

LOW COST IOT ARCHITECTURE FOR FAIL SAFE MONITORING AND CONTROL OF A SMALL SCALE HYDROPONIC SYSTEM

^{1*}R Vijayakumar,² Arjun K, ³ Mithunkumar M R , ⁴ Manjunath M, ⁵ Sasnidha C

Department of Electrical and Electronics Engineering, SNS College of Technology, Tamil Nadu, India

Abstract

Small hydroponic units require dependable nutrient circulation and timely fault reporting, but many published prototypes combine sensing and cloud dashboards without documenting fail-safe behavior, calibration, or performance metrics. This paper presents a low-cost Internet of Things architecture built around an ESP8266 controller, a reservoir-level sensor, a pH-sensing interface, an IRLZ44N pump driver, and the Blynk platform. The contribution is an edge-first control design in which pump protection remains operational when cloud connectivity is unavailable. The paper defines threshold and hysteresis logic, sensor plausibility checks, event logging, alarm priorities, and a reproducible validation protocol. A design-level analysis shows that the available prototype supports reservoir monitoring, protected pump switching, and remote status reporting; however, the current evidence does not yet establish pH-control accuracy, agronomic benefit, water saving, or long-term reliability. Accordingly, quantitative acceptance criteria are specified for calibration error, response time, packet delivery, pump duty cycle, energy use, and water balance. The architecture is appropriate for laboratory teaching and small nutrient-film or recirculating demonstrations, while publication-grade claims about crop performance require a replicated cultivation experiment. The revised design separates demonstrated functions from proposed extensions and provides a defensible path from prototype demonstration to experimentally validated smart hydroponics.

Keywords hydroponics, Internet of Things, ESP8266, pump protection, sensor calibration, edge control, precision agriculture

1 Introduction

Hydroponics supplies water and mineral nutrients directly to plant roots without field soil. Closed or recirculating systems can improve control over the root-zone environment and are attractive where cultivable land or water is constrained. These advantages are conditional rather than automatic: nutrient composition, pH, electrical conductivity, dissolved oxygen, temperature, light, crop choice, sanitation, and operator practice determine system performance [1–3]. Recent reviews therefore treat hydroponics as

a controlled-environment production method whose resource efficiency must be measured for a defined crop and boundary, not assumed from the technology label alone [2,3].

Low-cost microcontrollers and wireless sensors have made continuous monitoring accessible to teaching laboratories, households, and small growers. Existing smart-hydroponics studies commonly monitor pH, electrical conductivity, solution temperature, ambient conditions, and water level, then publish readings through a mobile or web interface [4–8]. The systematic review by Muhasin et al. [6] found a fragmented evidence base and emphasized the need for clearer implementation guidance. Sneineh and Shabaneh [7] provided a replicable ESP32 monitoring method, while recent AIoT work has extended monitoring toward crop recommendation and nutrient management [8]. These studies also reveal a recurring limitation: a connected dashboard does not by itself demonstrate safe control, agronomic improvement, or reliable operation.

The original prototype considered in this study integrates an ESP8266, water-level and pH sensing, a MOSFET-driven pump, and Blynk connectivity. Its practical value lies in low-cost component integration, but its scientific contribution requires a precise separation between implemented functions, design intentions, and experimentally verified outcomes. This paper therefore makes three contributions: (i) an edge-first architecture that protects the pump independently of cloud availability; (ii) explicit control, fault, calibration, and logging requirements; and (iii) a quantitative protocol for validating the prototype before agronomic claims are made. The scope is engineering design and validation readiness. Crop-yield improvement and resource savings are not claimed without comparative measurements.

2 Related Work and Research Gap

Early IoT hydroponic prototypes established the feasibility of inexpensive sensing, wireless transmission, and rule-based actuation [4,5]. Subsequent systems expanded the monitored variables and adopted mobile dashboards, data logging, artificial intelligence, and remote decision support [6–9]. The direction of the field is clear: credible systems increasingly combine plant-science variables with engineering metrics such as sensor error, communication reliability, actuator response, energy use, and fault recovery.

Table 1 Position of the proposed architecture relative to recent work

Study	Platform and variables	Control or analysis	Relevant limitation
Lakshmanan et al. [5]	IoT hydroponics; multiple environmental variables	Automated monitoring and control	Limited transferability across crops and installations

Study	Platform and variables	Control or analysis	Relevant limitation
Muhasin et al. [6]	Systematic review of smart hydroponics	Evidence synthesis	Reports inconsistent architectures and evaluation practices
Sneineh and Shabaneh [7]	ESP32; water-quality and environmental sensing	Real-time IoT monitoring	Method paper; crop-level benefit requires separate trials
Tajnin et al. [8]	AIoT nutrient monitoring and crop recommendation	Data-driven recommendation	Higher model and data requirements
Present study	ESP8266; level, pH interface, pump status, Blynk	Edge fail-safe switching and validation protocol	Empirical time-series and crop trial still required

The research gap is not the absence of another connected sensor node. It is the lack of a low-cost design whose claims are traceable to explicit tests and whose basic safety does not depend on the network. The present architecture addresses this engineering gap through local interlocks, hysteresis, plausibility screening, and structured event logging. It does not replace the additional pH, electrical-conductivity, dissolved-oxygen, temperature, and light controls required for commercial crop management.

3 Materials and Methods

3.1 System Boundary and Functional Requirements

The system boundary includes reservoir sensing, edge computation, pump switching, Wi-Fi communication, dashboard display, and alarm generation. Plant lighting, nutrient dosing, heating, cooling, aeration, and mechanical crop support are outside the implemented boundary. The controller must continue safe local operation during Wi-Fi or cloud interruption. Remote commands must never override the low-level dry-run interlock.

Table 2 Functional requirements and evidence needed for verification

ID	Requirement	Verification evidence
----	-------------	-----------------------

ID	Requirement	Verification evidence
FR1	Sample reservoir level and pH channels at a defined interval	Timestamped raw and filtered values
FR2	Prevent pump operation below the validated minimum level	Fault-injection test with low-water condition
FR3	Avoid rapid switching near a threshold	State log showing hysteresis or minimum dwell time
FR4	Publish measurements and states to Blynk	Packet delivery ratio and timestamp audit
FR5	Retain safe control during network loss	Offline test with recorded pump and alarm states
FR6	Flag implausible, stale, or disconnected sensor values	Open-circuit, short-circuit, and frozen-value tests

3.2 Hardware Architecture

The ESP8266 NodeMCU provides the processing and Wi-Fi interface. A water-level sensor supplies the primary dry-run protection signal. The pH channel is included as a monitoring interface; because the source prototype does not document a dosing actuator or calibration dataset, the revised system does not claim autonomous pH correction. An IRLZ44N logic-level N-channel MOSFET switches the 12 V DC pump. The gate circuit should include a series resistor and pull-down resistor, and an inductive pump requires a correctly rated flyback diode. The controller and sensor supply must share a defined reference while remaining adequately decoupled from pump transients [10,11].

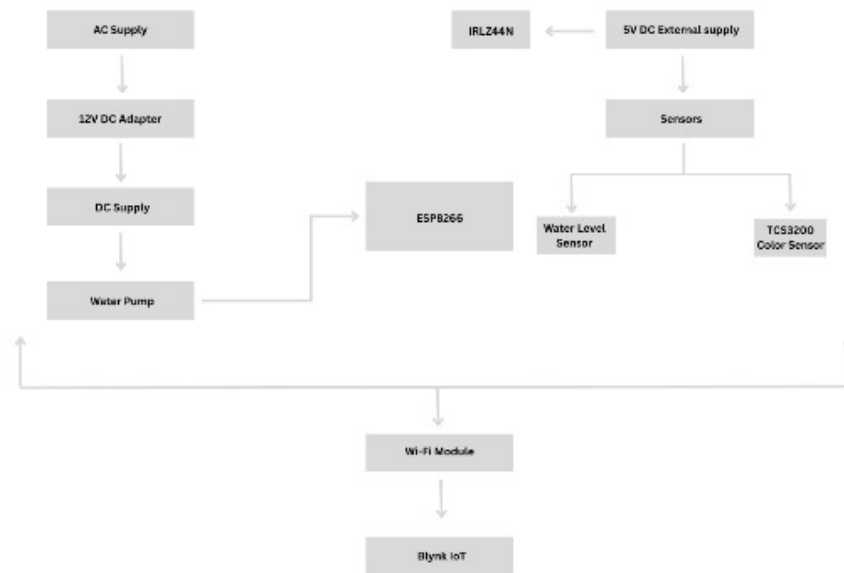


Figure 1 *Functional architecture of the source prototype*

Table 3 *Components, roles, and publication-critical specifications*

Component	Role	Specification that must be reported
NodeMCU ESP8266	Sampling, decision logic, Wi-Fi	Firmware version, sampling interval, ADC scaling, watchdog behavior
pH interface	Nutrient-solution acidity monitoring	Probe model, range, calibration buffers, slope, offset, temperature
Water-level sensor	Low-level detection	Sensor type, mounting height, hysteresis, repeatability
IRLZ44N and protection	Pump power switching	Gate network, diode, measured current, heat dissipation
12 V DC pump	Nutrient circulation	Rated flow, measured flow, head, current, duty cycle
Blynk platform	Dashboard and alarms	Template, update rate, disconnect handling, data

Component	Role	Specification that must be reported
		retention

3.3 Edge Control and Fail Safe Logic

Let $L(k)$ denote the filtered reservoir-level signal at sample k . Two thresholds, L_low and L_high , implement hysteresis. The pump permission state $P(k)$ is updated locally as follows:

$$P(k) = 0 \text{ if } L(k) \leq L_low; \quad P(k) = 1 \text{ if } L(k) \geq L_high; \quad \text{otherwise } P(k) = P(k-1).$$

The demanded pump state is combined with the permission state, sensor-health flag $H(k)$, and maximum-run timer T_max :

$$Pump(k) = Demand(k) \wedge P(k) \wedge H(k) \wedge [t_run < T_max].$$

This ordering makes dry-run protection dominant. A remote command can request circulation, but it cannot bypass a low-level or sensor-fault condition. A minimum off-time limits relay or MOSFET cycling, and a watchdog returns the output to OFF if the controller becomes unresponsive. Every transition should record timestamp, raw signal, filtered signal, threshold, command source, pump state, network state, and fault code. These records are essential for reproducibility and diagnosis.

3.4 Sensor Calibration and Data Quality

A two- or three-point pH calibration should be performed with fresh traceable buffers that bracket the working range. Voltage must be recorded after stabilization, and a linear calibration model $pH = aV + b$ should be fitted for each probe. Calibration performance should be reported as coefficient of determination, root-mean-square error, mean absolute error, bias, and maximum absolute error. An independent check buffer should be used after fitting. Because pH electrodes drift and respond to temperature, the experiment should report solution temperature, recalibration frequency, cleaning, and probe age [1,2].

$$RMSE = \sqrt{[(1/n) \sum_i (y_i - \hat{y}_i)^2]} \quad MAE = (1/n) \sum_i |y_i - \hat{y}_i|.$$

The level sensor should be tested through repeated fill-and-drain cycles. Threshold repeatability, false-trigger rate, response time, and mounting sensitivity should be measured. Median filtering can reject isolated spikes, but the window length must remain short enough to preserve protection response.

Missing, saturated, non-finite, or unchanged readings over a defined interval should raise a sensor fault rather than be treated as valid process measurements.

3.5 Communication and Data Management

Blynk receives telemetry and displays system state, but the network is supervisory rather than safety-critical. Packet delivery ratio is calculated from received and expected messages. End-to-end latency is the difference between the edge timestamp and platform receipt time when clocks are synchronized. Tests should include normal Wi-Fi, intermittent connectivity, router shutdown, platform unavailability, and reconnection. Credentials must not be embedded in a public code repository; tokens should be provisioned separately and changed before sharing firmware [12].

$$\text{Packet delivery ratio (\%)} = 100 \times N_{\text{received}} / N_{\text{expected}}.$$

3.6 Experimental Design for Quantitative Validation

Engineering validation should precede a crop experiment. The bench phase should include at least 30 repeated level transitions, repeated pump commands under each interlock state, a 24–72 h continuous run, controlled network outages, and pH checks spanning the intended range. A later agronomic trial should compare automated and manually managed units using the same crop, cultivar, planting date, nutrient formulation, reservoir volume, light schedule, and environmental conditions. Biological replicates and randomized unit positions are needed because plants within one reservoir are subsamples rather than independent experimental units.

Table 4 *Minimum quantitative dataset for a publication-grade evaluation*

Domain	Variables	Recommended analysis
Sensor	Reference value, raw output, calibrated value, temperature	Regression, RMSE, MAE, bias, Bland–Altman plot
Control	Command time, state transition, response time, fault code	Median, interquartile range, 95th percentile, success rate
Network	Expected and received packets, outage duration, latency	Delivery ratio, latency distribution, recovery time
Resources	Reservoir additions, discharge, pump	Daily totals; L kg ^{−1} fresh mass; kWh kg ^{−1}

Domain	Variables	Recommended analysis
	runtime, electrical energy	
Crop	Fresh and dry mass, leaf number, leaf area, root length, days to harvest	Mixed model or ANOVA with unit as replicate; effect size and confidence interval

For repeated observations, the hydroponic unit should be treated as a random effect and treatment as a fixed effect. Assumptions should be checked from residuals. If normality or equal variance is untenable, transformation or a suitable generalized model should be used. Exact p values, effect sizes, and 95% confidence intervals are more informative than declaring results significant or non-significant alone.

4 Design Verification and Analytical Results

4.1 Functional Traceability

Inspection of the documented prototype establishes a complete sensing–processing–actuation–communication path: level and pH interfaces connect to the ESP8266; the controller drives a 12 V pump through an IRLZ44N stage; and Wi-Fi links the node to Blynk. This supports the feasibility of local acquisition, pump switching, and remote display at the architectural level. The available manuscript does not provide firmware listings, time-series logs, calibration records, flow measurements, energy measurements, or crop observations. Consequently, accuracy, reliability, water saving, yield improvement, and cost-effectiveness cannot be quantified from the current evidence.

Table 5 Claim and evidence audit for the present prototype

Claim	Evidence in source material	Status after revision
Integrated IoT architecture	Component descriptions and block diagram	Supported as design evidence
Automatic pump switching	Controller–MOSFET–pump control path described	Functionally plausible; requires transition log
Dry-run prevention	Water-level input and proposed	Design intent; fault-injection test

Claim	Evidence in source material	Status after revision
	interlock	required
Real-time remote monitoring	ESP8266 and Blynk connection described	Architecture supported; latency and delivery unmeasured
Accurate pH monitoring	pH sensing stated but calibration absent	Not established
Water or energy saving	No baseline, flow, runtime, or energy data	Not established
Improved plant growth or yield	No crop, duration, replication, or biomass data	Not established

4.2 Control Robustness Analysis

A single-threshold controller can chatter when sensor noise causes repeated crossings. The two-threshold rule in Section 3.3 creates a neutral band and is therefore more robust. The state-retention term also makes the decision deterministic inside that band. Adding a maximum-run timer limits the consequence of a stuck demand, while the health flag converts implausible measurements into a safe OFF state. These properties can be established from the logic; their numerical performance still depends on measured noise, threshold placement, sampling interval, and hydraulic response.

The edge-first arrangement also removes the cloud platform from the safety loop. If Wi-Fi fails, local level protection and scheduled circulation can continue; telemetry becomes stale and an offline alarm is recorded for later transmission. This is preferable to a cloud-dependent actuator command because network delay cannot hold the pump ON against a local fault. Reconnection logic should transmit the latest state and a bounded backlog rather than flooding the platform with every stored sample.

4.3 Failure Mode and Effects Analysis

Table 6 Failure modes and proposed mitigations

Failure mode	Potential effect	Detection	Mitigation
--------------	------------------	-----------	------------

Failure mode	Potential effect	Detection	Mitigation
Level sensor open or short	False water state; pump risk	Range and plausibility check	Pump OFF; local and remote alarm
pH probe drift	Misleading nutrient assessment	Check-buffer error or rate-of-change limit	Recalibrate; do not automate dosing
Wi-Fi or cloud loss	Unavailable dashboard	Heartbeat timeout	Maintain local control; queue summary events
MOSFET short circuit	Pump remains ON	Command–current mismatch if current sensing is added	Fuse and independent cutoff
Pump blockage or dry run	No flow and possible damage	Flow/current anomaly if added	Stop pump; inspect reservoir and line
Controller reset	Uncontrolled startup transient	Reset-cause log	Default output OFF; watchdog; validated restart sequence

The analysis identifies two priority improvements. First, a flyback diode, fuse, pull-down resistor, watchdog, and default-OFF startup state are essential electrical protections. Second, current or flow sensing would convert several undetectable hydraulic and switch failures into observable faults. These additions contribute more to dependability than adding further dashboard variables without corresponding fault logic.

4.4 Agronomic Interpretation

The implemented variables are insufficient for closed-loop nutrient management. pH affects nutrient availability, but pH alone does not indicate total ionic concentration. Electrical conductivity is normally used as a practical proxy for solution strength, while solution temperature and dissolved oxygen influence root-zone physiology [1–3]. Ambient air temperature, relative humidity, and light are also required to interpret crop response. The TCS3200 color sensor shown in the source prototype may support exploratory color classification under controlled illumination, but it cannot be treated as a validated plant-

health sensor without a fixed optical geometry, color reference, labelled ground truth, and classification performance.

Resource-efficiency statements also require a defined comparison. Water use should include initial reservoir charge, additions, discharge, sampling loss, leakage, and harvested biomass. Energy should include the pump, controller, communication hardware, lighting, aeration, and environmental control within the selected boundary. Reporting only pump runtime would underestimate total production energy in indoor cultivation. Recent hydroponics research similarly emphasizes that yields and resource burdens depend on system design and operating context [2,3,9].

5 Discussion

The revised architecture is strongest as a teaching and small-scale monitoring platform. It uses accessible hardware, places a critical interlock at the edge, and can generate the event data needed for systematic evaluation. Its novelty is therefore methodological: it links every performance claim to a measurable variable and acceptance test. This is a more defensible contribution than describing connectivity as proof of intelligent control.

Comparison with recent work shows that the prototype has a narrower sensing scope than systems that include electrical conductivity, temperature, humidity, light, or AI-based recommendations [6–9]. That narrower scope can still be useful if reliability is demonstrated rigorously. A small system that reports calibration error, packet loss, actuator response, and fault recovery may offer stronger reproducible evidence than a larger sensor list without validation. The proposed protocol also makes future comparisons possible across firmware revisions and cultivation cycles.

Several claims in the earlier manuscript were too strong for the evidence. Automatic switching does not prove water saving; a mobile dashboard does not prove real-time reliability; component availability does not prove low lifecycle cost; and a hydroponic prototype does not prove higher yield. The revised discussion treats these as hypotheses for measurement. This distinction is important for peer review because it prevents engineering feasibility from being confused with agronomic efficacy.

6 Limitations and Future Experimental Work

This study is limited by the absence of raw sensor logs, reference-instrument readings, firmware, pump flow and power measurements, bill-of-materials data, cultivation conditions, crop observations, and replicated control units. The block diagram also shows a color sensor while the narrative refers to pH sensing, so the final hardware configuration must be reconciled before submission. No autonomous pH

dosing is demonstrated. The present results are therefore limited to architecture, control logic, verification requirements, and failure analysis.

The next experimental version should add a documented EC sensor, solution-temperature sensor, and flow or current feedback; publish the controller code and data dictionary; and record continuous data through controlled faults and at least one complete crop cycle. A replicated automated-versus-manual experiment should then report water input, energy, nutrient adjustments, uptime, fresh and dry biomass, quality indicators, and uncertainty. A cost table should separate prototype purchase price from replacement, calibration, communication, and energy costs.

7 Conclusion

A low-cost ESP8266 platform can integrate reservoir sensing, protected pump switching, and remote monitoring for a small hydroponic unit. The revised design improves technical defensibility by keeping dry-run protection at the edge, defining hysteresis and fault states, and specifying calibration, communication, resource, and crop metrics. The available prototype documentation supports architectural feasibility but does not yet support quantitative claims about pH accuracy, water saving, energy efficiency, reliability, or plant growth. Completion of the proposed bench tests and replicated cultivation trial is necessary before the manuscript can be presented as a fully validated experimental study. With those data, the architecture provides a clear basis for a reproducible and publishable smart-hydroponics evaluation.

References

- [1] D. R. Hoagland and D. I. Arnon, *The Water-Culture Method for Growing Plants Without Soil*, California Agricultural Experiment Station Circular 347, 1950.
- [2] G. Rajaseger et al., “Hydroponics: current trends in sustainable crop production,” *Bioinformation*, vol. 19, no. 9, pp. 925–938, 2023, doi: 10.6026/97320630019925.
- [3] S. Rajendran et al., “Hydroponics: exploring innovative sustainable technologies and applications across crop production, with emphasis on potato mini-tuber cultivation,” *Heliyon*, 2024.
- [4] A. Hadinata and Mashoedah, “Internet of Things-based hydroponic: literature review,” *Journal of Physics: Conference Series*, vol. 2111, 012014, 2021, doi: 10.1088/1742-6596/2111/1/012014.
- [5] R. Lakshmanan, M. Djama, S. Perumal, and R. Abdulla, “Automated smart hydroponics system using Internet of Things,” *International Journal of Electrical and Computer Engineering*, vol. 10, no. 6, pp. 6389–6398, 2020, doi: 10.11591/ijece.v10i6.pp6389-6398.

- [6] H. J. Muhasin, A. Y. Gheni, N. I. Ismarau Tajuddin, N. A. Izni, Y. Yah Jusoh, and K. Azhar Aziz, “A systematic literature review for smart hydroponic system,” *Bulletin of Electrical Engineering and Informatics*, vol. 13, no. 1, pp. 656–664, 2024, doi: 10.11591/eei.v13i1.4738.
- [7] A. A. Sneineh and A. A. Shabaneh, “Design of a smart hydroponics monitoring system using an ESP32 microcontroller and the Internet of Things,” *MethodsX*, vol. 11, 102401, 2023, doi: 10.1016/j.mex.2023.102401.
- [8] M. S. Tajnin et al., “An AIoT-based hydroponic system for crop recommendation and nutrient parameter monitorization,” *Smart Agricultural Technology*, vol. 8, 100472, 2024, doi: 10.1016/j.atech.2024.100472.
- [9] M. Dutta et al., “Evaluation of growth responses of lettuce and energy efficiency of the substrate and smart hydroponics cropping system,” *Sensors*, vol. 23, 2023.
- [10] Espressif Systems, ESP8266EX Datasheet, version 7.0, Espressif Systems, 2020.
- [11] Infineon Technologies, IRLZ44N HEXFET Power MOSFET Data Sheet, Infineon Technologies.
- [12] Blynk Inc., “Blynk IoT documentation,” <https://docs.blynk.io/> (accessed Sept. 11, 2026).
- [13] A. Al-Fuqaha et al., “Internet of Things: a survey on enabling technologies, protocols, and applications,” *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015, doi: 10.1109/COMST.2015.2444095.
- [14] L. Mainetti, L. Patrono, and A. Vilei, “Evolution of wireless sensor networks towards the Internet of Things: a survey,” in *Proc. 19th International Conference on Software, Telecommunications and Computer Networks*, 2011, pp. 1–6.
- [15] Food and Agriculture Organization of the United Nations, *The Future of Food and Agriculture: Trends and Challenges*, Rome, 2017.