Development of a Self-Diagnostic IoT-Based Aquaculture Monitoring System Using Raspberry Pi

Danny Chang Jun Wei
Faculty of Electronic Engineering Technology
Universiti Malaysia Perlis
Arau, Perlis, Malaysia
dannychang345@gmail.com

Asral Bahari Jambek
Faculty of Electronic Engineering Technology
Centre of Excellence for Micro System Technology
Universiti Malaysia Perlis
Arau, Perlis, Malaysia
asral@unimap.edu.my

Mohd Yusri Azli Mohd Yusof
Karamas Global Trading
Kangar, Perlis, Malaysia
yusri779@gmail.com

Mazlan Abbas
FAVORIOT Sdn Bhd
Puchong, Selangor, Malaysia
mazlan@favoriot.com

***Abstract*— This paper presents the design and development of a self-diagnostic Internet of Things (IoT) system using a Raspberry Pi for real-time environmental monitoring in aquaculture farming. The system integrates multiple sensors, including pH, total dissolved solids (TDS), water temperature, water level, ambient temperature and humidity sensors. The measured data is continuously collected and transmitted to Favoriot, an IoT platform. A key feature of this design is its built-in diagnostic capability, which detects sensor or communication failures before triggering either self-recovery actions or user alerts. The system is programmed in Python and includes scheduled monitoring, sensor calibration, and automatic data logging. These diagnostic functions help identify hardware and software faults, thus improving overall system reliability. Based on the seven-day experimental results, the proposed design is capable of monitoring the environment effectively, performing self-diagnostics and providing timely feedback to the user.**

Keywords— Internet of Things (IoT), Raspberry Pi, Aquaculture Monitoring, Self-Diagnostic System, Reliability

I. Introduction

IoT systems play a crucial role in modern aquaculture by enabling real-time monitoring, automation, and data-driven decision-making [1]. These technologies help maintain water quality while reducing labour and operational costs. The scalability and adaptability of IoT systems support various farm sizes and improve yields in today's competitive market [2].

Reliability and availability are key to successful IoT and automation deployment. The system must achieve high availability, whereby it can detect faults, isolate issues, perform self-recovery or recalibration, and send real-time alerts. This requires fully automated and reliable fault-handling processes, with potential hardware and communication redundancy [3].

This work focuses on developing a self-diagnostic IoT system for aquaculture. The proposed system integrates multiple sensors to continuously monitor water quality and environmental conditions, using a Raspberry Pi as the main controller and logging data to Favoriot, an IoT platform [4].

The rest of the paper is organised as follows. Section II reviews related work, while Section III outlines the research methodology used in this paper. Section IV presents experimental results and Section V concludes the paper.

II. Literature Review

Various researchers have explored self-diagnostic techniques to enhance system resilience in cyber-physical and sensor-based systems. The authors in [5] proposed a fault detection method using logical comparisons between expected (E) and observed (O) outputs. The system identifies faults when $E \neq O$, using Answer Set Programming (ASP) to formalise reasoning. While robust, the method is computationally intensive, making it unsuitable for microcontroller-level IoT devices.

In [6], a self-monitoring approach was introduced using logistic regression, artificial neural networks (ANNs), and support vector machines (SVMs). It achieved fault prediction accuracy of up to 99.7% and could estimate Remaining Useful Life (RUL). However, its reliance on extensive historical data and high computational demand limits its use in real-time embedded systems.

The authors in [7] focused on monitoring sensor wear via live resistance measurement ($R = V/I$). Deviations from expected resistance trigger alerts before performance degrades. Though effective, this method is limited to resistive sensors. In [8], a Self-Diagnostic Index (SDI) was developed to assess sensor health based on sensitivity loss. The system supports remote diagnostics via IoT, making it suitable for online aquaculture monitoring. However, its accuracy depends on stable network conditions and consistent signal acquisition.

The authors in [9] combined AI-based surrogate models with IoT water quality sensors to predict fish feeding behaviour and automate management. The system uses dissolved oxygen, pH, turbidity, and temperature data, applying Bayesian optimisation and deep learning to achieve

Authors' Final Manuscript.
Published paper link: https://doi.org/10.1109/ICSyS65149.2025.11379905

an R^2 of 0.94. While highly integrated, it requires extensive data training and lacks emphasis on offline reliability.

The paper [10] describes a microcontroller-based aquaculture monitoring system using Raspberry Pi or Arduino, integrated with sensors for pH, temperature, turbidity, and dissolved oxygen. Thresholds trigger automatic responses such as motor control or user alerts. Sensor data is transmitted via Wi-Fi to cloud platforms, with MATLAB used for visualisation and decision support.

The authors in [11] present an IoT system using NodeMCU for fish pond monitoring, featuring sensors for pH, temperature, and turbidity. Data is displayed via Android apps and Blynk integration. The system offers good accuracy and cost-effectiveness, making it suitable for rural use.

While existing methods mainly focus on collecting and analysing sensor data, few studies have addressed the development of reliable and dependable IoT systems for aquaculture [12]. Since this aspect is crucial for large-scale IoT deployment, this paper presents our approach to achieve it. The following section outlines our design methodology.

III. METHODOLOGY

Fig. 1 shows the overall block diagram of the proposed system. The system aims to perform automatic water monitoring for freshwater Australian red claw crayfish (*Cherax quadricarinatus*). The system comprises multiple sensors connected to a controller, actuators, IoT cloud and user alert system. Sensor data is read by the controller before it is sent to an IoT Cloud. In this work, we use an IoT platform called Favoriot [4]. Based on the analysed data, the control unit activates actuators (pumps) as needed.

Fig. 2 presents a 3D design illustrating the placement of the system's sensors, controller, and actuators, including the water tank, piping, and wiring. The full system connection schematic is shown in Fig. 3, while details of the device models are summarised in Table 1.

The system's software framework is built on Raspberry Pi OS Lite for its lightweight and stable performance on embedded hardware. Python 3 is used as the main programming language due to its strong support for GPIO control, HTTP requests, and JSON formatting, making it suitable for IoT applications. Each sensor or actuator is programmed in a separate script to enable modularity, easier debugging, and faster maintenance. If a reading remains invalid, the system retries up to three times with short delays and computes an average to reduce false alerts. Valid data is sent to Favoriot platform, while failed attempts are logged locally in JSON files to prevent data loss. When internet connectivity is restored, the system automatically pushes the data to the cloud sequentially.

Real-time notifications are sent via Telegram using the Bot API, including timestamps and sensor-specific details when abnormal conditions or failures are detected. For I2C-based sensors, a diagnostic routine performs an I2C address scan to verify hardware connectivity during failures. The CRONTAB command schedules script execution at fixed intervals.

To ensure reliable 24/7 data collection, each sensor is individually calibrated and verified before full system integration. The sensors calibration methods follow the procedures as recommended in the sensor manufacturers' datasheets. Sensor data is evaluated against preset thresholds values as recommended by local the aquaculture farmers as follows: pH range is 28 to 29 °C, maximum TDS value of 600 ppm and water level drop of maximum 50% from initial value. When values fall outside these ranges, the diagnostic routines will be triggered. Next, notifications and diagnostic results are sent to Favoriot platform, and alerts are delivered to users via Telegram in real time. If a sensor error is detected, the system performs internal diagnostics to isolate any hardware or communication failures. This includes performing I2C address scanning and automatic retries in reading the sensors.

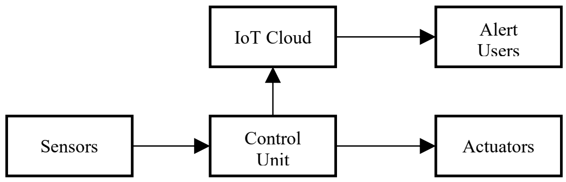

Fig. 1 Block Diagram of Full System`

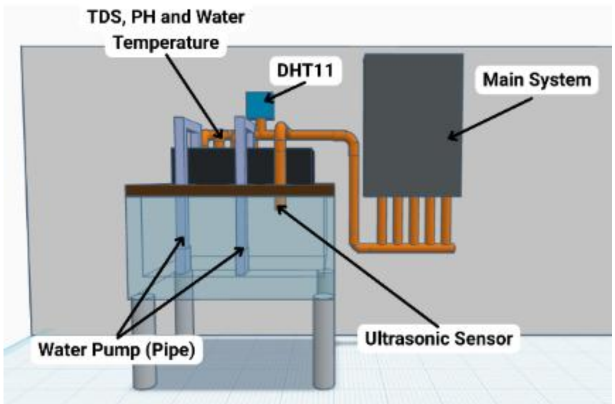

Fig. 2 Front View of 3D Design and Labelling

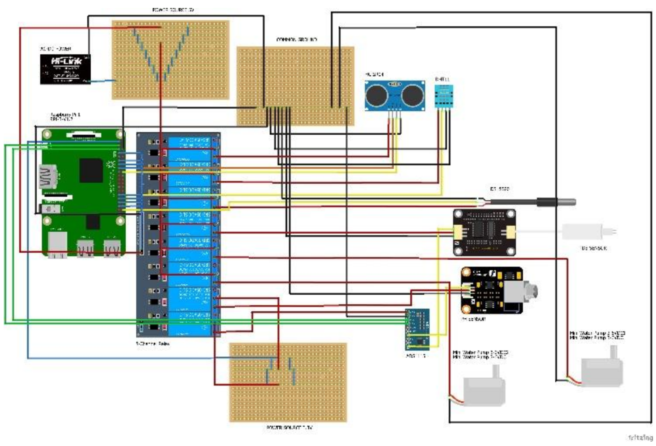

Fig. 3 Full Connection diagram

TABLE I. LIST OF COMPONENTS USED IN THIS PROJECT

Device	Model
Microcontroller	Raspberry Pi 3B
Analog to digital converter	ADS1115
Surrounding temperature and humidity	DHT11
Digital water temperature	DS18B20
Ultrasonic sensor to measure water level	HC-SR04
pH sensor	SEN0161-V2
TDS sensor	Keyestudio TDS Meter V1.0
Relay	8-channel relay module
Power supply adapter	240V AC to 5V DC.

IV. Results and Discussion

This section presents the results of testing the self-diagnostic IoT system in the simulated aquaculture environment as discussed in the previous section. After calibrating all the devices against commercial equipment, all components were assembled on an A3-sized acrylic board, as shown in Fig. 4. To ensure strong and reliable connections, pin headers and jumper wires were secured using hot glue and heat shrink tubing. Screw terminals were used for sensors requiring long or movable wiring, especially those controlled by relays, to help prevent disconnections during operation.

With the sensor arrangement shown in Fig. 2, the system was operated continuously for seven days in a laboratory with an automatic air conditioner. Sensor readings were recorded while the water container was filled with tap water. The pH, TDS, and water temperature sensors were submerged, and the ultrasonic sensor was placed 3 cm above the water surface.

Fig. 5 to 12 show sensor readings during this period. Fig. 5 shows the surrounding temperature (as recorded by the DHT11 sensor) and water temperature readings (from the DS18B20). Due to a temporary power interruption, the surrounding temperature and humidity recorded a single zero-value reading which is considered as an outlier. Based on the data, the water temperature sensor closely followed ambient changes in the lab where it dropped to 18 °C when the air conditioner was active starting at 8 am, and rose to 28 °C when the air-conditioner was inactive after 6 p.m.

Fig. 6 shows the TDS sensor readings. The values gradually varied between 330 ppm and 390 ppm, following the water temperature trend. This indicates that temperature fluctuations affect TDS readings and are inline as discussed in [13]. Fig. 6 also presents the water level readings from the ultrasonic sensor which is placed 3 cm above the water surface. The sensor reported values mostly between 2–3 cm, with occasional spikes above 4 cm. These anomalies may be caused by air bubbles or surface ripples in the tank. Fig. 7 shows the pH sensor readings of the water tank, which remained consistent between 7.6 and 7.9, a typical range for tap water used in this experiment [14].

In this work, Favoriot platform was successfully integrated with Raspberry Pi and Telegram for real-time alerts. Alert rules were configured based on crayfish farmer requirements as discussed in the previous section. The system notifies users when sensor values exceed these thresholds. Fig. 8 shows the rule setup on the Favoriot dashboard.

After seven days of testing, alerts were consistently sent to Telegram upon abnormal readings. The Favoriot dashboard was designed with six graphs: water level, TDS, pH, water temperature, and surrounding temperature and humidity. These graphs display real-time data from the system partially shown in Fig. 9.

To enhance system reliability and availability, a built-in diagnostic module was implemented for all sensors. This module is triggered only when a sensor records an extreme value, suggesting a possible malfunction. To test this feature, sensor thresholds were intentionally adjusted to activate diagnostics. The module was tested on both I2C and non-I2C sensors by setting threshold values to simulate abnormal conditions.

For I2C-based sensors such as the TDS and pH probes, thresholds were deliberately set to extreme ranges, i.e. TDS values between 5000 and 6700, and pH values that fall outside the recommended range, i.e. either smaller than 4.0 or greater than 9.0. When such values were detected, the diagnostic routine first performed an I2C address scan to verify communication, as shown in Fig. 10. If the sensor was detected, the system proceeded with three successive readings. If values remained outside acceptable bounds, an alert was sent by Favoriot via Telegram, as shown in Fig. 11.

For non-I2C sensors such as the ultrasonic sensor, DHT11, and DS18B20, the I2C check was bypassed. Instead, the system directly performed three readings and calculated the average. If abnormal readings persisted, the diagnostic module triggered a Telegram alert to Favoriot indicating a potential hardware issue or environmental anomaly, as shown in Fig. 12. The results confirm that the system can perform self-diagnostics for both I2C and non-I2C sensors, enabling effective fault monitoring.

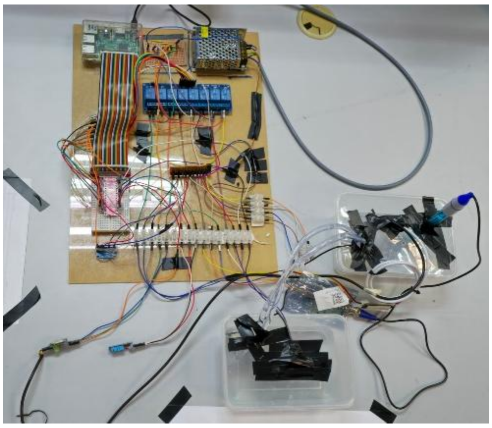

Fig. 4 Full system circuit connection

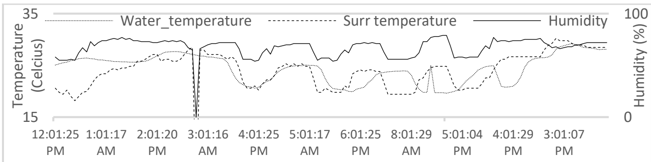

Fig. 5 Results for Surrounding Temperature, Water temperature and Humidity

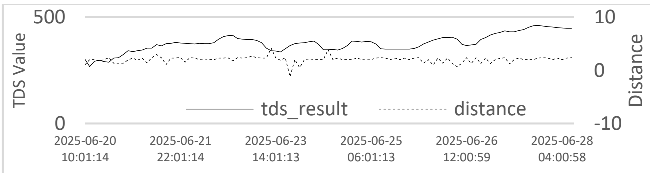

Fig. 6 Result for TDS Sensor and Ultrasonic Sensors

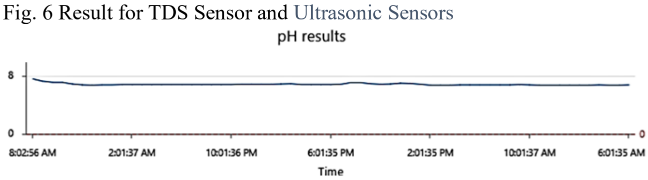

Fig. 7 Result for pH sensor

Activated	Rule Name	Method	Device Developer ID	Action
	High_PH	TELEGRAM	Ph_Sensor@udangkara1	
	Low_PH	TELEGRAM	Ph_Sensor@udangkara1	
	Low_TDS	TELEGRAM	TDS@udangkara1	
	High_TDS	TELEGRAM	TDS@udangkara1	
	Low_Temperature	TELEGRAM	WaterTemperature@udangkara1	

Fig. 8 Rules Configuration

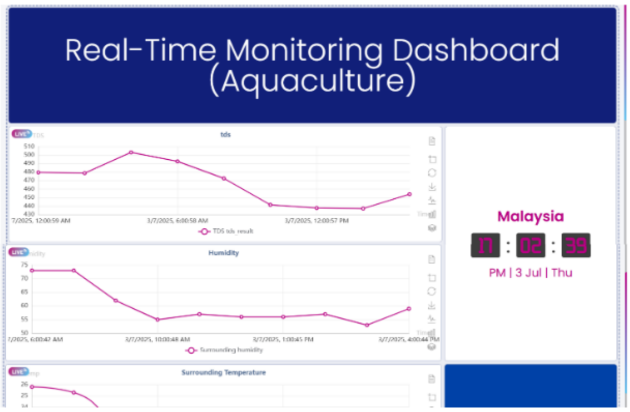

Fig. 9 Favoriot Monitoring Dashboard

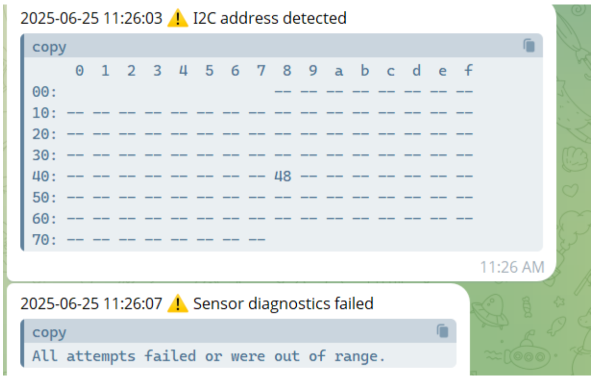

Fig. 10 Message for I2C Sensors

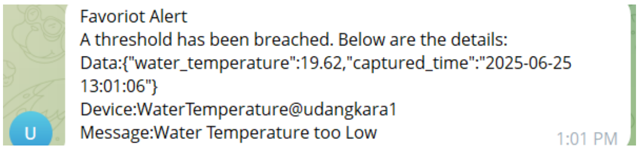

Fig. 11 Message Informed Through Telegram via Favoriot

V. Conclusion

This paper presents the development of a self-diagnostic IoT system using a Raspberry Pi for real-time monitoring in aquaculture. The system integrates multiple low-cost sensors with the Favoriot platform and includes features such as Telegram alerts, offline data logging, and automated diagnostics to enhance reliability. It effectively addresses issues like sensor failures and internet disconnection, ensuring minimal downtime. Testing confirmed the system's ability to detect anomalies, retry data collection, and promptly alert users. Hardware reliability was improved through secure connections for long-term stability. The system proves highly suitable for reducing manual effort and improving water quality control in aquaculture. Further improvements may include using higher-grade sensors, integrating solar power and battery backup to prevent outages, and applying machine learning for predictive analytics to enhance reliability and availability.

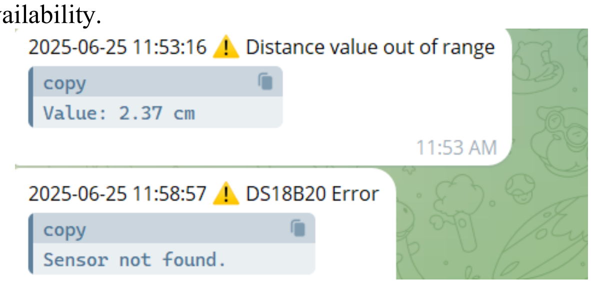

Fig. 12 Message for Non I2C Sen*sors*

References

[1] H. Rastegari et al., "Internet of Things in aquaculture: A review of the challenges and potential solutions based on current and future trends," Aug. 01, 2023, Elsevier B.V. doi: 10.1016/j.atech.2023.100187.

[2] M. G. C. Emerenciano, A. Miranda-Baeza, M. Martínez-Porchas, M. A. Poli, and F. do N. Vieira, "Biofloc Technology (BFT) in Shrimp Farming: Past and Present Shaping the Future," Front Mar Sci, vol. 8, Dec. 2021, doi: 10.3389/fmars.2021.813091.

[3] D. G. . Raheja and L. J. . Gullo, Design for Reliability. John Wiley & Sons, 2012.

[4] Favoriot Platform, [Online]. Available: https://www.Favoriot.com/

[5] D. Kaufmann, I. Nica, and F. Wotawa, Intelligent Agents Diagnostics—Enhancing Cyber - Physical Systems with Self - Diagnostic Capabilities," Advanced Intelligent Systems, vol. 3, no. 5, May 2021, doi: 10.1002/aisy.202000218.

[6] W. Kim, C. Lim, and J. Chai, "Development of a sdms (Self-diagnostic monitoring system) with prognostics for a reciprocating pump system," Nuclear Engineering and Technology, vol. 52, no. 6, pp. 1188–1200, Jun. 2020, doi: 10.1016/j.net.2019.12.001.

[7] F. Mazzoli, D. Alghisi, and V. Ferrari, "Self-Diagnostic and Self-Compensation Methods for Resistive Displacement Sensors Tailored for In-Field Implementation †," Sensors, vol. 24, no. 8, Apr. 2024, doi: 10.3390/s24082594.

[8] D. H. Jung, H. J. Kim, J. Y. Kim, S. H. Park, and W. J. Cho, "Water nitrate remote monitoring system with self-diagnostic function for ion-selective electrodes," *Sensors*, vol. 21, no. 8, Apr. 2021, doi: 10.3390/s21082703.

[9] M. C. Chiu, W. M. Yan, S. A. Bhat, and N. F. Huang, "Development of smart aquaculture farm management system using IoT and AI-based surrogate models," J Agric Food Res, vol. 9, Sep. 2022, doi: 10.1016/j.jafr.2022.100357.

[10] Godugolla Karthik, Puvvada Rupa Sri Sai, Meenapalli Sree Vaishnavi, and Kulluru Deepika, "Fully Automated Aqua Farm Monitoring System," International Journal of Advanced Research in Science, Communication and Technology, pp. 300–307, Feb. 2022, doi: 10.48175/ijarsct-2550.

[11] M. I. Bachtiar, R. Hidayat, and R. Anantama, "Internet of Things (IoT) Based Aquaculture Monitoring System," MATEC Web of Conferences, vol. 372, p. 04009, 2022, doi: 10.1051/matecconf/202237204009.

[12] A. Čolaković, "IoT systems modeling and performance evaluation," Nov. 01, 2023, Elsevier Ireland Ltd. doi: 10.1016/j.cosrev.2023.100598.

[13] Mihaela Flori, "Influence of Temperature on The Measurement of Total Dissolved Solids (TDS) In Water", Acta Technica Corviniensis – Bulletin of Engineering, 2024, pp: 79 – 82

[14] Ong, C., Ibrahim, S., & Sen Gupta, B. (2007). A survey of tap water quality in Kuala Lumpur. Urban Water Journal, 4(1), 29–41. https://doi.org/10.1080/15730620601145923