Indonesian Journal of Electrical Engineering and Computer Science
Vol. 34, No. 2, May 2024, pp. 1042~1051
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v34.i2.pp1042-1051

Water quality monitoring with an early warning system for enhancing the shrimp aquaculture production

Guruh Aryotejo, Prajanto Wahyu Adi, Eko Adi Sarwoko
Department of Informatics, Faculty of Science and Mathematics, Diponegoro University, Semarang, Indonesia

Article Info

Article history:

Received Nov 16, 2023
Revised Jan 23, 2024
Accepted Feb 4, 2024

Keywords:

Early warning system
ESP32
Internet of things
Monitoring system
Water quality

ABSTRACT

Aquaculture provided 43% of the aquatic animal food consumed by humans in 2007, and it is anticipated to expand even more to meet the increasing future demand. Marine Science Techno Park (MSTP) is one of the Techno Parks in Indonesia and is located in Teluk Awur, Jepara. MSTP has an intensive system of Vannamei shrimp farming activities. The challenge that has been faced annually is the organic matter of the remaining shrimp feed that accumulates in the waters can cause a decrease in pond water quality. Ammonia-N anions derived from the decomposition of feed residues can cause toxins in shrimp culture ponds which can further interfere with shrimp survival. The objective of this study is to design a self-controlled water quality monitoring system that is equipped with an early warning system based on internet of things (IoT) and to implement the design for analyzing the water quality including dissolved oxygen (DO), pH, and temperature in the Vannamei farming pond of MSTP UNDIP through integrating IoT which is equipped with an early warning system. If the water quality reaches a threshold value, the monitoring system will send a sound signal to the buzzer followed by sending an alert notification in real-time conditions automatically to a smartphone.

Corresponding Author:

Guruh Aryotejo
Department of Informatics, Faculty of Science and Mathematics, Diponegoro University
Prof Sudarto S.H. Street, Tembalang, Semarang, 50275, Indonesia
Email: guruh2000@yahoo.com

1. INTRODUCTION

Aquaculture supplied 43% of the aquatic animal food consumed by humans in 2007, and it is expected to further expand to meet the growing demand in the future [1]. Vannamei shrimp farming is a significant source of income in many countries, especially those near the equator and the tropics with coastal areas [2]. Proper management of these farms leads to economic growth for nearby communities, as it creates job opportunities in production, processing, marketing, transportation, and other related services [3]. Furthermore, toxins in shrimp culture ponds, resulting from the decomposition of feed residues, can be attributed to ammonia-N anions and may pose a threat to the survival of shrimp [4]. To ensure the growth of Vannamei shrimp and the profitability of the farm, monitoring water quality in the ponds is crucial. Periodic monitoring of the water quality parameters is necessary to achieve this control [5], [6]. For the example, maintaining DO levels between 5-7 ppm or mg/L and a minimum of 4-5 mg/L is essential. Additionally, water temperature, pH, and salinity (measured by electrical conductivity) play a critical role in water quality. The optimum temperature for shrimp growth ranges from 28-32 °C, and pH levels between 6.5-9.0 are recommended [6]. The health and survival of shrimp are directly linked to the physical and chemical properties, as well as the five-day biochemical parameters of the pond water [6], [7], making the upkeep of a healthy environment essential.

To achieve success in shrimp farming, it is crucial to keep a close eye on various water quality parameters, with dissolved oxygen (DO) being the most significant factor that affects shrimp growth and mortality [8]. Extremely high or low pH levels can cause stress and negatively impact survivability [9]. Furthermore, maintaining an ideal salinity concentration between 15-25 ppt is important, as a high concentration can reduce DO levels and alkalinity, leading to unfavorable conditions for aquatic growth [10].

At present, water quality monitoring in shrimp farms is primarily conducted through manual methods involving the use of handheld measuring devices to collect measurements from various ponds or sites [11]. Measurements are taken by an individual who walks around and notes the readings in a notebook for later analysis. This approach is demanding and challenging. To enhance accuracy and consistency of monitoring, provide real-time data for decision-making, and decrease the time and effort needed to collect and analyze data, a more efficient and automated system is required to monitor and help farmers identify low water quality, and make informed decisions regarding water quality management [12].

Time series data can also be useful in predicting changes in water quality [13]. Farmers are gradually gaining a deeper understanding of water quality parameters, which can be combined to suit the specific features of individual farm sites to achieve successful crop growth and enhance economic profitability [14]. However, obtaining real-time agriculture or aquaculture information for monitoring crops and predicting yields can be challenging [15].

To address this challenge, integrating internet of things (IoT) technologies is vital for the transformation of conventional farming methods into smart farming systems [16], [17]. Proficiency in computer networking is essential for the successful implementation of IoT [18]. Employing virtualization for simulating computer networks, including IoT, can minimize implementation errors and is achievable in a home setting [19]. If IoT implementation leads to substantial internet bandwidth usage, the adoption of bandwidth management strategies proposed by Mufadhol *et al.* [20], Aryotejo and Mufadhol [21] can effectively streamline internet bandwidth.

2. METHOD

2.1. Research method

The research method of this study is as follows Figure 1. The hardware system design is an overview of the water quality monitoring system hardware presented, while the software system design is a general perspective on the development of software for monitoring within the system and web. After the completion of hardware and software design, the next step is to implement the design into a prototype to ensure the validity and accuracy of the design. We will examine the sensors and the entire system using the black box method.

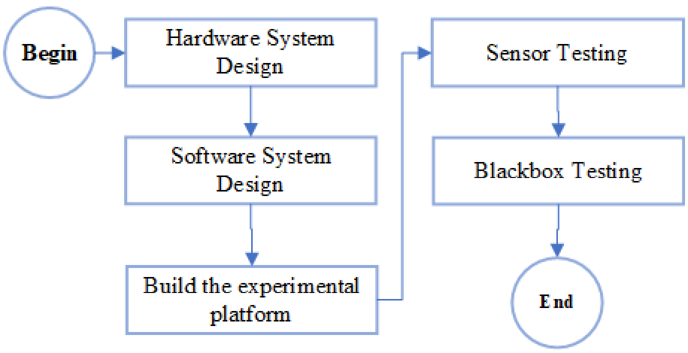


Figure 1. Research methods

2.1.1. Development tools

In this research, we employed the Arduino Uno as the main controller and the ESP32 as the communication module. The sensors employed include the DS18B20 for temperature sensing, an analog pH sensor, and a DO sensor. The OLED and Buzzer will be used as an early warning system, as illustrated in Figure 2. The specification of the sensor can be seen in Table 1. The development of the system's software uses the Arduino IDE, Laravel framework, and ThingSpeak.

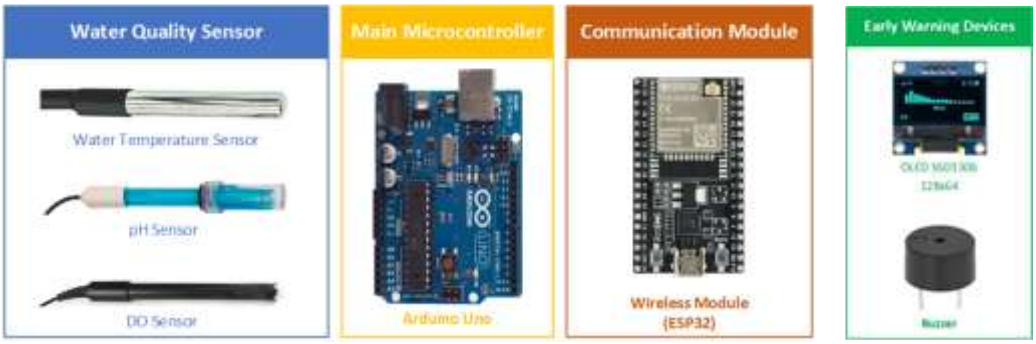


Figure 2. Water temperature, pH, DO sensor, Arduino Uno, ESP32, OLED, and Buzzer

Table 1. The sensors used in the study

Sensor	Model	Range	Accuracy
pH	SEN0161	0 – 14 pH	±0.1 pH (25 ℃)
DO	SEN0237-A	0~20 mg/L	±0.05 mg/L
Temperature	DS18B20	-55 ℃～125 ℃	±0.5 °C

3. RESULTS AND DISCUSSION

3.1. Monitoring system

The data gathered by the water quality sensors were sent to the main microcontroller through an Arduino Uno board, which is equipped to establish communication with numerous sensors through different communication protocols such as Universal asynchronous receiver-transmitter (UART) and inter-integrated circuit, eye-squared-C (I2C). The main microcontroller is in charge of collecting and sending the measured data to the gateway, which is detailed in the communication layer. The data collection from the sensors occurs every minute to capture the measuring data. Once collected, the measuring data is transmitted to the gateway through a UART standard, as seen in Figure 3.

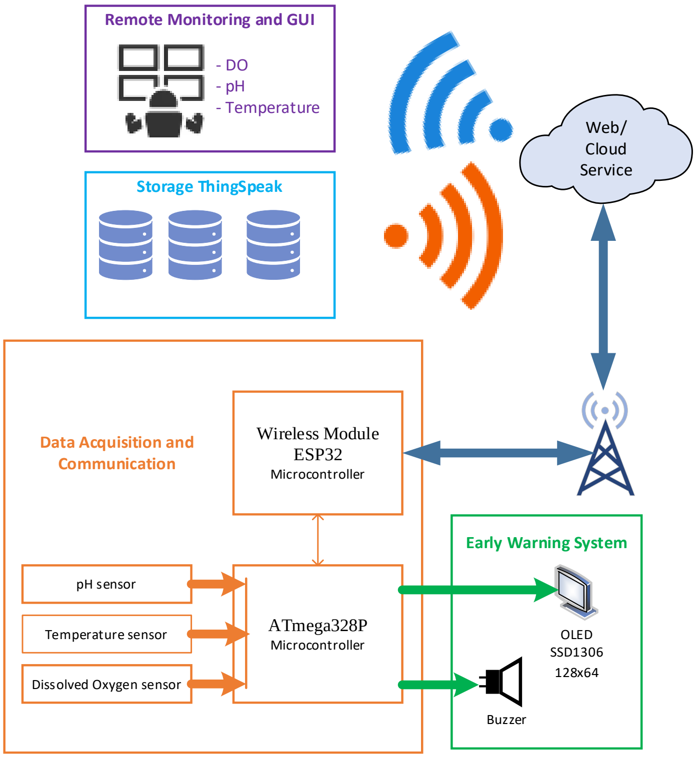


Figure 3. An overview of the water quality monitoring system

3.1.1. Communication layer

The Firebeetle ESP32 is a low-cost system on chip (SoC) microcontroller developed by espressif systems, the same company behind the renowned ESP8266 SoC used in NodeMCU [17]. It serves as the successor to the ESP8266 SoC and features a 32-bit Xtensa LX6 microprocessor with integrated Wi-Fi and Bluetooth capabilities [22]. One of the notable advantages of ESP32, similar to ESP8266, is the integration of RF components such as power amplifier, low-noise receive amplifier, antenna switch, and filter. This integration simplifies hardware design for ESP32 as it requires minimal external components. An important aspect to note about ESP32 is that it is manufactured using 40 nm ultra-low-power TSMC technology. This allows it to be operated with common batteries, similar to those used in audio devices, monitoring equipment or smartwatches.

3.1.2. Storage ThingSpeak

Primarily, the ThingSpeak API is used to connect an object to the IoT [23], [24]. This interface provides simple communication capabilities for IoT objects as well as a variety of additional applications, such as ThingTweet [25]. It provides users with support for near-real-time data collection, processing, and simple visualizations. There are four fields designated for positional data: description, latitude, longitude, and elevation. All incoming information is timestamped and given a sequential identifier. Data can be published by calling the ThingSpeak API with a unique alphanumeric string referred to as a ‘write key’ for authentication. Similarly, if the channel is configured to keep its data private (the default setting), a ‘read key’ is required to access its data. If a channel is set to public, a read key is not required to access it [26].

3.1.3. Remote monitoring and GUI

The primary viewer for observing this self-controlled water quality monitoring system is the SIMAPeDI. It retrieves sensor data from Thingspeak, an IoT platform, and API designed to store sensor data in real-time. The subsequent display device is a 128×64 OLED placed close to the monitoring system to facilitate direct system monitoring.

3.1.4. Early warning system

The early warning system comprises two elements: a 128×64 OLED display and a buzzer. When the monitoring system detects data surpassing the predetermined threshold, it activates the buzzer and showcases the sensors with exceeded data on the OLED display. Concurrently, the monitoring system sends notifications to the website, ensuring that users promptly receive alerts regarding the exceeded threshold data.

3.2. Hardware system design

The system has been successfully developed to enable the sensor to gather temperature, pH, and DO information from the surrounding environment, which is then displayed on an OLED screen and web. Table 2 shows the pin configuration for Arduino UNO and ESP32, while Figure 4 is a self-controlled water quality monitoring circuit diagram. The Arduino Uno is used as the main controller responsible for reading data from pH, temperature, and DO sensors. Additionally, it is also assigned the task of displaying the readings from these sensors on an OLED screen. If any parameters deviate from the predefined values, the Arduino Uno will send a signal to the buzzer, as in Figure 5. Later, Arduino Uno sends the processed sensor data to the ThingSpeak website using the assistance of the ESP32 module.

Table 2. The pin configuration for Arduino UNO and ESP32

No.	Microcontroller	Pin No.	Function
1	Arduino Uno	A0	Connected to analog pH sensor on data pin to measure the pH level of pond water
2	Arduino Uno	A1	Connected to analog DO sensor on data pin to measure the DO level of pond water
3	Arduino Uno	A4	Connected to OLED display SSD1306 128×64 on SDA (serial data) pin to display the current reading of the sensors
4	Arduino Uno	A5	Connected to OLED display SSD1306 128×64 on SCL (serial clock) pin to display the current reading of the sensors
5	Arduino Uno	D0	Connected to ESP32 on #17 pin
6	Arduino Uno	D1	Connected to ESP32 on #16 pin
7	Arduino Uno	D4	Connected to DS18B20 sensor on SCL (serial clock) pin to display the current reading of the sensors
8	Arduino Uno	D~9	Connected to buzzer on data pin
9	ESP32	16	Connected to Arduino Uno on #D1 pin
10	ESP32	17	Connected to Arduino Uno on #D0 pin

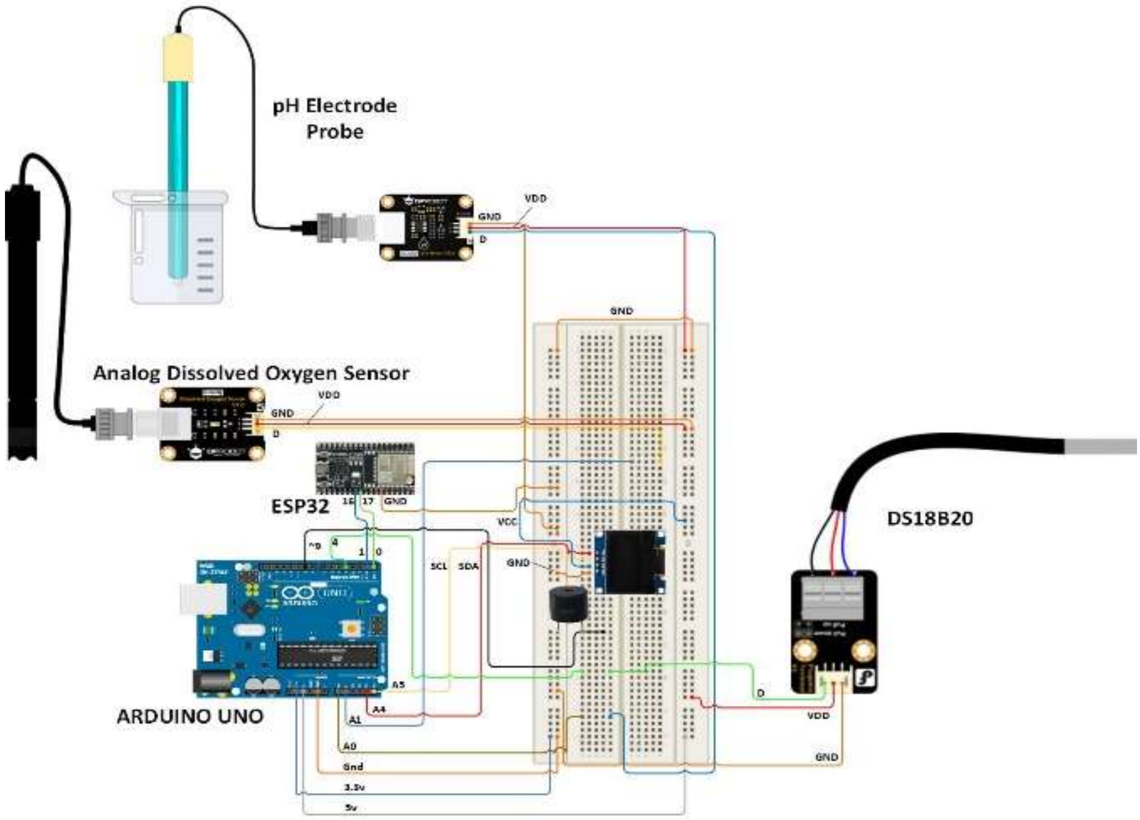


Figure 4. Self-controlled water quality monitoring circuit diagram

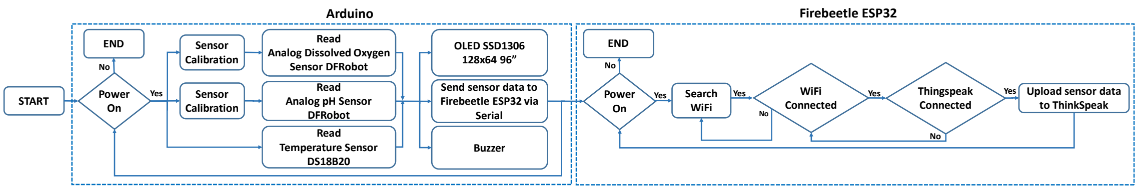


Figure 5. An overview of the self-controlled water quality monitoring system software

3.3. Software system design

The workflow from the software perspective is explained through the flowchart depicted in Figure 6. When the Arduino Uno is powered on, its first task is to calibrate the analog sensors and DO sensor. Afterward, the DS18B20 temperature sensor, analog pH sensor, and analog DO sensor will collect data. Each successfully retrieved data will be displayed on the OLED screen. If it exceeds the specified limits, the buzzer will sound.

Every successfully retrieved sensor data is immediately forwarded to the ESP32 through serial communication. If the ESP32 is already powered on, it will establish a connection with Wi-Fi. Upon successful connection, the sensor data is uploaded to ThingSpeak, as in Figure 5.

Every sensor value sent by the self-controlled water quality monitoring to Thingspeak can be directly viewed within Thingspeak and on a custom website called SIMAPeDI (independent water monitoring system with early warning/in Indonesian: *sistem monitoring air mandiri dengan peringatan dini*). This custom website is accessible through standard browsers on desktops or smartphones. Given that smartphones serve as the primary tool for information retrieval in cyberspace [27]. Users can log in, view the dashboard, and check temperature, pH, DO, and view/edit notifications, as shown in Figures 6 and 7.

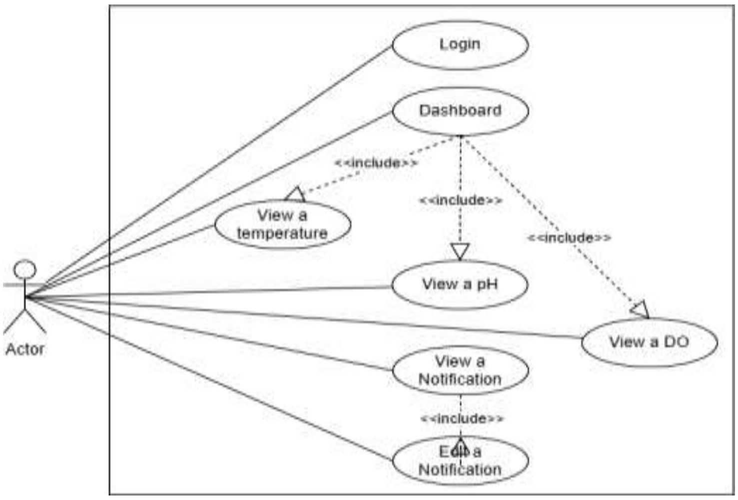


Figure 6. The use case of SIMAPeDI

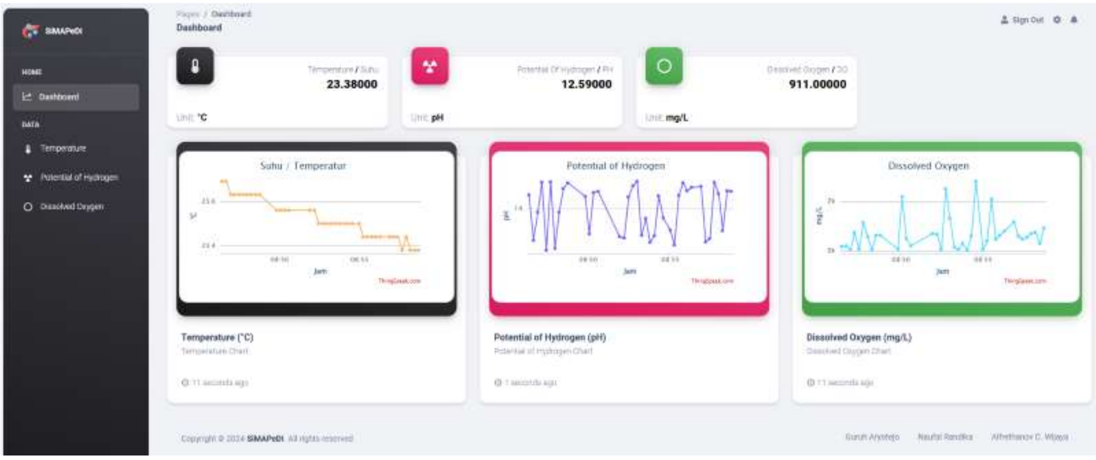


Figure 7. The dashboard of SIMAPeDI

3.4. Build the experimental platform

The water quality monitoring with an early warning system has been successfully built to enable the sensor to gather temperature, pH, and DO information, as seen in Figure 8. The overall system is shown in Figure 8(a), where the sensor data are displayed on an OLED and buzzer screen, as seen on Figure 8(b). In addition, this prototype also sends sensor reading data to the SIMAPeDI site.

Figure 8. Prototype self-controlled water quality monitoring circuit (a) overview and (b) buzzer and OLED

3.5. Sensors testing

Sensor testing is crucial to guarantee their proper functioning under different environmental conditions and to obtain precise and dependable data. The first sensor to be tested is the temperature sensor DS18B20. The testing was conducted by considering the comparison between the readings obtained from the DS18B20 temperature sensor and the Dr. Gray Digital Thermometer on the shrimp pond water The average relative error value from eleven temperature reading is approximately 3%.

The second sensor to be tested is the DFRobot analog pH sensor. A correct pH meter calibration necessitates a minimum two-point calibration, utilizing distinct buffer solutions (pH 4 and pH 7). After rinsing with distilled water, the pH probe is immersed in the solution until the pH level stabilizes, ensuring that the pH meter accurately aligns with the pH value of the calibration solution. After calibrating the pH sensor, the next step is to test the accuracy of the pH sensor by comparing the values obtained from the analog pH sensor (DFRobot) and the calibrated PH-001(I)A pen type pH meter using shrimp pond water. The average relative error value from eleven pH reading experiments is approximately 3.1%, as presented in Table 3.

The second sensor to be tested is the DFRobot analog pH sensor. A correct pH meter calibration necessitates a minimum two-point calibration, utilizing distinct buffer solutions (pH 4 and pH 7). After rinsing with distilled water, the pH probe is immersed in the solution until the pH level stabilizes, ensuring that the pH meter accurately aligns with the pH value of the calibration solution. After calibrating the pH sensor, the next step is to test the accuracy of the pH sensor by comparing the values obtained from the analog pH sensor (DFRobot) and the calibrated PH-001(I)A pen type pH meter using shrimp pond water. The average relative error value from eleven pH reading experiments is approximately 3.1%, as presented in Table 3.

The last sensor to be tested is the DO sensor. The examination involved evaluating the readings from the DFRobot gravity: analog DO sensor in comparison with those obtained from the YSI Pro 20. The data from the tests is displayed in Table 4. It was found that the analog DO sensor readings relatively match the results from the YSI Pro 20. The average relative error, calculated from eleven temperature reading experiments, is approximately 2%.

Table 3. The results of temperature sensor (DS18B20) with Dr. Gray digital thermometer accuracy testing

Time (Minutes)	DS18B20	Dr. Gray digital thermometer	*Error* (%)
0	29.25	28.50	3%
1	29.25	28.30	3%
2	29.19	28.30	3%
3	29.19	28.30	3%
4	29.25	28.40	3%
5	29.19	28.30	3%
6	29.25	28.40	3%
7	29.25	28.40	3%
8	29.25	28.40	3%
9	29.19	28.30	3%
10	29.19	28.30	3%
Average			3%

Table 4. The results of DFRobot pH sensor (DS18B20) with PH-001(I)A pen type PH meter accuracy testing

Time (minutes)	Gravity DFRobot Analog DO sensor	YSI Pro 20	Error (%)
0	5.63	5.95	3%
1	4.62	5.64	2%
2	4.82	5.38	2%
3	5.75	5.05	1%
4	4.66	5,22	1%
5	5.46	4.91	1%
6	4.52	5.52	2%
7	5.11	5.8	1%
8	4.25	4.79	3%
9	6.77	5.53	4%
10	5.93	5.32	2%
Average			2%

3.6. Blackbox testing

A testing method that does not rely on any information about the internal design or implementation details of the subject being evaluated [28]. This method evaluates an application's functionality without delving into its internal mechanisms or operations. This approach can be utilized across various stages of software testing, including unit, integration, system, and acceptance testing [29].

3.6.1. Buzzer

The purpose of the buzzer testing was to verify whether the buzzer system performs as intended in different situations, as outlined in the predetermined scenarios. The outcomes of this examination are displayed in Table 5, which indicates that all four tested scenarios produced acceptable results. These results indicate that the buzzer is ready to be used as intended.

Table 5. The results of buzzer testing

No	Scenario	Expected result	Summary
1	Give commands when the temperature condition read by the sensor exceeds 30 °C (>30 °C)	The buzzer sounds when the temperature condition read by the sensor exceeds 30 °C (>30 °C).	[□] Valid [ ] Invalid
2	Give commands when the temperature condition read by the sensor is below 26 °C (<26 °C)	The buzzer sounds when the temperature condition read by the sensor is below 26 °C (<26 °C)	[□] Valid [ ] Invalid
3	Give commands when the pH condition read by the sensor exceeds 9.00 (>9.00)	The buzzer sounds when the pH condition read by the sensor exceeds 9.00 (>9.00)	[□] Valid [ ] Invalid
4	Give commands when the pH condition read by the sensor is below 6.50 (<6.50)	The buzzer sounds when the pH condition read by the sensor is below 6.50 (<6.50)	[□] Valid [ ] Invalid

3.6.2. OLED

This test is conducted to evaluate whether the LCD (OLED SSD1306 0.96") operates according to the predetermined scenarios. The results of this test are displayed in Table 6. It can be observed from Table 6 that one of the executed scenarios yielded valid results.

Table 6. The results of OLED testing

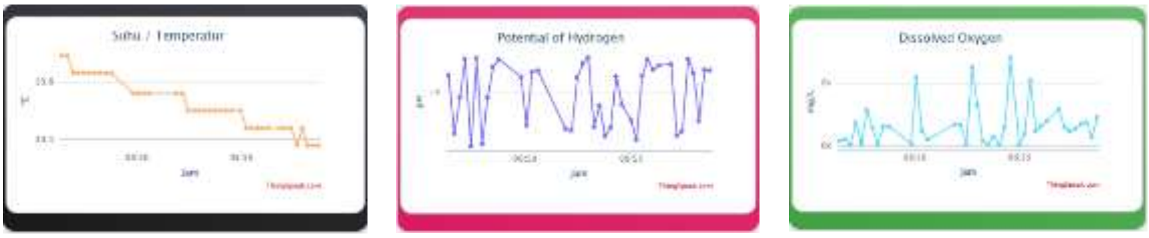
No	Scenario	Expected result	Summary
1	Provide a command to display the temperature sensor readings as ‘Temperature: (c): <value>’ and the pH sensor readings as ‘pH Value: <value>’.	The LCD can display the temperature sensor readings as ‘Temperature: (c): <value>’ and the pH sensor readings as ‘pH Value: <value>’	[✓] Valid [ ] Invalid

3.6.3. Uploaded sensors data

This assessment is performed to determine the functionality of uploading sensor data to ThingSpeak in accordance with predefined scenarios. The outcomes of this evaluation are presented in Table 7 and the display for the successful data transmission is presented in Figure 9. Figure 9(a) specifically illustrates the temperature sensor (DS18B20), Figure 9(b) for the analog pH sensor and Figure 9(c) for the analog DO sensor. These results indicate that the sensor data transmission to ThingSpeak has been successful and is ready to be used as intended.

Table 7. The results of sensors data uploaded to ThinkSpeak testing

No	Scenario	Expected result	Summary
1	Provide a command to upload the recent data from temperature sensor to ThinkSpeak.	ThinkSpeak receive the recent uploaded temperature sensor readings and displayed at graph ‘Temperature’	[✓] Valid [ ] Invalid
2	Provide a command to upload the recent data from pH sensor to ThinkSpeak.	ThinkSpeak receive the recent uploaded pH sensor readings and displayed at graph ‘pH’	[✓] Valid [ ] Invalid
3	Provide a command to upload the recent data from DO sensor to ThinkSpeak.	ThinkSpeak receive the recent uploaded DO sensor readings and displayed at graph DO’	[✓] Valid [ ] Invalid



(a) (b) (c)

Figure 9. Uploaded sensor readings and displayed at ThinkSpeak graph during sensor testing; (a) temperature graph, (b) pH graph, and (c) DO graph

4. CONCLUSION

The system has been successfully developed and implemented to enable the sensor to gather temperature, pH, and DO information from the surrounding environment. Based on the results and discussion of accuracy and Blackbox testing on the prototype of the IoT-based water quality monitoring application that we developed, we can conclude that this system can be used to monitor the temperature and pH of water in aquaculture ponds or shrimp ponds. The system is capable of measuring water temperature, pH, and DO with a high level of accuracy, with an average relative error of 3.0% for the DS18B20 temperature sensor, 3.1%

for the DFRobot pH sensor, and 2% for the DFRobot DO sensor. Furthermore, functional testing of hardware and software on the LCD, buzzer, and website using black box testing methods also showed valid results.

ACKNOWLEDGEMENTS

This research is supported by Institute for Research and Community Services, Universitas Diponegoro, Semarang, Indonesia with the research subject "Design and implementation of self-controlled water quality monitoring with an early warning platform based on NB-IoT for enhancing the shrimp aquaculture production system" and grant number is 753-06/UN7.D2/PP/IV/2023.

REFERENCES

[1] J. Bostock *et al.*, "Aquaculture: global status and trends," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 365, no. 1554, pp. 2897–2912, 2010, doi: 10.1098/rstb.2010.0170.
[2] J. J. Carbajal and L. P. Śanchez, "Classification based on fuzzy inference systems for artificial habitat quality in shrimp farming," *7th Mexican International Conference on Artificial Intelligence - Proceedings of the Special Session, MICAI 2008*, vol. 1996, pp. 388–392, 2008, doi: 10.1109/MICAI.2008.70.
[3] N. C. Ferreira, C. Bonetti, and W. Q. Seiffert, "Hydrological and water quality indices as management tools in marine shrimp culture," *Aquaculture*, vol. 318, no. 3–4, pp. 425–433, Aug. 2011, doi: 10.1016/J.AQUACULTURE.2011.05.045.
[4] M. Zhao, D. Yao, S. Li, Y. Zhang, and J. J. Aweya, "Effects of ammonia on shrimp physiology and immunity: a review," *Reviews in Aquaculture*, vol. 12, no. 4, pp. 2194–2211, 2020, doi: 10.1111/raq.12429.
[5] K. N. A. K. Adnan *et al.*, "Water quality classification and monitoring using e-nose and e-tongue in aquaculture farming," *2014 2nd International Conference on Electronic Design, ICED 2014*, pp. 343–346, 2014, doi: 10.1109/ICED.2014.7015827.
[6] J. J. C. Hernández, L. P. S. Fernández, and O. Pogrebnyak, "Assessment and prediction of water quality in shrimp culture using signal processing techniques," *Aquaculture International*, vol. 19, no. 6, pp. 1083–1104, 2011, doi: 10.1007/s10499-011-9426-z.
[7] I. D. Oviedo, C. A. Altamirano, M. E. Millan, N. X. Arreaga, and V. S. Padilla, "Development of a prototype of an automatic system for shrimp farming using telecontrol as a viable proposal for small producers," *Proceedings - 2017 European Conference on Electrical Engineering and Computer Science, EECS 2017*, 2018, pp. 477–483, doi: 10.1109/EECS.2017.94.
[8] R. A. Bórquez-Lopez, R. Casillas-Hernandez, J. A. Lopez-Elias, R. H. Barraza-Guardado, and L. R. Martinez-Cordova, "Improving feeding strategies for shrimp farming using fuzzy logic, based on water quality parameters," *Aquacultural Engineering*, vol. 81, no. August 2017, pp. 38–45, 2018, doi: 10.1016/j.aquaeng.2018.01.002.
[9] J. J. Carbajal-Hernández, L. P. Sánchez-Fernández, L. A. Villa-Vargas, J. A. Carrasco-Ochoa, and J. F. Martínez-Trinidad, "Water quality assessment in shrimp culture using an analytical hierarchical process," *Ecological Indicators*, vol. 29, pp. 148–158, Jun. 2013, doi: 10.1016/J.ECOLIND.2012.12.017.
[10] M. S. Shekhar, J. Kiruthika, and A. G. Ponniah, "Identification and expression analysis of differentially expressed genes from shrimp (Penaeus monodon) in response to low salinity stress," *Fish and Shellfish Immunology*, vol. 35, no. 6, pp. 1957–1968, 2013, doi: 10.1016/j.fsi.2013.09.038.
[11] "HI-9828 multi-parameter water quality portable meter," [Online]. Available: https://www.keison.co.uk/hannainstruments_hi9828.shtml.
[12] M. Oberle, S. Salomon, B. Ehrmaier, P. Richter, M. Lebert, and S. M. Strauch, "Diurnal stratification of oxygen in shallow aquaculture ponds in central Europe and recommendations for optimal aeration," *Aquaculture*, vol. 501, pp. 482–487, Feb. 2019, doi: 10.1016/J.AQUACULTURE.2018.12.005.
[13] P. Shi, G. Li, Y. Yuan, G. Huang, and L. Kuang, "Prediction of dissolved oxygen content in aquaculture using clustering-based softplus extreme learning machine," *Computers and Electronics in Agriculture*, vol. 157, pp. 329–338, Feb. 2019, doi: 10.1016/J.COMPAG.2019.01.004.
[14] I. Sivaraman, M. Krishnan, and K. Radhakrishnan, "Better management practices for sustainable small-scale shrimp farming," *Journal of Cleaner Production*, vol. 214, pp. 559–572, Mar. 2019, doi: 10.1016/J.JCLEPRO.2018.12.172.
[15] S. V. Gaikwad, A. D. Vibhute, K. V. Kale, and S. C. Mehrotra, "An innovative IoT based system for precision farming," *Computers and Electronics in Agriculture*, vol. 187, p. 106291, Aug. 2021, doi: 10.1016/J.COMPAG.2021.106291.
[16] V. S. Narwane, A. Gunasekaran, and B. B. Gardas, "Unlocking adoption challenges of IoT in Indian agricultural and food supply chain," *Smart Agricultural Technology*, vol. 2, p. 100035, Dec. 2022, doi: 10.1016/J.ATECH.2022.100035.
[17] M. Mejia-Herrera, J. S. Botero-Valencia, D. Betancur-Vásquez, and E. A. Moncada-Acevedo, *Low-cost system for analysis pedestrian flow from an aerial view using Near-Infrared, Microwave, and Temperature sensors*, vol. 13, 2023.
[18] M. Mufadhol, G. Aryotejo, and D. E. Kurniawan, "The network planning concept for increase quality of service using packet tracer," *Proceedings of the 2019 2nd International Conference on Applied Engineering, ICAE 2019*, pp. 8–13, 2019, doi: 10.1109/ICAE47758.2019.9221675.
[19] G. Aryotejo, E. A. Sarwoko, A. Sugiharto, and M. M. Hakim, "Performance of virtual machine managers for computer network learning," *5th International Conference on Informatics and Computational Sciences (ICICoS)*, pp. 155–159, 2021, doi: 10.1109/icicos53627.2021.9651899.
[20] M. Mufadhol, G. Aryotejo, and A. Wibowo, "Netscan and networx for management bandwidth and traffic with simple routing," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 15, no. 1, pp. 464-470, 2017.
[21] G. Aryotejo and M. Mufadhol, "Static and dynamic alliance: The solution of reliable internet bandwidth management," *Journal of Physics: Conference Series*, vol. 1217, no. 1, p. 012126, 2019, doi: 10.1088/1742-6596/1217/1/012126.
[22] S. Kocer, O. Dundar, and R. Butuner, *Programmable Smart Microcontroller Cards*, ISRES Publishing, 2021.
[23] M. A. A. Razali, M. Kassim, N. A. Sulaiman, and S. Saaidin, "A ThingSpeak IoT on real time room condition monitoring system," *2020 IEEE International Conference on Automatic Control and Intelligent Systems, I2CACIS 2020 - Proceedings*, no. June, pp. 206–211, 2020, doi: 10.1109/I2CACIS49202.2020.9140127.
[24] T. Zachariah, N. Klugman, and P. Dutta, "ThingSpeak in the Wild: exploring 38K visualizations of IoT data," *SenSys 2022 - Proceedings of the 20th ACM Conference on Embedded Networked Sensor Systems*, pp. 1035–1040, 2022, doi: 10.1145/3560905.3567766.
[25] L. Alhmoud and H. Al-Zoubi, "IoT applications in wind energy conversion systems," *Open Engineering*, vol. 9, no. 1, pp. 490–499, 2019, doi: 10.1515/eng-2019-0061.

[26] L. D. Nardis, G. Caso, and M. G. D. Benedetto, "ThingsLocate: a ThingSpeak-based indoor positioning platform for academic research on location-aware internet of things," *Technologies*, vol. 7, no. 3, 2019, doi: 10.3390/technologies7030050.
[27] A. Wibowo, G. Aryotejo, and M. Mufadhol, "Accelerated mobile pages from JavaScript as accelerator tool for web service on e-commerce in the e-business," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 8, no. 4, pp. 2399–2405, 2018, doi: 10.11591/ijece.v8i4.pp2399-2405.
[28] M. Sholeh, I. Gisfas, Cahiman, and M. A. Fauzi, "Black box testing on ukmbantul.com page with boundary value analysis and equivalence partitioning methods," *Journal of Physics: Conference Series*, vol. 1823, no. 1, 2021, doi: 10.1088/1742-6596/1823/1/012029.
[29] V. C. Hu and R. Kuhn, "Verification and test methods for access control policies/models NIST special publication 800-192 verification and test methods for access control policies / models," *NIST*, vol. 800, no. 192, 2017.

BIOGRAPHIES OF AUTHORS

Guruh Aryotejo Graduate of Computer Science and Informatics from Budi Luhur University, Jakarta, Indonesia, and a Master of Science from ASIA University, Taichung, Taiwan. Department of Informatics, Universitas Diponegoro, Semarang, Indonesia. He served as a lecturer at the Department of Informatics, Faculty of Science and Mathematics, Universitas Diponegoro. Research Interests are in computer networks, database systems, IoT, software engineering, and information systems. He can be contacted at email: guruh2000@gmail.com.

Prajanto Wahyu Adi earned his Bachelor's degree in Computer Science from Universitas Stikubank, Indonesia, in 2011. Subsequently, he completed a dual-degree program in Information Technology at Universitas Dian Nuswantoro, Indonesia, and Computer Science at Universiti Teknikal Malaysia Melaka, Malaysia, graduating in 2014. He began his career as a lecturer at the Department of Information Technology within the Faculty of Computer Science at Universitas Dian Nuswantoro, and presently, he holds a teaching position at the Department of Informatics in the Faculty of Science and Mathematics at Universitas Diponegoro. His areas of expertise encompass cryptography, digital image classification, steganography, and watermarking. For contact purposes. He can be contacted at email: prajanto@live.undip.ac.id.

Eko Adi Sarwoko Graduate of mathematics from Universitas Diponegoro, Indonesia in 1991 and the Master of computer science at Universitas Indonesia in 2003. He served as a lecturer at the Department of Informatics, Faculty of Science and Mathematics, Universitas Diponegoro. Research interests are in machine learning, computer systems, and information systems. He can be contacted at email: ekoadisarwoko@lecturer.undip.ac.id.