

Design and Implementation of HidroPond: An IoT-Based Information System for Aquaponics Monitoring and Control

Lendy Rahmadi¹, Kusnita Yusmiarti², Heriansyah³

^{1,2,3}Information Systems Department, Faculty of Technology and Science, Lembah Dempo University, Pagar Alam, South Sumatera, Indonesia

Received:

May 19, 2026

Revised:

September 13, 2026

Accepted:

September 30, 2026

Published:

October 24, 2026

Corresponding Author:

Author Name*:

Lendy Rahmadi

Email*:

lendy@lembahdempo.ac.id

DOI:

10.63158/journalisi.v8i5.1934

© 2026 Journal of Information Systems and Informatics. This open access article is distributed under a (CC-BY License)



Abstract. Aquaponics requires timely monitoring and traceable remote control of water and feeding operations. This study designed, implemented, and functionally validated HidroPond, an IoT-based information system for aquaponics monitoring and control. Dedicated Sensor and Actuator ESP32 devices, an Arduino Nano for pH acquisition, a FastAPI backend, a Supabase database, and a Vercel-hosted web application form the architecture. Six parameters (pH, total dissolved solids, water temperature, ambient temperature, humidity, and surface distance as a water-level indicator) were aggregated for transmission at a configured three-minute interval. The actuator supports water filling, draining, pH Up and pH Down dosing, and fish feeding. During a one-day evaluation at a single aquaponics installation, all ten predefined functional checkpoints were verified across telemetry, cloud storage, web monitoring, remote control, device connectivity, and execution feedback. Command-linked feedback distinguishes submitted requests from device-reported outcomes and provides a basis for inspecting unconfirmed operations before further intervention; completion reports do not independently verify water flow, dosing volume, or feed delivery. The contribution is a field-deployed bidirectional information workflow linking telemetry, command records, execution reports, and device connectivity across physical devices and cloud services. Future work will quantify network performance, sensor accuracy, and long-term reliability and separately evaluate AIoT-based predictive control using task-specific outcome data.

Keywords: Aquaponics, Internet of Things, Information System, Real-Time Monitoring, Bidirectional Communication.

1. INTRODUCTION

The combination of aquaculture and hydroponics into one integrated water-based production system is what aquaponics is all about. The system operates by using fish waste to feed the plants and using the plants to purify the water at the same time. The high efficiency and circularity of this system make it a great candidate for sustainable food production, especially when water, land, and food systems are being heavily utilized [1] [2] [3]. However, its successful operation is highly dependent on the maintenance of good water-quality and environmental conditions such as pH, total dissolved solids (TDS), water temperature, water level, ambient temperature, and humidity. Manual checking of conditions at intervals can give insight only to a limited extent of these dynamic factors, leading to the necessity for constant surveillance and fast adjustments in operations.

Internet of Things (IoT) technologies provide a practical means of connecting physical sensors, embedded devices, communication networks, data services, and user interfaces into integrated monitoring environments [4], [5]. In aquaponics, Ali et al. [6] implemented a multi-layer monitoring and control framework that integrates sensing, communication, data storage, web interaction, and remote control, while Naputol et al. [7] combined IoT-based monitoring with an automated feeder. Related smart-agriculture studies have also emphasized field-to-cloud connectivity, data management, interoperability, and analytics as important elements of agricultural IoT infrastructures [8], [9].

Prior work demonstrates IoT sensing, monitoring, remote control, and field-to-cloud information management in aquaponics and related agricultural systems [6]–[9]. The implementation question examined here is how telemetry, user requests, execution reports, and device-connectivity records can be linked and functionally verified as one traceable operational workflow after deployment. HidroPond therefore evaluates continuity between the field installation and the application across upstream monitoring and downstream control, without claiming that the underlying IoT functions are individually new.

Water filling, draining, pH dosing, and feeding affect the shared fish–plant environment. A submitted command and an online device status do not by themselves establish whether a programmed action completed. Linking each request to device-reported

execution and connectivity records provides a basis for inspecting unconfirmed operations before intervening again. This physical–digital traceability supports accountable operation of an aquaponics installation and is relevant to sustainable management; effects on water consumption, fish health, and crop productivity were not measured here.

HidroPond executes this workflow via two ESP32 devices, Sensor and Actuator, connected to a web backend, Supabase cloud database, and web application. Sensor data get delivered in the application, while commands for control are sent to the actuator device alongside the responses of device execution being reported. Presence and heartbeat log connectivity apart from command results. A firmware completion report simply means that the programmed sequence was completed and is different from measurement of water flow, amount dispensed, or feed delivered.

To achieve this, the authors conducted research focused on the creation, implementation, as well as functioning of HidroPond, an end-to-end IoT information system for aquaponics monitoring and remote control. Specifically, one of the goals of this paper is to integrate sensing, edge processing, telemetry, backend services, cloud-based data management system, and real-time web monitoring. Another one is to set up two-way (downward and upward) communication through which one can get the actuator control commands, status information execution, and device connectivity update messages. The third purpose is to implement and functionally test the whole system in the real environment for aquaponics.

The mentioned objectives are tackled with three research questions: (RQ1) How is it possible to integrate sensors (sensing), edge data processing, transmitting data (telemetry), servers/remote services (backend services), storage and management of cloud data, and web monitoring all in a smart aquaponics information system? (RQ2) In which ways can the communication going up, from IoT devices, and going down from the website application enhance the control, actuation, and data transfer functions? (RQ3) How would the combined system perform and be validated in a real life aquaponics setting?.

The research gap identified above concerns the field implementation and validation of traceable links among measurements, application requests, device-reported outcomes, and connectivity records. The stated objectives and RQ1–RQ3 guide HidroPond's design, bidirectional integration, and functional evaluation in an operational aquaponics installation. Its specific integration-level contribution is a field-deployed information workflow connecting physical aquaponics operations with cloud and web records for operator review. The one-day evaluation establishes functional continuity in the observed setting; it does not quantify sensor accuracy, long-term reliability, water flow or dosing volume, or autonomous AIoT performance.

2. METHODS

2.1. Research Design and System Development Workflow

This study followed a design–implementation–deployment–functional validation approach to develop HidroPond as an integrated IoT-based information system for aquaponics monitoring and control. The unit of evaluation was the complete system, encompassing physical sensing and actuation, embedded processing, communication, backend services, cloud storage, and web interaction, rather than an individual component. Integrated sensing and control approaches in aquaponics provide the methodological context for this system-level evaluation [10], [11]

The development process consisted of system requirement identification, architecture design, IoT device development, backend and database engineering, web application development, system integration, pre-deployment functional verification, field deployment, and post-deployment end-to-end validation. Hardware and software components were progressively integrated through the backend API so that each development stage contributed to establishing continuity between physical sensing, telemetry transmission, cloud data storage, web-based monitoring, actuator command delivery, and execution feedback.

The overall research and system-development workflow is presented in Fig. 1. The workflow distinguishes development and component-level verification from field deployment and end-to-end functional validation. The latter was used to evaluate whether the integrated system maintained the intended upstream telemetry,

downstream control, and device-connectivity information flows under actual aquaponics operating conditions.

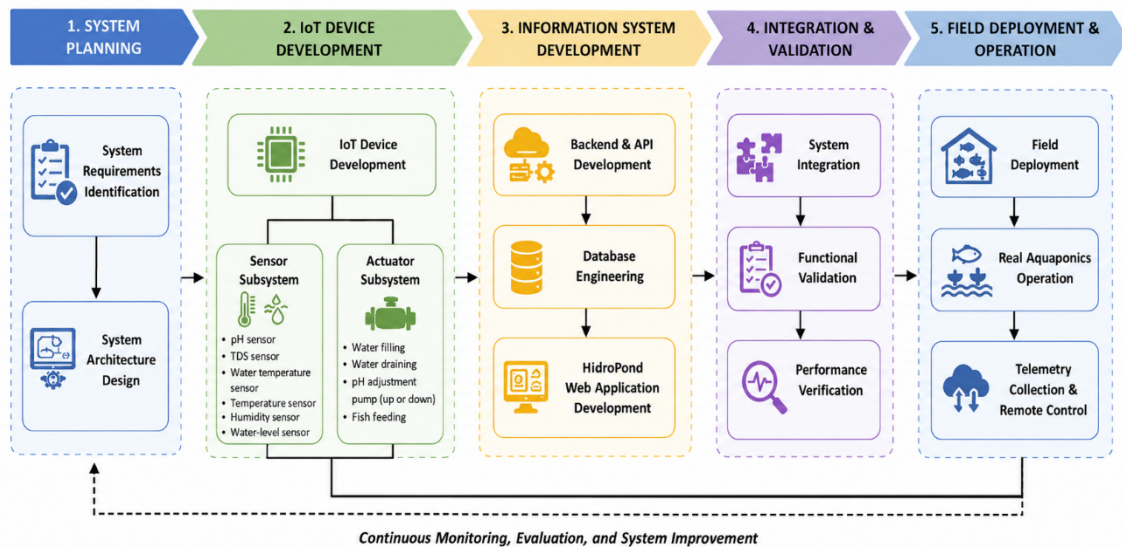


Figure 1. Research and System Development Workflow

The resulting workflow provided the basis for the development of the sensing and actuation devices described in the following subsection. Recent aquaponics IoT implementations have also demonstrated modular architectures integrating sensing, edge or embedded processing, remote communication, cloud-based services, and application-level monitoring, supporting the use of an end-to-end development approach for smart aquaponics systems [12], [13]

2.2. IoT Device Development

The HidroPond IoT infrastructure was developed as the physical interface between the aquaponics environment and the Information-System layer. It consists of separate sensing and actuation devices that acquire environmental and water-quality information, communicate with the backend services, and execute physical control commands. The use of sensing and actuation components for remote aquaponics monitoring and control is consistent with previous IoT-based implementations [10], [11], [13].

2.2.1. Sensor and Actuator Subsystem

The sensing subsystem acquires six operational parameters: pH, total dissolved solids (TDS), water temperature, ambient temperature, humidity, and surface distance as a

water-level indicator. A dedicated Sensor ESP32 serves as the primary acquisition, integration, and Wi-Fi communication controller. The PH-4502C pH sensor is processed through an Arduino Nano and transmitted to the Sensor ESP32 through UART, whereas the TDS, DS18B20, DHT22, and ultrasonic sensors are interfaced directly with the ESP32. An LCD 20×4 I2C provides local presentation of the acquired measurements. These parameters are consistent with those commonly used for IoT-based aquaponics monitoring [10] [12], [14].

At initial installation, the pH sensor was manually calibrated using buffer solutions labelled pH 4.01, 6.86, and 9.18. The TDS sensor was calibrated using solutions labelled 500 ppm and 1,000 ppm NaCl, with reference values specified at 25 °C. Sensor readings were also compared with a reference calibration instrument, and calibration was repeated at two-week intervals during operation. The temperature stated on the solution labels denotes the reference condition for the standards; the temperature during the field calibration was not recorded in this study. Calibration supported routine sensor operation, whereas the reported one-day functional evaluation did not quantify measurement error or long-term sensor drift.

The subsystem that handles actuation has a dedicated Actuator ESP32 which it uses to convert commands from the backend into actual operations of the system. Together with the pump-related features that are relay-based, there are water filling, water draining, pH Up and pH Down dosing functions. In addition, fish-feeding actuation that is carried out by an SG90 servo is also included in the system. These commands are conveyed by the backend API and the Actuator ESP32 then maps them appropriately to the relay or servo channel. It is also true to note that similar remote control pumps and feeding techniques have been put to use in earlier IoT-based aquaponics works [6], [13], [15]. The key hardware parts from both the subsystems are presented in Table 1.

Table 1. Hardware Components of the HidroPond IoT Device

Subsystem	Component	Function / Parameter	Interface / Controller
Sensor	PH-4502C	Water pH	Arduino Nano

Subsystem	Component	Function / Parameter	Interface / Controller
Sensor	DFRobot TDS	Total dissolved solids	Sensor ESP32
Sensor	DS18B20	Water temperature	Sensor ESP32
Sensor	DHT22	Ambient temperature and humidity	Sensor ESP32
Sensor	Ultrasonic sensor	Surface distance (water-level indicator)	Sensor ESP32
Sensor	LCD 20×4 I2C	Local sensor display	Sensor ESP32
Sensor	Arduino Nano	Dedicated pH acquisition	UART to Sensor ESP32
Sensor	Sensor ESP32	Sensor integration and API communication	Wi-Fi
Actuator	Actuator ESP32 + relay module	Command processing and pump switching	Wi-Fi / GPIO
Actuator	Fill, drain, pH Up, and pH Down pumps	Water handling and pH dosing	Relay
Actuator	SG90 servo	Fish-feeding actuation	ESP32 PWM

2.2.2. Integrated IoT Hardware Design

The integrated hardware assigns acquisition, aggregation, and telemetry to the Sensor ESP32, while the Actuator ESP32 handles command polling, pump and servo timing, and execution reporting. Separating these workloads keeps actuator-side processing out of the sensing program loop and allows the two firmware modules to be maintained separately. This supports modularity and limits direct workload interference. Both controllers nevertheless depend on Wi-Fi and backend availability, and this study did not compare their reliability with a single-controller design. Figure 2 presents the sensor, controller, power, and actuator connections used to assemble the field control panel.

The complete hardware configuration is shown in Fig. 2. The schematic defines the principal sensor, controller, actuator, power, and communication connections used as the technical reference for assembling the HidroPond control panel before field deployment. The assembled hardware subsequently served as the field-side infrastructure connected to the backend and application services described in Section 2.3.

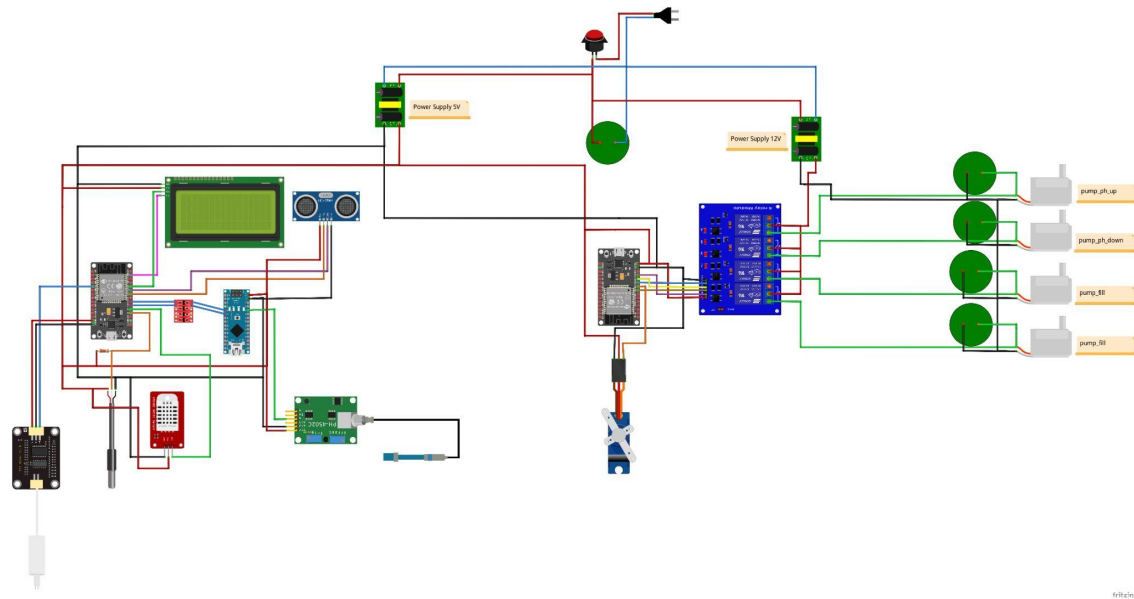


Figure 2. Integrated Hardware Schematic of HidroPond IoT Device

2.3. Backend, API, and Database Engineering

The backend infrastructure was developed as the integration layer connecting the deployed IoT devices, cloud data storage, and the HidroPond web application. An API-centered architecture was used so that physical devices and the user-facing application communicate through backend services rather than directly with one another. Sensor telemetry is received by the backend and persisted in the cloud database, while monitoring information and actuator-control requests are exchanged with the web application through the same integration layer. This separation provides distinct device, communication, data-management, and presentation functions within the overall Information System.

2.2.1. Backend

A FastAPI backend mediates communication among field devices, the Supabase database, and the HidroPond web application. Devices transmit telemetry and retrieve pending

commands through backend endpoints; the application retrieves monitoring data and submits user requests through the same service rather than communicating directly with the devices. Layered communication and cloud-connected services have also been used in related aquaponics platforms [6], [13], [16]. Figure 3 shows the upstream telemetry path, downstream control path, and separate device-status exchange.

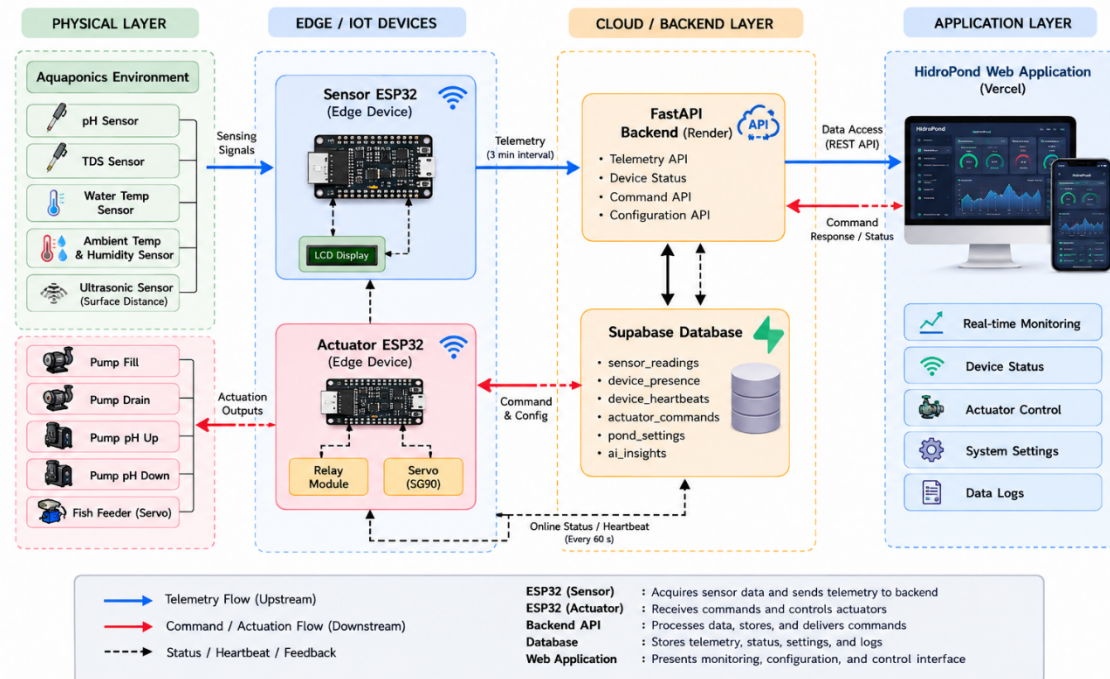


Figure 3. End-to-End Architecture of the HidroPond IoT-Based Information System

2.2.2. API Communication and Telemetry Processing

Real-time monitoring in this paper refers to access on the web to the measurement data that is refreshed based on the telemetry-interval setting; however, it is not to be confused with the hard real-time control type. Real-time (RT) is a system characteristic where the output of a system must meet certain criteria of timeliness. The API allows for telemetry data ingestion, fetching the monitoring data, submitting and fetching commands, reporting the execution-result, and keeping track of the device-connection.

The Sensor ESP32 reads its connected sensors and receives pH measurements from the Arduino Nano through UART. It maintains separate sums and sample counts for each parameter and checks averaged values for upload at the configured 3 min interval. After a valid upload to FastAPI, the aggregate buffer is cleared. A failed upload or a value that

does not satisfy the implemented validity conditions leaves the buffer in memory. A later average may therefore cover more than 3 min. The buffer retains aggregates rather than a persistent queue of individual samples [12], [13].

Downstream control begins when a user submits an actuator request through the web application. FastAPI records the request with a command identifier, and the Actuator ESP32 polls its device-specific endpoint for pending commands. For a supported action, the controller maps the request to the relevant relay or SG90 feeder servo and starts its locally timed sequence. The device reports acceptance when execution begins and completion under the same identifier when the programmed sequence ends. Unsupported actions or actuator identifiers can generate a failure report. Device-presence and heartbeat records are maintained separately from command results. When Wi-Fi connectivity is interrupted, the actuator device suspends new command polling and enters its recovery routine. A connection attempt has a configured timeout of 15 s, followed by a 10 s retry interval; once Wi-Fi becomes available, API health checks are scheduled at 5 s intervals. A successful health check permits polling to resume. These are firmware settings, not measured recovery-time guarantees.

To make a pump and feeder sequence actuator, timing will not block the Actuator ESP32 program loop. The sequences run time is set on firmware level. So, changing the sequence through the server does not influence the Actuator ESP32 timing in this implementation. Once a command is obtained from the server, it is executed by the local loop. If there is communication over the network, one cannot be sure that physical actuators have been actuated, even if the local execution has been completed. An implementation that lacks real-time physical control cannot make physical acts of machines. So in Actuator ESP32 the command is a set of operations which is executed locally until completion, but while it is working, a synchronous network request may get stuck the loop.

Reporting the outcomes of execution also needs a connection. If a result from the Actuator ESP32 does not reach the FastAPI, an Actuator ESP32 has no local retry queue. Thereby, the Actuator ESP32 could just finish a sequence, but the execution report will not be stored in the backend. When connection is restored, however, there is no way to guarantee to retransmit the feedback generated during the disconnect period. Hence, in such cases, a missing backend completion report does not, solely, prove that a physical

actuation has failed. It is likely that the physical actuation has been executed, but there is not much you can do, as the backend has not received and stored the signal.

2.2.3. Database Design and Cloud Deployment

Supabase records data related to time from different devices, for example, presence of devices, heartbeat signals, the status of actuator commands and pond settings. As we can see from the data model in the figure 4, *sensor_readings* stores the measured data with device, pond, and timestamp information; *actuator_commands* links requests to implementation status; and *device_presence* and *device_heartbeats* help check whether the connection is maintained correctly. In addition to these entities, the created db also stores an additional table, *ai_insights*, but in this study the AI-powered decision making and control are not discussed. The stored data in the cloud allows one to get current and historical records through the backend API. [13], [16]

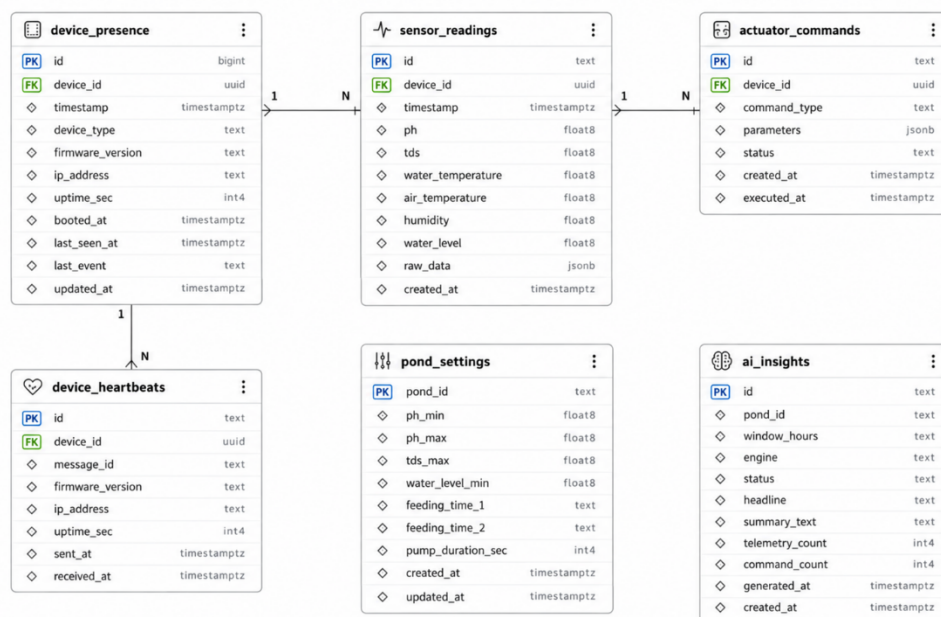


Figure 4. Database Structure

The software services were deployed separately in accordance with their respective roles: Supabase serves as the cloud database provider, the FastAPI backend is deployed on Render, and the HidroPond web app is deployed on Vercel. This arrangement effectively decouples data persistence, communication, presentation, yet retains a seamless integration via the backend API.

2.4. HidroPond Web Application

The HidroPond web application was developed as the user-facing layer of the IoT-based Information System, providing access to current and historical telemetry, device status, operational settings, and remote actuator control. The application communicates exclusively through the backend API rather than directly with the physical sensing or actuation devices. Monitoring information is retrieved through backend services, while user-generated control requests are submitted to the backend and subsequently delivered to the corresponding Actuator ESP32. Similar web- and mobile-based interfaces have been employed in IoT aquaponics systems to provide remote monitoring and control capabilities [6], [13], [15]

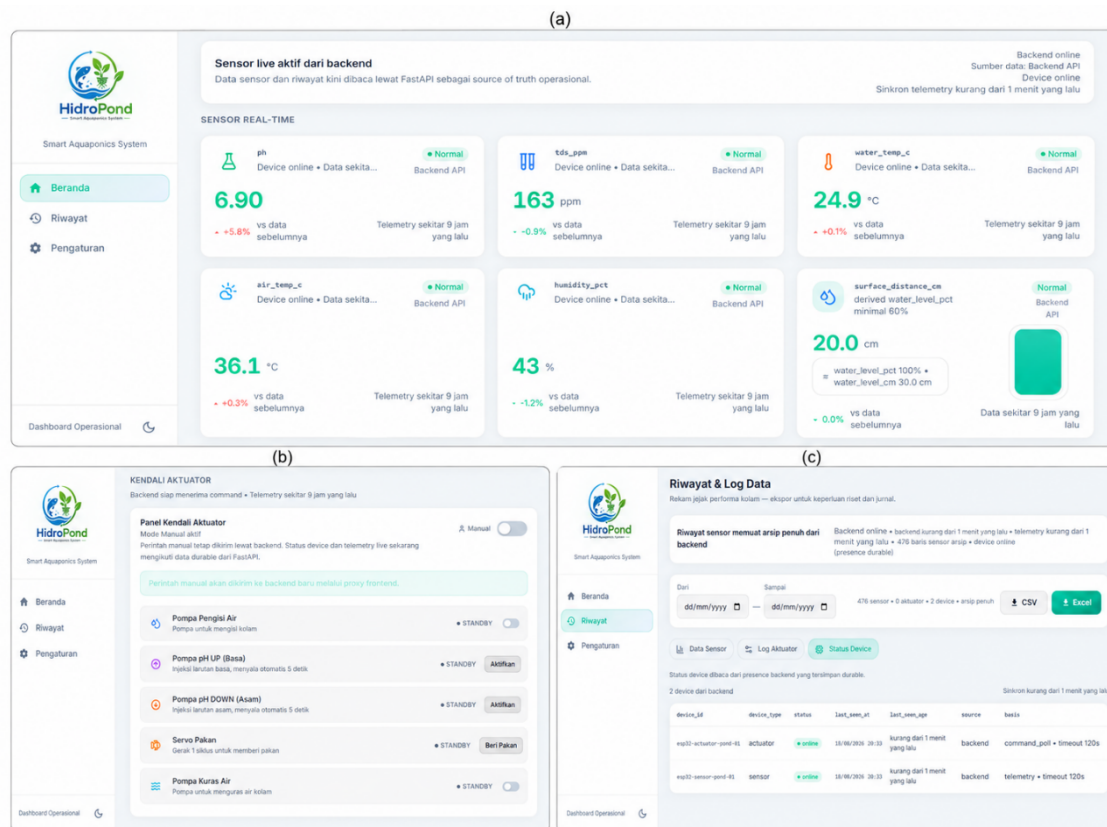


Figure 5. HidroPond Web Interfaces : (a) Monitoring Dashboard, (b) Manual Actuator Control Interface, and (c) Operational History and Device Status.

The principal user interfaces are shown in Fig. 5. The live monitoring dashboard presents the latest water-quality and environmental measurements obtained through the upstream telemetry pipeline (Fig. 5(a)). The manual actuator-control interface enables

users to submit water-management, pH-adjustment, and feeding commands through the backend (Fig. 5(b)), while the operational-history and device-status interface provides access to archived sensor records and device-connectivity information (Fig. 5(c)). Together, these interfaces provide a unified application layer for monitoring, manual remote control, operational traceability, and device observation. Automated machine-learning-based control is not evaluated in the present study and is reserved for subsequent development.

2.5. System Integration and Field Deployment

Following the integration of the IoT devices, backend, database, and web application, the complete HidroPond system was deployed in an actual aquaponics environment. Field deployment was conducted to verify that sensing, communication, cloud data management, web-based monitoring, and physical actuator control could operate as an integrated system under operational conditions rather than only during component-level development [12], [14].



Figure 6. Physical Integration and Field Deployment of the HidroPond IoT System: (a) Integrated Control Panel, (b) Actuator Installation, (c) Sensor Installation, and (d) Operational Field Deployment.

HidroPond was installed at an operational aquaponics installation with a fish-culture tank of approximately 50 cm × 150 cm × 40 cm and a supporting reservoir of approximately 40 cm × 40 cm × 50 cm. The sensor and actuator controllers, relays, communication and power components were assembled in an integrated control panel. Installation included positioning the water-quality probes and ultrasonic sensor, connecting the water-management and pH-dosing pumps and feeder, and establishing communication with the backend. Following installation, the integrated HidroPond system underwent a one-day functional observation period on 18 August 2026. During this field test, sensor acquisition, telemetry transmission, backend and database communication, device connectivity, web-based monitoring, actuator execution, and execution feedback were examined under actual aquaponics operating conditions.

2.6. Functional Testing and Validation

Functional validation was conducted after field deployment to determine whether the hardware and software components operated together as an integrated end-to-end Information System under actual aquaponics conditions. The evaluation focused on three complementary aspects: upstream telemetry, downstream actuator control, and device connectivity. Similar system-level evaluations in IoT-based aquaponics have examined sensing operation, remote data availability, communication, and integrated system behavior under deployed conditions [12], [13], [14]. A functional checkpoint was classified as verified when the expected observable outcome could be confirmed across the corresponding end-to-end pathway during deployed system testing. Verification was based on operational evidence appropriate to each function, including sensor acquisition, backend and database records, application-level monitoring, physical actuator response, command-execution feedback, and device-presence or heartbeat information. The evaluation therefore assessed functional continuity between components rather than quantitative system-performance characteristics.

2.6.1. Upstream Telemetry Validation

Upstream validation traced sensor information from physical acquisition to application-level presentation. Measurements acquired by the sensing subsystem were processed and validated by the Sensor ESP32 before transmission to the Web API. The validation then examined whether the telemetry was successfully received by the backend, persisted in the cloud database, and subsequently retrievable through the HidroPond

web application. Measurements were accumulated and averaged before transmission; when communication failed, the accumulated information was retained for a subsequent transmission attempt.

2.6.2. Downstream Actuator Validation

Downstream validation traced control requests from the application and backend to the Actuator ESP32. The evaluation covered water filling, water draining, pH Up dosing, pH Down dosing, and fish feeding. Verification examined command delivery, the observable pump or servo response, and the associated execution information returned to the backend. Command identifiers linked device reports to the originating requests. Physical observation during the functional test and device-reported execution status were treated as complementary evidence: a firmware completion report indicates that the programmed output sequence has finished, but does not independently quantify water flow, delivered dosing volume, or feed quantity.

2.6.3. Device Connectivity Validation

Device connectivity was observed independently from sensor telemetry and actuator-command records. After network and API connectivity were established, the IoT devices reported their online status and transmitted periodic heartbeat information to the backend. The end-to-end validation was therefore considered functionally successful when sensor information could progress from the physical environment to the web application, control requests could reach and activate the intended physical actuators, execution information could return to the backend, and device connectivity remained observable through presence and heartbeat records. The evaluation was intentionally functional and did not quantify network latency, packet-loss rate, long-term availability, sensor accuracy, or machine-learning and autonomous-control performance, which were outside the scope of the present study.

3. RESULTS AND DISCUSSION

3.1. End-to-End System Implementation Results

The deployed HidroPond system connected field sensing, backend services, cloud storage, and web-based monitoring. Measurements acquired by the sensing subsystem were processed by the Sensor ESP32, transmitted to the FastAPI backend, stored in Supabase,

and made available through the HidroPond web application. Figure 7(a) shows cloud-stored telemetry records, while Fig. 7(b) shows sensor information presented through the web dashboard. Together, these observations demonstrate the implemented upstream information path and address RQ1.

The sensor firmware was configured to upload aggregated measurements at three-minute intervals. This interval refers to how often data is intended to be transmitted rather than the frequency at which the dashboard will be updated for sure. In case of a failed upload or measurements which could not be validated through the implemented validity conditions, the accumulation of data points continued. The average, therefore, could be of a longer duration. It follows from this that the values you are seeing are the most recent summaries that have been successfully received and not, without a doubt, the results of actual measurements right now.

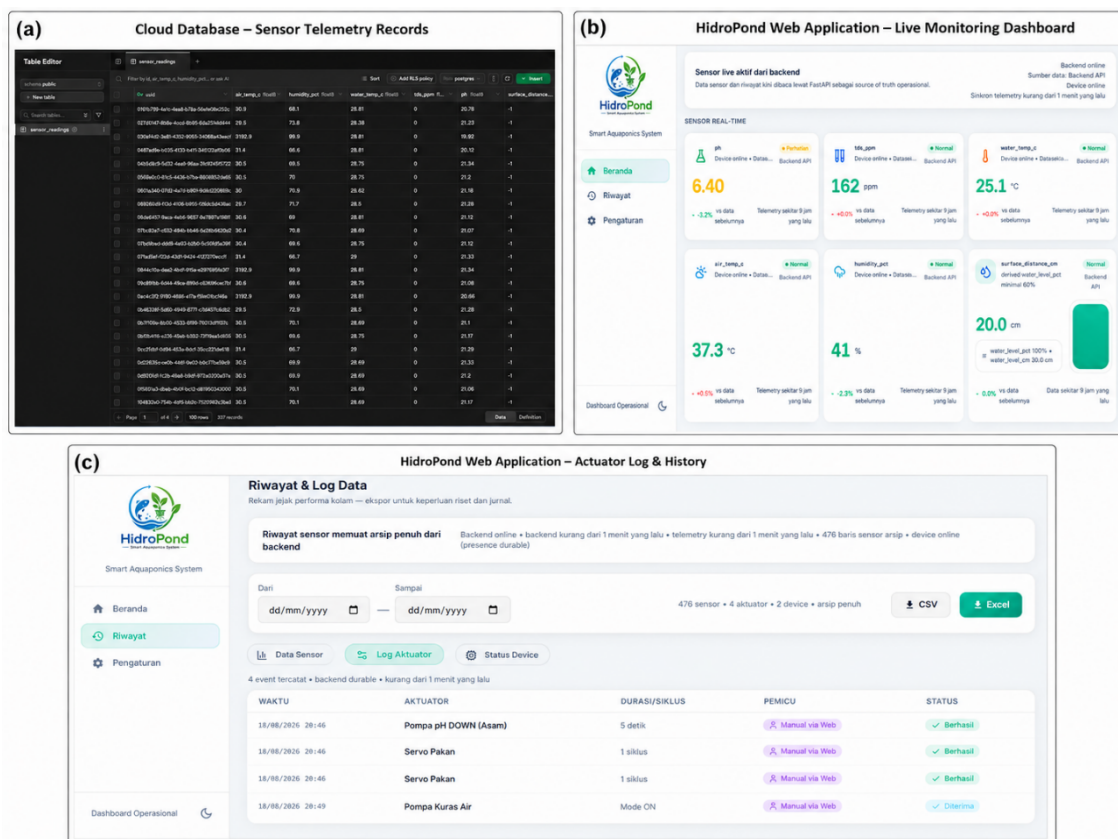


Figure 7. Operational Evidence from HidroPond: (a) Telemetry records Stored in Supabase, (b) Sensor information displayed in the web application, and (c) Actuator command and status records.

3.2. Bidirectional Monitoring and Control Validation

The downstream workflow was also verified. Control requests submitted through the application were made available by the backend to the Actuator ESP32 through polling. The device then performed the programmed pump or servo action and submitted execution information associated with the originating command identifier. Acceptance and completion represent different stages: completion indicates that the firmware finished its programmed output sequence.

Table 2 summarizes the ten predefined functional checkpoints. All were verified during the reported field observation, including sensor acquisition, telemetry transmission, database persistence, web monitoring, device connectivity, command delivery, observable actuator response, and execution feedback. These results establish that the specified functions operated in the tested deployment; they are not a statistical success rate for repeated operation. Figure 7(c) provides application-level evidence of command-related states. Presence and heartbeat records separately describe device connectivity. Together, these results address RQ2 by demonstrating information flow from the aquaponics environment to the application and from the application to field actuators, followed by device-reported feedback.

Table 2. End-to-End Functional Verification Results of the HidroPond System

No.	Validation Item	Observed Result	Status
1	Sensor acquisition	Measurements received by Sensor ESP32	Verified
2	Sensor-data validation	Valid measurements processed before transmission	Verified
3	Telemetry transmission	Telemetry received by backend API	Verified
4	Database persistence	Telemetry stored in Supabase	Verified
5	Web monitoring	Telemetry presented in HidroPond web application	Verified
6	Device connectivity	Presence and heartbeat received by backend	Verified
7	Command transmission	Commands delivered to Actuator ESP32	Verified
8	Actuator execution	Requested pump/servo action executed	Verified

No.	Validation Item	Observed Result	Status
9	Execution feedback	Execution result returned to backend	Verified
10	End-to-end operation	Upstream and downstream functions operated after deployment	Verified

Note: "Verified" indicates that the expected observable outcome was confirmed during the reported functional test. It does not represent a measured reliability rate or independently quantified sensor accuracy.

3.3. Field Deployment and Functional Validation

Field deployment confirmed that the integrated HidroPond system remained operational after installation in the aquaponics environment described in Section 2.5. The deployed sensing subsystem continued to acquire water-quality and environmental measurements, and the resulting telemetry was transmitted through the established backend and cloud-data pipeline for retrieval through the HidroPond web application. The actuator subsystem also remained accessible through the implemented control workflow, with water filling, water draining, pH Up, pH Down, and fish-feeding functions responding to commands and returning execution information to the backend. These results demonstrate that the sensing, communication, monitoring, and physical-control functions established during development were maintained after deployment.

Variation in actuator response waiting time was observed during field operation under differing Wi-Fi conditions. The requested actuator functions remained operational during the documented inspection. However, response time, signal strength, packet loss, and availability were not systematically measured, so the observation does not establish the magnitude or cause of the variation. It identifies a performance question for subsequent testing rather than demonstrating a particular network effect or a guaranteed response time.

Collectively, the field results address RQ3 by demonstrating that HidroPond could be transferred from component-level development to an operational aquaponics installation while maintaining sensing, telemetry transmission, cloud data persistence, web-based monitoring, device connectivity, actuator control, and execution feedback. The validation therefore confirms that the proposed architecture was implemented as a field-deployed IoT-based Information System rather than remaining a conceptual design or isolated laboratory prototype.

3.4. Discussion and Comparison with Related Studies

The results demonstrate that the principal contribution of HidroPond does not originate from the introduction of an individual sensing device, communication protocol, cloud platform, or web interface. Instead, its contribution lies in the implementation of these components as an integrated Information System connecting the physical aquaponics environment with cloud-based information services and remote physical control. The functional results presented in Sections 3.1–3.3 show that this integration was maintained across sensing, telemetry transmission, backend processing, cloud persistence, web-based monitoring, command delivery, actuator execution, and device-level feedback. Recent systematic reviews further indicate that IoT-based aquaponics research increasingly emphasizes integrated monitoring, environmental control, and connected information services [17], [18].

This positioning is important because previous aquaponics studies have already demonstrated several forms of IoT-enabled monitoring and control. Therefore, the contribution of HidroPond should be interpreted in terms of its particular implementation of the physical–digital information flow and operational integration rather than as the first application of IoT or remote control in aquaponics.

3.4.1. End-to-End Information-System Integration

Previous research has established several architectural approaches for connecting aquaponics sensing with remote information services. Reyes-Yanes et al. [10], through their review of automated aquaponics, emphasized the increasing role of sensing, IoT, and smart systems for monitoring and control. Khaoula et al. [11] subsequently proposed an IoT architecture incorporating multiple environmental and water-quality sensors together with actuators for aquaponics monitoring and control. These studies provide an important architectural foundation for the integration of sensing, communication, and automation in aquaponics systems.

More implementation-oriented systems have also been reported. Alselek et al. [12] developed a modular water-IoT monitoring platform integrating multiple sensing parameters with wide-area communication technologies and remote monitoring infrastructure. Wan et al. [13] implemented an end-edge-cloud aquaponics architecture in which sensing and edge-processing components communicate with cloud storage and a

WeChat Mini Program for monitoring and remote control. Their work demonstrates a relatively mature integration of physical sensing, cloud services, data storage, and application-level interaction.

The HidroPond results are consistent with this broader transition from stand-alone sensing toward connected aquaponics platforms. However, the implementation in the present study emphasizes an Information-System architecture in which two dedicated field devices—the Sensor ESP32 and Actuator ESP32—are explicitly separated from the FastAPI integration layer, Supabase data-management layer, and HidroPond web application. Figure 7(a)–(b) provides operational evidence of the upstream information path by showing cloud-stored telemetry and its presentation in the HidroPond web application.

Separating the Sensor ESP32 and Actuator ESP32 provides a division of processing responsibilities. Measurement acquisition, aggregation, and telemetry are handled by the sensing controller, while command polling, output timing, and execution reporting are handled by the actuation controller. This arrangement isolates sensor-side processing from the actuator controller's program loop. It is particularly relevant when actuator-side communication occupies processing time, because those operations do not directly execute within the sensing controller's loop. Shared network and backend dependencies nevertheless remain, so controller separation does not eliminate communication-related interruptions.

3.4.2. Operational Value and Boundary of Execution Feedback

The second key point in the results is that information system and aquaculture system environment are able to communicate each other in a two-way manner. Ali et al [6]., for instance, have developed a platform called AEMOS offering real-time data monitoring, historical data access, remote device control, notifications and rule engine which they have demonstrated that web-based aquaponics systems should only passively monitor but rather can also control up to the application-level. Likewise, Wan et al. [13] have been providing a remote control through anIoT application layer, while Zaini et al. [15] has done smart device-based monitoring and control work for NFTP (Nutrient Film Technique Aquaponics) environment. Consequently, remote actuation features alone will not make HidroPond unique.

The operational value of execution feedback lies in distinguishing a submitted control request from an outcome reported by the field device. When an operator requests water filling, draining, pH adjustment, or feeding, an unconfirmed outcome leaves uncertainty about whether the intended intervention occurred. Command-linked records provide a basis for inspecting that uncertainty before further action is taken. For example, the absence of a completion report can prompt a check of the device or communication path, although the absence alone cannot establish whether physical actuation failed.

This difference is important in aquasonic systems because the handling of water, the giving of feedings, and water treatments constitute actions in a mutual habitat of fish and plants. The availability of a trackable command can assist in these processes' oversight and can help reduce reliance on the idea that a certain request was done. This particular assessment didn't test if these interventions reduced water toxicity, fish nutrition, underfeeding, or resource use, so these should be considered as probable subjects for future operational checking rather than being presented as proven benefits.

Feedback provided has limitations too. Having received a complete signal just means that firmware executed its programmed cycle; it doesn't check water flow, dosing volume, or feed delivery. Likewise presence and heartbeat messages from devices only show if a connection is active, not that an actuator was successfully carried out. The results therefore support a traceable bidirectional information workflow, while detection of mechanically stalled pumps or unsuccessful material delivery would require additional sensing and dedicated failure-condition testing.

3.4.3. Network Conditions and Field Operation

Field deployment provides another important perspective for interpreting the HidroPond results. Alselek et al. [12] progressed beyond conceptual architecture by developing and evaluating a complete aquaponics monitoring platform, while Wan et al. [13] conducted continuous operation and field testing of their modular aquaponics monitoring system. More recently, Ortiz-Arreola et al. [14] experimentally evaluated a remote IoT water-quality monitoring system in a pilot-scale aquaponics installation over an extended operating period, demonstrating the importance of validating IoT monitoring under actual aquaponics conditions. Recent implementation studies have also demonstrated

IoT-based monitoring and closed-loop control in operational aquaponics environments , [19], [20]

The polling workflow makes command response dependent on when the Actuator ESP32 next retrieves a pending request. Wireless connectivity, backend processing, and HTTP exchanges can also affect the time until a command or its resulting status becomes visible. These are plausible contributors to the waiting-time variation observed in the field, but their individual effects were not measured. In particular, a network interruption can delay command retrieval or prevent a completed local action from being reported, because the implemented firmware does not maintain a local retry queue for failed execution-result submissions. Unlike studies that quantified network behavior, such as Wan et al. [13], this evaluation establishes functional operation under the observed conditions without a quantified latency distribution or maximum response-time guarantee.

The waiting time variation that is found can be related to the polling architecture that is in use here. A newly submitted command becomes available to the actuator after the next polling request. However, wireless communication, backend processing and synchronous HTTP requests can also lead to delay. These factors are quite probably responsible for variation in waiting time, but none of their individual impacts were quantitated in this work. Wan et al. [13] did a related work, which quantitatively assessed packet losses, showing thereby a dimension of network performance that has not been considered in our present functional characterization. Therefore, the HidroPond results show that remote-control operation can be realized under the conditions that were observed here without setting a maximal response time or giving a hard real-time execution guarantee.

3.4.4. Position of HidroPond Relative to Existing Systems

To further clarify the position of HidroPond, Table 3 compares selected related studies that are particularly relevant to the architectural, monitoring-and-control, field-deployment, and information-flow characteristics evaluated in this study. The comparison describes the evidence reported for each implementation; it does not rank overall system quality. HidroPond combines web monitoring and remote physical control with command-

linked device reports in a one-day field-validated information workflow. The limited duration of that evaluation should be considered when interpreting the comparison.

Table 3. Documented Monitoring, Control, and Feedback Characteristics of Related Aquaponics Systems.

Study	Documented and Monitoring	Command-linked device Feedback	Reported Validation Context
Ali et al. [6]	Web Monitoring and Remote Control	Not reported	System Implementation
Khaoula et al. [11]	Proposed sensing and actuation architecture	Not reported	Architecture Proposal
Alselek et al. [12]	Remote water monitoring	Not reported	Field evaluation
Wan et al. [13]	End-edge-cloud monitoring and control	Not reported	Field evaluation
Zaini et al. [15]	Mobile monitoring and control	Not reported	Experimental implementation
Ortiz-Arreola et al. [14]	Pilot-scale water monitoring	Not applicable to reported monitoring focus	Extended field evaluation
HidroPond	Web monitoring and remote control of five actuator functions	Command-linked, firmware-reported results	One-day, single-site functional test

Note: "Not reported" means that the cited account did not explicitly document this characteristic; it does not establish that the capability was absent. HidroPond's device-reported result does not independently measure physical delivery.

As summarized in Table 3, HidroPond shares several documented capabilities with previous IoT-based aquaponics systems, including remote monitoring, cloud-based data management, and physical control. These capabilities are therefore not considered standalone novelty. The distinction of the present implementation lies in the explicit

integration of separate sensing and actuation devices, API-mediated communication, persistent telemetry and command records, device-presence and heartbeat mechanisms, execution feedback, web-based interaction, and field deployment within the same operational Information System.

3.4.5. Telemetry Interval and Communication Trade-offs

The configured three-minute aggregation interval reduces the number of upload attempts and stored records relative to transmitting every acquired sample during normal connectivity. The resulting averages summarize operating conditions, but brief changes may be smoothed and their exact timing may be lost. Thus, the interval balances the frequency of communication and records against the temporal detail available to an operator. This study did not compare alternative intervals or measure bandwidth, energy consumption, or anomaly-detection performance.

After a failure to upload the Sensor ESP32 preserves the cumulative totals as well as sample counts for a subsequent resending. Hence, the average that is eventually sent can cover a period of more than three minutes resulting in even less detail in time when a communication interruption occurs. Next versions might retain for instance aggregation-window timestamps and sample counts and also investigate event-triggered reporting besides regular summaries. Such modifications have to be thoroughly tested prior to any assertion of improved detection and response capabilities.

3.5. Threats to Validity and Future Development

The study assessed predefined functions over one day at one aquaponics installation. This supports a conclusion of observed functional integration in that setting, but it does not establish long-term reliability or generalizability to different installations and network conditions. Network latency, packet loss, availability, and actuator-response-time distributions were not quantified. The performance of the implemented reconnection procedure also requires controlled interruption tests.

In line with the pH and T DS manual measurement procedures, the calibration was also done, but no quantitative sensor-accuracy assessment of a benchmark measurement against a standard reference device or determination of a drift in time over long-term was performed. Similarly, the results of pump and servo operation as well as software

reports do not tell how much water, dosing solution, or feed were actually supplied. Therefore, in longer experiments repeated system and actuator testing together with comparison measurements drift estimation, and direct observation of the relevant physical results should be carried out.

The network characteristics like latency, packet-loss rate, availability or actuator response-time distribution were not quantified at all. As a result, the fluctuation waiting times that were noticed is a qualitative field observation only. The actuator firmware although has automatic reconnection but, the effectiveness of its recovery from a network breakdown in a systematic test was not evaluated, as the network was interrupted for a short time to see if the system would get recovery in the described assessment. Besides, the fact that the system does not maintain a list of unconfirmed transactions to submit later locally means that an actuation may locally finish without the result being known to the center backend.

Preliminary studies have demonstrated that measurement validity can be a limitation in the evaluation of sensor performance. The pH and TDS sensors were manually calibrated on the first day after their installation and every other week by using calibration solutions and also the data from an independent reference instrument. However, the reported study did not involve independent quantitative sensor-accuracy benchmarking or check for sensor drift over several months. It can be seen that regular calibration is only a method for carrying out the measurement procedures. It does not, in any way, establish or quantify sensor accuracy or long-term stability of the equipment across different environmental conditions.

Moreover, physical observability of execution reports was limited. Firmware-based completion indicates that the programmed output sequence has been completed, instead of actually measuring water flow, dosing volume, feed delivery. The study did not look into areas such as mechanical-failure detection, dosing effectiveness, fish-health, plant productivity, or resource savings. The above-listed outcomes are only examples of possible areas for study that would require specialized measurements methods and study designs, rather than a mere technical functional integration assessment.

Related hydroponic studies provide methodological starting points for future predictive analysis. Rahmadi et al. [21] compared ARIMA and Prophet for hydroponic lettuce growth simulation and implemented the selected Prophet model in a Streamlit application. A subsequent study compared support vector machines, k-nearest neighbors, random forests, and gradient boosting for predicting lettuce growth patterns in NFT hydroponic systems [22]. These studies motivate the evaluation of alternative predictive approaches for HidroPond; however, their findings require separate validation under aquaponic conditions.

Future AIoT development can build on the historical telemetry and command records maintained in Supabase. Data preparation should address invalid or missing measurements, variable aggregation periods, and timestamp alignment between observations and actuator events. Growth prediction would additionally require corresponding plant-growth measurements as target data. Prediction targets or control-decision labels must be defined independently of firmware completion status, which does not establish that an intervention was agronomically appropriate. Candidate models should be evaluated using chronological data separation and appropriate baselines before controlled integration with the actuator interface. Machine-learning performance and autonomous control were not evaluated in the present study.

4. CONCLUSION

This study designed, implemented, and functionally validated HidroPond as an IoT-based information system for aquaponics monitoring and remote control. The integration of dedicated Sensor and Actuator ESP32 devices, a FastAPI backend, Supabase storage, and a web application established a traceable bidirectional information workflow linking field measurements, control requests, device connectivity, and device-reported execution outcomes. All ten predefined functional checkpoints were verified during a one-day observation at a single aquaponics installation. The principal contribution is the operational linkage between monitoring information and command-linked feedback, which allows operators to distinguish a submitted request from a device-reported completion. Firmware completion does not independently verify water flow, dosing volume, or feed delivery. The evaluation also did not establish network latency, sensor accuracy or drift, long-term reliability, productivity improvement, or autonomous-control

performance. Future work should extend field evaluation, strengthen delivery of execution reports during network interruptions, and assess predictive models using independently measured outcomes before their controlled integration with actuation.

ACKNOWLEDGMENT

The authors thank the Ministry of Higher Education, Science, and Technology of the Republic of Indonesia for funding this research, Universitas Lembah Dempo for institutional support, and the technical staff for assistance with the HidroPond installation and field evaluation.

REFERENCES

- [1] A. Faisal *et al*, "Towards sustainable aquaponics: A bibliometric analysis of trends, scientific developments, and thematic evolution," *Aquac. Int*, vol. 33, no. 6, 2025, doi: 10.1007/s10499-025-02100-x.
- [2] C. Bibbiani, A. Campiotti, L. Incrocci, A. Pardossi, B. Fronte, and C. Viola, "Aquaponic as sustainable innovation for food production," *Riv. Studi Sostenib*, no. 2, pp. 249–258, 2017, doi: 10.3280/RISS2016-002022.
- [3] N. Mchunu, G. Lagerwall, and A. Senzanje, "Aquaponics in South Africa: Results of a national survey," *Aquac. Rep*, vol. 12, pp. 12–19, 2018, doi: 10.1016/j.aqrep.2018.08.001.
- [4] P. D. Deshpande, "Revolutionizing intelligent agriculture using Industry 5.0 and Society 5.0," in *Industry 5.0 for Society 5.0: Revolutionizing Smart Farming, Manufacturing, and Green Computing (Part 2)*, 2025, pp. 22–38. doi: 10.2174/9798898810849125010005.
- [5] K. Jabroot, H. Rohilla, D. Saxena, and R. Kazimi, "The role of IoT in promoting sustainability in green agriculture," in *Smart Agriculture for Sustainable Practices*, 2025, pp. 143–160. doi: 10.1201/9781003594093-10.
- [6] M. E. Ali, J. K. Sultan, and A. Hafeez, "A smart monitoring and observation framework for aquaponics ecosystem," *ARPN J. Eng. Appl. Sci*, vol. 19, no. 13, pp. 864–869, 2024, doi: 10.59018/072417.

- [7] A. T. Naputol, J. I. Ituriaga, B. M. Yu Jeco-Espaldon, G. Guanzon, K. Co, and M. G. Giner, "Development of an IoT-based aquaponics monitoring system with automated feeder," in *2024 IEEE Symposium on Industrial Electronics & Applications (ISIEA)*, 2024, pp. 1–6. doi: 10.1109/ISIEA61920.2024.10607265.
- [8] R. Rajora, A. Rajora, R. Singh, and R. Gupta, "IoT integration in agricultural infrastructure: From fields to clouds," in *2023 International Conference on Research Methodologies in Knowledge Management, Artificial Intelligence and Telecommunication Engineering (RMKMATE)*, 2023, pp. 1–5. doi: 10.1109/RMKMATE59243.2023.10369026.
- [9] S. Tamta and V. Vimal, "Integrating IoT-based environmental monitoring and data analytics for crop-specific smart agriculture management: A multivariate analysis," in *2023 3rd International Conference on Technological Advancements in Computational Sciences (ICTACS)*, 2023, pp. 368–373. doi: 10.1109/ICTACS59847.2023.10390277.
- [10] A. Reyes-Yanes, P. Martinez Rodriguez, and R. Ahmad, "Towards automated aquaponics: A review on monitoring, IoT, and smart systems," *J. Clean. Prod.*, vol. 263, 2020, doi: 10.1016/j.jclepro.2020.121571.
- [11] T. Khaoula, R. Ait Abdelouahid, I. Ezzahoui, and A. Marzak, "Architecture design of monitoring and controlling of IoT-based aquaponics system powered by solar energy," *Procedia Comput. Sci.*, vol. 191, pp. 493–498, 2021, doi: 10.1016/j.procs.2021.07.063.
- [12] M. Alselek, J. M. Alcaraz-Calero, J. Segura-Garcia, and Q. Wang, "Water IoT monitoring system for aquaponics health and fishery applications," *Sensors*, vol. 22, no. 19, 2022, doi: 10.3390/s22197679.
- [13] S. Wan, K. Zhao, Z. Lu, J. Li, T. Lu, and H. Wang, "A modularized IoT monitoring system with edge-computing for aquaponics," *Sensors*, vol. 22, no. 23, 2022, doi: 10.3390/s22239260.
- [14] J. Ortiz-Arreola *et al.*, "Real-time water quality monitoring system in an aquaponics pilot culture," *Appl. Sci.*, vol. 16, no. 11, 2026, doi: 10.3390/app16115638.
- [15] A. Zaini, A. Kurniawan, and A. D. Herdhiyanto, "Internet of Things for monitoring and controlling nutrient film technique (NFT) aquaponic," in *2018 International Conference on Computer Engineering, Network and Intelligent Multimedia (CENIM)*, 2018, pp. 167–171. doi: 10.1109/CENIM.2018.8711304.

- [16] S.-C. Wang, W.-L. Lin, and C.-H. Hsieh, "To improve the production of agricultural using IoT-based aquaponics system," *Int. J. Appl. Sci. Eng.*, vol. 17, no. 2, pp. 207–222, 2020, doi: 10.6703/IJASE.202005_17(2).207.
- [17] M. A. H. B. Zamnuri, et al "Integration of IoT in Small-Scale Aquaponics to Enhance Efficiency and Profitability: A Systematic Review," *Animals*, vol. 14, no. 17, 2024, doi: 10.3390/ani14172555.
- [18] P. Chandramenon, A. Aggoun, and F. Tchienbou-Magaia, "Smart approaches to Aquaponics 4.0 with focus on water quality—Comprehensive review," *Comput. Electron. Agric.*, vol. 225, 2024, doi: 10.1016/j.compag.2024.109256.
- [19] M. M. M. Mahmoud, R. Darwish, and A. M. Bassiuny, "Development of an economic smart aquaponic system based on IoT," *J. Eng. Res.*, vol. 12, no. 4, pp. 886–894, 2024, doi: 10.1016/j.jer.2023.08.024.
- [20] J. A. C. Jose, et al. "An automated small-scale aquaponics system design using a closed loop control," *Environ. Chall.*, vol. 19, 2025, doi: 10.1016/j.envc.2025.101127.
- [21] L. Rahmadi, Hadiyanto, and R. Sanjaya, "Automated hydroponic growth simulation for lettuce using ARIMA and Prophet models during rainy season in Indonesia," *Int. J. Adv. Comput. Sci. Appl.*, vol. 15, no. 11, pp. 383–402, 2024, doi: 10.14569/IJACSA.2024.0151137.
- [22] L. Rahmadi, Hadiyanto, and R. Sanjaya, "Prediction of hydroponic lettuce growth patterns using comparative machine learning approaches in hydroponics NFT systems," *J. Eur. Syst. Autom.*, vol. 58, no. 3, pp. 449–466, 2025, doi: 10.18280/jesa.580304.