 2017.

[7] X. Tian, J. Li, and D. Li, "Design of automatic watering system based on STC89C52 MCU," *Journal of Computational Methods in Science and Engineering,* vol. 18, no. 4, pp. 949-961, 2018.

[8] Y. Wang, S.-d. Lin, and S.-y. Tang, "Design and application of small-sized intelligent irrigation system based on single-chip microcomputer," 2023.

[9] K. S. Nemali and M. W. van Iersel, "An automated system for controlling drought stress and irrigation in potted plants," *Scientia horticulturae,* vol. 110, no. 3, pp. 292-297, 2006.

[10] E. Wang *et al.*, "Development of a closed-loop irrigation system for sugarcane farms using the Internet of Things," *Computers and Electronics in Agriculture,* vol. 172, p. 105376, 2020.

[11] D. D. Avgoustaki, I. Avgoustakis, C. C. Miralles, J. Sohn, and G. Xydis, "Autonomous mobile robot with attached multispectral camera to monitor the development of crops and detect nutrient and water deficiencies in vertical farms," *Agronomy,* vol. 12, no. 11, p. 2691, 2022.

[12] N. K. Nawandar and V. R. Satpute, "IoT based low cost and intelligent module for smart irrigation system," *Computers and electronics in agriculture,* vol. 162, pp. 979-990, 2019.

[13] C. Anagha, P. M. Pawar, and P. Tamizharasan, "Cost-effective IoT-based intelligent irrigation system," *International Journal of System Assurance Engineering and Management,* vol. 14, no. Suppl 1, pp. 263-274, 2023.

[14] Y. Zhang *et al.*, "Review of the field environmental sensing methods based on multi-sensor information fusion technology," *International Journal of Agricultural and Biological Engineering,* vol. 17, no. 2, pp. 1-13, 2024.
[15] I. A. Lakhiar, G. Jianmin, T. N. Syed, F. A. Chandio, N. A. Buttar, and W. A. Qureshi, "Monitoring and control systems in agriculture using intelligent sensor techniques: A review of the aeroponic system," *Journal of sensors,* vol. 2018, no. 1, p. 8672769, 2018.
[16] M. N. Murshed, "Smart Plant Watering System: A Mobile Robotic Solution for Precise Plant Hydration in Indoor Environments," 2024.
[17] C. Maraveas, "Incorporating artificial intelligence technology in smart greenhouses: Current State of the Art," *Applied Sciences,* vol. 13, no. 1, p. 14, 2022.
[18] M. B. Wasserman, "The Plant Water Management Experiments," Portland State University, 2022.
[19] S. Ishak, N. N. Abd Malik, N. A. Latiff, N. E. Ghazali, and M. Baharudin, "Smart home garden irrigation system using Raspberry Pi," in *2017 IEEE 13th Malaysia International Conference on Communications (MICC)*, 2017: IEEE, pp. 101-106.
[20] K. P. Paudel, M. Pandit, and R. Hinson, "Irrigation water sources and irrigation application methods used by US plant nursery producers," *Water Resources Research,* vol. 52, no. 2, pp. 698-712, 2016.
[21] R. Karthikamani and H. Rajaguru, "IoT based smart irrigation system using Raspberry Pi," in *2021 Smart Technologies, Communication and Robotics (STCR)*, 2021: IEEE, pp. 1-3.
[22] J. K. Ng, H. Li, V. C. Cheng, and W. K. Cheung, "An environmental-adaptive Wi-Fi localization approach with low start-up cost for the exhibition industry," in *2019 IEEE 25th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*, 2019: IEEE, pp. 1-8.
[23] L. Zhang and H. Wu, "Application of single chip technology in internet of things electronic products," *Journal of Intelligent & Fuzzy Systems,* vol. 40, no. 2, pp. 3223-3233, 2021.
[24] W.-Y. Chung, J. F. Villaverde, and J. Tan, "Wireless Sensor Network Based Soil Moisture Monitoring System Design," in *FedCSIS (Position Papers)*, 2013, pp. 79-82.
[25] I. M. Kulmány *et al.*, "Calibration of an Arduino-based low-cost capacitive soil moisture sensor for smart agriculture," *Journal of Hydrology and Hydromechanics,* vol. 70, no. 3, pp. 330-340, 2022.
[26] S. Chandran and R. Gokaraju, "Design considerations for relay and circuit breaker," in *Switchgear Design, Operation, and Maintenance Using Industry Standards*: Elsevier, 2025, pp. 71-94.