International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 01 Issue: 01 December 2023
Page No: 24 - 31
https://irjaeh.com

Water Monitoring System for Aquatic Organisms

Dr. M. Kanthimathi[1], Morrven R[2], Madhan N[3], Nataraj T R[4],

[1,2,3,4] Department of Electronics and Communication Engineering, Sri Sai Ram Engineering College, West Tambaram, Chennai, India.

Email: hod.ci@sairam.edu.in[1], sec20ec021@sairamtap.edu.in[2], nagarajanmadhan7@gmail.com[3],

trnataraj02@gmail.com[4]

Abstract

Maintaining the health and sustainability of aquatic ecosystems demands vigilant water quality monitoring. In response, we introduce an advanced Water Quality Monitoring System tailored for real-time data acquisition and analysis, contributing to effective water resource management. A wide range of vital indicators, including temperature, pH, dissolved oxygen, and more, are integrated into this system. These sensors are placed at strategic locations inside bodies of water. The Water Quality Monitoring System offers a spectrum of indispensable features to elevate monitoring capabilities. It ensures continuous data collection, empowering early identification of fluctuations in water quality and rapid response to potential pollution incidents. Automated alerts are incorporated, promptly notifying stakeholders of abnormal conditions using the GSM module. Its modular architecture guarantees scalability, simplifying integration with existing monitoring infrastructures and accommodating additional sensors as needed. The system accommodates deployment in remote or challenging terrains, ensuring uninterrupted vigilance over water quality.

Keywords: *Turbidity, pH, GSM Module, Ecosystems, Water Quality Monitoring System.*

1. Introduction

Within the Internet of Things domain, where connectivity additionally with data-driven decision-making reigns supreme, our project embarks on a mission to create a robust and innovative solution. We are utilizing Node-MCU, an ESP8266-based microcontroller, to monitor vital parameters: temperature and pH levels in aquatic environments. This endeavor serves a dual purpose: to enhance our understanding of these ecosystems and to actively safeguard them. Our Node-MCU setup continuously collects temperature and pH data, transmitting it to the cloud-based platform Things-Speak for real-time visualization and analysis. This integration empowers us to observe trends, detect anomalies, and take timely action. Furthermore, our system boasts an intelligent alert mechanism. If the recorded parameters surpass predetermined thresholds, Node-MCU triggers SMS alerts to a designated phone number, ensuring prompt attention to adverse conditions and proactive intervention. In a world where aquatic ecosystems are increasingly vulnerable, this project aligns with the broader goals of conservation and sustainability. By seamlessly combining IoT technology, data analysis, and communication, we aim to foster a harmonious coexistence between human activities and aquatic environments, where the well-being of all inhabitants, from microorganisms to macrofauna, is prioritized.

2. Literature Survey

S. Pasika, and S.T. Gandla, in 2023, presented a novel monitoring system designed to assess various water quality measuring parameters including water temperature, environmental humidity, tank level of water, pH levels, additionally turbidity. These sensors interface seamlessly with a Microcontroller Unit, and further data processing is undertaken by a Personal Computer. Subsequently, this is collected through the Internet of Things, and data is transferred to the ThingSpeak cloud-based platform for continuous monitoring of water quality. Looking ahead, future efforts should expand to encompass the analysis of additional characteristics of the water, including dissolved oxygen

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 01 Issue: 01 December 2023
Page No: 24 - 31
https://irjaeh.com

concentrations, free residual chlorine, electrical conductivity, and nitrates [1]. Unnikrishna Menon and colleagues in 2022, This method is founded upon wireless sensor networks, enabling continuous surveillance of crucial water quality measuring parameters. Within this system, a dedicated nomadic wireless sensor is engineered for continuous pH monitoring, recognized as a pivotal parameter influencing water quality. Four modules: a power module, a wireless communication module, a processing module, and a signal conditioning module are among the key modules included in the sensor node's architectural design. To facilitate seamless data transmission, data using the wireless communication module—more especially, the Zigbee module— is sent to the base station from the pH sensor. While adhering to crucial signal processing and conditioning protocols. The pathway development of the nodes is carefully considered during the design and simulation phases and hardware prototype construction, employing suitable circuit components. This meticulous approach significantly minimizes power requirements for the system, delivering a reasonably priced platform for the continuous evaluation of the water quality in our valuable water resources [2]. Guntur Samodro, Rini Kurnia, Marti Widya Sari, and Prahenusa Wahyu Ciptadi in 2023 This study suggests an Arduino-based real-time water quality monitoring system with five different kinds of sensors, along with data analysis. Thing-Speak is an open-source Internet of Things software. The goal of this project was to create a monitoring system for assessing water quality to gauge the degree of groundwater pollution in that area. The Arduino IDE, a programming language for C, was used to create this Android-based well water quality monitoring system. The development of this system made use of observation, interviews, and literature study as data collection techniques. Phases of analysis, system design, implementation, and testing are all part of application development. A variety of sensors, including sensors for pH, temperature, and TDS, are used in the implementation of this program to measure the temperature, pH, and amount of dissolved solids. Additionally, this system has a Wi-Fi module that allows data to be sent from the device to the created application.[3]. M. H. Abd Razak, J. Jamaludin, I. Ismail, W. Z. Wan Ismail, and N. A. Razman in 2023, The literature contains several studies using various methods for monitoring water quality. We can classify the proposed approaches to measure water quality in three categories: Sensor Node, GSM Module, and Wireless Sensor Network. N. A. suggested a monitoring system that measures the necessary parameters using sensor nodes. Cloete sensors are connected to a microcontroller through a signal conditioning circuit for sensor signal processing. Demetillo et al. proposed a monitoring system in which the pH, temperature, and sensor node design measure the dissolved oxygen content of water. These sensors were chosen because they are readily available, reasonably priced, and compatible with Arduino.[4]. Sutawas Janreung, Natthakun Wisitthiwong, Nontanan Yonsiri, Chaowanan Jamroen, and Thitiworada Odthon in 2023. Recently, water quality monitoring systems for aquaculture environments have been developed by previous researchers. A remote monitoring system was created by Wang et al. to enhance the aquaculture cages that are open to the ocean's automation management. Third-generation (3G) cellular technology was paired with an Android-based mobile operating system. To measure the factors associated with water quality, Vaddadi et al. used WSNs to develop portable water quality measurement units that were mounted on a floating platform. Nam et al. combined code division multiple access technology with the ZigBee network protocol to create a WSN-based remote water quality monitoring system for offshore aquaculture. An industry standard for packet-oriented mobile data is general packet radio service additionally an Android mobile phone platform was utilized by Huan et al. to present a wireless remote method for monitoring the water quality in aquaculture [5-7].

3. Working Procedure and Design

First, the ESP8266 must establish a station mode (STA) configuration by connecting to the router (local network). To accomplish this, update the source code to match what we have for the network SSID and password. Following a

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 01 Issue: 01 December 2023
Page No: 24 - 31
https://irjaeh.com

successful connection—the router should, of course, establish a connection to the Internet—the ESP8266 assumes the role of a client and begins communicating with the server. The block diagram is shown in Fig 1.

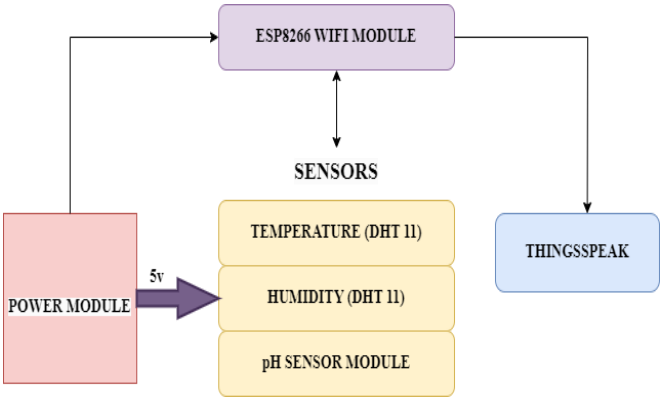


Fig 1: Block diagram

3.1 Algorithm for Establishing Wi-Fi Connection

Step 1: Initialize the serial port to communicate with the ESP8266.s
Step 2: Print the message "DHT-xx test!"
Step 3: Initialize the Wi-Fi module.
Step 4: connect to the Wi-Fi network.
Step 5: Print the message "_ _" when the Wi-Fi connection is not established.
Step 6: Delay for 500 milliseconds.
Step 7: Exit the loop and proceed to the next step.

3.2 Assemble data using DHT11

The successful establishment of an Internet connection does not guarantee that the data is automatically sent to the server. It must pass through a few stages before data is sent to the server. The data must first be accessible. Thus, the DHT sensor ought to be able to convert information from the physical world into data in the following ways:

3.2.1 Algorithm to Collect Data from DHT11 Sensor

Step 1: Initialization:
- Declare the library for the DHT sensor.

Step 2: Sensor Configuration:
- Define the sensor's pin (DHTPIN) and type (DHTTYPE), e.g., DHT11.

Step 3: Sensor Initialization:
- Initialize the DHT sensor instance with the configured pin and type using the DHT library.

Step 4: Setup Function:
- In the setup function:
- Initialize the DHT sensor using `dht. Begin ().

Step 5: Main Loop Function:
- In the loop function, perform the following steps in a continuous loop:

Step 6: Delay:
- Add a delay of 2 seconds (2000 milliseconds) to control the reading frequency.

Step 7: Read Sensor Data:
- Read the humidity (h), the temperature in Celsius (t), and the temperature in Fahrenheit (f) from the DHT sensor and store them in respective variables.

Step 8: Data Validity Check:
- Check if any of the sensor data values (h, t, or f) is not a number (NaN), indicating a failure to read data.

Step 9: Data Valid Path:
- If the data is valid, proceed with processing or using the sensor data.

Step 10: Data Invalid Path:
- If any of the sensor data is invalid (NaN):
- Print a message to the serial monitor: "Failed to read from DHT sensor!"
- Return from the loop function.

3.2.2Steps were taken at what time - ESP8266 uses a GET request to send data to the server.

Here, a GET request is useful once DHT11 has finished gathering data. To put it briefly, the server can update data by performing the following actions after receiving a URL from the ESP8266 containing the parameters temperature and humidity:

1. In the `loop` function, a URL request is constructed based on temperature (`t`) and humidity (`h`) values. The request is in the format of a URL with query parameters [8].
2. The `HTTP Request` function is called, passing the constructed URL request as an argument.
3. In the `HTTP Request` function:

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 01 Issue: 01 December 2023
Page No: 24 - 31
https://irjaeh.com

a. It checks if the Wi-Fi connection is established (`Wi-Fi. Status () == WL_CONNECTED`).
b. It creates a Bear-SSL Wi-Fi Client-Secure instance using a smart pointer and configures it to accept insecure (self-signed) certificates using client->set-Insecure ().
c. An `HTTP-Client` instance named `https` is created.
d. It checks if an HTTPS connection can be established with the specified URL using `https. begin(client, request)`.

4. If the HTTPS connection is established successfully, it proceeds to make an HTTP GET request using `https. GET ().
5. The code logs the constructed request URL, and the HTTPS response code, and if the response code is greater than 0 (indicating a successful request), it reads the response payload using `HTTPS.getString() and logs it.
6. If the response code is not greater than 0, indicating a failed request, it logs an error message with the details of the error using `https.errorToString(httpsCODE).c_str()`.
7. Finally, it ends the HTTPS connection using `https. End ().

These steps demonstrate how the code is structured to make an HTTPS GET request to a specified server with the given URL, handle the response, and log the relevant information. https.end (); To make things simpler, we can learn what the ESP8266 does by following these steps:

1. Establish a URL that updates every two seconds and contains the temperature and humidity values.
2. Launch HTTP and specify the location where the ESP8266 will operate.

The GET request method is used. Verify whether the request was successful or not. Print the https code if it doesn't work, and the payload otherwise.

3.3 Render chart from browser

3.3.1 Browser contacts server with GET request

The user must visit this URL to view the chart: https://khamec14thesis.000webhostapp.com/showdht11.php. As we can see, the browser sends a GET request to showdht11.php right away when the user accesses the URL above. The purpose of this page is to render, or "draw," the chart. However, there is one item under the hood that needs to be updated for the data to be updated appropriately [9].

1. Begin the 'load' function
2. Initialize variables
3. Set up an interval to repeatedly execute the following code:
   i. Create a new XML-Http-Request object and assign it to the variable 'xml-http'.
   ii. Set the URL variable 'URL' to "data.json".
   iii. Override the response MIME type to "application/json".
   iv. Define an 'onreadystatechange' function:

- Check if the ready state of the request ('this.readyState') is 4 (completed) and the status code ('this. status') is 200 (OK).

- If both conditions are met, proceed.

- Parse the response text as JSON.

- Extract the 'temp' and 'humi' properties from 'myArr' and assign them to 'json_temp' and 'json_humi' respectively.

- Print the values of 'json_temp' and 'json_humi' to the console.

   v. Open a new GET request with the 'xml-http' object using the 'URL' and set it to asynchronous (true).
   vi. Send the GET request.

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 01 Issue: 01 December 2023
Page No: 24 - 31
https://irjaeh.com

4. End of the 'load' function.

There is a section of this algorithm known as data JSON. It makes a GET request to data, as we can see by looking at the network tab in the client browser. Additionally, JSON via showdht11.php, though not directly [10].

4. Server Responds

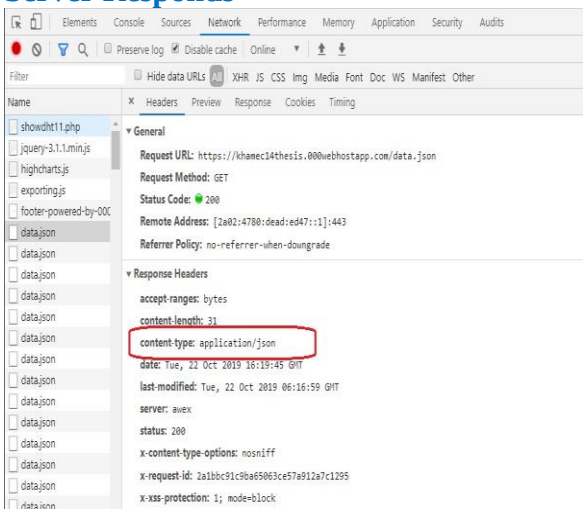


Fig 2: The server answers the request with a JSON string

The server answers the request with a JSON string, as seen in Fig 2. We can examine the Console tab for further information. The server response value is shown in Fig 3.

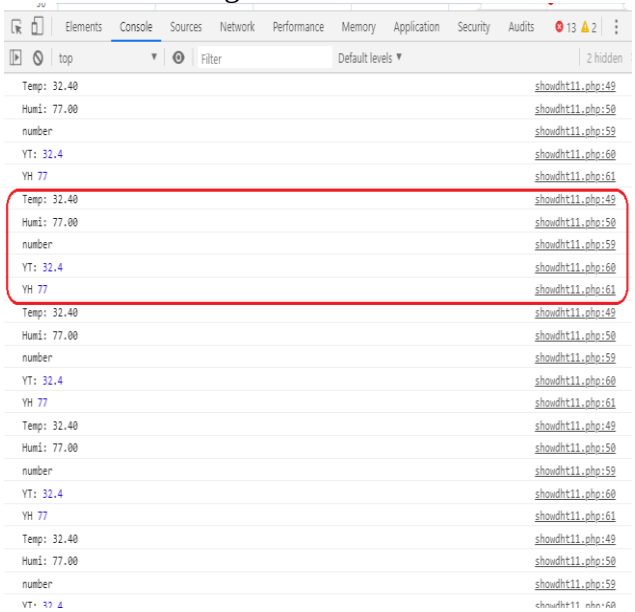


Fig 3 Server response value

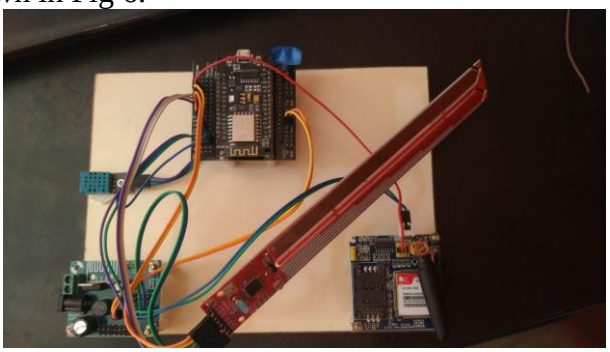
Thus, after parsing, this is the message's value. They correspond to lines 49, 50, 59, 60, and 61 in the code. The code is shown in Fig 4.

```
39  chart = this;
40  setInterval(function () {
41          var xmlhttp = new XMLHttpRequest();
42          var url = "data.json";
43          xmlhttp .overrideMimeType("application/json");
44          xmlhttp.onreadystatechange = function() {
45              if (this.readyState == 4 && this.status == 200) {
46                  var myArr = JSON.parse(this.responseText);
47                  json_temp = myArr['temp']
48                      json_humi = myArr['humi']
49                  console.log("Temp:", json_temp );
50              console.log("Humi:", json_humi );
51              }
52          };
53          xmlhttp.open("GET", url, true);
54          xmlhttp.send();
55
56          var x = (new Date()).getTime(),
57          y_temp = Number(json_temp),
58          y_humi = Number(json_humi);
59          console.log(typeof(y_temp));
60          console.log("YT:", y_temp);
61          console.log("YH", y_humi);
62
63          series_temp.addPoint([x, y_temp], true, true);
64          series_humi.addPoint([x, y_humi], true, true);
65          activeLastPointToolip(chart);
```

Fig 4: Code

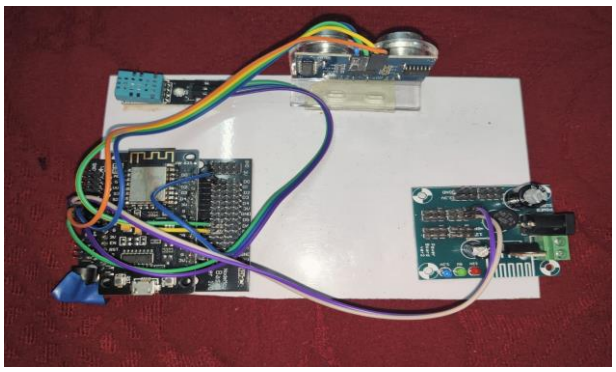
5. Experimented Results and Execution

After the number of trials and experiments are done to get the required results. Finally, the essential results are obtained which is shown in Fig 5. It is the prototype that is being suggested [11]. The output is shown in Fig 6.



(a)

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 01 Issue: 01 December 2023
Page No: 24 - 31
https://irjaeh.com



(b)

Fig 5 (a) and (b) prototype of the proposed model

Fig 6 Temperature and Humidity output

6. Future Scope

In the future, our project envisions an expansion that includes an automatic pH stabilizer unit. This advanced addition will significantly enhance the capabilities of our system, furthering its mission to monitor and protect aquatic environments. The key feature of this upcoming phase is the implementation of an intelligent auto-stabilizer unit. This unit will function in real-time, continuously assessing pH levels within the aquatic environment. When the pH strays from the desired range, the unit will take immediate action by engaging a motor-driven mechanism [12-14]. This motor-driven mechanism will be responsible for precisely adjusting the pH levels. It will carefully introduce either acidic or basic water into the environment, effectively counteracting any fluctuations in pH. The system will be programmed to execute these adjustments gradually, preventing sudden and potentially harmful shifts in pH that could negatively impact aquatic life. To ensure the system's effectiveness and adaptability, we plan to incorporate a feedback loop. Sensors will monitor the pH levels after stabilization, providing valuable data to the auto-stabilizer unit. This feedback mechanism will allow for fine-tuning of the motor's operations, ensuring that the pH remains within a narrow, predefined range. Our project will also expand its data analysis capabilities to encompass the monitoring and logging of actions taken by the auto-stabilizer unit. This data will offer insights into the system's performance and assist in refining its algorithms over time [15].

7. Conclusion

Amidst growing environmental concerns, our project represents a significant stride toward sustainable resource management. The ability to monitor temperature and pH levels in aquatic ecosystems with precision has far-reaching implications. It enables us to identify early warning signs of environmental stressors, such as pollution or climate change effects, and take targeted corrective actions. By harnessing the power of Node MCU and IoT technology, we're not just collecting data; we're fostering a new era of adaptive conservation practices. Moreover, our project embodies the spirit of collaboration and knowledge sharing. We understand that the challenges facing our planet's aquatic ecosystems are too immense for any one entity to tackle alone. As such, we're committed to open-source principles. All aspects of our Node MCU-based system, from hardware schematics to code, will be made readily available to the global scientific and conservation community. By sharing our innovations, we hope to catalyze a collective effort toward the preservation of these vital ecosystems, transcending geographical and

disciplinary boundaries. Together, we can drive positive change, promoting a future where aquatic environments thrive and sustain life for generations to come.

References

[1]. Pasika, S., & Gandla, S. T. (2020). Smart water quality monitoring system with cost-effective use of IoT. Heliyon, 6(7).

[2]. Menon, K. U., Divya, P., & Ramesh, M. V. (2012, July). Wireless sensor network for river water quality monitoring in India. In 2012 Third International Conference on Computing, Communication and Networking Technologies (ICCCNT'12) (pp. 1-7). IEEE.

[3]. Kurnia, R., Sari, M. W., Ciptadi, P. W., & Samodro, G. (2023, June). Android-based water quality measurement monitoring system development to determine the level of water pollution in Kranggan Village, Kulon Progo. In AIP Conference Proceedings (Vol. 2491, No. 1). AIP Publishing.

[4]. Razman, N. A., Wan Ismail, W. Z., Abd Razak, M. H., Ismail, I., & Jamaludin, J. (2023). Design and analysis of water quality monitoring and filtration systems for different types of water in Malaysia. International Journal of Environmental Science and Technology, 20(4), 3789-3800.

[5]. Jamroen, C., Yonsiri, N., Odthon, T., Wisitthiwong, N., & Janreung, S. (2023). A standalone photovoltaic/battery energy-powered water quality monitoring system based on narrowband internet of things for aquaculture: Design and implementation. Smart Agricultural Technology, 3, 100072.

[6]. Manoj, M., Dhilip Kumar, V., Arif, M., Bulai, E. R., Bulai, P., & Geman, O. (2022). State of the art techniques for water quality monitoring systems for fish ponds using IoT and underwater sensors: A review. Sensors, 22(6), 2088.

[7]. Susanti, N. D., Sagita, D., Apriyanto, I. F., Anggara, C. E. W., Darmajana, D. A., & Rahayuningtyas, A. (2022, January). Design and implementation of water quality monitoring system (temperature, pH, TDS) in aquaculture using IoT at low cost. In 6th International Conference of Food, Agriculture, and Natural Resource (IC-FANRES 2021) (pp. 7-11). Atlantis Press.

[8]. Junaidi, A., & Kartiko, C. (2020, March). Design of pond water quality monitoring system based on the Internet of things and pond fish market in real-time to support the Industrial Revolution 4.0. In IOP Conference Series: Materials Science and Engineering (Vol. 771, No. 1, p. 012034). IOP Publishing.

[9]. Chen, C. H., Wu, Y. C., Zhang, J. X., & Chen, Y. H. (2022). IoT-based fish farm water quality monitoring system. Sensors, 22(17), 6700.

[10]. Huan, J., Li, H., Wu, F., & Cao, W. (2020). Design of water quality monitoring system for aquaculture ponds based on NB-IoT. Aquacultural Engineering, 90, 102088.

[11]. Zaini, A., Wulandari, D. P., & Wulandari, R. (2020, November). Data Visualization on Shrimp Pond Monitoring System Based on Temperature, pH, and DO (Dissolved Oxygen) with IoT. In 2020 International Conference on Computer Engineering, Network, and Intelligent Multimedia (CENIM) (pp. 1-6). IEEE.

[12]. Wongmeekaew, T., Boonkirdram, S., & Phimphisan, S. (2019, August). Wireless Sensor Network for Monitoring of Water Quality for Pond Tilapia. In 2019 Twelfth International Conference on Ubi-Media Computing (Ubi-Media) (pp. 294-297). IEEE.

[13]. Postolache, O., Pereira, J. D., & Silva Girão, P. (2014). Wireless sensor network-based solution for environmental monitoring: water quality assessment case

study. IET Science, Measurement & Technology, 8(6), 610-616.

[14]. Shareef, Z., & Reddy, S. R. N. (2019, March). Design and wireless sensor Network analysis of water quality monitoring system for aquaculture. In 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC) (pp. 405-408). IEEE.

[15]. Ma, Y., & Ding, W. (2018, October). Design of intelligent monitoring system for aquaculture water dissolved oxygen. In 2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC) (pp. 414-418). IEEE.