An IoT-Based Smart Irrigation System for Home Gardening: Towards Water-Efficient and Low-Effort Household Sustainability

Tan Kai Ting[1], Aine Izzati Tarmizi[1*], Anis Niza Ramani[1], Radi Husin Ramlee[2]

[1]Faculty of Electrical Technology and Engineering, Universiti Teknikal Malaysia Melaka (UTeM), Melaka, Malaysia

[2]Faculty of Electronics and Computer Technology and Engineering, Universiti Teknikal Malaysia Melaka (UTeM), Melaka, Malaysia

***Corresponding Author**

DOI: https://doi.org/10.47772/IJRISS.2026.101000544

Received: 26 October 2026; Accepted: 31 October 2026; Published: 07 November 2026

ABSTRACT

Home gardeners often water by habit and visual judgement, which can lead to overwatering, underwatering, and wasted water; watering also depends on the gardener being available at the right time. This study aimed to design, build and test a low-cost Internet of Things (IoT) smart irrigation system for small-scale home gardens that waters according to a time schedule and measured environmental conditions. An ESP32 microcontroller reads two capacitive soil moisture sensors, a YL-83 raindrop sensor, and a YF-S201 flow sensor to decide when to water each of two potted-plant zones, and switches two submersible pumps through a relay module that feeds a drip arrangement. Watering follows a user-set daily schedule but is withheld when the soil is already moist or rain is detected; users can monitor readings, set pump durations, and override the schedule through the Blynk mobile application. The prototype was verified sensor by sensor and then in scheduled and manual modes, with sensor and pump states logged over five days and plant height recorded over seven days. The sensors responded correctly to dry, wet and rainy conditions, and the controller withheld water in four of the ten zone-level watering decisions logged because the soil was sufficiently moist or rain was present. Plants in both zones grew steadily, reaching 5 cm and 4 cm after seven days. Because the trial was short, small, and uncontrolled, we did not quantify water savings or growth benefits. The results nevertheless show that an affordable, sensor-driven, remotely manageable system can reduce manual watering effort and unnecessary watering and may help households adopt more sustainable gardening routines.

Keywords: Smart irrigation; Home gardening; Internet of Things; Water conservation; ESP32

INTRODUCTION

Home gardening is becoming more popular as people seek to grow ornamental plants, vegetables and herbs in the limited spaces around their homes. Among the many aspects of plant care, watering is one of the most important, because different plants need different amounts of water to grow well. Smart irrigation systems have therefore gained attention as a modern way to water plants: they monitor surrounding conditions, follow an adjustable schedule, and water by time and area[1].

Traditional watering depends on manual effort and human judgment, using watering cans, hoses, or other simple tools. Because people estimate how much water a plant needs rather than measure it, they often waste water; one industry source reports that as much as 50% of water is lost to overwatering through inefficient manual watering in the United States[2]. Inefficient practices can also harm plants through overwatering or underwatering. Smart garden irrigation systems have been developed to provide a reliable, fully automated alternative that regulates watering based on what plants actually need[3].

The problem is especially visible in home gardens. Many home gardeners have limited understanding of different plants' water requirements, so they base watering decisions on personal judgment and visual

observation rather than actual plant conditions. Irregular watering can reduce photosynthetic efficiency, damage cells, and cause root rot[4]. Watering also depends on the gardener being available at the right time, which is difficult for people with busy routines. From a household-sustainability perspective, daily watering is a small but repeated resource-use decision, and a system that makes that decision based on measured conditions rather than habit offers a practical way to conserve water and reduce effort.

This paper reports the design, implementation, and testing of an IoT-based smart irrigation system for home gardening. The objectives were to design and develop a system that automates plant watering based on a schedule and surrounding environmental conditions. The objectives were to implement an IoT-based platform for remote monitoring and control of irrigation and to test how effectively the system manages water use in a home-gardening setting. The scope is limited to a small-scale home garden with a few plant pots watered by drip irrigation, which delivers water slowly to the roots with minimal loss; large-scale field and commercial farming are outside the scope.

The paper first reviews relevant concepts, components, and earlier studies; then describes the system design and test procedure; presents the results; discusses their implications and limitations; and concludes with directions for future work.

LITERATURE REVIEW

Irrigation and IoT in the Home Garden

An irrigation system supplies water to plants when natural rainfall is not enough, delivering it from a storage or supply source in a controlled manner. Common methods include surface (flood), drip, sprinkler, centre-pivot, and sub-irrigation[5]. Among these, drip irrigation is considered the most efficient and water-saving because it delivers water where it is needed. A smart irrigation system adds sensing, control, and communication to this delivery method so plant conditions can be monitored and acted upon[6].

The Internet of Things connects physical devices to the internet so that they can collect and exchange data. It is commonly described through four elements: devices (sensors and processors), connectivity (for example, Wi-Fi or Bluetooth), a platform that receives and analyses data and returns instructions, and a user interface such as a dashboard[7]. In a home garden, these elements allow real-time monitoring and remote operation without constant presence. Such systems also let gardeners receive alerts and review irrigation history on a smartphone, supporting a combined approach of local sensing and cloud-based data storage[8]. In this architecture, the microcontroller acts as the edge node that reads soil moisture, water level, and rainfall signals and decides locally whether to run a pump, while the cloud platform provides a remote interface for scheduling, threshold adjustment, and manual override[9], [10].

Core Components

The ESP32 is a low-power, dual-core microcontroller with built-in Wi-Fi and Bluetooth, up to 520 KB of SRAM, and a wide range of GPIO, ADC, and DAC functions, making it suitable as the central unit of an IoT system[11]. Capacitive soil moisture sensors measure the change in dielectric properties caused by soil water; because their electrodes are not exposed to the soil, they suffer less corrosion and last longer than resistive probes[6]. The YL-83 raindrop module, based on an LM393 comparator, detects rain by changes in conductivity across its sensing pad and is widely used in educational and prototype systems because it is inexpensive and simple to interface. In the water-level and pump-control stage of the present design, the ESP32 reads the sensor's digital output. It compares it against a calibrated threshold before energising the relay module, so the pump switches on only when soil or reservoir conditions warrant it. This control logic means the pump is never run on a fixed timer alone, so water is applied only when measured conditions indicate a need, reducing unnecessary irrigation and limiting waste [12]. Because the controller combines a schedule with live sensor readings, the same logic can also be adapted to other inputs, such as the soil moisture at several depths or the quality of the water being supplied[13], [14].

The YF-S201 is a Hall-effect water flow sensor whose rotor produces electrical pulses in proportion to flow; its datasheet gives the pulse frequency as 7.5 times the flow rate in liters per minute[15]. A relay module lets a low-power controller switch a high-power load and provides electrical isolation between them. An LM2596 buck converter efficiently steps a battery or other higher-voltage supply down to the lower voltages needed by the controller and sensors[16].

Related Studies

Three earlier studies illustrate the range of design choices. In one study, they built a home-garden system around a Raspberry Pi that reads soil moisture, light, and ultrasonic water-level sensors and starts watering from an Android application; when the water level falls below 15 cm, it sends an email asking the user to open the valve[8], [17], [18]. Researchers used an ESP32 with the Arduino IoT Cloud and Alexa voice commands to combine environmental monitoring (room temperature, soil moisture, and gas) with fan and pump control, and developed a ZigBee-based system with sensor nodes, a master controller, and a server that can run in manual mode or in fuzzy-logic mode that decides watering based on soil conditions.

These studies share a common architecture of sensors, a controller, and an actuator linked to an IoT interface, but they differ in cost, communication method, and actuator type. The Raspberry Pi and tap-fed solenoid-valve designs are comparatively costly or depend on water pressure, whereas a tank-fed submersible pump needs no external pressure. Suggestions from those studies, such as push notifications, an automatic override, and additional environmental sensors, indicate what a low-cost home system should offer. This work therefore combines a low-cost ESP32 controller, rain-aware scheduling, per-zone soil sensing, flow measurement, and manual override in a single tank-fed drip system.

Social and Behavioural Perspective

Because the related studies are engineering papers, their social and behavioural assumptions are implicit rather than examined. Researchers keep the gardener in the loop: the system alerts the user by email and relies on the user to respond, which assumes a smartphone, regular email use, and prompt action. Masum *et al.* [17] place garden care within a connected household and add voice commands, which may lower the technical skill needed to operate the system, although they did not test this with users. Al-Ali *et al.* [18] let users choose between manual control and an automatic mode, raising the question of how much control people want to hand to a system.

Several questions of interest to the social sciences remain open in this literature. None of the three studies reports user trials, so acceptance, ease of use, and trust in automated watering remain unknown. None measure whether households water differently or use less water after adopting the system, so the link between such systems and sustainable behaviour is assumed rather than shown. Automation may also reduce gardeners' own attention to plant needs, which manual modes only partly address. Finally, the designs depend on smartphones, wireless connectivity and, in some cases, relatively costly hardware, and none discusses who can realistically adopt them. The motivation that busy people find it hard to water regularly appears throughout this work, but it is supported by industry sources such as [2] rather than by peer-reviewed behavioural research. The present study therefore treats the social benefits of smart irrigation as a motivation and a hypothesis. It identifies user studies of acceptance, water-use habits and affordability as a necessary next step.

METHODOLOGY

Development proceeded through hardware design, software development, and system testing. The prototype was built for two plant pots (Zone 1 and Zone 2), each fed by its own pump and drip emitter from a shared water tank.

System Architecture and Hardware

Fig. 1 shows the block diagram. A battery pack supplies the buck converter, which provides regulated power to the ESP32 and the connected devices. The ESP32 reads the soil moisture sensors, the rain sensor, and the flow

sensor, and switches the water pumps through the relay module. Fig. 2 shows the circuit, drawn in an online circuit-design tool.

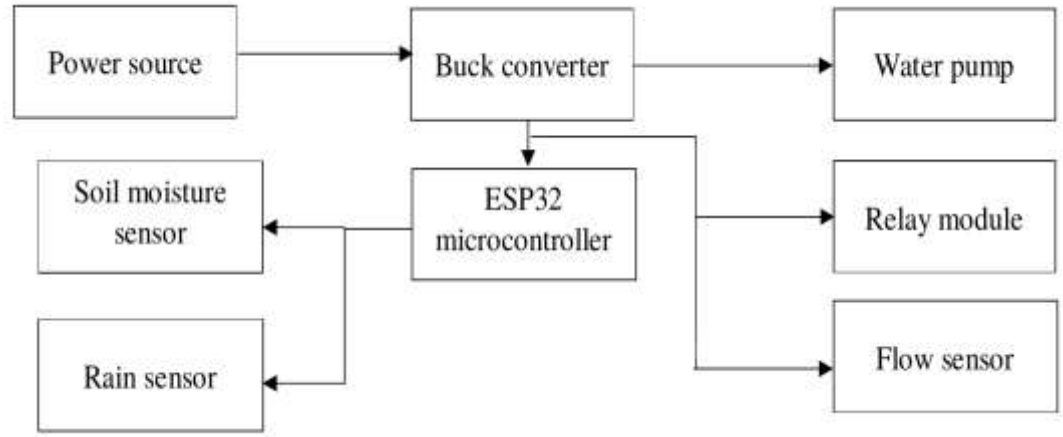


Fig. 1 Block diagram of the smart irrigation system

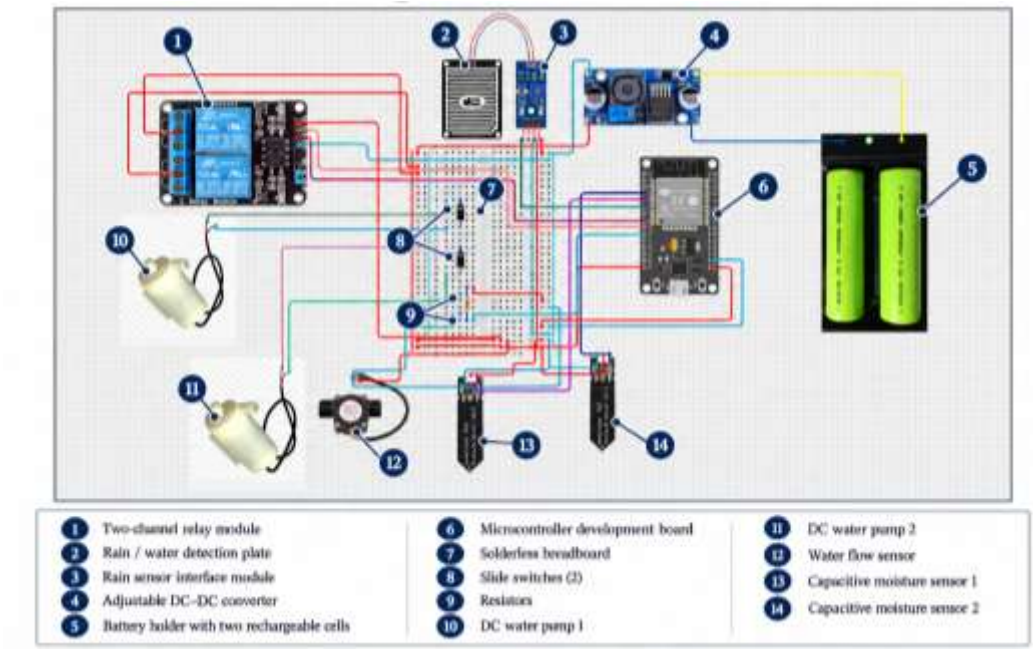


Fig. 2 Circuit diagram of the system

The main components were an ESP32 microcontroller, two capacitive soil moisture sensors (v1.2), a YL-83 raindrop module, a YF-S201 flow sensor (about 1–30 L/min), two 3–6 V DC mini submersible pumps, a 5 V 2-channel relay module, and an LM2596 adjustable buck converter. A 10 kΩ pull-up resistor stabilises a digital input, a 1N4007 diode protects against inductive spikes from the pump motors, and a 47 µF capacitor smooths the supply. The ESP32 was preferred to an Arduino UNO for its built-in wireless connectivity and higher processing capability. The capacitive moisture sensor was preferred to a resistive one for durability, and the relay module to a MOSFET switch or solid-state relay for its simple interface and stable switching. Submersible pumps were chosen for their low power consumption and because, unlike solenoid valves, they need no external water pressure.

Control Logic and Software

The firmware was written in C++ using Arduino IDE 2.3.10, and the Blynk IoT platform provides the cloud service and the mobile dashboard. On start-up, the program initialises the pins, sensors, and flow-sensor interrupt, connects to Wi-Fi and Blynk (reconnecting automatically if either drops), and synchronises the clock

with an NTP server. The main loop then continuously reads the sensors, publishes the values to Blynk, updates the flow measurement and checks the irrigation schedule.

At the user-set time, the system evaluates each zone independently. Because a capacitive sensor gives a higher reading in drier soil, a zone is irrigated only if its soil moisture reading exceeds the threshold of 1000 (raw 12-bit ADC value, 0–4095) and no rain is detected; the pump then runs for the duration set for that zone and switches off automatically to prevent overwatering. An interrupt routine counts flow-sensor pulses, converts them to litres per minute, and accumulates total consumption. In manual mode, the user can start both pumps from the Blynk application, and they run for their set durations. The dashboard shows soil moisture for both zones, rain status, flow rate, pump status, the irrigation time, and the pump-duration sliders.

Testing and Validation

Testing had three stages. At first, each sensor was checked individually: the moisture sensors in dry and wet soil, the raindrop module with water droplets, and the flow sensor with water passing through the pipe, with outputs read on the serial monitor and the Blynk application. Second, the program was uploaded to the ESP32, and relay control was verified through sensor-based and user-initiated pump switching. Third, the whole system was validated in scheduled and manual modes, and the sensor and pump states were logged over five days. Plant growth was then observed over seven days by recording plant height in each zone daily from photographs. During the growth observation, pump durations were set to 25 s for Pump 1 and 15 s for Pump 2.

RESULTS

Sensor Verification

All three sensor types produced readings consistent with the expected behaviour (Table 1). The soil moisture sensor gave high readings in dry soil and low readings in wet soil; the raindrop module output fell as more water covered the pad, and the flow sensor generated pulses only when water flowed. All outputs were displayed in real time on the serial monitor and in the Blynk application.

Table 1 Sensor readings under test conditions

Sensor	Condition	Actual reading	Expected	Result
Soil moisture	Dry soil	>2500	High	OK
	Wet soil	<1500	Low	OK
Raindrop module	No water	4095	High	OK
	Little water	2000–3000	Medium	OK
	Much water	<2000	Low	OK
Flow sensor	No flow	0 L/min	0 pulse	OK
	Water flowing	1.5–25 L/min	Pulses	OK

Scheduled and Manual Irrigation

In the scheduled test, Zone 1 read 794 at the set time, below the threshold and indicating wet soil, so Pump 1 stayed off even though no rain was present. Zone 2 read 1057 with no rain, so Pump 2 switched on and watered for the 10 s set in the application (Fig. 3). In manual mode, both pumps switched on when the user activated the control and watered both zones for their set durations.

ISSN No. 2454-6186 | DOI: 10.47772/IJRISS | Volume X Issue X October 2026

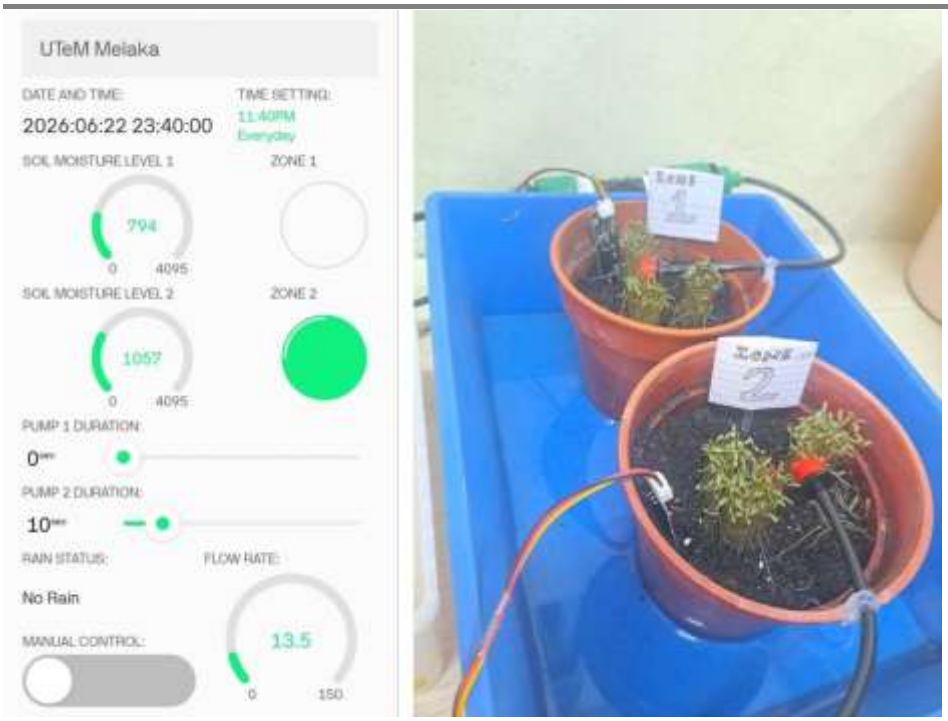


Fig. 3 (a) Blynk dashboard at the scheduled time and (b) the two plant zones with drip emitters during the scheduled-watering test

Table 2 lists the sensor and pump states logged over five days. When a zone's reading was medium or high, and no rain was present, the pump switched on; when the reading was low (wet soil), or light rain was detected, the pump stayed off. Of the ten zone-level watering decisions in the log, six resulted in watering, and four were withheld: Pump 1 on Day 2, Pump 2 on Day 3, and both pumps on Day 4, when light rain was detected.

Table 2 Sensor and pump states over five days

Day	Soil 1	Soil 2	Flow (L/min)	Rain	Pump 1	Pump 2
1	High	High	21.8	No rain	ON	ON
2	Low	Medium	–	No rain	OFF	ON
3	Medium	Low	24.3	No rain	ON	OFF
4	Medium	Medium	–	Light rain	OFF	OFF
5	Medium	High	19.7	No rain	ON	ON

Reading classes: High >1500; Medium 1000–1499; Low <1000. A dash indicates no flow because no pump was on.

Plant Growth

Plant height was recorded daily for seven days (Fig. 4). Zone 1 grew from 0 cm on Day 1 to 5 cm on Day 7, and Zone 2 from 0 cm (0.3 cm on Day 3) to 4 cm. Both plants showed continuous increases in height throughout the observation period.

ISSN No. 2454-6186 | DOI: 10.47772/IJRISS | Volume X Issue X October 2026

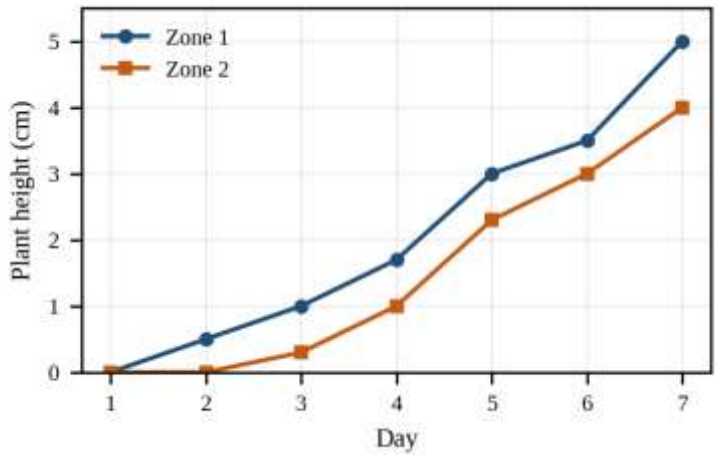


Fig. 4 Plant height in Zone 1 and Zone 2 over seven days

DISCUSSION

The results show that the system performed its intended function: it measured soil and rain conditions, decided zone by zone whether to water, delivered water for a limited duration, and reported its state remotely. The most useful behaviour for water conservation was withholding water. In four of ten zone-level decisions, the system did not irrigate because the soil was already moist or rain was present, which are the situations in which a manual or purely timer-based routine would typically water anyway. The rain sensor and the per-zone soil sensors therefore directly address the overwatering described in the Introduction. The flow sensor adds visibility into how much water is delivered, and the duration limit automatically stops each watering event.

The system also lowers the effort of caring for plants. Watering follows a schedule without the gardener being present; readings and pump status can be checked from a smartphone, and manual override is available when the user wants to water immediately. For households, this is a practical way of making a small daily resource-use decision based on measurement rather than habit. The low-cost components, open programming environment, and free-tier IoT platform suggest that such a system could be affordable for home users, although this study did not analyse cost.

The growth observation shows that plants in both zones kept growing under the automatic regime, but it cannot show that the system improved growth. The two pumps ran for different durations (25 s and 15 s), so the zones received different amounts of water, which may partly explain the height difference; seed emergence time and individual plant variation may also contribute.

Demonstrated Functionality and Anticipated Benefits

Read the results as evidence of technical functionality, not environmental or household benefit. Table 4 separates what this study demonstrated from what is anticipated but not yet shown.

Table 4 Demonstrated functionality and anticipated benefits

Demonstrated in this study	Anticipated, not yet demonstrated
Soil, rain, and flow sensors responded as expected in bench checks	Lower total water use than with manual or timer-based watering
Zone-level watering decisions followed the soil and rain rule over five days	Fewer irrigation events or more stable soil moisture than with a timer
Scheduled and manual operation, and remote	Reduced effort and more sustainable watering habits in

monitoring through Blynk	households
Plants in both zones grew during a seven-day observation	Better plant growth than with other watering methods
Pumps stopped automatically after the set duration	Accurate water-volume measurement; cost advantage; long-term reliability

Social Benefit and Behavioural Perspective

The results suggest three ways in which the system could benefit households, each tied to what was observed in the trial. First, the watering decision moved from the gardener's judgment to measured conditions. In the five-day log, the controller watered a zone only when its soil reading was medium or high and no rain was present, and it withheld water in four of ten zone-level decisions, including the day with light rain (Table 2). A household using this system would not need to check the soil or weather before each watering, addressing the identified problem that watering is often decided by habit and visual estimation.

Second, the system reduced the need for the gardener to be present. Scheduled watering ran at the time set in the Blynk application, and the dashboard showed soil moisture, rain status, flow rate, and pump settings on a smartphone (Fig. 3). For people with busy routines, this shifts the task from daily attendance to occasional checking. Third, the user stayed in control: the application let users adjust watering time, each pump's duration, and manual override, and manual mode watered both zones when the user chose. This matters behaviourally because an automated system that cannot be overridden may be less acceptable to the people who use it.

Socio-technical Considerations

Using IoT in irrigation ties the technical design to how households actually live with the system, and several design choices in this study carry social consequences. The first is dependence on connectivity. Remote monitoring and control rely on home Wi-Fi and the Blynk cloud platform, so the household's convenience depends on a network and a third-party service. The firmware reconnects automatically if either drops, but the trial did not examine what a household would experience during a prolonged outage or if the platform changed.

Next is the balance between automation and user control. The system makes routine decisions, while the user keeps the schedule, the pump durations, and a manual override. Showing soil moisture, rain status, flow rate, and pump state on the dashboard lets users check what the system is doing, which builds trust. Automation also changes the work rather than removing it: the gardener no longer waters daily but must refill the tank, maintain the pumps, tubing and sensors, and keep the battery supply running.

Finally, adoption depends on conditions outside the device. Users need a smartphone, a wireless connection, and basic digital skills, and building and configuring this prototype required electronics knowledge most home gardeners do not have, so wider use would require a ready-made, easily configured product. The components are low-cost, but cost was not analysed, and readings are stored on a cloud platform over which the household has limited control. These points mean the trial supports the system's technical feasibility, not its acceptance, trust, or effect on behaviour, which would require a study with home gardeners.

Limitations

Several limitations should be noted. The trial was short, consisting of five days of pump logs and seven days of growth data using two pots and had no control group watered manually or by timer, so water savings and growth benefits are not quantified; the reduction in unnecessary watering is inferred from the logged decisions and from the 50% overwatering figure reported in[2], not measured. The flow sensor used the factory conversion constant and was not calibrated against a measured volume, so the logged flow rates should be treated as indicative. The soil moisture thresholds were fixed values rather than calibrated for specific plant

species or soil types, and the system depends on Wi-Fi for remote control and on a battery pack for power. Finally, no user study was conducted, so claims about ease of use for non-expert gardeners remain untested.

CONCLUSION

This paper presented an IoT-based smart irrigation system for home gardening that combines an ESP32 controller, capacitive soil moisture sensors, a rain sensor, a flow sensor, relay-switched submersible pumps, and a Blynk mobile interface. The system monitored conditions in real time, watered each zone only when the soil was dry and no rain was detected, and allowed remote monitoring and manual override. In bench testing, the sensors and pumps behaved as intended; the system withheld water in four of ten logged decisions, and plants in both zones grew throughout a seven-day observation.

Future work should test the system for longer periods with control groups to quantify water savings; calibrate the flow sensor and soil thresholds; replace multiple pumps with solenoid valves for multi-zone control; add push notifications and weather-forecast-based scheduling; supply the system with solar panels and rechargeable batteries; and apply sensor filtering and regular calibration to improve accuracy. User studies with home gardeners would also show whether such systems help households adopt more water-efficient routines.

ACKNOWLEDGMENT

The authors thank Universiti Teknikal Malaysia Melaka and the Faculty of Electrical Engineering and Technology (FTKE) for providing the facilities, resources, and learning environment for this work, and the project course coordinators for their guidance.

REFERENCES

1. N. M. Abdikadir, A. A. Hassan, H. O. Abdullahi, and R. A. Rashid, "Smart Irrigation System," International Journal of Electrical and Electronics Engineering , vol. 10, no. 8, pp. 224–234, Sep. 2023, doi: 10.14445/23488379/ijeee-v10i8p122 .
2. H. Muntz and K. Kopp, "Influencing Water Consumption Through the Water Check Program," Journal of Extension , vol. 57, no. 4, Aug. 2019, doi: 10.34068/joe.57.04.09 .
3. Shahapure, R. Siddha, T. Tewary, A. Shahapure, S. Mantri, and A. A. Kamble, "Smart Garden Watering and Lighting System Integrated with Machine Learning Model to Maintain Urban Gardens," pp. 664–670, Feb. 2025, doi: 10.1109/icccit62592.2025.10927752 .
4. T. Ji et al. , "Proper irrigation amount for eggplant cultivation in a solar greenhouse improved plant growth, fruit quality and yield by influencing the soil microbial community and rhizosphere environment," Frontiers in Microbiology , vol. 13, p. 981288, Sep. 2022, doi: 10.3389/fmicb.2022.981288 .
5. S. Yadav, P. Kumar, A. K. Mishra, M. Tiwari, U. Kumar, and G. Singh, "Smart Agriculture Irrigation System using IoT," Computer Science & Engineering An International Journal , vol. 15, no. 1, pp. 73–81, Mar. 2025, doi: 10.5121/cseij.2025.15109 .
6. S. Mansoor, S. Iqbal, S. M. Popescu, S. L. Kim, Y. S. Chung, and J. Baek, "Integration of smart sensors and IOT in precision agriculture: trends, challenges and future prospectives," Frontiers in Plant Science , vol. 16, p. 1587869, May 2025, doi: 10.3389/fpls.2025.1587869 .
7. M. B. Kiran and M. Wynn, "The Internet of Things in the Corporate Environment," in Advances in e-business research series , IGI Global, 2021, pp. 132–148. doi: 10.4018/978-1-7998-7712-7.ch008 .
8. H. Mestouri, K. Baraka, J. Ezzahar, and M. Madiafi, "Design and implementation of an intelligent system for smart garden watering using fuzzy logic," Frontiers in Artificial Intelligence , vol. 9, p. 1789499, Jun. 2026, doi: 10.3389/frai.2026.1789499 .
9. R. Jain, "Experimental performance of smart IoT-enabled drip irrigation system using and controlled through web-based applications," Smart Agricultural Technology , vol. 4, p. 100215, Mar. 2023, doi: 10.1016/j.atech.2023.100215 .

10. G. B. HMIDA, "Enhancing Home Automation: Developing an IoT-Driven Smart Irrigation System," International Research Journal of Innovations in Engineering and Technology , vol. 10, no. 2, p. 110, Jan. 2026, doi: 10.47001/irjiet/2026.102019 .
11. K. L. Nisha, V. Pradeep, P. P. K. Nair, S. Pillai, M. Arunachalam, and R. T. S. Babu, "Optimising battery life: a TinyML approach to lithium-ion battery health monitoring," IAES International Journal of Artificial Intelligence , vol. 14, no. 5, p. 3858, Oct. 2025, doi: 10.11591/ijai.v14.i5.pp3858-3868 .
12. M. Ragab, "IOT based Smart Irrigation System," International Journal of Industry and Sustainable Development , vol. 0, no. 0, p. 0, Aug. 2022, doi: 10.21608/ijisd.2022.148007.1021 .
13. A. A. Abdelmoneim, C. M. A. Kalaany, G. Dragonetti, B. Derardja, and R. Khadra, "Comparative Analysis of Soil Moisture- and Weather-Based Irrigation Scheduling for Drip-Irrigated Lettuce Using Low-Cost Internet of Things Capacitive Sensors," Sensors , vol. 25, no. 5, p. 1568, Mar. 2025, doi: 10.3390/s25051568 .
14. Mahmood A. Al-Shareeda, L. B. Najm, H. Ali, S. A. Mushtaq, and H. Ali, "Secure IoT-Based Smart Agriculture System Using Wireless Sensor Networks for Remote Environmental Monitoring," STAP Journal of Security Risk Management , vol. 2024, no. 1, pp. 56–66, Jan. 2024, doi: 10.63180/jsrm.thestap.2024.1.4 .
15. M. E. H. Chowdhury et al. , "Design, Construction and Testing of IoT Based Automated Indoor Vertical Hydroponics Farming Test-Bed in Qatar," Sensors , vol. 20, no. 19, p. 5637, Oct. 2020, doi: 10.3390/s20195637 .
16. Texas Instruments, "LM2596 SIMPLE SWITCHER power converter 150-kHz 3-A step-down voltage regulator," Texas Instruments, 2023. [Online]. Available: www.ti.com
17. R. Babis et al. , “DEVELOPMENT OF CLOUD-BASED SMART PLANT WATERING SYSTEM,” IJIREEICE , vol. 13, no. 6, May 2025, doi: 10.17148/ijireeice.2025.13601 .
18. Joice et al. , "Applications of Raspberry Pi for Precision Agriculture—A Systematic Review," Agriculture , vol. 15, no. 3, p. 227, Jan. 2025, doi: 10.3390/agriculture15030227 .