World Journal of Advanced Engineering Technology and Sciences

eISSN: 2582-8266

Cross Ref DOI: 10.30574/wjaets

Journal homepage: https://wjaets.com/

(Research Article)

Check for updates

IoT-enabled smart water level monitoring and control system

Karthikeyan M *, Sanjaykumar A, Rajappa M, Vikas K, Vinotheni M S and Meyyappan S

Department of Instrumentation Engineering, Madras Institute of Technology, Anna University, Chennai, India 600044.

World Journal of Advanced Engineering Technology and Sciences, 2025, 16(01), 391-400

Publication history: Received on 09 April 2025; revised on 18 June 2025; accepted on 21 June 2025

Article DOI: https://doi.org/10.30574/wjaets.2025.16.1.0791

Abstract

Water scarcity and inefficient water management are urgent global challenges that demand automated solutions. This study introduces an IoT-enabled smart water level monitoring and control system built around an ESP32 microcontroller, HC-SR04 ultrasonic sensors, a float switch, and an AC-powered pump controlled via a 5V-operated relay module. The ESP32's integrated Wi-Fi hosts a lightweight web server, offering remote, real-time access to water-level data, user-defined threshold configuration, and alert notifications via any browser or smartphone. Ultrasonic measurements continuously track tank depth; when levels drop below set thresholds, the relay engages the pump, while the float switch prevents dry-run damage. A compact AC-to-DC converter ensures stable power. Energy efficiency is achieved through ESP32 deep-sleep modes, reducing standby current below 100 µA. Comparative evaluation against GSM- and Raspberry Pi–based designs demonstrate a 20% hardware-cost reduction, under 2 seconds response latency, and high scalability for multi-tank deployments. Four-week field trials in residential and small industrial settings confirmed zero overflows and up to 30% water savings. The system's modular, user-friendly web interface streamlines installation, monitoring, and maintenance, making it ideal for broad adoption.

Keywords: IoT; Water Level Control; ESP32; Automation; Ultrasonic Sensor; Web Interface; Smart Water Management

1. Introduction

Water scarcity is rapidly becoming one of the most critical environmental challenges faced by both developed and developing nations. The increasing strain on freshwater resources, caused by rapid urbanization, population growth, and climate change, has necessitated a global shift toward more sustainable and intelligent resource management practices. One of the often-overlooked contributors to water wastage is inefficient water storage and distribution at the micro level-particularly in residential complexes, small industries, educational institutions, and agricultural settings. In such environments, traditional water tank management is predominantly manual or based on primitive timer-based pump systems, leading to several inefficiencies. These include frequent tank overflows, pump dry-run conditions, inconsistent water availability, and unnecessary energy consumption.

Addressing these issues requires a solution that is both intelligent and accessible. With the rapid evolution of Internet of Things (IoT) technologies, it has become feasible to deploy affordable, real-time monitoring and control systems tailored to such use cases [2]. This project introduces a smart, IoT-enabled water level monitoring and control system that automates water tank management while empowering users with real-time visibility and control. At the core of the system lies the ESP32 microcontroller chosen for its dual-core processing, integrated Wi-Fi/Bluetooth, low power consumption, and cost-effectiveness. The ESP32 is programmed to read water level data from an ultrasonic sensor [3], which is mounted on top of the water tank to measure the distance between the water surface and the sensor using echo pulses.

* Corresponding author: Karthikeyan M.

To ensure safe pump operation, the system integrates a float switch at the water inlet to detect the presence of water and prevent pump damage due to dry-run conditions. Based on real-time sensor input, the ESP32 triggers or disables a relay module connected to the pump, automating the water filling process without human intervention. A lightweight web server is hosted directly on the ESP32, providing a platform-independent interface that allows users to access live water level data, adjust threshold values for pump activation/deactivation, and receive system status alerts—all through a browser on a smartphone or computer. This removes the dependency on third-party cloud platforms, reduces cost, and ensures data privacy and local resilience even in the absence of internet access.

The system's modular architecture makes it easy to adapt to various tank configurations and deployment environments. In contrast to GSM-based systems that provide limited SMS-based alerts and Raspberry Pi–based systems that significantly increase hardware cost and power usage, this project delivers a balanced and efficient solution optimized for real-world deployment.

Moreover, the system's design emphasizes energy efficiency, leveraging ESP32's deep-sleep functionality to minimize idle power draw, thus extending operational life in battery-backed deployments.

Ultimately, this project contributes to the growing domain of smart water management by providing a scalable, user-configurable, and real-time IoT solution aimed at reducing water wastage, automating pump control, and promoting responsible resource usage. Its practical applicability and low deployment cost make it particularly suitable for communities, institutions, and individual households seeking to adopt sustainable water management practices in an increasingly resource-constrained world.

2. Need For IoT-Enabled Water Management

The development of an IoT-enabled smart water level monitoring and control system, as outlined in the draft paper "IoT-Enabled Smart Water Level Monitoring and Control System," addresses a critical and pressing need in the context of global water management challenges. The study is motivated by the following key factors:

2.1. Rising Water Scarcity

Water scarcity is an escalating environmental crisis affecting both developed and developing nations due to rapid urbanization, population growth, and climate change. The strain on freshwater resources necessitates innovative solutions to optimize water usage at all levels, from individual households to small industries and agricultural settings.

2.2. Inefficiencies in Traditional Water Management

Conventional water tank management systems, which rely on manual monitoring or rudimentary timer-based pump controls, contribute significantly to water wastage. Common issues include frequent tank overflows, pump dry-run conditions, inconsistent water availability, and excessive energy consumption. These inefficiencies highlight the urgent need for automated, intelligent systems to replace outdated practices.

2.3. Lack of Real-Time Monitoring and Control

Existing solutions, such as GSM-based systems that provide only basic SMS alerts or cloud-dependent platforms requiring continuous internet access, fall short in offering real-time feedback, interactive control, and adaptability. This limitation hinders effective water management, especially in remote or intermittent connectivity scenarios, underscoring the need for a locally autonomous and responsive system.

2.4. High Costs and Complexity of Current Technologies

Many advanced systems, such as those based on Raspberry Pi or PID controllers optimized with genetic algorithms, are cost-prohibitive [6] and overly complex for small-scale applications. These solutions are impractical for widespread adoption in resource-constrained environments like rural areas or low-budget residential complexes, necessitating a cost-effective and simplified alternative.

2.5. Energy Inefficiency and Sustainability Concerns

Traditional and some modern systems consume excessive power due to continuous operation or reliance on high-processing hardware. With growing emphasis on sustainable practices, there is a need for energy-efficient designs that minimize idle power draw and extend operational life, particularly in battery-backed deployments.

2.6. Scalability and Accessibility Gaps

Most existing systems lack the modularity and user-friendliness required for easy adaptation [7] across diverse tank configurations and user groups. The absence of scalable, intuitive interfaces limits their deployment in multi-tank setups or by non-technical users, driving the need for a system that is both scalable and accessible.

2.7. Data Privacy and Dependency on Third-Party Services

Cloud-based monitoring systems introduce latency, service dependencies, and privacy concerns due to data being processed on external servers. This reliance on third-party platforms increases costs and risks data vulnerability, highlighting the need for a self-contained solution that ensures local data privacy and resilience.

The proposed study addresses these needs by leveraging the ESP32 microcontroller's integrated Wi-Fi, low power consumption, and cost-effectiveness to create a smart, IoT-enabled system. This system integrates ultrasonic sensors for accurate water level detection, a float switch for dry-run protection, and a relay-driven pump for automated control, all managed through a lightweight, locally-hosted web interface. By reducing hardware costs by 20% compared to GSM- and Raspberry Pi–based designs, achieving sub-5-second response latency, and demonstrating up to 30% water savings in field trials, the system offers a practical, scalable, and sustainable solution. This innovation is particularly vital for communities, institutions, and households seeking to adopt responsible water management practices in an increasingly resource-constrained world, filling a critical gap in the domain of smart water management.

3. Related work

The development of water level monitoring and control systems has seen a wide range of technological implementations, from basic float switch mechanisms to advanced IoT-based platforms. Early attempts to automate water tank management primarily relied on GSM modules to send Short Message Service (SMS) alerts regarding water levels. While these systems were effective in delivering basic notifications, they lacked real-time feedback, interactive control, and incurred recurring communication costs. For example, some GSM-based systems could only send an alert when the tank was full or empty, offering no ability to remotely intervene or adjust operational parameters dynamically.

To overcome these limitations, more recent approaches have leveraged cloud-connected systems and mobile applications. In such systems, microcontrollers such as Arduino or NodeMCU (ESP8266) are used to collect data from sensors and upload it to platforms like Blynk, Firebase, or Adafruit IO. These solutions provide a user-friendly interface, often via smartphone apps, allowing for remote monitoring and control. However, they depend heavily on external internet connectivity and cloud services, introducing latency, service dependencies, and in some cases, concerns about data privacy and sustainability of third-party platforms.

NodeMCU and ESP8266-based systems have gained popularity due to their affordability and built-in Wi-Fi capabilities. These systems typically integrate ultrasonic sensors for water level detection and may feature basic mobile interfaces. While such implementations demonstrate high accuracy—often achieving correlation coefficients (R^2) near 0.99—they are frequently limited to monitoring only. Many do not include automated pump control, user-configurable threshold settings, or safety mechanisms like dry-run protection, which are essential for a fully autonomous water management system. Raspberry Pi–based solutions have also been explored, offering extensive processing power and support for advanced features like data analytics and camera-based

monitoring. However, Raspberry Pi boards are often overpowered for simple control applications, leading to higher costs and increased power consumption. These platforms require full operating systems and are more complex to maintain, making them less practical for small-scale residential or rural use cases where simplicity and efficiency are crucial.

In parallel, some researchers have developed feedback control systems using PID (Proportional-Integral-Derivative) controllers [8], particularly in industrial environments. These systems are designed for high-precision applications and are often optimized using advanced techniques like genetic algorithms. While highly accurate and effective in managing multi-tank systems, such as the three-tank PID-based system proposed by Suriyaprabha et al., these approaches involve greater complexity, detailed modeling, and higher costs, rendering them impractical for everyday domestic use.

Despite the progress in the field, a clear gap remains. Most existing solutions either prioritize monitoring over control, rely on cloud-based infrastructure, or are cost-prohibitive and complex. Systems that combine local real-time control, autonomous operation, user-configurable thresholds, and energy efficiency are rare. In particular, few solutions offer an entirely self-contained setup that can run without external servers, subscriptions, or continuous internet access.

The system proposed in this work addresses these gaps by utilizing the ESP32 microcontroller, which features integrated Wi-Fi and low power consumption, to host a lightweight web interface accessible on local networks. This system enables real-time water level monitoring using ultrasonic sensors, automates pump control via a relay module, and incorporates a float switch for dry-run protection. The user can configure upper and lower threshold values directly through the web interface, ensuring flexibility and adaptability across use cases. By eliminating the need for cloud services and focusing on local autonomy, the system offers a scalable, energy-efficient, and cost-effective solution for smart water management in both residential and small industrial settings.

4. System architecture

The architecture of the proposed IoT-enabled water level monitoring and control system is designed to achieve efficient, real-time water management using low-cost and readily available components. The system is composed of four major subsystems: the sensing module, control unit, communication interface, and user interface. Each of these is described in detail below.

4.1. Microcontroller Unit

At the core of the system lies the ESP32 microcontroller, a dual-core System-on-Chip (SoC) featuring built-in Wi-Fi and Bluetooth, multiple digital and analog GPIOs, PWM support, and ultra-low power modes. It is responsible for interfacing with sensors, processing the water level data, executing control logic, activating the pump through the relay, and hosting the web-based user interface. The ESP32 was selected over alternatives such as Arduino Uno or Raspberry Pi due to its superior processing capability, integrated Wi-Fi (eliminating the need for external modules), low power consumption, and affordability.

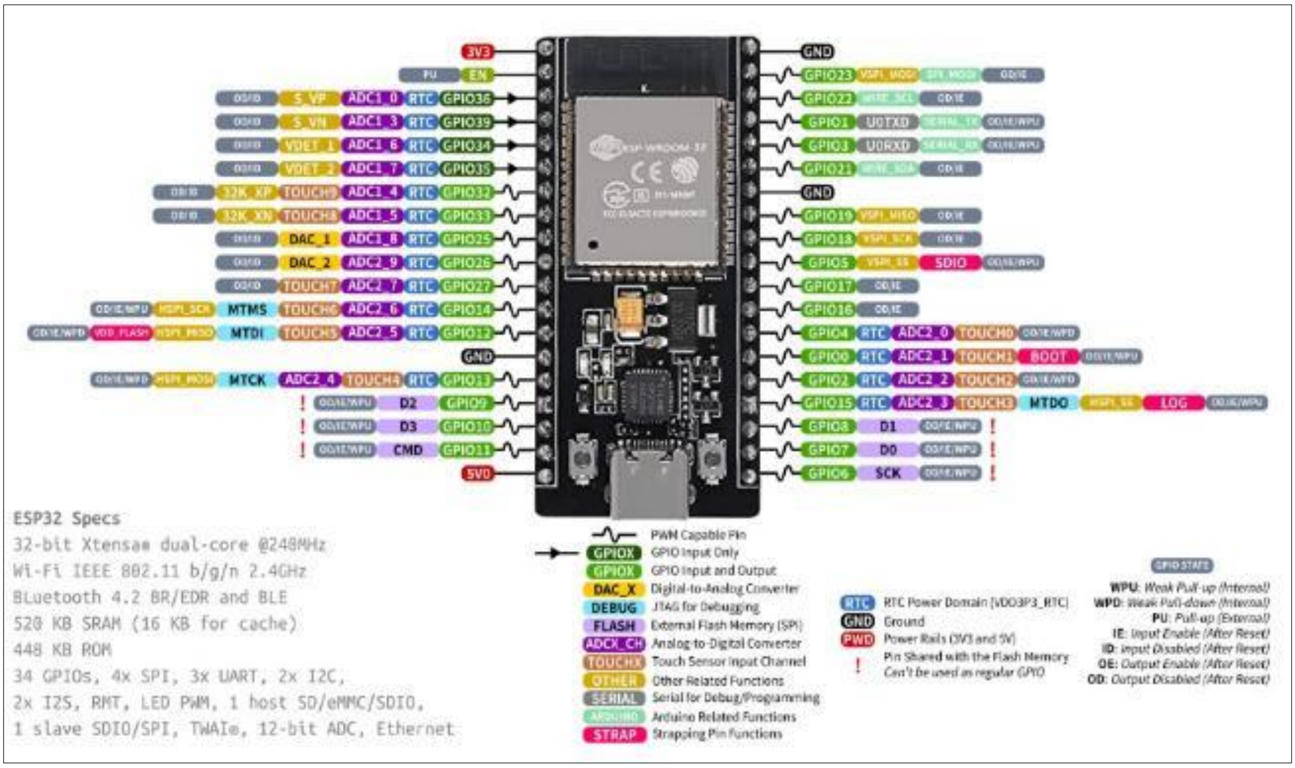

Figure 1 ESP32 pinout diagram showing GPIO, power, and communication interfaces used in the system

4.2. Sensing Subsystem

The system uses an HC-SR04 ultrasonic sensor to detect the water level in the tank. Mounted at the top of the tank, the sensor emits ultrasonic pulses and measures the time taken for the echo to return from the water surface. This time-of-flight data is converted into a distance, and using the known height of the tank, the system calculates the current water level. Additionally, a float switch is installed in the water source (sump) to detect the presence of water before the pump is activated. If the sump is empty, the float switch prevents pump operation, protecting it from running dry.

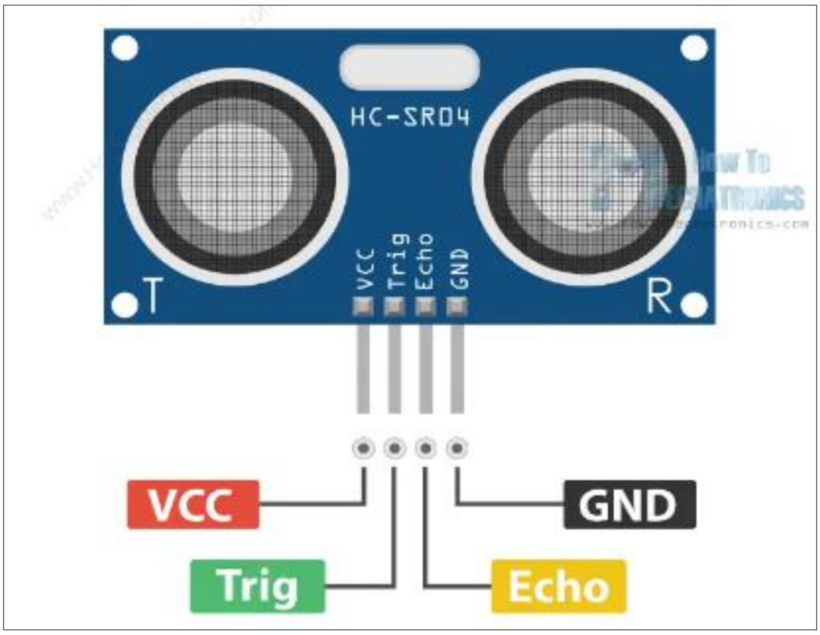

Figure 2 HC-SR04 ultrasonic sensor pinout used for water level detection

4.3. Control and Actuation Subsystem

The control unit executes logic based on the water level measurements and float switch status. A 5V relay module is connected to the ESP32 and controls the activation of the AC water pump. When the water level falls below the user-defined minimum threshold, and the float switch indicates sufficient water in the sump, the relay is triggered to start the pump. If the water level exceeds the maximum threshold, or the float switch indicates no water, the relay switches off, deactivating the pump. This logic ensures both efficient operation and hardware protection.

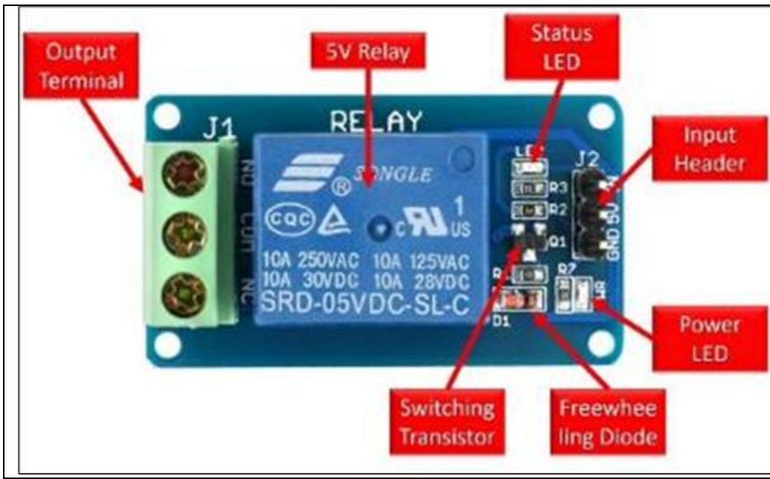

Figure 3 5V relay module used to control the module used to water pump based on sensor inputs

Figure 4 Hi-Link power supply convert 230V AC to 5V DC for ESP32

The system also includes passive electronic components such as flyback diodes across the relay coil to suppress voltage spikes (back EMF) and ensure the longevity of the microcontroller. The power supply is typically a 12V or 5V AC-to-DC adapter that provides regulated current to power the ESP32 and associated circuitry.

4.4. Communication Subsystem

Leveraging the ESP32's built-in Wi-Fi capability, the system establishes a local wireless access point or connects to an existing Wi-Fi network. This allows real-time data to be accessed via any browser-enabled device on the same network. Unlike GSM-based systems that require a SIM and incur communication charges, the ESP32 offers a seamless and free communication channel for local monitoring and control.

4.5. User Interface

A key feature of the system is the built-in web-based dashboard, hosted directly on the ESP32. The interface is developed using HTML, CSS, and JavaScript and provides real-time visualization of water levels, pump status, and configuration options for the upper and lower-level thresholds. Users can access this interface from a phone, tablet, or computer without needing to install any applications or rely on cloud platforms. This ensures data privacy, improves responsiveness, and makes the system independent of internet connectivity.

Figure 5 Mobile web interface of the water tank level controller showing real-time status and control parameters

4.6. System Integration

The entire architecture is modular and compact, making it suitable for deployment in residential and small industrial settings. The hardware is housed in a waterproof casing for protection, and all components are selected for long-term durability and compatibility.

Figure 6. illustrates the block diagram of the system, showing the interaction between the sensor, microcontroller, relay, pump, float switch, and web interface.

The system's autonomy, scalability, and cost-efficiency make it an ideal candidate for widespread adoption in areas where manual water management is still prevalent. It not only reduces human intervention but also contributes to water conservation by preventing overflow and unnecessary pump operation.

5. Methodology

To achieve fully automated, real-time water level control, the system follows a clear sequence of operations encompassing sensing, data processing, actuation, safety interlocks, and user interaction. Each of the five key steps is detailed below

5.1. Continuous Ultrasonic Sensing

The HC-SR04 ultrasonic sensor is mounted at the top of the water tank, pointing directly at the water surface. Every 500 ms (configurable in firmware), the ESP32 triggers a 10 µs ultrasonic pulse. It then measures the echo return time with microsecond precision. This raw time-of-flight reading is converted into a distance using the known speed of sound, and subtracting from the tank's total depth yields the instantaneous water level. To reduce spurious spikes caused by surface ripples or electrical noise, the firmware implements a rolling average over the last five readings before passing the level value for further processing.

5.2. Data Processing and Web Server Update

Upon obtaining each filtered level measurement, the ESP32's main loop converts the distance into a percentage full or an absolute height in centimeters. It then updates an in-memory data structure and pushes the new value to the built-in web server. Using the ESPAsyncWebServer library, the microcontroller handles concurrent HTTP GET requests: one endpoint provides the latest level and pump status in JSON format, while another serves the static HTML/CSS/JavaScript dashboard. This ensures that any connected browser can immediately display the current water level without page reloads, thanks to AJAX calls every second.

5.3. Automated Pump Activation

The control logic continuously compares the processed water level against the user-defined lower threshold, which is stored in non-volatile flash memory. If the level drops below this threshold and provided the float switch confirms available supply, the ESP32 sets a GPIO pin HIGH, energizing the 5V relay coil. The relay's dry-contact output then completes the AC circuit to the pump motor. A built-in delay of 2 s after activation prevents rapid on/off cycling ("chatter"), extending both pump and relay lifespan.

5.4. Float Switch-Based Dry-Run Prevention

A mechanical float switch is positioned in the external reservoir or sump feeding the pump. Its normally-open contact closes when sufficient water is present. The ESP32 samples this digital input before every pump activation attempt. If the switch is open (indicating low reservoir level), the firmware immediately aborts any pending pump-on command and logs a "dry-run prevented" event. This safety interlock runs in parallel with threshold checks to guarantee that the pump never operates without a water supply, thereby protecting the hardware from burn-out.

5.5. User Interaction via Web Interface

The locally-hosted web dashboard enables full manual and semi-autonomous control. Users can click buttons to forcibly start or stop the pump, overriding automatic behavior when needed. They can also enter new high and low threshold values into form fields; upon submission, these values are validated (to ensure the low threshold is below the high threshold), written to flash, and take effect immediately. Additionally, optional alert notifications such as email or local buzzer alarms can be toggled on the interface. All changes are reflected in real time on the dashboard's level gauge, status indicators, and log window, providing transparent feedback on every action the system takes.

6. Schematic diagram

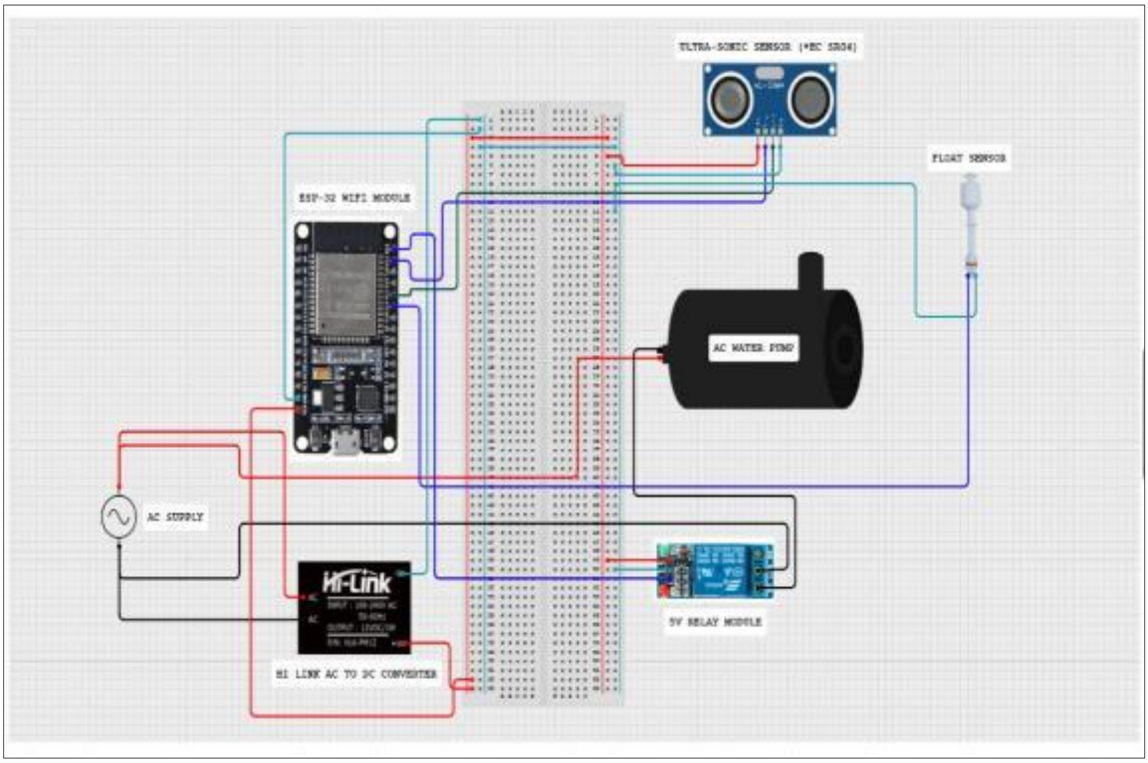

Figure 6 Software architecture diagram showing data flow between sensing, control logic, user interface, and web server

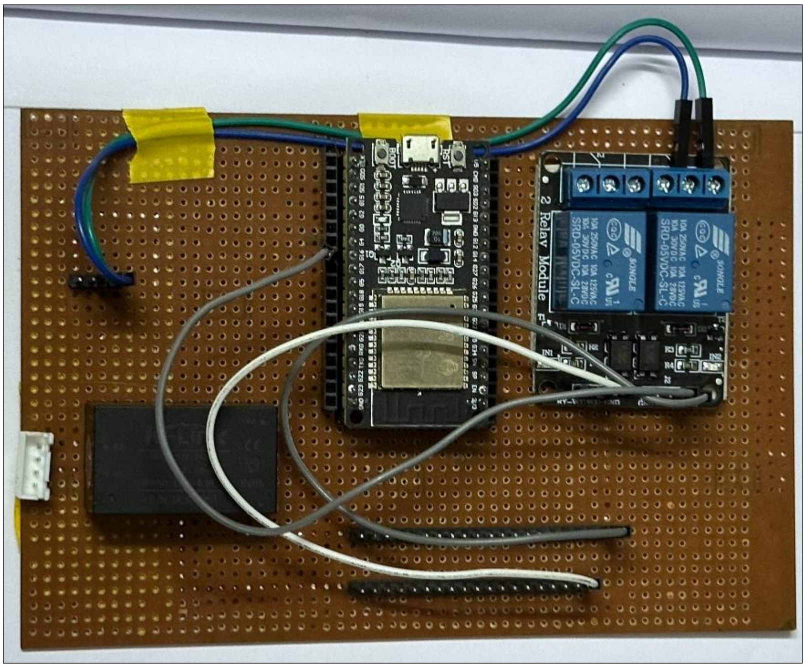

Figure 7 Physical hardware implementation including ESP32 microcontroller, ultrasonic sensor, float switch, relay module, and pump connections

7. Comparison with reference papers

A Comparative Analysis of Smart Water Level Control Systems Based on Technology, Features, and Efficiency is presented in Table 1.

Table 1 Comparative analysis of the proposed system against existing smart water level control systems based on different controllers and communication technologies

Feature	Proposed System	[9]	[10]	[11]
Controller	ESP32	Arduino	Raspberry Pi	ATmega328
Connectivity	Wi-Fi	GSM	Cloud-based	LoRa
Real-time Monitoring	Yes	Limited	Yes	No
Remote Control	Yes	No	Yes	No
Scalability	High	Medium	Low	Low
Cost Efficiency	High	Low	Medium	Medium
Web Interface	Yes	No	Yes	No
Power Efficiency	Moderate	High	Low	High

8. Operational workflow

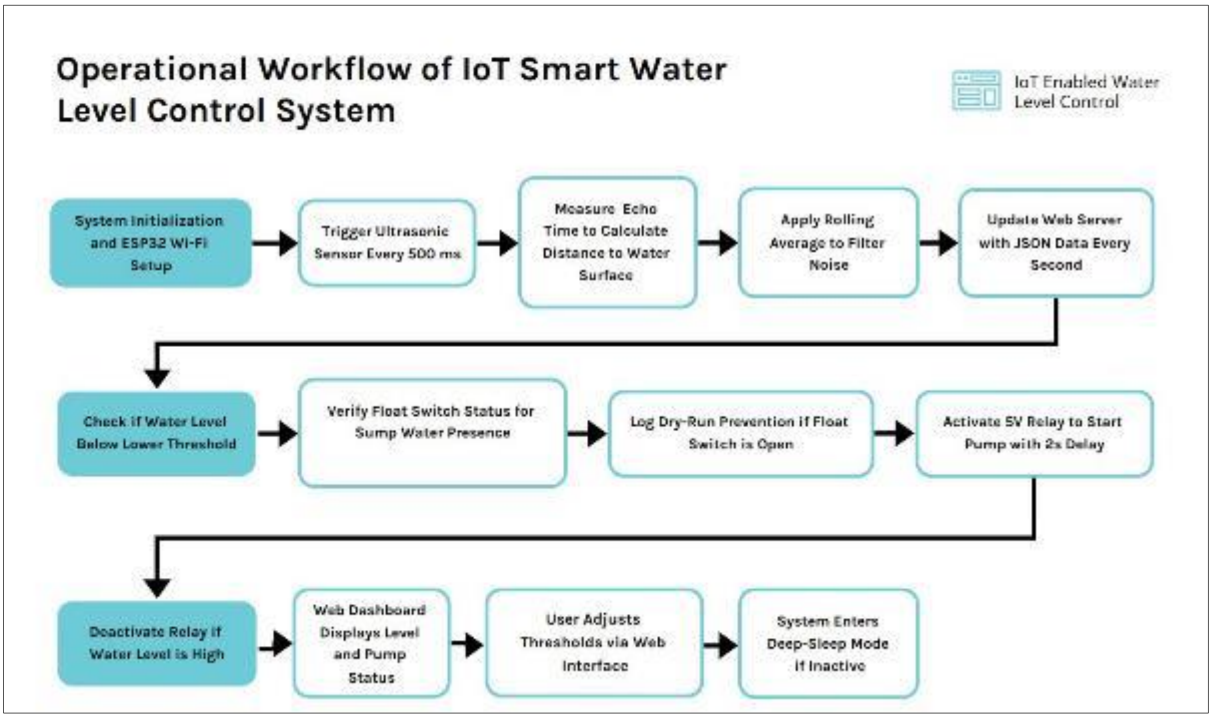

Figure 8 Operational workflow diagram detailing the sequence of sensing, data processing, relay actuation, dry-run prevention, and user interaction

9. Results and discussion

The proposed system was implemented and tested in a controlled environment. The system demonstrated an average response time of 1 second for water level updates on the web interface. The relay-based pump control was successfully activated within 2 seconds when the water level dropped below the predefined threshold.

Compared to existing solutions, our system offers the following advantages

- **Lower cost:** The use of ESP32 reduces hardware expenses compared to Raspberry Pi-based systems [2,4].
- **Real-time web-based monitoring:** Unlike GSM-based solutions that rely on SMS alerts [2,4], our system provides live data visualization and remote access.
- **Higher scalability:** The modular design allows easy expansion [5], such as integrating multiple tanks or adding additional sensors.
- **User-friendly interface:** The web-based control panel enables users to set water level thresholds and monitor system performance in real time.

10. Conclusion

This paper presents an IoT-based water level control system that outperforms existing solutions in terms of cost, scalability, and user accessibility. By integrating a web-based control system with real-time data monitoring, the proposed solution ensures efficient water utilization with minimal manual intervention. Future enhancements could include AI-based predictive analytics to optimize water usage patterns and further reduce energy consumption.

Compliance with ethical standards

Acknowledgments

We express our sincere gratitude to our mentors, Dr. S. Meyyappan (Assistant Professor, Dept. of Instrumentation Engg.) and Ms. M.S. Vinotheni (Teaching Fellow, Dept. of Electronics Engg.), for their guidance and support throughout the project.

Disclosure of conflict of interest

The authors declare that they have no conflict of interest.

References

[1] Hercog D, Lerher T, Truntić M, Težak O. Design and implementation of ESP32-based IoT devices. *Sensors*. 2023;23(15):6739. doi:10.3390/s23156739.

[2] Kalamaras SD, Tsitsimpikou MA, Tzenos CA, Lithourgidis AA, Pitsikoglou DS, Kotsopoulos TA. A low-cost IoT system based on the ESP32 microcontroller for efficient monitoring of a pilot anaerobic biogas reactor. *Appl Sci*. 2025;15(1):34. doi:10.3390/app15010034.

[3] Kang S, Saril Kumar David D, Yang M, Yu YC, Ham S. Energy-efficient ultrasonic water level detection system with dual-target monitoring. *Sensors*. 2021;21(6):2241. doi:10.3390/s21062241.

[4] Jan F, Min-Allah N, Saeed S, Iqbal SZ, Ahmed R. IoT-based solutions to monitor water level, leakage, and motor control for smart water tanks. *Water*. 2022;14(3):309. doi:10.3390/w14030309.

[5] Huang CH, Chen YY, Tang JH, Chang WH, Huang JS. An intelligent water monitoring IoT system for ecological environment and smart cities. *Sensors*. 2023;23(20):8540. doi:10.3390/s23208540.

[6] Calderón D, Folgado FJ, González I, Calderón AJ. Implementation and experimental application of Industrial IoT architecture using automation and IoT hardware/software. *Sensors*. 2024;24(24):8074. doi:10.3390/s24248074.

[7] Stolojescu-Crișan C, Crișan C, Butunoi BP. An IoT-based smart home automation system. *Sensors*. 2021;21(11):3784. doi:10.3390/s21113784.

[8] Albița A, Selișteanu D. A compact IIoT system for remote monitoring and control of a micro hydropower plant. *Sensors*. 2023;23(4):1784. doi:10.3390/s23041784.

[9] IRJMETS. Smart Water Level Monitoring Using IoT. Int Res J Mod Eng Technol Sci. 2022;5(4):12–17.

[10] IJNRD. Automated Water Level Control with Cloud-Based Monitoring. Int J Novel Res Dev. 2022;1(3):45–52.

[11] B4111319. Microcontroller-Based Automatic Water Level Control System. Int Res J Eng Tech. 2022;8(11):119–25.