

Precision Agriculture for Cucumber Cultivation: An Internet of Things Approach

Mustafa Manap^{1*}, Mohd Hatta Jopri¹, Shamsul Fakhar Abd Gani², Mohammad Syafiq Yusaini³

¹Fakulti Teknologi dan Kejuruteraan Elektrik, Universiti Teknikal Malaysia Melaka (UTeM), Hang Tuah Jaya, 76100 Durian Tunggal, Melaka, Malaysia

²Fakulti Teknologi dan Kejuruteraan Elektronik dan Komputer, Universiti Teknikal Malaysia Melaka (UTeM), Hang Tuah Jaya, 76100 Durian Tunggal, Melaka, Malaysia

³Dagang Net Technologies Sdn. Bhd., Cyberjaya, 63000 Selangor, Malaysia

DOI: <https://doi.org/10.47772/IJRISS.2026.100900382>

Received: 22 September 2026; Accepted: 27 September 2026; Published: 12 October 2026

ABSTRACT

Cucumber growers who irrigate by a fixed daily schedule routinely water a plant that is already wet, because the schedule cannot see the soil it is watering. This paper reports the design, hardware realisation, and field testing of an Internet of Things (IoT) control and monitoring system that replaces the schedule with a threshold triggered by an actual soil-moisture reading. A NodeMCU ESP32 microcontroller reads a capacitive soil-moisture probe, a DHT11 temperature and humidity sensor, and a light-dependent resistor (LDR) module, and switches a 12 V DC water pump through a relay whenever soil moisture falls below a 35% threshold; every reading is pushed over WiFi to a MySQL database and displayed on a role-based (administrator and gardener) web dashboard built in PHP and Morris.js.

The control circuit was first modelled and verified in Fritzing before any hardware was built, the driver board was laid out and etched as a custom printed circuit board (PCB) in-house, and the completed unit was fitted to a potted cucumber plant for field trial. Over a two-day comparative trial, threshold-triggered automatic irrigation opened the water pump 3 times against 12 times for a fixed 4-times-daily schedule, while holding soil moisture within the same 35-60% band, a 75% reduction in pump activations for an equivalent moisture outcome.

The prototype also survived an unplanned stress test: after heavy rain and a fungal leaf infection, the automatically irrigated plant recovered following a mancozeb fungicide application, continuing growth to the end of the 12-day observation window. The results support threshold-based, sensor-driven irrigation as a practical route to water saving in small-scale and home-garden cucumber cultivation, and the paper closes with a quantitative evaluation protocol, covering longer trial duration, larger plant samples, WiFi reliability, and pest and disease pressure, that should precede any larger deployment.

Keywords: Internet of Things, Precision Agriculture, Soil Moisture Sensor, Automatic Irrigation, Cucumber Cultivation

INTRODUCTION

Background

Agriculture remains one of the largest contributors to employment and income in Malaysia even as the services sector has come to dominate national GDP, and the government's Smart Farm Revolution agenda treats the integration of modern sensing and communication technology into farming as the route back to a globally competitive agricultural sector (Mat et al., 2019).

Within that agenda, irrigation is one of the more tractable problems to automate: unlike yield or pest pressure, soil moisture is a single, continuously measurable quantity, and a wrong irrigation decision shows up quickly, either as a wilted plant or as wasted water.

Most smallholder and home-garden irrigation in Malaysia is still managed by a fixed schedule or by a farmer's own judgement of how the soil feels, because the sensing and control hardware needed to do better has historically been priced for commercial farms rather than a single polybag on a balcony. A fixed schedule waters at the same times regardless of whether the plant needs it, which produces the two failure modes documented across the smart-irrigation literature: over-watering when rain or humidity has already kept the soil wet, and under-watering when a hot, dry stretch outpaces the schedule (Khanna & Kaur, 2019; Ali et al., 2025). Cucumber is a useful crop on which to study this problem because it is shallow-rooted and comparatively sensitive to both drought stress and waterlogging, so its soil moisture needs to be kept inside a fairly narrow band rather than simply "wet" or "dry".

The falling cost of WiFi-capable microcontrollers, low-cost capacitive soil-moisture probes, and cloud-hosted databases has made sensor-driven, threshold-based irrigation feasible at exactly this small scale (Soussi et al., 2024; Hercog et al., 2023). A capacitive probe reads the dielectric permittivity of the soil around it, which tracks moisture content without the corrosion problems of a resistive probe, and a microcontroller can compare that reading against a set threshold and switch a pump automatically, with no scheduling logic and no farmer judgement call required. What is missing from most classroom-scale demonstrations of this idea, however, is a side-by-side comparison against the schedule it is meant to replace, run on a real plant rather than a benchtop model.

Related Work

The Internet of Things (IoT) literature on precision agriculture has grown quickly since 2020, and recent reviews converge on the same architecture used in this work: a low-power sensor node reads environmental parameters, a microcontroller applies a control rule, and a wireless link carries the data to a dashboard or database for the farmer to monitor remotely (Atalla et al., 2023; Senoo et al., 2024; Mansoor et al., 2025). Singh and Sharma (2024) reviewed more than a hundred IoT-based precision-agriculture systems published between 2018 and 2024 and found soil-moisture-triggered irrigation to be the single most common control strategy, precisely because it needs only one reliable sensor to produce a usable decision rule.

Several recent studies have quantified the water saving that threshold-based control delivers over a fixed schedule. Behzadipour et al. (2024) compared intelligent and conventional irrigation scheduling in a greenhouse cucumber trial and reported a substantial reduction in applied water with no loss of yield under sensor-triggered control, a result consistent with the pump-activation reduction reported in this paper.

Yascaribay et al. (2022) compared WiFi against LoRa and other low-power wide-area options for agricultural telemetry and found WiFi adequate for the short-range, mains-powered case addressed here, while noting that WiFi's higher power draw becomes a limitation once a deployment moves to battery-powered, multi-node fields, a limitation this paper's system shares. Morchid et al. (2025) demonstrated an embedded, server-sent-events based irrigation controller that removed the need for the client to poll the server for updates, an architectural refinement that the PHP-and-database dashboard used here does not yet implement.

Closer to the hardware side, Hercog et al. (2023) documented the design trade-offs of building ESP32-based IoT nodes for exactly this class of low-cost sensing task, and their guidance on power budgeting and sensor debouncing is consistent with the design decisions taken for the NodeMCU ESP32 node reported below. On the pest and disease side, which this paper's field trial encountered directly and unexpectedly, Nayagam et al. (2023) and Nyakuri et al. (2025) both report IoT-linked approaches to earlier pest and disease detection, an extension this paper's discussion returns to when interpreting the fungal infection observed during the field trial.

Problem Statement and Objectives

Two problems motivate this work. First, manual and fixed-schedule irrigation in small-scale cucumber cultivation wastes water and gives the plant an uneven moisture supply, because the schedule cannot see the soil it is watering. Second, threshold-based alternatives are reported widely in the literature but are comparatively rarely demonstrated end to end, from a simulated circuit through a custom-etched PCB to a field trial on a living plant with a quantified comparison against the schedule it replaces.

This paper addresses both problems with the following objectives: (i) to design and build an IoT-based control and monitoring system for a cucumber plant that measures soil moisture, temperature, humidity, and light intensity and switches irrigation automatically on a moisture threshold, and (ii) to evaluate the effectiveness of that threshold-based control against fixed-schedule irrigation for small-scale agriculture, using pump-activation count and soil-moisture stability as the comparison metrics.

Scope

The system described here is scoped to a single cucumber plant grown in a polybag, monitored by one soil-moisture probe, one DHT11 temperature and humidity sensor, and one LDR light sensor, with irrigation delivered by a single 12 V DC pump through a drip line. Sensor data is transmitted over WiFi to a MySQL database and displayed on a PHP-based web dashboard with separate administrator and gardener roles. The system does not address multi-node field deployment, fertigation, or automated pest control; these are identified as future work in the concluding section.

METHODOLOGY

Project Overview

Development followed a waterfall sequence of four stages: (i) requirement definition, drawing on a site visit to a cucumber farm at Kampung Tumbuk Darat, Tanjung Sepat, Selangor, to establish the functional and non-functional requirements listed below; (ii) system design and simulation, in which the sensor-to-actuator circuit was modelled in Fritzing before any physical wiring; (iii) system implementation, covering hardware selection, PCB design and etching, and the software stack; and (iv) system integration and testing, covering enclosure assembly, the web dashboard, and the field trial reported in the Results section.

The functional requirements were: the system must supply water from a dedicated reservoir through a low-flow drip line suited to cucumber's preference for well-drained, evenly moist soil; the water pump must be sized for approximately 1 litre per minute; and the system must expose live temperature, humidity, light-intensity, and soil-moisture status together with pump state. The non-functional requirements were: user access must be gated by login authentication with separate administrator and gardener roles; sensor status must be visible to the gardener through the web dashboard; and the system must refresh its readings at a 5-second interval to keep the displayed state close to real time.

System Design and Simulation

Figure 1 shows the overall system architecture. The soil-moisture probe, the DHT11 sensor, and the LDR module all feed into the NodeMCU ESP32, which applies the moisture threshold, drives the relay that switches the water pump, and pushes every reading to the cloud database over WiFi; the database in turn serves the PC and smartphone dashboard views.

The soil-moisture threshold was fixed at 35%, the level an agronomist consulted during development judged appropriate for cucumber; irrigation is triggered whenever the probe reads below that value and stops once the reading recovers above it.

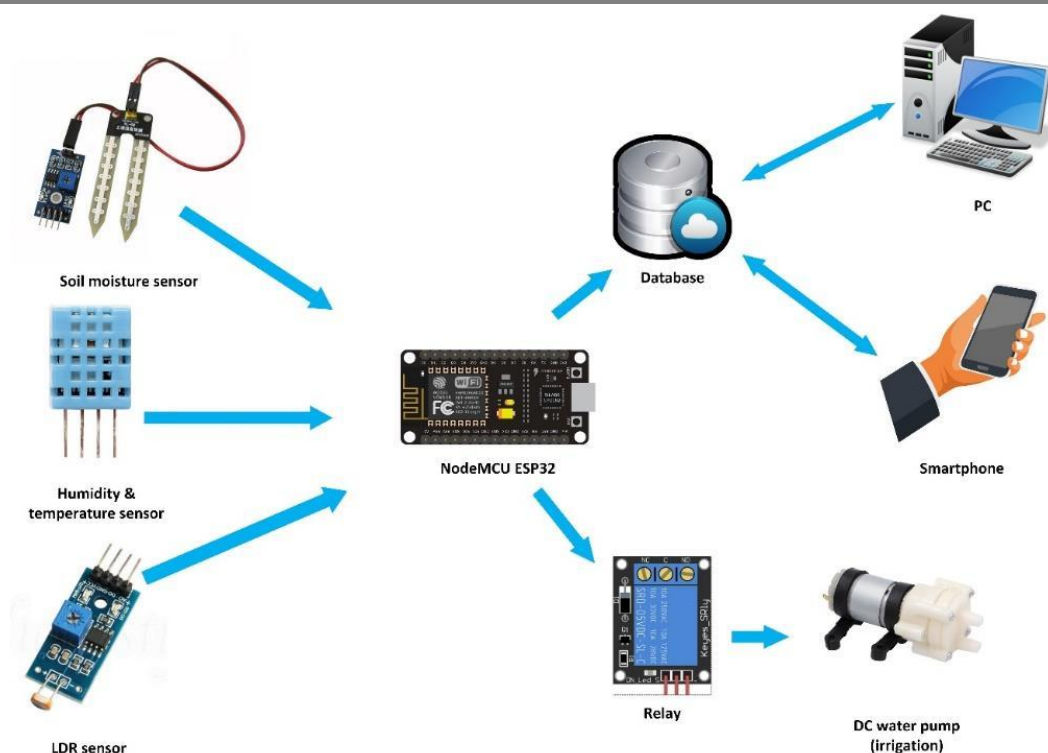


Figure 1: Overall System Architecture, Showing Input Sensors, the NodeMCU ESP32 Processing Unit, and the Database, Actuator, and Client Outputs

Before any hardware was built, the input-to-output chain was modelled in Fritzing, shown in Figure 2. The NodeMCU ESP32 is wired to the DHT11 sensor, the capacitive soil-moisture probe, the LDR module, a relay module driving the water pump, and an OLED display used for on-device status. Running the sensor inputs through their expected range in the simulated circuit confirmed the wiring and the relay-triggering logic before any component was soldered, which is the same verification role that Proteus and similar schematic-capture tools serve in comparable ESP32-based IoT builds (Hercog et al., 2023).

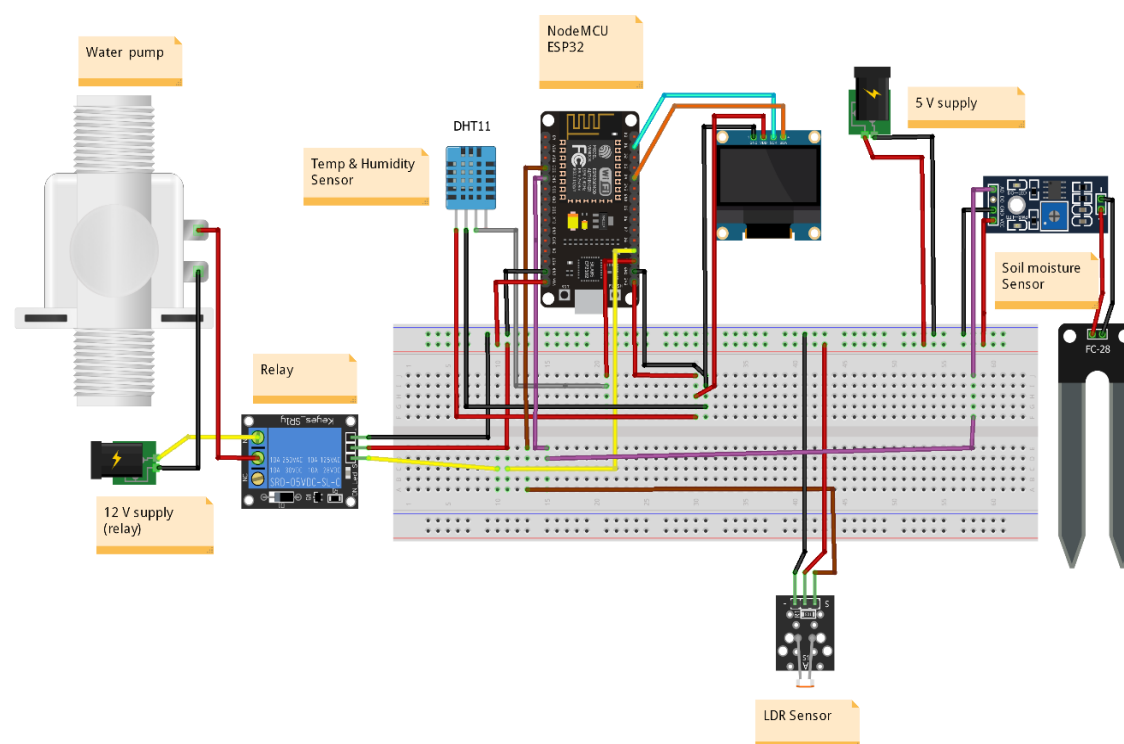


Figure 2: Simulated Sensor-to-Actuator Circuit (Fritzing), Verifying Wiring and Relay-Triggering Logic Before Hardware Assembly

Hardware Selection

Table 1 summarises the sensing and actuation hardware selected for the prototype.

Table 1: Hardware Components and Their Function

Component	Function
DHT11 temperature and humidity sensor	Digital-output capacitive/thermistor sensor for ambient temperature and humidity around the plant
Capacitive soil-moisture sensor (SEN0193)	Measures volumetric soil-water content by capacitive rather than resistive sensing, avoiding electrode corrosion
LDR sensor module	Detects ambient light intensity to distinguish day from night operation
12 V DC 385 water pump	Centrifugal drip-irrigation pump rated at approximately 3 L/min
2-channel 5 V relay module	Switches the 12 V pump circuit under 5 V logic control from the microcontroller
NodeMCU ESP32	3.3 V, WiFi-integrated microcontroller that reads all sensors, applies the control rule, and reports to the database

A NodeMCU ESP32 was chosen over a plain Arduino board specifically for its integrated WiFi radio, which removes the need for a separate communication module and keeps the sensor node to a single board.

System Implementation: PCB Design and Fabrication

The sensor and relay connections were consolidated onto a custom PCB rather than left on a breadboard, to give the finished unit the mechanical robustness needed for outdoor use. Figure 3 shows the PCB layout, routed with header positions for the relay input and output, the DHT11, LDR, and soil-moisture connections, the drive motor terminal, and the 12 V and 5 V supply rails. Figure 4 shows the resulting board after etching on a positive-board substrate in the faculty's PCB laboratory, with the copper traces and mounting holes visible. Etching the layout in-house, rather than ordering a commercial board, kept the prototype cost low and is consistent with the low-cost, DIY-hardware approach that recent IoT-agriculture work (Hercog et al., 2023; Muthuramalingam et al., 2024) treats as one of the field's practical advantages over legacy commercial farm automation.

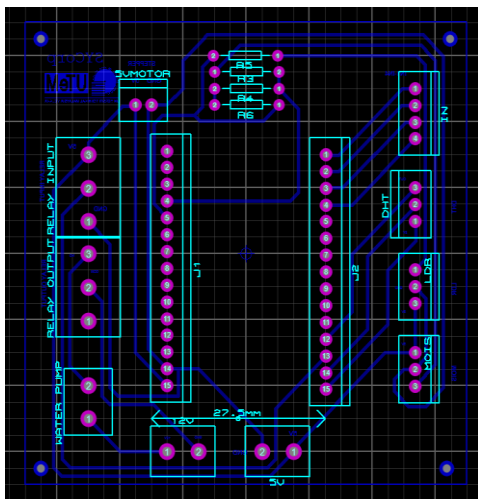


Figure 3: PCB Layout Showing Relay, Sensor, and Power Header Positions

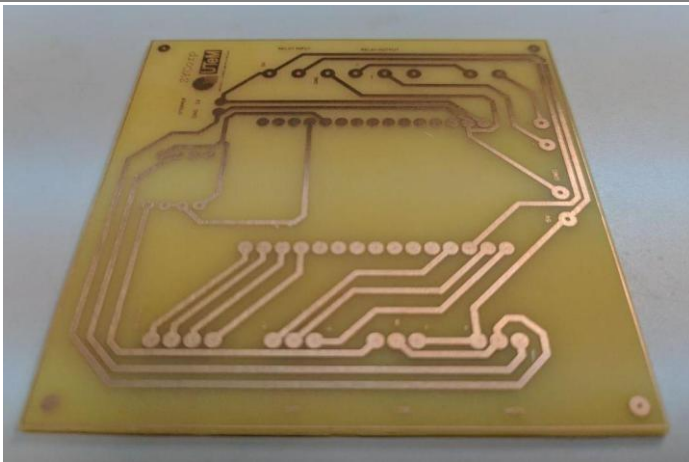


Figure 4: Custom PCB After In-House Etching on a Positive-Board Substrate

Once the board was etched, each connection was re-verified against the Fritzing simulation before the LED status indicators, the switch, and the OLED display were added, and the assembly was housed in a PVC junction box to protect the circuit from moisture and dust during outdoor operation.

Software Implementation

The microcontroller firmware, written in C/C++ under the Arduino IDE, polls all three sensors on a fixed interval, evaluates the moisture threshold, drives the relay accordingly, and sends every reading to the server as an HTTP GET request; a serial-monitor response of 200 OK confirms that the database write succeeded. On the server side, a MySQL database, administered through phpMyAdmin, stores four tables (temperature and humidity, light intensity, soil moisture, and user accounts), and a PHP-based web front end queries those tables to render the dashboard. Two account roles are supported: an administrator role that can manage user and admin accounts and review all sensor history, and a gardener role limited to viewing the dashboard, current sensor status, and pump state, and editing their own profile. Live-updating line charts of temperature, light intensity, and soil moisture are rendered client-side with Morris.js, refreshed on the 5-second interval set out in the non-functional requirements.

Prototype Integration

Figure 5 shows the completed prototype fitted to a potted cucumber plant for the field trial: the control box housing the PCB, OLED display, and status LEDs; the 12 V water pump drawing from a reservoir container; the soil-moisture probe inserted at the plant's root zone; and the combined LDR and DHT11 sensor head mounted above the canopy. This is the physical configuration used for every result reported in the next section.

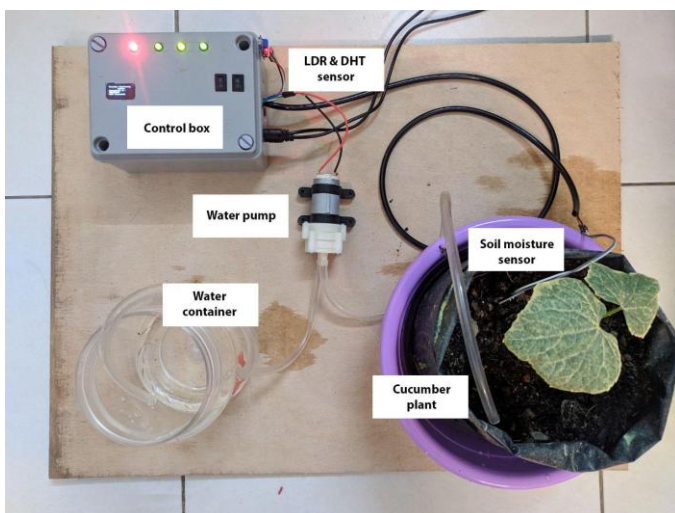


Figure 5: Completed Prototype Fitted to a Potted Cucumber Plant for Field Trial

RESULTS AND DISCUSSION

Web Dashboard Verification

Figure 6 shows the role-based login screen that gates access to the system: a blank or incorrect email and password is rejected with an on-screen alert, a first-time gardener can self-register through the linked sign-up form, and an administrator reaching this same screen can subsequently add, edit, and delete both admin and gardener accounts, with each change reflected immediately in the underlying database.



Figure 6: Role-Based Login Screen for the Web Dashboard

Once authenticated, the gardener is taken to the dashboard shown in Figure 7. The four status tiles across the top give the current temperature, light intensity, soil-moisture percentage, and pump state at a glance, refreshed on the 5-second interval set out in the non-functional requirements, and the live temperature trend beneath them is rendered with Morris.js directly from the same sensor log later plotted in Figure 8. Throughout testing, all four tiles and their charts updated correctly at the configured interval, confirming that the WiFi-to-database-to-dashboard path worked end to end and that a gardener with no access to the physical prototype could still monitor its state remotely. This path currently relies on the browser polling the server every 5 seconds; Discussion returns to this design choice below.

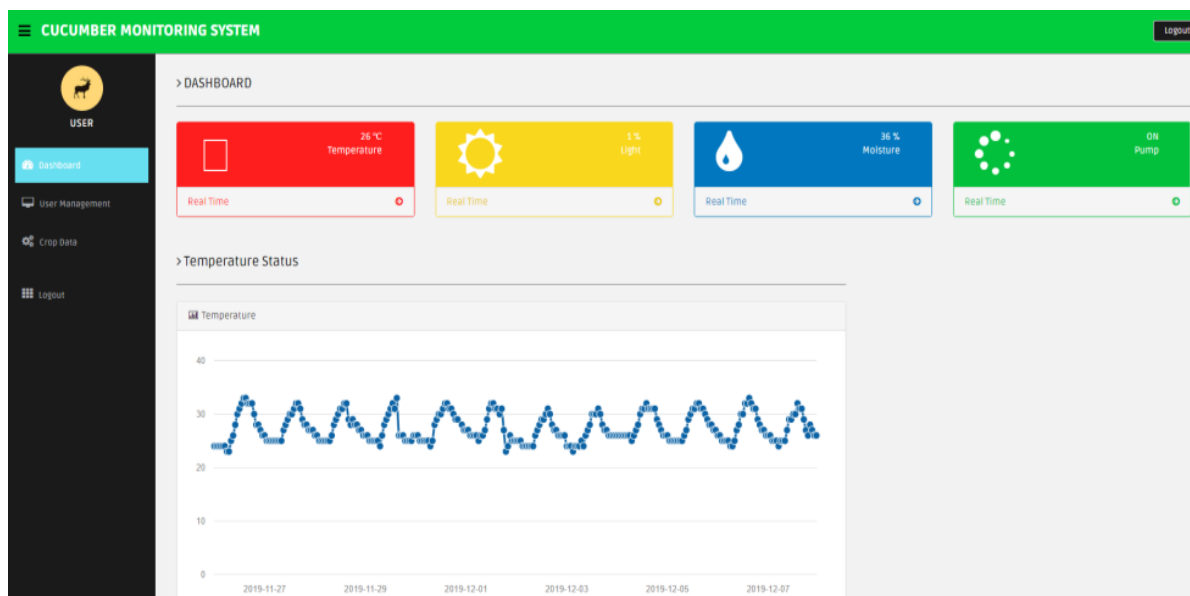


Figure 7: Gardener Dashboard View, Showing Real-Time Status Tiles and the Live Temperature Trend Chart

Irrigation Comparison: Schedule versus Threshold-Based Control

The central test compared two irrigation strategies on two cucumber plants grown under otherwise identical conditions in polybags, each fed by a single drip line: a fixed schedule that opened the pump four times a day, at 08:00, 10:00, 16:00, and 18:00, for five minutes each time, and threshold-based automatic irrigation that opened the pump whenever the soil-moisture probe read below 35% and closed it once the reading recovered above that value. Soil moisture was logged at one-hour intervals for both plants over the same two-day window, 26 to 28 November.

Figure 8 shows the ambient temperature recorded over the trial period, confirming the expected daytime peak and overnight trough that governs how quickly the soil dries between irrigation events; soil moisture fell fastest during the hottest part of each day and recovered only through irrigation.

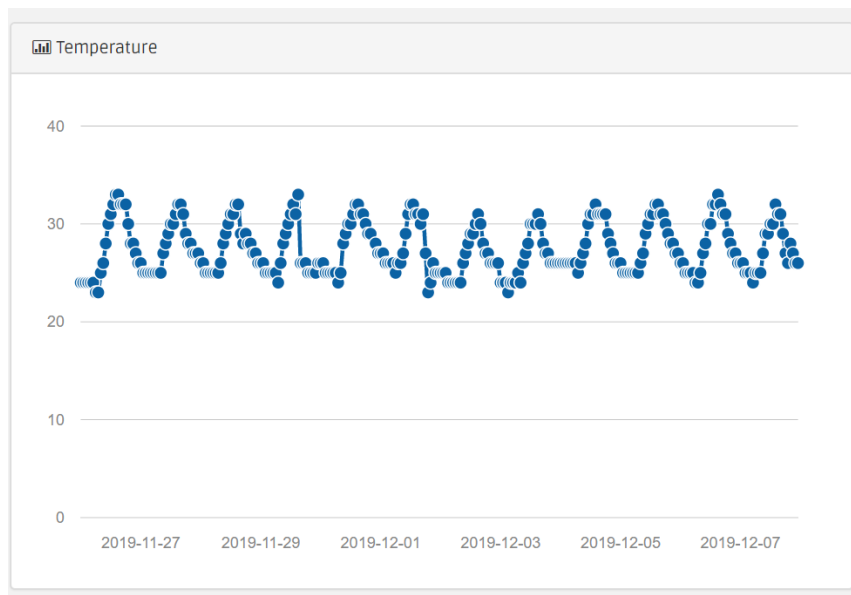


Figure 8: Ambient Temperature Recorded During the Two-Day Trial Period (26-28 November)

Figure 9 plots the resulting soil-moisture traces for both strategies side by side. Under the fixed schedule (left panel), moisture rose and fell across the full 35-60% range on every one of the twelve scheduled irrigation events across the two days, including at least one event, the first watering at 08:00 on 26 November, that occurred while the reading was still comfortably above the 35% threshold, an unambiguous case of water applied to soil that did not yet need it. Under threshold-based control (right panel), the pump activated only when the reading reached 35%, which happened three times across the same two-day window, and each activation returned the reading to the same 50-58% recovery band achieved by the scheduled waterings.

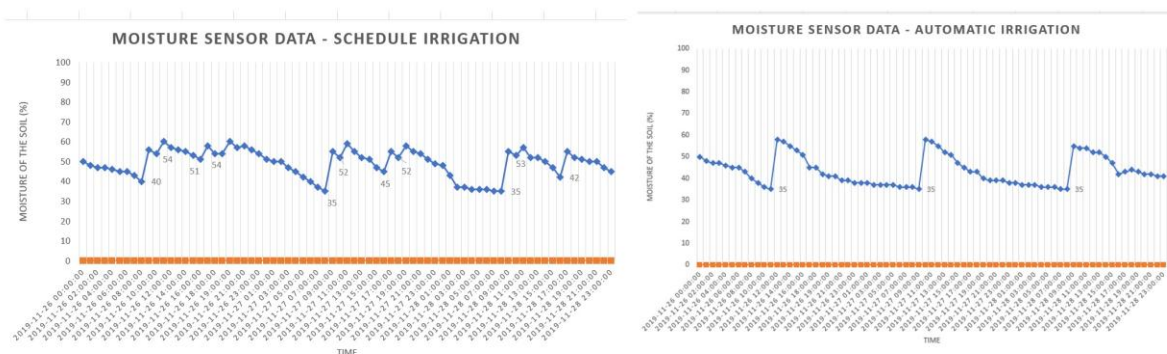


Figure 9: Soil-Moisture Readings Over Two Days Under Fixed-Schedule Irrigation (Left) and Threshold-Based Automatic Irrigation (Right)

Comparing the two panels, threshold-based control held soil moisture within essentially the same working range as the fixed schedule, 35% to roughly 58-61%, while triggering the pump only 3 times against 12 times for the schedule, a 75% reduction in irrigation events for an equivalent moisture outcome. This comparison is stated deliberately in activations rather than litres: the scheduled pump ran for a fixed five minutes per activation, but the automatic pump's run time per activation, the time needed for the reading to climb back above 35%, was not logged separately in this trial, so a precise water-volume figure cannot be derived from the data collected here. Activation count is therefore reported as the available proxy for water applied, not as a substitute for a direct volume measurement, and Table 2 adds inline flow metering as the fix for this gap in the next trial. With that caveat, the reduction in activation count still demonstrates the core claim of this paper: a soil-moisture threshold reproduces the moisture outcome of a conservative fixed schedule while calling for irrigation substantially less often. This is consistent with the substantial water-use reductions reported for sensor-triggered irrigation in a directly comparable greenhouse cucumber trial that did measure volume directly (Behzadipour et al., 2024), and with the wider pattern documented across smart-irrigation reviews that a measured trigger outperforms a fixed timetable whenever weather or soil conditions vary from day to day (Ali et al., 2025; Mansoor et al., 2025).

Field Trial on a Living Plant

Figure 10 shows the two cucumber plants after twelve days of monitored growth, the automatically irrigated plant on the left and the schedule-irrigated plant on the right. Growth under automatic irrigation was visibly slower over this window than under the fixed schedule, an outcome that runs counter to the water-saving result above and is explained by an event outside the irrigation logic rather than by the control strategy itself: following a period of heavy rain, the automatically irrigated plant developed yellow leaf spotting consistent with a fungal infection, and growth stalled until a mancozeb fungicide application brought the plant back to active growth, albeit at a reduced rate for the remainder of the observation window.



Figure 10: Cucumber Plants After Twelve Days of Monitored Growth Under Automatic Irrigation (Left) and Scheduled Irrigation (Right)

Both plants stood on the same plot and were exposed to the same rain event, so the two irrigation strategies did not differ in the amount of rainfall each plant received; only the automatically irrigated plant went on to show fungal symptoms. That pattern is consistent with, but does not establish, a causal link between moisture-optimised watering and increased fungal risk: light bushy growth and locally humid conditions around the leaf canopy, the same conditions this watering regime can inadvertently sustain by keeping the root zone consistently moist, are known to favour the insect and fungal pressure observed here (Nayagam et al., 2023), but with one plant per strategy the result is equally consistent with ordinary biological variation between two individual plants. This trial cannot distinguish between those two explanations, and Table 2 lists pest and

disease incidence across a replicated sample as the test that would. A production deployment of this system would in any case benefit from pairing the existing moisture threshold with the kind of IoT-linked early pest and disease detection reported by Nyakuri et al. (2025) and Nayagam et al. (2023), so that a fungal outbreak of this kind is flagged before it visibly damages the canopy, whatever its underlying cause.

LIMITATIONS

The system's main operational limitation observed during testing was WiFi robustness: when the NodeMCU ESP32 lost its WiFi connection it did not reliably reacquire it without a manual power cycle, which would interrupt both data logging and irrigation control in an unattended deployment. This is a known limitation of consumer WiFi radios relative to purpose-built low-power wide-area protocols for agricultural telemetry (Yascaribay et al., 2022). The tested firmware did not include automatic recovery from this state; the straightforward fix is a `WiFi.onEvent()` disconnect callback that calls `WiFi.reconnect()` on a short retry interval, backed by a hardware watchdog (`esp_task_wdt`) that forces a full reset if no sensor reading has been logged within a set window, and reconnection time under that fix is the first metric listed in the evaluation protocol proposed below.

A second limitation is architectural rather than operational: the dashboard in Figures 6 and 7 updates by having the browser poll the PHP server every 5 seconds, which is simple to implement but wastes requests when nothing has changed and caps how quickly a reading can reach the screen. Morchid et al. (2025) report an embedded controller built around Server-Sent Events that pushes a new reading to the client only when one exists; adopting the same push-based approach, or a WebSocket connection, in place of polling is a straightforward upgrade path for a future version of this dashboard and is noted as such rather than implemented here.

A third limitation is the evidence base itself: every result in this paper, the activation-count comparison in Figure 9 and the growth outcome in Figure 10 alike, rests on a single plant per irrigation strategy, observed for two days of logged soil moisture and twelve days of visible growth, at one site. That is enough to demonstrate that the control mechanism works and to illustrate the kind of outcome it produces, but it is not a statistically powered comparison, and none of the percentages reported above should be read as a generalisable water-saving or yield figure beyond the case reported here. Table 2 sets out the larger-sample, longer-duration trial needed before that generalisation can be made.

Proposed Evaluation Protocol

Because the trial reported above was a functional and comparative demonstration rather than a statistically powered agronomic trial, Table 2 sets out the additional evaluation that should precede a larger deployment.

Table 2: Proposed Quantitative Evaluation Protocol

Test	Purpose	Suggested Metric
Extended-duration trial	Confirm the water-saving result over a full cropping cycle rather than two days	Total litres applied per plant per week
Multi-plant replication	Support a statistical comparison of yield and water use between strategies	Yield and water use, mean and standard deviation across $n \geq 6$ plants per group
WiFi reconnection behaviour	Quantify how long the node stays offline after a dropped connection, and whether it recovers without manual intervention	Time to reconnection; irrigation events missed while offline
Pest and disease incidence	Establish whether moisture-optimised irrigation changes pest or fungal	Incidence rate per plant per trial

	pressure relative to schedule-based watering	
Threshold sensitivity	Identify whether a threshold other than 35% further reduces water use without stressing the plant	Water applied and visible plant condition across a range of threshold values

CONCLUSION

This paper presented the design, hardware realisation, and field testing of an IoT-based control and monitoring system that replaces fixed-schedule cucumber irrigation with a soil-moisture threshold. A NodeMCU ESP32 node, verified first in Fritzing simulation and then built around a custom-etched PCB, read soil moisture, temperature, humidity, and light intensity, switched a 12 V water pump through a relay, and reported every reading to a MySQL database displayed on a role-based PHP web dashboard. Fitted to a potted cucumber plant for a two-day comparative field trial, threshold-based control held soil moisture within the same working range as a conservative fixed schedule while triggering the pump only 3 times against 12 times for the schedule, a 75% reduction in irrigation events for an equivalent moisture outcome, supporting sensor-driven irrigation as a practical, low-cost water-saving measure for small-scale and home-garden cucumber cultivation. An unplanned fungal infection during the trial, resolved with a fungicide application, also underlined that moisture control alone does not address every threat to plant health, and that pest and disease sensing is a natural next addition to a system like this one. These findings should be read within the trial's actual scope: one plant per irrigation strategy, two days of logged soil moisture, twelve days of observed growth, and a single site, which is sufficient to demonstrate the control mechanism but not to support a statistical water-saving or yield claim, and the reported activation-count reduction is a proxy for water applied rather than a directly measured volume. Future work should carry out the extended, multi-plant, and WiFi-reliability testing set out in Table 2, with water applied measured directly by inline flow metering rather than inferred from activation count; should examine pairing the moisture threshold with IoT-linked pest and disease detection, along the lines reported by Nyakuri et al. (2025) and Nayagam et al. (2023), so that the system responds to more than moisture alone; and should consider replacing the current polling-based dashboard with a push-based architecture along the lines demonstrated by Morchid et al. (2025).

Ethical Considerations

This work involved the design and field testing of an electromechanical irrigation prototype on cultivated cucumber plants and did not involve human participants or animal subjects; no ethical approval was therefore required.

Conflict of Interest: The authors declare no conflict of interest.

Data Availability

The sensor logs and design files supporting this study are available from the corresponding author upon reasonable request.

ACKNOWLEDGMENT

The authors thank Universiti Teknikal Malaysia Melaka (UTeM) for their support of this work.

REFERENCES

1. Ali, A., Hussain, T., & Zahid, A. (2025). Smart irrigation technologies and prospects for enhancing water use efficiency for sustainable agriculture. *AgriEngineering*, 7(4), 106. <https://doi.org/10.3390/agriengineering7040106>

2. Anoop, E. G., & Bala, G. J. (2024). IoT and ML-based automatic irrigation system for smart agriculture system. *Agronomy Journal*, 116(3). <https://doi.org/10.1002/agj2.21344>
3. Atalla, S., Tarapiah, S., Gawanmeh, A., Daradkeh, M., Mukhtar, H., Himeur, Y., Mansoor, W., Hashim, K. F. B., & Daadoo, M. (2023). IoT-enabled precision agriculture: Developing an ecosystem for optimized crop management. *Information*, 14(4), 205. <https://doi.org/10.3390/info14040205>
4. Behzadipour, F., Ghasemi-Nejad-Raeini, M., Abdanan Mehdizadeh, S., Taki, M., Khalili Moghadam, B., & Zare Bavani, M. R. (2024). Optimizing water use efficiency in greenhouse cucumber cultivation: A comparative study of intelligent irrigation systems. *PLOS ONE*, 19(10), e0311699. <https://doi.org/10.1371/journal.pone.0311699>
5. Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, 29(7), 1645-1660. <https://doi.org/10.1016/j.future.2013.01.010>
6. Halim, S., Utama, D. N., & Wicaksono, P. (2026). IoT-based smart irrigation system using DHT-11 and soil moisture sensors with real-time monitoring and automated water pump control. In *Proceedings of the 2026 14th International Conference on Information and Communication Technology (ICoICT)*. <https://doi.org/10.1109/icoict71118.2026.11664831>
7. Hercog, D., Lerher, T., Truntić, M., & Težak, O. (2023). Design and implementation of ESP32-based IoT devices. *Sensors*, 23(15), 6739. <https://doi.org/10.3390/s23156739>
8. Holzman, M. E., Rivas, R., & Piccolo, M. C. (2014). Estimating soil moisture and the relationship with crop yield using surface temperature and vegetation index. *International Journal of Applied Earth Observation and Geoinformation*, 28(1), 181-192. <https://doi.org/10.1016/j.jag.2013.12.006>
9. Khanna, A., & Kaur, S. (2019). Evolution of Internet of Things (IoT) and its significant impact in the field of Precision Agriculture. *Computers and Electronics in Agriculture*, 157, 218-231. <https://doi.org/10.1016/j.compag.2018.12.039>
10. Mansoor, S., Iqbal, S., Popescu, S. M., Kim, S. L., Chung, Y. S., & Baek, J.-H. (2025). Integration of smart sensors and IoT in precision agriculture: Trends, challenges and future perspectives. *Frontiers in Plant Science*, 16, 1587869. <https://doi.org/10.3389/fpls.2025.1587869>
11. Mat, I., Mohd Kassim, M. R., Harun, A. N., & Yusoff, I. M. (2019). Smart agriculture using internet of things. In *2018 IEEE Conference on Open Systems (ICOS 2018)* (pp. 54-59). IEEE.
12. Morchid, A., Et-taibi, B., Oughannou, Z., El Alami, R., Qjidaa, H., Ouazzani Jamil, M., Boufounas, E.-M., & Abid, M. R. (2025). IoT-enabled smart agriculture for improving water management: A smart irrigation control using embedded systems and Server-Sent Events. *Scientific African*, 27, e02527. <https://doi.org/10.1016/j.sciaf.2024.e02527>
13. Muthuramalingam, R., Velu, R. R., Baskar, H., & Saminathan, M. H. V. (2024). An IoT-based smart irrigation system. *Engineering Proceedings*, 66(1), 13. <https://doi.org/10.3390/engproc2024066013>
14. Nayagam, M. G., Vijayalakshmi, B., Somasundaram, K., Mukunthan, M. A., Yogaraja, C. A., & Partheeban, P. (2023). Control of pests and diseases in plants using IoT technology. *Measurement: Sensors*, 26, 100713. <https://doi.org/10.1016/j.measen.2023.100713>
15. Nyakuri, J. P., Nkundineza, C., Gatera, O., Nkurikiyeyezu, K., & Mwitende, G. (2025). AI and IoT-powered edge device optimized for crop pest and disease detection. *Scientific Reports*, 15(1), 22905. <https://doi.org/10.1038/s41598-025-06452-5>
16. Saha, G., Shahrin, F., Khan, F. H., Meshkat, M. M., & Azad, A. K. M. A. M. (2025). Smart IoT-driven precision agriculture: Land mapping, crop prediction, and irrigation system. *PLOS ONE*, 20(3), e0319268. <https://doi.org/10.1371/journal.pone.0319268>
17. Săcăleanu, D.-I., Matache, M.-G., Roșu, Ș.-G., Florea, B.-C., Manciu, I.-P., & Perișoară, L.-A. (2024). IoT-enhanced decision support system for real-time greenhouse microclimate monitoring and control. *Technologies*, 12(11), 230. <https://doi.org/10.3390/technologies12110230>
18. Senoo, E. E. K., Anggraini, L., Kumi, J. A., Karolina, L. B., Akansah, E., Sulyman, H. A., Mendonça, I., & Aritsugi, M. (2024). IoT solutions with artificial intelligence technologies for precision agriculture: Definitions, applications, challenges, and opportunities. *Electronics*, 13(10), 1894. <https://doi.org/10.3390/electronics13101894>
19. Singh, G., & Sharma, S. (2024). A comprehensive review on the Internet of Things in precision agriculture. *Multimedia Tools and Applications*, 84(17), 18123-18198. <https://doi.org/10.1007/s11042-024-19656-0>

-
20. Soussi, A., Zero, E., Sacile, R., Trincherro, D., & Fossa, M. (2024). Smart sensors and smart data for precision agriculture: A review. *Sensors*, 24(8), 2647. <https://doi.org/10.3390/s24082647>
 21. Taniushkina, D., Lukashevich, A., Shevchenko, V., Belalov, I. S., Sotiriadi, N., Narozhnaia, V., Kovalev, K., Krenke, A., Lazarichev, N., Bulkin, A., & Maximov, Y. (2024). Case study on climate change effects and food security in Southeast Asia. *Scientific Reports*, 14(1), 16150. <https://doi.org/10.1038/s41598-024-65140-y>
 22. Yascaribay, G., Huerta, M., Silva, M., & Clotet, R. (2022). Performance evaluation of communication systems used for Internet of Things in agriculture. *Agriculture*, 12(6), 786. <https://doi.org/10.3390/agriculture12060786>