International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 11 | Nov - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

IOT Based Greenhouse Monitoring System Using ESP32

Darshan M, Omkar Gowda, Chandra Vardhan AN, Pavan Lakshmana Kuri

Guide: Prof. Vinay Kumar

Department of Electronics and Communication Engineering, Aditya College of Engineering and Technology

---

Abstract - Low-cost IoT greenhouse systems built on ESP32 or similar boards and cloud apps such as Blynk are widely reported, yet a survey of nine recent studies shows recurring weaknesses: single-threshold relay control that can chatter, little or no sensor-fault handling, dependence on cloud or power availability, uncalibrated sensors, absent alerts, and no defined rules for combining manual and automatic control. This paper proposes a firmware-level architecture on the ESP32 DevKit V1 that addresses these gaps without extra hardware: signal conditioning, three sensor-fault checks with a degraded timed-irrigation mode, hysteresis with burst-and-soak pump control, a three-state mode arbiter (AUTO, MANUAL, TIMED-MANUAL) with safety interlocks, offline-first execution with non-volatile state, and rate-limited Blynk alerts. The hardware (DHT11, soil sensor and LDR via ADS1115, relay-driven pump, fan and LED, I²C LCD) is assembled and firmware is under development. In a 24-hour simulation of the control logic, relay switching fell from 231–438 to 42 per day and a frozen-sensor fault was detected in 58 s, limiting pump run-time to 500 s instead of about 65,000 s. Hardware validation is pending.

Key Words: IoT, Smart Greenhouse, ESP32, Blynk, Fault Detection, Offline-First Control.

1. INTRODUCTION

Greenhouse yield depends on keeping temperature, humidity, light and root-zone moisture inside a narrow band, and the Internet of Things (IoT) makes it practical to sense these quantities and act on them remotely [14]. Low-cost microcontrollers such as the ESP32 [11], together with mobile dashboards such as Blynk [10], have made such systems easy to build, and many student and research prototypes have been reported.

However, a prototype that works on the bench is not the same as a system that can be left running. A noisy soil sensor can make a relay chatter; a disconnected sensor can keep a pump running until the pot floods; a router or cloud outage can stop control; and an app switch can conflict with an automatic rule. These operational aspects are rarely addressed in low-cost designs.

The problem is especially relevant for small growers and student projects. Their systems typically run unattended for days, are powered from ordinary adapters, connect over a home or farm Wi-Fi link that is not always stable, and use inexpensive sensors whose readings drift or fail. A control strategy for such a node must therefore be robust first and feature-rich second.

This paper (i) reviews nine recent greenhouse IoT studies and extracts their reported or evident limitations; (ii) consolidates them into seven gaps; and (iii) proposes a firmware architecture for the ESP32 DevKit V1 that targets those gaps using only the components already in the prototype. The contributions are the architecture itself, a gap-to-mechanism mapping, and a simulation-based comparison of the control logic with baseline strategies. We do not claim to remove every limitation in the literature; items such as disease detection or solar power are outside our scope.

Project status. The hardware has been assembled. Firmware is still under development, so all quantitative results in this paper come from a simulation of the control logic and are labelled as such. No field measurements are reported.

2. RELATED WORK

Table -1 summarises nine studies. Statements about limitations are either those stated by the authors or observations from the published description; where a page does not mention a feature, we say it is not described rather than claiming it is absent.

Table -1: Summary of existing studies and their gaps

Ref.	Approach	Limitation / gap
[1] 2015	Temperature, light, soil and water-level sensing; web page with real-time plots and storage	Web/cloud centred; local autonomy, fault handling and evaluation not described
[2] 2021	Raspberry Pi, ThingSpeak, SMS/GSM, Faster R-CNN disease detection (80.4 % reported)	Heavier and costlier hardware; 1,000-image dataset; unclear threshold units; no sensor-fault handling described
[3] 2023	ESP8266, Blynk, pump at soil < 50 % or temp > 34 °C	Single threshold, no hysteresis; temperature drives the pump; one reading per condition; alerts left as future work
[4] 2026	Raspberry Pi, ADC, relays, cloud and Android app	Inconsistent hardware description; no test details; cost noted as barrier
[5] 2025	ESP32, wireless sensors, early-warning	No quantitative limits or performance in abstract
[6] 2026	Fuzzy controller, validated sensors (< 3 % deviation), US$47.10	Crop-specific (cucumber); test duration and fault behaviour not stated
[7] 2025	ESP32 + Blynk, solar power, SD logging	DHT11 error 38 % and light error 32.5 % in first trial; one week of data, three plants; water-supply status unknown
[8] 2026	ESP32 + Blynk, hysteresis 38/40 %	No power backup; no long-term storage; pump only
[9] 2026	ZigBee, ESP32 edge, Home Assistant; fault detection, energy-aware	Multi-component stack; temperature control focus; single prototype

Three trends emerge from the table. First, ESP-class boards with Blynk dominate the low-cost end [3], [7], [8], but their control logic is usually a threshold test, which is easy to implement and easy to break under noise. Second, studies that add intelligence, such as fuzzy control [6], deep learning [2] or fault detection [9], gain capability at the price of hardware, data or integration effort. Third, evaluation is typically short: one reading per condition [3], one week of data [7], or a single prototype [9]. This leaves a niche for a lightweight design that adds robustness through careful firmware rather than extra hardware.

Looking at the individual studies in more detail, the Blynk-based irrigation work of Parma et al. [3] is the simplest: a soil reading below 50 % or a temperature above 34 °C switches the pump, so one noisy sample is enough to change the relay and a hot day waters the plant regardless of soil condition. Astutik et al. [8] improve this with a 38/40 % hysteresis band, but a 2 % band is narrow compared with typical sensor noise and no run-time limit is described. Galon et al. [7] add solar power and logging, yet the 38 % and 32.5 % sensor errors they report before calibration show how much an uncalibrated low-cost sensor can mislead a controller. Li Zeping et al. [9] are the only authors in the set to detect sensor faults, but their solution relies on a gateway, a home-automation server and smart plugs, which is beyond the reach of many small growers.

3. GAP ANALYSIS

From Table -1 we identify seven gaps (G1–G7).

- **G1 Chattering and unguarded actuation.** A single threshold [3] switches the relay on every noisy sample; hysteresis [8] helps but does not limit run-time or allow for the delay before water reaches the sensor.
- **G2 Sensor faults.** Only [9] reports fault detection, and it needs a heavier stack. A frozen or disconnected soil sensor can otherwise cause continuous watering.
- **G3 Cloud and power dependence.** Several designs centre on cloud or web services [1], [2], [4]; [8] reports no power-outage backup.
- **G4 Calibration.** Large early sensor errors were reported in [7] and validation is offline in [6]; a user-friendly in-field calibration is not described.
- **G5 Alerts.** Notifications are future work in [3] and [8].
- **G6 Cost and complexity.** Cameras and GSM [2], extra gateways [9] and imported equipment [4] raise cost.
- **G7 Manual versus automatic control.** None of the descriptions explains how an app override interacts with automatic rules or safety limits.

4. PROPOSED SYSTEM

4.1 Hardware

The prototype uses an ESP32 DevKit V1, a DHT11 for air temperature and humidity [13], a soil moisture sensor and an LDR module read through a 16-bit ADS1115 [12] (the ESP32 ADC2 channels are unusable while Wi-Fi is active [11]), a 4-channel 5 V relay module switching a 12 V pump, 12 V fan and LED strip from a 12 V adapter, and a 16×2 I²C LCD sharing the I²C bus with the ADS1115. A pipe and tank supply water. The prototype is a tabletop greenhouse with potted plants, and the electronics are mounted on a separate board outside the enclosure so that only the sensors, pipe and loads are exposed to humidity. Fig -1 shows the architecture.

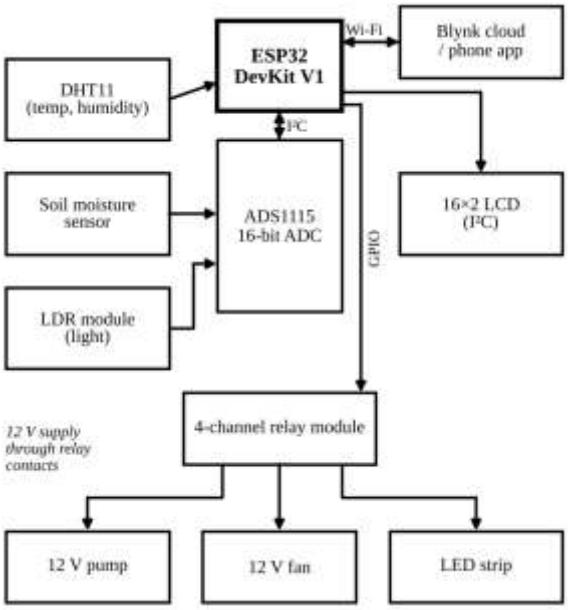


Fig -1: Hardware block diagram

Table -2: Pin allocation and device addresses

Signal	Connection
DHT11 data	GPIO 4 (10 kΩ pull-up)
I²C SDA / SCL	GPIO 21 / GPIO 22
ADS1115 (soil A0, LDR A1)	I²C address 0x48
16×2 LCD (I²C backpack)	I²C address 0x27 (or 0x3F)
Relay IN1 pump / IN2 fan	GPIO 26 / GPIO 27
Relay IN3 LED / IN4 spare	GPIO 25 / GPIO 33

Electrical design notes. The relay contacts isolate the 3.3 V logic from the 12 V load circuit. The pump and fan are inductive loads, so a flyback diode across each load is recommended to suppress switching spikes that can reset the ESP32 or corrupt I²C traffic. The ESP32 and relay module are powered from a regulated 5 V source (USB or a step-down converter from the 12 V adapter), and the adapter rating should exceed the sum of the pump, fan and LED currents with margin; these currents are to be measured on the prototype. The relay inputs are active-low on most boards, so GPIO pins are driven HIGH at boot to keep every load off.

4.2 Control Pipeline

Fig -2 shows the firmware pipeline, executed every 2 s on the ESP32 itself. Blynk is only the user interface; it is not in the control path.

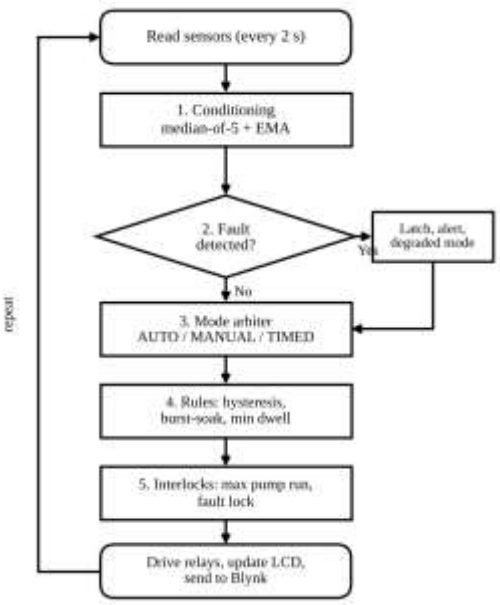


Fig -2: Proposed control pipeline

The loop of Fig -2 is summarised below (period 2 s; the DHT11 is read at most once per 2 s as its datasheet requires [13]):

```
read DHT11, soil (A0), LDR (A1)
soil_f = EMA(median5(soil_raw))
fault = range OR stuck OR ineffective
if fault: latch, alert, pump = DEGRADED
mode = arbiter(app events, timeout)
cmd = rules(mode, soil_f, T, RH, light)
cmd = interlocks(cmd, dwell, max_run)
write relays; refresh LCD
if Wi-Fi up: push data, send events
save changed settings to NVS
```

Conditioning (G1). Each analog value is passed through a median-of-5 filter followed by an exponential moving average, removing spikes before any decision is taken.

Fault detection (G2). Three checks run on the soil channel: (a) a range check (reading outside 0–100 % after calibration, or an ADS1115 that does not answer on I²C); (b) a stuck-at check (standard deviation of the last 30 raw samples below 0.05 %); and (c) an ineffective-irrigation check (the pump has run for at least 90 s within 20 minutes but the conditioned moisture has risen by less than 3 points). The third check also catches a floating, noisy input that the stuck-at check cannot. On a fault the pump is locked, an alert is raised and the controller enters a degraded mode that waters for 40 s every 2 h until the user acknowledges in the app. DHT11 read failures are counted and the fan defaults to ON after a set number of consecutive failures.

Rules (G1). Irrigation starts when conditioned soil moisture falls below 38 % and stops at 42 % (hysteresis). Water is applied in bursts of at most 20 s followed by a 90 s soak, because moisture reaches the sensor with a delay; this prevents overshoot. The pump depends only on soil moisture, not on temperature. The fan uses its own hysteresis band (for example ON above 30 °C, OFF below 28 °C) and the LED uses a light threshold with hysteresis; every relay has a minimum dwell time.

Mode arbiter and interlocks (G7). The system has three modes. AUTO follows the rules. MANUAL follows the Blynk switches. TIMED-MANUAL is entered when the user toggles a switch while in AUTO and reverts to AUTO after a configurable time (for example 30 minutes), so a forgotten override cannot leave the greenhouse uncontrolled. Interlocks apply in every mode: the pump has a maximum continuous run time and is blocked when a fault is latched or the tank is empty (if a level input is added).

Offline-first operation (G3). The control loop never waits for Wi-Fi; reconnection to Blynk is non-blocking. Thresholds, mode, calibration and fault flags are stored in the ESP32 non-volatile storage, and relays are initialised OFF at boot, so after a power cut the node resumes with its configuration and without an uncontrolled pump start.

Calibration and alerts (G4, G5). Two Blynk buttons store the raw reading in dry air and in water as the 0 % and 100 % points in non-volatile storage, avoiding re-flashing. Blynk events notify the user of faults, high temperature, dry soil and offline recovery, with a minimum interval per event type to avoid alert floods.

Cost (G6). The design adds no hardware beyond the components already listed: the novelty lies in the firmware logic.

Table -3: Gap-to-mechanism mapping

Gap	Mechanism
G1	Median+EMA, hysteresis, burst-soak, min dwell
G2	Range, stuck-at, ineffective-irrigation checks; degraded mode
G3	Local control, NVS state, relays OFF at boot
G4	Two-point soil calibration from Blynk
G5	Rate-limited Blynk events
G6	Firmware-only, no extra hardware
G7	AUTO / MANUAL / TIMED-MANUAL, interlocks

Table -4: Default control parameters

Parameter	Default
Control period	2 s
Soil ON / OFF thresholds	38 % / 42 %
Pump burst / soak	20 s / 90 s
Ineffective-irrigation check	≥ 90 s ON in 20 min, rise < 3 points
Stuck-at check	σ of 30 samples < 0.05 %
Degraded watering	40 s every 2 h
Fan ON / OFF	30 °C / 28 °C
TIMED-MANUAL timeout	30 min

4.3 Blynk Datastreams

Table -5: Blynk datastreams

Pin	Type	Function
V0–V1	Read	Temperature, humidity
V2–V3	Read	Soil moisture, light level
V4–V6	Switch	Pump, fan, LED (MANUAL)
V7	Selector	AUTO / MANUAL mode
V8–V9	Button	Soil dry / wet calibration
V10	Text	Status: mode and fault

Events (fault latched, high temperature, dry soil, device back online) are sent with Blynk event logging, each limited to one notification per fixed interval.

4.4 Failure-Mode Handling

Table -6 lists the main failures considered at design time and the intended response. It is the basis for the hardware fault-injection tests of Section 6.

Table -6: Failure modes and designed responses

Failure	Designed response
Soil sensor unplugged or frozen	Stuck-at / range check; pump locked; alert; timed watering
Floating or drifting soil input	Ineffective-irrigation check; degraded mode

Tank empty or pipe blocked	Moisture does not rise: same check; pump locked
ADS1115 not answering on I²C	Range check fails; analog loads set to safe state
DHT11 read errors	Last good value held; fan ON after N failures
Wi-Fi or Blynk cloud loss	Local control continues; non-blocking reconnect
Power cut and restart	Settings restored from NVS; relays OFF at boot
Forgotten manual override	TIMED-MANUAL reverts to AUTO

4.5 Firmware Organisation

The firmware is organised as a cooperative, non-blocking loop with no long delays, so that sensing, control and networking never block one another. Four small modules are planned: a sensing module (drivers for the DHT11 and ADS1115 with the filters of Section 4.2); a safety module (fault checks, dwell timers and interlocks); a control module (mode arbiter and rules); and a connectivity module (Wi-Fi, Blynk and event limiting). A hardware watchdog timer restarts the ESP32 if the loop stalls, and the relays are forced OFF in the start-up code before Wi-Fi is attempted. User settings (thresholds, calibration points, mode and fault flags) are written to non-volatile storage only when they change, to limit flash wear.

Sensor sampling runs every 2 s, the LCD refreshes every 2 s, and data are pushed to Blynk every 5–10 s to stay within the platform message limits. These periods are configuration constants that can be changed without altering the logic.

5. EVALUATION

5.1 Simulation Setup

Because the firmware is not yet complete, the control logic was evaluated in a Python simulation of one day (2 s time step, 10 random seeds). Soil moisture was modelled with a diurnal evaporation of about 2.9 % per hour, a pump adding 0.15 % per second, and a sensor lagging the true moisture by a first-order delay of 45 s with Gaussian noise ($\sigma = 0.8$ %). This model is our assumption and is not calibrated to a real pot. Three controllers were compared: B1, a single threshold at 40 % on raw readings (as in [3]); B2, hysteresis 38/40 % on raw readings (as in [8]); and P, the proposed pipeline. Three scenarios were run: normal operation; a sensor fault from hour 6 (frozen at 25 %, and a floating noisy 20 % reading); and a 10-hour cloud outage, in which a cloud-dependent controller B3 (B2 logic that holds its last relay state when disconnected) is compared with P.

5.2 Results

Table -7: Simulation results over 24 h (mean of 10 runs)

Scenario / controller	Switches	Pump ON (s)	Max moist. (%)
Normal, B1	438	442	45
Normal, B2	231	430	45
Normal, P	42	420	45
Frozen sensor, B2	64	64,921	100
Frozen sensor, P	32	500	54.7
Floating sensor, B2	64	64,921	100
Floating sensor, P	40	568	64.7

In normal operation P cut relay switching by 82 % against B2 and 90 % against B1 for about the same water use (Fig -3a), because filtering removes noise-driven toggling and bursts are separated by a soak. With a frozen sensor, B1 and B2 kept the pump on for about 65,000 s and drove the pot to saturation, whereas P detected the fault after 58 s on average and limited pump time to 500 s. For a floating noisy reading, the ineffective-irrigation check detected the fault after 443 s on average (10 of 10 runs) with a pump time of 568 s. In the outage scenario B3 let soil moisture fall to about 1 % and kept it inside the 35–50 % band for only 64 % of the day, while P, which does not depend on the cloud, stayed in band for the whole day (minimum 38 %).

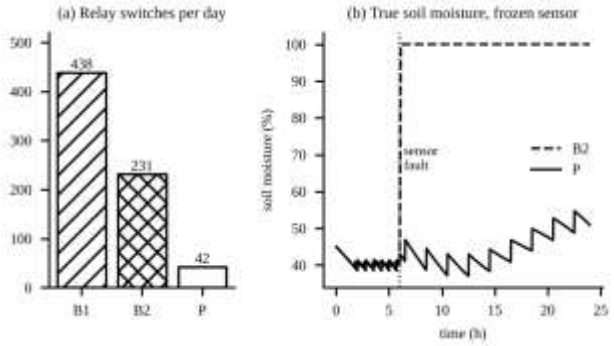


Fig -3: (a) Relay switches per day; (b) true soil moisture under a frozen-sensor fault

5.3 Sensitivity to Sensor Noise

The noise level was varied from $\sigma = 0.4$ % to 2.5 % in normal operation (Table -8). When the sensor is very clean, hysteresis alone (B2) is almost as good as P, which is an honest limit of the proposed logic: its advantage grows with noise. As noise rises, B1 and B2 toggle the relay hundreds of times per day while P stays near 42–44. The fault checks raised no false alarm in any of the 40 normal runs.

Table -8: Relay switches per day versus sensor noise (10 runs)

Noise σ (%)	B1	B2	P
0.4	428	41	42
0.8	438	231	42
1.5	450	386	44
2.5	467	437	44

5.4 Limitations of the Simulation

These figures reflect an idealised model: real soils differ in infiltration and the effect of a sensor fault will depend on the failure mode. The B3 baseline is a deliberately simple assumption. The 82–90 % switching reduction mainly shows the value of filtering and burst-soak under the assumed noise level. The simulation also does not model temperature, humidity or light control, relay wear, or electrical interference from the pump motor. These effects can only be assessed on the hardware.

5.5 Reproducibility

All simulation settings are listed in Section 5.1 and Table -4. Each controller was run for 43,200 steps of 2 s with random seeds 0–9, and results are means over those seeds. The pump ON time is reported in seconds because the simulated flow of 0.15 % moisture per second is an assumption; absolute values would scale with the real pump flow, but the ratios between controllers do not depend on it in the fault scenarios, where the baselines keep watering until the pot is saturated.

6. PLANNED HARDWARE VALIDATION

Once the firmware is complete, the assembled prototype will be tested as in Table -9. Target values are initial acceptance goals chosen by the authors, not results.

Table -9: Hardware validation plan

Test	Procedure	Target
Sensor accuracy	DHT11 vs reference thermometer / hygrometer	Within datasheet tolerance [13]
Soil calibration	Dry-air and water points via Blynk	Readings span 0–100 %

Chatter	Count relay switches, threshold-only vs proposed firmware, 24 h	Clear reduction
Fault injection	Unplug soil sensor during watering	Pump locked within minutes; alert received
Cloud outage	Switch off router for hours	Control continues; data resumes
Power cut	Remove and restore supply	Settings kept; pump off at boot
Manual control	Toggle Blynk switches; time response	Relay changes within a few seconds

The development order planned for the remaining work is: firmware for sensing and LCD, relay control with the safety module, Blynk datastreams and events, in-field calibration, and finally the fault-injection and outage tests of Table -9. Each step can be verified on the bench before the next is added.

7. COMPARISON AND DISCUSSION

Table -10 compares the proposed design with the closest feature in the surveyed studies. It also lists what the proposed design does not include.

Table -10: Feature comparison with the surveyed studies

Feature	Closest prior work	Proposed
Hysteresis control	[8] 38/40 % band	Yes, plus burst-soak
Sensor-fault detection	[9] (ZigBee + Home Assistant stack)	Yes, firmware-only
Local control when offline	Not described in most	Yes, by design
Manual / auto arbitration	Not described	Three modes, interlocks
Alerts	[2] SMS via GSM	Blynk events
In-field calibration	Offline in [6], [7]	Blynk buttons
Data logging	[7] SD card	Future work
Solar / battery supply	[7]	Future work
Disease detection	[2] deep learning	Not included

The proposed design is therefore not a superset of the earlier work; it focuses on operational robustness at low cost, and leaves sensing intelligence and energy autonomy as extensions.

The proposed architecture deliberately trades features for robustness. It does not use machine learning, and its thresholds are fixed numbers that a grower must set for the crop, whereas fuzzy control [6] adapts more smoothly. Threats to validity include the unvalidated soil model, the choice of baselines, which are simplified versions of strategies described in [3] and [8] rather than reimplementations of those systems, and the use of a single soil sensor type. In return the logic is transparent: every relay decision can be explained by one rule and checked on the LCD. The fault checks are generic rather than sensor-specific, which makes them simple but means some subtle failures, such as slow drift, will not be detected and should be handled by periodic recalibration.

For a new crop, the grower should set the ON threshold slightly above the permanent wilting level, the OFF threshold below field capacity, and then adjust the burst and soak times so that moisture at the sensor stops rising within one soak period. Because the soil takes time to respond, the soak time is the most important parameter to tune on the real pot.

The design can be extended without changing the pipeline: a water-level float switch on the spare relay-board GPIO would give a direct tank-empty interlock; an SD card or a time-series service would add the long-term logging missing in [8]; a battery or solar supply as in [7] would add power resilience; and multiple nodes could report to one Blynk template. The same architecture applies to other small-scale controlled environments such as nurseries, hydroponic racks and terrace gardens.

8. CONCLUSIONS

A review of nine greenhouse IoT studies exposed seven operational gaps. A firmware-only architecture for the ESP32 DevKit V1 and Blynk was proposed to address them: conditioning, fault checks with a degraded mode, hysteresis with burst-and-soak irrigation, a three-state mode arbiter with interlocks, offline-first execution with stored state, in-field calibration and rate-limited alerts. Simulation indicates fewer relay switches, rapid fault containment and unbroken control during a cloud outage. Hardware validation, a tank-level input, battery backup and crop trials are the next steps.

Future scope

- **Field trials.** Run the prototype on a real crop for a full growth cycle and compare soil moisture, water use and relay switching against threshold-only firmware.
- **Tank-level input.** Add a float switch to the spare GPIO so that an empty tank is detected directly rather than inferred from ineffective irrigation.
- **Better sensing.** Replace the DHT11 with a more accurate temperature and humidity sensor and use a corrosion-resistant capacitive soil probe.
- **Logging and analytics.** Store readings locally or in a time-series service and use them to tune thresholds per crop.
- **Power resilience.** Add a battery or solar supply to ride through mains outages for longer than the stored-state recovery described here.

ACKNOWLEDGEMENT

The authors thank Prof. Vinay Kumar and the Department of Electronics and Communication Engineering, Aditya College of Engineering and Technology, for guidance and laboratory facilities.

REFERENCES

1. Rao, K., Srijana, K., Prasad, D., Chandrasekaran, K.: Ubiquitous greenhouse monitoring system. In: Proc. Int. Conf. Emerging Research in Electronics, Computer Science and Technology (ICERECT), IEEE (2015) 335–339
2. Fatima, N., Siddiqui, S.A., Ahmad, A.: IoT-based smart greenhouse with disease prediction using deep learning. Int. J. Adv. Comput. Sci. Appl. 12(7) (2021) 113–121
3. Parma, S.K., Alfian, Y., Manurung, E.M., Mufit, C.: IoT-based automatic plant watering system with the Blynk application. In: Proc. 4th Int. Seminar and Call for Paper (ISCP UTA'45 Jakarta 2023), SCITEPRESS (2024) 427–431. DOI 10.5220/0012584100003821
4. Sivaranjani, A.: IoT and cloud computing in a smart greenhouse. Int. J. Eng. Res. Technol. (IJERT) 14(7), NCIRTAI 2026, IJERTCONV14IS070040
5. Boussaoula, S., Jbari, A., Achmamad, A., Bellarbi, L., Yaakoubi, N.: Connected greenhouse for experimental validation of IoT control and monitoring system. EPJ Web Conf. 330 (2025) 04001. DOI 10.1051/epjconf/202533004001
6. Wayo, E., Ruang-on, S., Songsri-in, K., Tina, F.W., Nuengmatcha, P.: Design and implementation of a smart sensing IoT system for cost-effective greenhouse environmental monitoring and control. Int. J. Smart Sens. Intell. Syst. 19(1) (2026). DOI 10.2478/ijssis-2026-0008
7. Galon, M.L.Q., Tumaliwan, M.V.R., Sejera, M.M.: Automated monitoring and control system of solar greenhouse using ESP32

and Blynk application. Eng. Proc. 92(1) (2025) 57. DOI 10.3390/engproc2025092057
8. Astutik, R.P., Gunawan, B., Mas'ud, A.A.: IoT based watering system activation on smart garden. AUT J. Electr. Eng. 58(Special Issue 1) (2026) 17–26. DOI 10.22060/eej.2025.24495.5716
9. Li Zeping, Abdullah, N., Ishak, M.K.: Smart greenhouse climate control with real-time fault detection and energy-aware automation. Smart Agric. Technol. 13 (2026) 101707. DOI 10.1016/j.atech.2025.101707
10. Blynk Inc.: Blynk IoT Platform Documentation. https://docs.blynk.io
11. Espressif Systems: ESP32 Series Datasheet. https://www.espressif.com
12. Texas Instruments: ADS111x 16-Bit I²C ADCs Datasheet (SBAS444)
13. Aosong Electronics: DHT11 Humidity & Temperature Sensor Datasheet
14. Gubbi, J., Buyya, R., Marusic, S., Palaniswami, M.: Internet of Things (IoT): A vision, architectural elements, and future directions. Future Gener. Comput. Syst. 29(7) (2013) 1645–1660