RESEARCH ARTICLE OPEN ACCESS

# Design and Implementation of an IoT-Based Water Level Monitoring System

Halima Sulaiman Saleh, Imrana Abubakar, Hamza Ahmad Abubakar, Rofihat Ibrahim, Hassan Muhammad, and Andrew Ezra Garba
*(Department of Computer Engineering, Kaduna Polytechnic, Nigeria)*

----------------------------------------************************----------------------------------

**Abstract:**
Continuous monitoring of water storage tanks is important for reducing overflow, preventing inadequate water supply, and improving the management of pumping systems. This study presents the design and implementation of an Internet of Things (IoT)-based water level monitoring system using an ESP32 microcontroller, two float-switch sensors, a DC pump interface, and the Blynk Cloud platform. The sensing unit detects minimum and maximum water-level conditions, while the ESP32 processes the sensor states and transmits the monitored status through Wi-Fi for remote observation. The prototype incorporates a solar/battery power arrangement and regulated voltage stages for the electronic components. Functional and simulated performance tests were conducted for water-level detection, IoT communication, system stability, and response time. The reported test sequence produced 100% detection across the three monitored level conditions, 98% overall IoT communication reliability, and average pump ON and OFF response times of 0.79 s and 0.76 s, respectively. The results indicate that the proposed architecture provides a compact approach to remote water-level monitoring; however, the communication performance remains dependent on the availability and quality of the Wi-Fi/internet connection.

***Keywords* —** Blynk Cloud, ESP32, Float Switch, Internet Of Things, Remote Monitoring, Solar Power, Water Level Monitoring.

----------------------------------------************************----------------------------------

## I. INTRODUCTION

Water is essential to domestic, agricultural, industrial, and commercial activities, and reliable storage and distribution are therefore important to effective water management. Water pumping systems are commonly used to transfer water from reservoirs, boreholes, or underground sources to storage tanks. Without continuous observation, these systems may experience overflow, insufficient supply, unnecessary pump operation, and associated energy or equipment losses [1], [2].

The development of embedded systems, wireless communication, cloud computing, and the Internet of Things (IoT) has enabled water systems to be monitored remotely. In a typical IoT architecture, sensors acquire environmental information, a microcontroller processes the data, and a communication network transfers the information to a cloud platform for visualization and monitoring [3], [4].

The ESP32 is particularly suitable for such applications because it combines processing capability with integrated Wi-Fi and Bluetooth connectivity and multiple programmable input/output interfaces. Its single-chip wireless capability can reduce hardware complexity in IoT prototypes [5], [6].

Previous water-monitoring studies have used ultrasonic, conductive, and other sensing technologies, as well as different microcontrollers and communication modules. These approaches demonstrate the feasibility of automated monitoring, but they may introduce additional hardware, sensing complexity, or cost depending on the application [7]–[9]. The present work therefore focuses on a relatively simple architecture based on float switches and an ESP32 for real-time remote water-level monitoring.

## II. RELATED WORK

Yelekar et al. [9] developed an IoT-based smart water-level monitoring system using an ESP32, ultrasonic sensing, GSM communication, and ThingSpeak. Their reported system incorporated additional functions such as temperature compensation and leakage detection. Ahmed et al.

  

[7] used float switches, an 89S52 microcontroller, an ESP8266 Wi-Fi module, relays, and a web server for multilevel water-tank monitoring and control. Barbade et al. [8] presented an automatic tank-filling system based on level sensing and relay control but without IoT-based remote observation.

Related architectures have also targeted broader water-distribution monitoring and control beyond single-tank level sensing [11], and recent systematic reviews of IoT-based monitoring and control frameworks summarize the range of sensing, communication, and control strategies adopted across this domain [12]. These studies demonstrate that automated level detection and remote observation can reduce dependence on manual inspection. The present design differs principally in combining the sensing, processing, and Wi-Fi communication functions in the ESP32 while using two float switches to identify the minimum and maximum monitored levels. This provides a straightforward architecture for remote status monitoring without requiring a separate Wi-Fi controller.

## III. SYSTEM DESIGN AND IMPLEMENTATION

### A. *System Architecture*

The proposed system consists of a power supply unit, sensing unit, monitoring unit, switching interface, and IoT communication unit. The float switches provide discrete level signals to the ESP32. The controller processes these signals and communicates the resulting status to Blynk Cloud through Wi-Fi. The switching interface forms part of the prototype pumping arrangement.

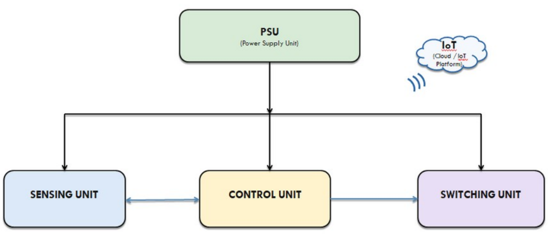


Fig. 1. Overall architecture of the IoT-based water level monitoring system.

### B. *Hardware Components*

The principal components used in the prototype are summarized in Table I. The design uses a 24 V battery source, two DC-DC buck-converter stages, an ESP32 controller, float switches, a relay interface, and associated protection and switching components.

TABLE I. PRINCIPAL COMPONENTS AND FUNCTIONS

| Component | Specification | Function |
|---|---|---|
| ESP32 microcontroller | Wi-Fi/Bluetooth enabled | Sensor processing and IoT communication |
| Float switches | Two level switches | Minimum and maximum level detection |
| Relay interface | JQX-16F (T91), 30 A class | Pump-status/ switching interface |
| Buck converters | 24→12 V; 12→5 V | Voltage regulation |
| DC pump | 60 W | Water pumping |
| Battery | 24 V, 220 Ah | Primary prototype power source |
| Diode / transistor / resistor / capacitor | 1N4007 / BC547 / 4.7 kΩ / 1000 µF, 50 V | Protection, switching and filtering |

### C. *Power Supply*

The prototype uses a 24 V battery as its primary source. A first buck-converter stage reduces the battery voltage to 12 V for the relay/pumping interface, while a second stage provides 5 V for the ESP32 and low-voltage electronics. The source report estimated a total connected load of 62 W and calculated a minimum design capacity of 12.91 Ah after applying a 25% design margin. A 24 V, 220 Ah battery was adopted for the prototype, providing a substantial capacity margin relative to that calculation.

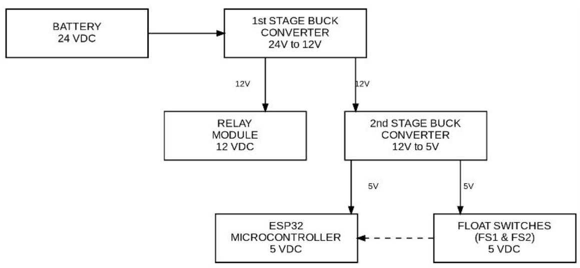


Fig. 2. Power-supply block diagram.

### D. *Sensing Unit*

Two float switches are positioned at predetermined levels in the storage tank. Float Switch 1 represents the minimum monitored level, while Float Switch 2 represents the maximum level. Movement of the float changes the electrical state of the switch, producing a discrete input that is read by the ESP32. This approach avoids the signalprocessing requirements associated with continuous-distance sensing and is consistent with the use of float switches for liquid-level detection [10].

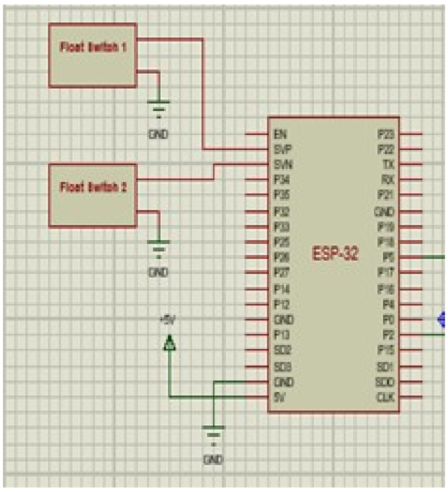


Fig. 3. Float-switch sensing circuit connected to the ESP32.

### *E. Monitoring and IoT Communication*

The ESP32 functions as the central monitoring controller. The two sensor inputs are read continuously, the corresponding water-level condition is determined, and the status is transmitted to Blynk Cloud through the ESP32's built-in Wi-Fi interface. The pin assignment used in the prototype is shown in Table II.

TABLE II. ESP32 PIN ASSIGNMENT

| Interface | ESP32 Pin |
|---|---|
| Float Switch 1 | SVP (GPIO 36) |
| Float Switch 2 | SVN (GPIO 39) |
| Pump-status interface | GPIO 5 |
| LED indicator | GPIO 2 |
| Wi-Fi | Built-in |
| Supply | 5 V |

### *F. Complete Hardware and Software Implementation*

The electronic components were assembled according to the system design. The float switches were connected to the ESP32 input pins, while the regulated supply powered the controller and associated electronics. The relay, transistor, diode, resistor, and capacitor were incorporated into the pumping-system interface. The complete hardware arrangement is shown in Fig. 4.

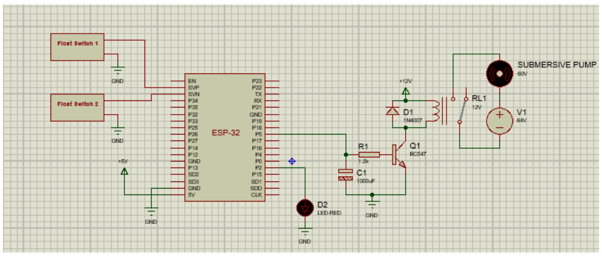


Fig. 4. Complete hardware connection diagram.

The control program was developed in the Arduino IDE using C/C++. The software initializes the ESP32 interfaces, establishes Wi-Fi connectivity, connects to Blynk Cloud, reads both float switches, updates the monitored status, and repeats the process continuously. The corresponding flow sequence is shown in Fig. 5.

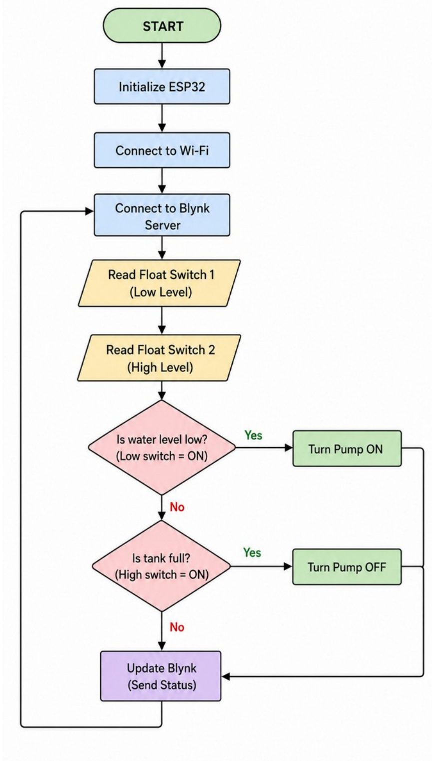


Fig. 5. Software flowchart for the monitoring process.

After testing, the electronics were enclosed to reduce exposure to dust, moisture, and accidental contact. The completed prototype and internal wiring are shown in Figs. 6 and 7.

  

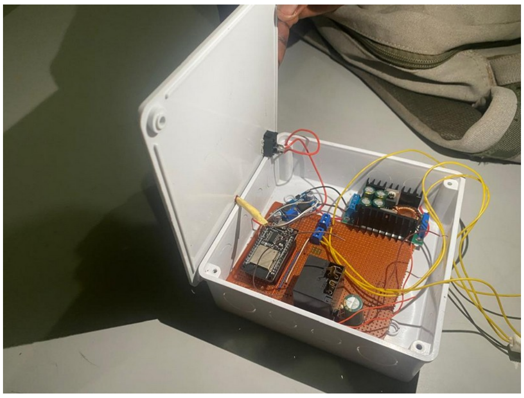

Fig. 6. Internal wiring of the packaged prototype.

Fig. 7. Packaged prototype enclosure.

## IV. RESULTS AND DISCUSSION

### A. Functional Performance

The functional test sequence examined low-level detection, high-level detection, Wi-Fi connection, Blynk monitoring, and the power supply. Each reported condition produced the expected response in the source report, as summarized in Table III.

TABLE III. FUNCTIONAL TEST OF THE DEVELOPED SYSTEM

| Test Parameter | Expected Response | Observed Response | Status |
|---|---|---|---|
| Low-level detection | Pump ON | Pump ON | Pass |
| High-level detection | Pump OFF | Pump OFF | Pass |
| Wi-Fi connection | ESP32 connects | Connected | Pass |
| Blynk monitoring | Status updated | Updated | Pass |
| Solar/battery supply | Stable supply | Stable | Pass |

### B. Water-Level Detection

Five trials were reported for each of three monitored conditions: below minimum, between minimum and maximum, and at maximum. All five trials in each condition were recorded as correct, giving 100% reported detection for each condition. Because the source document explicitly describes these trials as simulated, the results should be interpreted as prototype/simulation results rather than a statistically validated field accuracy measurement.

TABLE IV. WATER-LEVEL DETECTION TEST RESULTS

| Condition | Trials | Successful | Reported Accuracy |
|---|---|---|---|
| Below minimum | 5 | 5 | 100% |
| Between minimum and maximum | 5 | 5 | 100% |
| At maximum | 5 | 5 | 100% |

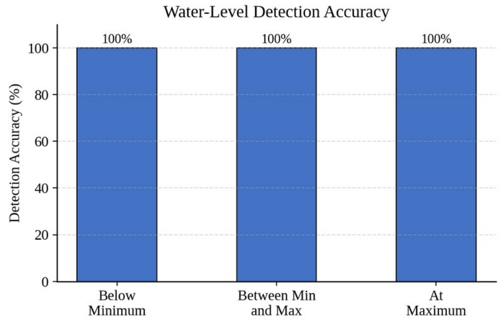


Fig. 8. Reported water-level detection accuracy.

### C. IoT Communication Performance

The IoT communication tests reported 100% success for Wi-Fi connection and Blynk server connection over 10 attempts each. Water-level updates and pump-status updates each recorded 49 successful events out of 50 attempts, while general communication recorded 98 successful events out of 100 attempts. The resulting reported communication reliability was therefore 98% for the overall test set.

TABLE V. IoT COMMUNICATION TEST RESULTS

| Parameter | Attempts | Success | Failed | Reliability |
|---|---|---|---|---|
| Wi-Fi connection | 10 | 10 | 0 | 100% |
| Blynk server connection | 10 | 10 | 0 | 100% |
| Water-level updates | 50 | 49 | 1 | 98% |

  

| | | | | |
|---|---|---|---|---|
| Pump-status updates | 50 | 49 | 1 | 98% |
| General communication | 100 | 98 | 2 | 98% |

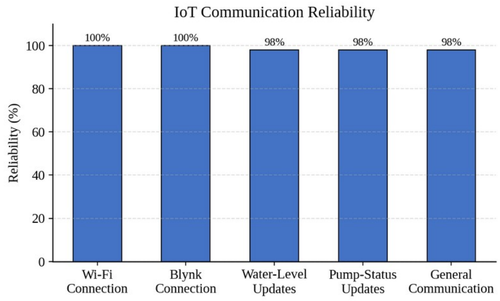


Fig. 9. Reported IoT communication reliability.

The reported 98% communication reliability indicates that the prototype was able to transmit most monitored events during the stated test sequence. In practical deployment, however, performance will depend on Wi-Fi coverage, internet availability, cloud-service connectivity, and local network conditions.

### *D. Response Time and System Stability*

The reported five-trial response-time measurements gave an average of 0.79 s from low-level detection to pump ON and 0.76 s from high-level detection to pump OFF. The source report also presented an overall stability/reliability calculation of 445 successful operations out of 450 operations, corresponding to 98.89%. These figures describe the reported prototype test sequence and should not be generalized beyond the stated test conditions.

TABLE VI. REPORTED RESPONSE-TIME RESULTS

| Trial | Low-level to Pump ON (s) | High-level to Pump OFF (s) |
|---|---|---|
| 1 | 0.82 | 0.74 |
| 2 | 0.76 | 0.77 |
| 3 | 0.79 | 0.75 |
| 4 | 0.81 | 0.79 |
| 5 | 0.78 | 0.76 |
| Average | 0.79 | 0.76 |

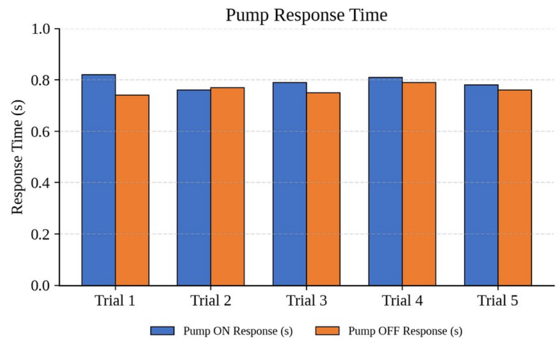


Fig. 10. Reported pump response times across five trials.

## V. CONCLUSION

An IoT-based water level monitoring system was designed and implemented using an ESP32 microcontroller, two float switches, a pumping-system interface, and Blynk Cloud. The architecture integrates water-level sensing, local processing, Wi-Fi communication, and remote status observation in a compact prototype. The reported test sequence produced correct detection for all simulated water-level trials, 98% overall IoT communication reliability, and average response times of 0.79 s for pump ON and 0.76 s for pump OFF.

The results support the feasibility of the proposed architecture for remote monitoring of storage-tank water levels. Nevertheless, the reported experiments were simulated/prototype tests, and actual field performance should be established through longer-duration trials under varying water-level cycles, network conditions, power availability, and environmental conditions.

## REFERENCES

[1] F. Jan, N. Min-Allah, S. Saeed, S. Z. Iqbal, and R. Ahmed, "IoT-based solutions to monitor water level, leakage, and motor control for smart water tanks," Water, vol. 14, no. 3, p. 309, 2022.

[2] A. Mangalraj, "Smart water management system powered by the Internet of Things (IoT) for real-world applications," in Internet of Things, pp. 129–142, Chapman and Hall/CRC, 2023.

[3] A. Paul and R. Jeyaraj, "Internet of Things: A primer," Human Behavior and Emerging Technologies, vol. 1, no. 1, pp. 37–47, 2019.

[4] C. Z. Zulkifli, S. Garfan, M. Talal, A. H. Alamoodi, A. Alamleh, I. Y. Ahmaro, and H. H. Chiang, "IoT-based water monitoring systems: A systematic review," Water, vol. 14, no. 22, p. 3621, 2022.

[5] D. Hercog, T. Lerher, M. Truntič, and O. Težak, "Design and implementation of ESP32-based IoT devices," Sensors, vol. 23, no. 15, p. 6739, 2023.

[6] A. A. Sneineh and A. A. Shabaneh, "Design of a smart hydroponics monitoring system using an ESP32 microcontroller and the Internet of Things," MethodsX, vol. 11, p. 102401, 2023.

  

[7] A. Ahmed, V. D. Chaudhari, I. S. Jadhav, M. N. Patil, and A. D. Vishwakarma, "Multilevel water level control & monitoring of multiple tanks," Journal Impact Factor, vol. 3, p. 66, 2018.

[8] G. M. Barbade, C. Shreyas, V. Vedant, V. Vaibhav, and P. Umesh, "Automatic water tank filling system with water level indicator," Indian Journal of Microprocessors and Microcontroller (IJMM), vol. 1, no. 2, pp. 1–7, 2021.

[9] R. Yelekar, T. D. Tency, S. Madbushi, and S. Chaudhari, "IoT-based smart water level monitoring," in Proc. 2023 IEEE 20th India Council International Conference (INDICON), pp. 1136–1142, Dec. 2023.

[10] V. Shenoy and S. Krishnan Venkata, "An in-depth analysis of liquid level measurement techniques and performance evaluation using computational fluid dynamics," Journal of Sensors, vol. 2025, p. 4412250, 2025.

[11] J. G. Natividad and T. D. Palaoag, "IoT based model for monitoring and controlling water distribution," IOP Conference Series: Materials Science and Engineering, vol. 482, no. 1, p. 012045, 2019.

[12] D. Witczak and S. Szymoniak, "Review of monitoring and control systems based on Internet of Things," Applied Sciences, vol. 14, no. 19, p. 8943, 2024.