

Monitoring System of Water Level Based on IoT

Alfian Syam¹, Wahyudi^{2*)}

^{1,2}Departement of Electrical Engineering, Faculty of Engineering, Universitas Diponegoro, Semarang, Indonesia

^{*)}wahyuditinom@elektro.undip.ac.id

Abstract: The water level monitoring system is an important component in water tank management to maintain water availability and prevent the water in the tank from becoming empty or overflowing. The monitoring process using IoT is very important to increase the efficiency and sustainability of system operations. This paper presents the design and implementation of an Internet of Things (IoT)-based water level monitoring system that provides automatic, real-time status updates accessible remotely. The system utilizes an ESP32 microcontroller integrated with an HC-SR04 ultrasonic sensor to measure the distance to the water surface; this data is processed to determine water level, volume, and tank capacity percentage. Measurement results are displayed on a 16x2 I2C LCD and transmitted via WiFi to the Blynk platform for monitoring on Android devices, with a buzzer included as a warning indicator. Control and data processing algorithms are implemented on the ESP32 microcontroller within the IoT framework. Test results demonstrate that the system accurately reads water levels in real-time with stable responsiveness, the sensor maintains an average error rate of less than 3%, and the system issues accurate warning notifications when preset minimum and maximum water levels are reached.

Keywords—Internet of Things; ESP32; level sensor; water monitoring; Blynk; active buzzer

1. INTRODUCTION

Clean water supply is a fundamental public service requirement, particularly for water utility companies responsible for distributing water to the community. Within a distribution network, the stability of water volume in storage reservoirs directly impacts service continuity, as fluctuations in water levels can trigger supply disruptions and pressure instability. Inadequate monitoring of water levels can lead to operational risks; such as water shortages, tank overflows, and distribution failures, thereby directly diminishing service quality. In addition to causing consumer dissatisfaction, these operational problems also increase infrastructure maintenance costs due to repeated mechanical failures caused by pressure surges within the piping system [1][2]. The process of managing clean water, which is still carried out manually by field personnel, has several limitations, including reliance on physical inspections, delayed information during rapid changes in water levels, the potential for reading errors due to human factors, and the inability to access data in real-time, all of which result in less responsive operational decision-making [1-4]. The development of the Internet of Things (IoT) is widely applied in water level monitoring across rivers, reservoirs, and storage tanks. Integrating ultrasonic sensors with microcontrollers such as the ESP32 has been proven capable of producing accurate, cost-effective monitoring systems accessible remotely via cloud platforms or mobile applications, with ultrasonic sensor accuracy reported to reach 82–85% under both stable and wavy water conditions [3-5]. Ultrasonic sensors were selected because they can measure water surface distance without physical contact, withstand humid environments, and easily integrate with microcontrollers [6]. Other studies show that integrating LCDs, buzzers, and Blynk dashboards on Arduino/ESP32-based systems facilitates users in monitoring system

conditions in real-time and receiving anomaly notifications [1][2][5].

This paper contributes to the design and implementation of an IoT-based water level monitoring system prototype using an ESP32, an HC-SR04 ultrasonic sensor, an I2C LCD, a buzzer, and the Blynk platform. Water levels in the tank can be monitored accurately in real-time and accessed remotely, thereby enhancing operational effectiveness [1-3][5].

2. METHODE

2.1 Hardware Design

Hardware design was carried out by establishing the system's functional requirements: the ability to detect water surface distance in real-time, process this data to determine water level and volume, and display the information via both local and remote interfaces. Based on these requirements, a block diagram was created to illustrate the system's overall workflow, incorporating the HC-SR04 ultrasonic sensor for distance measurement, the ESP32 as the main processing unit, and an LCD, buzzer, and the Blynk platform as output units, as shown in Fig. 1. This block diagram served as the fundamental reference for determining component interconnections prior to installation. This structured architecture ensures smooth data flow and seamless integration among all embedded hardware modules. Each component is connected to the ESP32 via specific I/O pins according to its characteristics and communication protocols. The HC-SR04 ultrasonic sensor is connected to pins GPIO13 (*Trigger*) and GPIO12 (*Echo*) to transmit and receive sound wave signals.

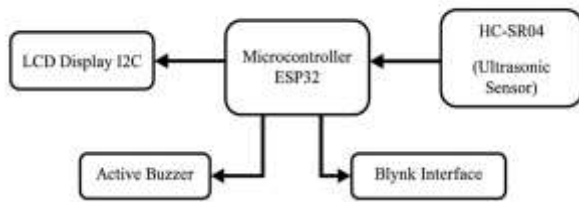


Fig. 1. System block diagram.

The 16x2 LCD uses an I2C interface module connected to pins GPIO21 (SDA) and GPIO22 (SCL) to save microcontroller GPIO pin usage. The active buzzer is directly controlled via digital pin GPIO25 as an audio indicator trigger. This dedicated pin mapping optimizes hardware resource allocation while maintaining electrical stability during continuous system operation. Details of pin interconnection mapping between hardware components are presented in Table 1. Wiring all components to the ESP32 microcontroller forms an integrated electrical schematic. The electrical wiring diagram of this prototype is shown in Fig. 2.

Table 1: Components and pins connected to ESP32.

No	Component	Component Pin	Connected to ESP32
1	Sensor ultrasonik HC-SR04	VCC	5V/VIN
		GND	GND
		TRIG	GPIO13
		ECHO	GPIO12
2	Buzzer Aktif	(+)	GPIO 25
		(-)	GND
		VCC	5V/VIN
3	LCD I2C 16X2	GND	GND
		SDA	GPIO21
		SCL	GPIO22

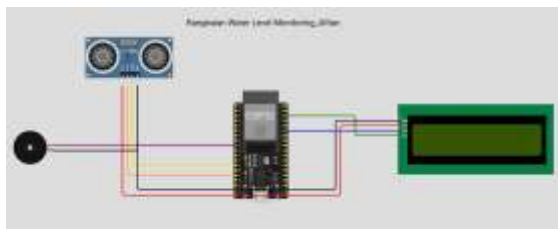


Fig. 2. System wiring diagram.

Physical component placement is designed separately between the measurement unit and the processing unit to protect electronic circuits in humid environments. The

ultrasonic sensor is mounted statically at the top of the container facing perpendicular to the water surface, while the ESP32 microcontroller and LCD are placed inside an external control box (Box X6). The physical appearance of the assembled and packaged prototype is shown in Fig. 3, while the schematic design of the sensor placement position on the tank is presented in Fig. 4.



Fig. 3. Physical prototype appearance.

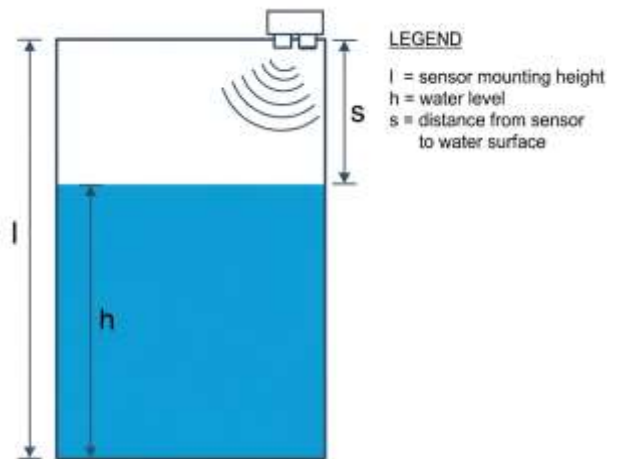


Fig. 4. Prototype placement design.

Calculations for water height and capacity percentage parameters are based on physical tank dimensions and sensor position. Water height (h) is obtained from the difference between the sensor mounting height from the tank bottom (l) and the water surface distance read by the sensor (s). The mathematical formulations for converting water height and volume percentage (%) are expressed in Equation (1) and Equation (2).

$$h = l - s \quad (1)$$

$$\text{Water volume (\%)} = \left(\frac{h}{l}\right) \times \text{Tank volume} \times 100\% \quad (2)$$

2.2 Software Design

Software design was implemented using the Arduino framework to manage the ESP32 microcontroller's workflow in a structured manner. The program utilizes a repetitive execution cycle, enabling simultaneous sensor readings, display updates, alarm activation, and IoT data transmission. Distance data processing from the ultrasonic sensor employs a two-stage digital filtering process to eliminate noise caused by water ripples. The first stage uses a median sampling method to select the median value from multiple readings. The second stage applies an Exponential Moving Average (EMA) algorithm to ensure that distance fluctuations remain stable and smooth. The system classifies water level conditions into five operational status levels based on the calculated water volume (%). These status categories are determined by specific water volume (%) thresholds, as summarized in Table 2.

Table 2: Water level status determination.

Water volume (%)	Status
Water volume ≤ 5	Empty
$5 < \text{Water volume} \leq 15$	Nearly Empty
$15 < \text{Water volume} \leq 85$	Medium
$85 < \text{Water volume} \leq 95$	Nearly Full
$95 < \text{Water volume}$	Full

The active buzzer-based audio alarm indicator is controlled dynamically using four different sound patterns to distinguish specific water volume crisis levels. The triggering of these audio alarm rhythms is arranged linearly based on status and water volume (%) evaluation as follows.

1. Sound Pattern 1 (Very fast beep): Emits intermittent beep tones with a 100 ms interval when water is at Empty status ($\leq 5\%$) as a danger warning of depleted water supply.
2. Sound Pattern 2 (Medium beep): Emits intermittent beep tones with a 300 ms interval when water is at Nearly Empty status (6% - 15%) as an early warning signal of low water supply.
3. Sound Pattern 3 (Slow beep): Emits intermittent beep tones with an 800 ms interval when water is at Nearly Full status (86% - 94%) to inform that the container is approaching maximum capacity.
4. Sound Pattern 4 (Continuous beep): Emits a continuously active tone without interruption (HIGH) when water reaches Full status ($\geq 95\%$) as the main emergency alarm to prevent overflow.
5. Normal/Medium Condition: Under Medium status (16% - 85%), the buzzer is positioned in an OFF state because the water volume is within the safe operational range.

Data display on the 16x2 LCD applies an automatic page-switching (*paging system*) every 4 seconds to alternate between water height and volume (Liters) data, as well as status and water volume (%) data. Simultaneously, data packets are wirelessly transmitted to the Blynk platform on Virtual Pins V0, V1, and V2 via a WiFi connection. The overall software control logic flow is comprehensively illustrated in the flowchart in Fig. 5.

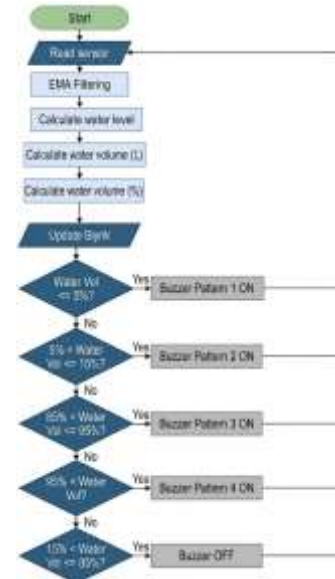


Fig. 5. System flowchart.

2.3 Software Design

Calibration of the HC-SR04 ultrasonic sensor was performed through comparative testing between digital sensor readings and an analog precision measuring tool (ruler) over a distance range of 5 cm to 30 cm. Testing results show that the sensor has an average error below 3% with excellent reading consistency for measuring water level. The linear correlation between sensor readings and actual distances was analyzed using the least-squares linear regression method, resulting in the calibration equation shown in Equation (3).

$$y = 1.0319x - 0.3986 \quad (3)$$

The variable y represents the estimated actual distance in centimeters (cm), while variable x represents the raw reading value obtained from the ultrasonic sensor. A slope value (a) of 1.0319, which is very close to one, and a minimal intercept value (b) of -0.3986 indicate that the sensor has a highly proportional and linear response to real distance changes. This mathematical regression equation is integrated directly into the ESP32 microcontroller programming algorithm as an automatic correction factor (offset) to minimize systematic error accumulation during water volume conversions in liters and percent. Thus, reading precision remains maintained to support continuous monitoring operations under stable water surface conditions. The calibration test data distribution along with the linear regression curve between sensor readings and actual values is shown in Fig. 6.

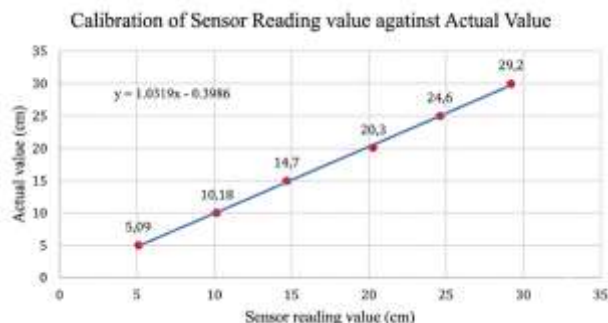


Fig. 6. Sensor calibration.

3. RESULT AND DISCUSSION

3.1 Monitoring

Data visualization functionality testing was conducted to ensure reliable real-time water level information presentation on both the local display and IoT dashboard. Testing was executed by step-by-step conditioning the water level in the tank across five main operational conditions: very low level (empty), low (nearly empty), normal range (medium), high (nearly full), and very high (full). The ESP32 microcontroller processes distance data from the HC-SR04 ultrasonic sensor into numerical parameters of water height (cm) and volume estimate (Liters), as well as water volume (%) and operational status. This step-by-step evaluation verifies that the processing algorithm consistently converts physical distance measurements into accurate operational indicators across all volume ranges.

This evaluation also tested character space allocation efficiency on the local LCD display and wireless data transmission consistency to the Blynk application. The 16x2 LCD screen was evaluated on executing its page-switching method (*paging system*) every 4 seconds to present numerical and qualitative data alternately. Line clearing execution on the LCD successfully eliminated leftover characters during text transitions. Concurrently, IoT data transmission was tested to confirm that water volume (%) (Virtual Pin V0), volume (Virtual Pin V1), and water height (Virtual Pin V2) were transmitted synchronously to the mobile application. Successful data synchronization ensures all main parameters can be monitored constantly from a distance without experiencing packet loss (*data loss*). In addition, executing these concurrent tasks demonstrates that the ESP32 handles dual-interface data presentation efficiently without causing display lag or communication timeouts. Details of test scenarios, evaluated water level conditions, and output indicators consisting of LCD display photos and Blynk interfaces are fully presented in Table 3.

Table 3: Test scenarios and indicators.

Scenario	Tested Condition	Indicator	
		LCD Display	Blynk Interface
Test 1	Very low water level		
Test 2	Water level in normal range		
Test 3	Low water level		
Test 4	High water level		
Test 5	Very high water level		

Evaluation results indicate that data presentation on both the local LCD interface and Blynk dashboard exhibits high precision across all scenarios. Height, volume, and percentage values displayed on the Blynk app are synchronized with local LCD numerical data. Although a wireless transmission delay ranging from 1 to 2 seconds occurs due to WiFi network latency on Blynk, data updates remain continuous and stable for remote monitoring.

3.2 Warning Alarm

Dynamic testing of the active buzzer audio alarm was conducted to evaluate sound rhythm triggering precision during continuous water volume changes. The first test was executed during a tank filling process starting from an initial 50% water volume up to maximum capacity. At 45 seconds into filling, the water volume crossed the 86% threshold (nearly full), automatically activating the buzzer with Sound

Pattern 3 (slow beep with an 800 ms interval). As filling continued to the critical 95% mark (full) at 60 seconds, the system instantly shifted tone modulation to Sound Pattern 4 (continuous beep without stopping) as an emergency alert to prevent overflow. Water volume increase trends and buzzer sound pattern transition trigger points during tank filling are shown in Fig. 7.

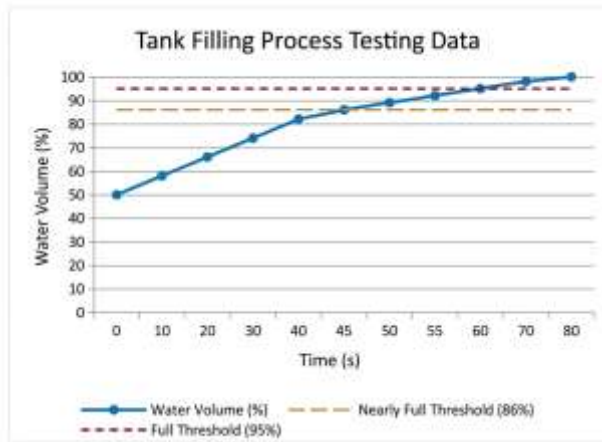


Fig. 7. Buzzer response on tank filling process.

The second test was executed during a tank drainage process to observe alarm signal precision as water volume gradually depleted from 50% capacity. When water drainage reached 15% at 40 seconds, the system crossed the nearly empty threshold and instantly triggered Sound Pattern 2 (medium beep with a 300 ms interval). Further volume decrease down to 5% at 60 seconds crossed the empty threshold, automatically instructing the system to emit Sound Pattern 1 (very fast beep with a 100 ms interval) as a critical water depletion alert. Meanwhile, within the 16% - 85% volume range (medium status), the buzzer remained OFF. Water volume depletion trends along with buzzer warning threshold lines during drainage are shown in Fig. 8. Dynamic test results visualized in these graphs confirm that threshold conditioning logic in the software program executes buzzer pattern transitions promptly. Non-blocking timing logic using `millis()` proved effective in running alarm rhythms without causing processing bottlenecks (*blocking*) during sensor reading or display updates.

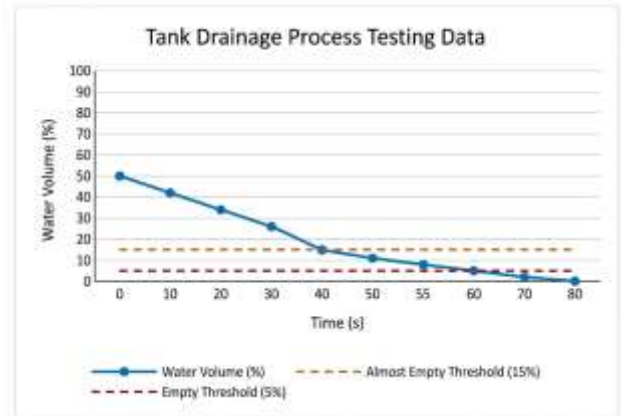


Fig. 8. Buzzer response on tank drainage process.

3.3 System Performance

A technical advantage of this system is the implementation of an Exponential Moving Average (EMA) filter, which effectively suppresses noise caused by water surface ripples and prevents drastic output fluctuations. The combination of local and remote monitoring results in a system that is far more responsive than manual methods. However, Blynk data transmission performance remains dependent on WiFi network stability, and the HC-SR04 sensor is susceptible to reduced accuracy in high-humidity environments; consequently, further evaluation of protective measures is required for industrial-scale implementation [7-10]. The remote monitoring system enables company management to detect critical water anomalies at an early stage—unconstrained by time or location—while also serving as a foundation for the transition toward modernized smart water management [11-12].

4. CONCLUSION

An IoT-based water level monitoring system has been designed to enhance monitoring efficiency. Test results demonstrate that the HC-SR04 ultrasonic sensor accurately measures the water surface distance, maintaining an average error rate of less than 3% compared to standard measuring instruments; this data is processed by an ESP32 microcontroller into real-time parameters—including water level, estimated volume, capacity percentage, and five tank status classifications. The processed information is synchronized with a local 16x2 I2C LCD screen and a Blynk mobile dashboard via WiFi, supported by an active buzzer alarm that responsively generates four distinct sound patterns based on critical capacity thresholds: Pattern 1 (rapid beeping, 100 ms interval) for "Empty" ($\leq 5\%$); Pattern 2 (moderate beeping, 300 ms interval) for "Near Empty" ($6\% - 15\%$); Pattern 3 (slow beeping, 800 ms interval) for "Near Full" ($86\% - 94\%$); Pattern 4 (continuous, uninterrupted beeping) for "Full" ($\geq 95\%$); and an "OFF" state for "Moderate" levels ($16\% - 85\%$). For future industrial application development, it is recommended to upgrade the system using industrial-grade waterproof/submersible sensors and to incorporate features

such as data logging, push or email notifications, an IP65-rated waterproof protective housing, a backup power supply (UPS), and direct relay integration for automatic pump actuation.

REFERENCES

- [1] C. P, S. Chethana, R. Manasa, K. M. Meghana, and D. D. Nikhitha, "IoT-based smart water distribution and monitoring system," *International Journal of Research and Analytical Reviews (IJRAR)*, vol. 10, no. 1, pp. 674–678, Mar. 2023.
- [2] S. Geetha and S. Gouthami, "Internet of things enabled real time water quality monitoring system," *IEEE Access*, vol. 5, pp. 12963–12970, 2017, doi: 10.1109/ACCESS.2017.2726390.
- [3] Hafni, M. Zen, dan Irwan, "Implementation of an Internet of Things (IoT) Based Water Level Monitoring System Using Ultrasonic Sensors and the Blynk Application," dalam *Proc. 2nd International Conference on Islamic Community Studies (ICICS)*, pp. 2658–2665.
- [4] P. Raskar, N. Sonawane, and S. Thorat, "IoT+Camera based water tank level and quality monitoring system," *International Journal on Science and Technology (IJSAT)*, vol. 16, no. 4, pp. 1–5, Oct.–Dec. 2025.
- [5] V. Sharma, A. Padale, Y. Bhosale, Y. Jangid, K. Patel, and N. Pawar, "Water level monitoring system using IoT," *International Research Journal of Innovations in Engineering and Technology (IRJIET)*, vol. 9, no. 3, pp. 272–274, Mar. 2025, doi: 10.47001/IRJIET/2025.903038.
- [6] A. Mulyani, D. Kurniadi, and R. E. Saputra, "IoT-based real-time river monitoring and early flood warning using ESP32 and HC-SR04," *Journal of Novel Engineering Science and Technology (JNEST)*, vol. 5, no. 1, pp. 55–62, Apr. 2025, doi: 10.56741/jnest.v5i01.1252.
- [7] R. Vadrucchio, G. Salvadori, and A. Tumino, "Smart metering and Internet of Things for efficient water management," *Procedia Computer Science*, vol. 219, pp. 1–9, 2023.
- [8] J. H. Ng, H. Seah, and C. M. Pang, *Digitalising Water: Sharing Singapore's Experience*. London, U.K.: IWA Publishing, 2020, doi: 10.2166/9781789061871.
- [9] Shahad K, A. A. Kamara, R. Giri, H. C. Ineza, Y. J. Djabatassiba, and S. Kallam, "Smart water management technology: An IoT-based solution for efficient water resource management," *International Journal of Innovative Research in Technology (IJIRT)*, vol. 11, no. 10, pp. 3128–3137, Mar. 2025.
- [10] M. S. Heriyawan, T. Santiko, and E. Pujianto, "Design and implementation of tank level monitoring system using Internet of Things (IoT)," *International Journal of Artificial Intelligence Research (IJAIR)*, vol. 9, no. 1.1, pp. 1–14, 2025.
- [11] Y. Dai, Z. Huang, N. Khan, and M. S. Labbo, "Smart water management: Governance innovation, technological integration, and policy pathways toward economic and ecological sustainability," *Water*, vol. 17, no. 13, art. no. 1932, Jun. 2025, doi: 10.3390/w17131932.
- [12] S. Howell, Y. Rezgui, and T. Beach, "Water Utility Decision Support through the Semantic Web of Things," *Environmental Modelling and Software*, vol. 102, pp. 94–114, 2023, doi: 10.1016/j.envsoft.2018.01.006.