International Journal of Scientific Research and Engineering Development-– Volume 9 Issue 5, Sep-Oct 2026
Available at www.ijsred.com

RESEARCH ARTICLE OPEN ACCESS

IoT-Based Water Level Monitoring and Control System: An Implementation Study

Maimunat Muhammad Imam, Mingi Mathew Haruna, Ahmad Shuaibu, Ahmad Abubakar, Stephen Hussaini, Abraham Terry Akumeh, and Ibrahim Shuaibu
(Department of Computer Engineering, Kaduna Polytechnic, Nigeria)

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

Abstract:
Reliable water supply in residential, institutional and agricultural settings depends on the continuous monitoring of storage tanks and on the correct operation of the pumps that fill them. Manual tank inspection and manually operated pumps remain common, and they lead to tank overflow, dry running of pumps, delayed fault detection and avoidable energy and water losses. This paper presents the design and implementation of a low-cost Internet of Things (IoT) based water level monitoring and control system built around an ESP32 microcontroller. Two float switches installed at the minimum and maximum levels of the tank provide the level information, a BC547-driven 12 V, 30 A electromechanical relay switches a 60 W DC submersible pump, and a two-stage buck converter conditions the 24 V solar-charged battery supply into the 12 V and 5 V rails required by the relay and the controller. Level and pump status are transmitted over the built-in Wi-Fi interface of the ESP32 to the Blynk Cloud, where they can be monitored remotely. The prototype was evaluated through repeated fill-and-empty test sequences. Level detection was correct in all trials, the mean pump switching delay was 0.79 s for switch-on and 0.76 s for switch-off, cloud communication reliability was 98%, and the overall operational reliability of the system was approximately 98.9%. The results indicate that a float-switch based ESP32 design offers a simple, robust and inexpensive alternative to ultrasonic and conductive sensing for automatic tank management.

Keywords — **Automatic Pump Control, Blynk Cloud, ESP32, Float Switch, Internet of Things, Solar-Powered Pumping, Water Level Monitoring.**

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

I. INTRODUCTION

Rising demand for potable water, ageing distribution infrastructure and rapid urban expansion have made the efficient management of water supply systems a pressing concern. Modern distribution networks are expected to deliver continuous and safe service while minimising losses and operational inefficiency, and meeting that expectation requires effective monitoring, maintenance and control strategies [1].

Water supply systems rely on electrical energy to drive the pumps that transfer water from boreholes and reservoirs into storage tanks. Alternating current has traditionally been preferred for this duty because of its availability and its ability to drive high-power motors efficiently [2]. Frequent outages, rising energy costs and the growth of renewable generation have, however, encouraged the adoption of direct current pumps in solar-powered installations [3, 4]. Direct current pumping offers better energy efficiency and greater siting flexibility, but its performance is sensitive to variations in solar irradiance, battery condition, equipment faults and changing demand, which makes continuous supervision of pump operation and tank level necessary.

Water monitoring is the continuous observation and measurement of the key parameters of a supply system [5], while control refers to the actions taken to regulate that system on the basis of the information obtained. Control actions typically include switching pumps on and off, operating valves, regulating distribution and preventing both tank overflow and dry running. Traditionally these functions have been carried out by manual inspection or by basic sensing devices [6], with mechanical float switches, timers or operator intervention performing the control. Such arrangements work for simple installations, but they respond slowly to changing conditions and demand frequent human attention.

Delayed fault detection, dependence on manual inspection and limited accessibility have created the need for more intelligent solutions [7]. Advances in digital sensing, wireless communication and embedded systems have made automation of these functions practical, and the Internet of Things (IoT) now allows monitoring and control systems to acquire, transmit and analyse data in real time while users observe system performance remotely [8]. This is particularly valuable for overhead tanks at dispersed locations, where IoT-based control removes the need for an operator to switch the motor on when a tank is empty and off when it overflows.

Despite these developments, many households and institutions still lack an affordable means of monitoring their tanks. Manual methods are slow and prone to human error, while several automated alternatives are complex, expensive or offer no remote access. The work reported here therefore addresses the design and implementation of an IoT-based water level monitoring and control system built from an ESP32 microcontroller, two float switches, a 12 V, 30 A relay and a 60 W DC submersible pump supplied from a solar-charged battery. Float switches were adopted for level sensing because of their simple operating principle, waterproof construction, high reliability, negligible power consumption and low maintenance requirement in continuous service.

The specific objectives of the work were to design and implement the sensing unit, the control unit and the switching unit, and to implement a cloud-based interface through which the system can be monitored remotely. The remainder of this paper reviews related work, describes the design and implementation of the system, presents and discusses the test results, and concludes.

II. RELATED WORK

The Internet of Things describes a network of physical devices embedded with sensors, software and communication modules that collect and exchange data over the internet with minimal human intervention [9]. A typical IoT deployment comprises sensing devices for data acquisition, a processing unit, a communication technology such as Wi-Fi, a cloud service for storage and processing, and a web or mobile interface for the user [10]. Programmable logic controllers and supervisory control and data acquisition systems can also automate tank management, but they are comparatively expensive, require specialised installation and are better matched to large industrial plants, whereas IoT platforms are flexible, scalable and cost-effective at residential and institutional scale.

[11] developed an automated water level detection and regulation system based on a PIC16F84A microcontroller and conductive level probes, with indicator LEDs and relay-based pump control. The system reduced waste, prevented overflow and limited human involvement, but offered no remote monitoring, cloud storage or network connectivity, which restricts its usefulness in modern installations. Conductive probes are also susceptible to corrosion because their electrodes remain permanently in contact with the water.

[12] implemented an IoT-enabled smart drip irrigation system using the ESP32 and the Blynk application, transmitting sensor data over the built-in Wi-Fi interface so that irrigation could be monitored and controlled remotely. The study demonstrated the effectiveness of the ESP32 and Blynk combination for real-time monitoring and automated control, although its application domain is field irrigation rather than the management of stored water. [13] reported a comparable embedded irrigation controller that streams events to a server, again confirming the suitability of the ESP32 platform for low-cost water applications.

[7] developed an IoT-based tank water monitoring system that transmits level data to an online platform for remote inspection, showing that IoT reduces manual inspection and water wastage. The work concentrates on monitoring, however, and gives limited attention to automatic pump control. [14] likewise surveyed IoT solutions for

monitoring water level, leakage and motor control in smart tanks and observed that most reported systems address a single function rather than integrating sensing, actuation and remote supervision.

[15] designed a secure IoT-based real-time level monitoring system for reservoirs, dams and industrial storage, pairing an ESP32 with an HC-SR04 ultrasonic sensor and adding authentication and encrypted transport to protect the monitoring data. The system achieved accurate non-contact measurement, but the authors acknowledged dependence on stable connectivity, sensor calibration errors and power management difficulties during long deployments. Ultrasonic sensing is in general more expensive than switch-based detection and can return erroneous readings in the presence of condensation, foam, turbulence or irregular tank geometry.

The reviewed literature shows that existing studies concentrate either on level monitoring, on automatic pump control, or on remote supervision, and that several rely on hardware whose cost and complexity are difficult to justify for domestic and institutional tanks. The present work combines automatic level detection, relay-based pump control, solar-battery operation and cloud monitoring in a single low-cost design, and uses float switches rather than ultrasonic or conductive sensing to obtain a direct, corrosion-free and calibration-free indication of the two level thresholds that automatic pump control actually requires.

III. SYSTEM DESIGN AND IMPLEMENTATION

A. System Architecture

The system automatically monitors the level in a storage tank, controls a DC submersible pump and reports status to a cloud dashboard. It comprises five functional units: the power supply unit, the sensing unit, the control unit, the switching unit and the IoT communication unit. The components used in the prototype are listed in Table I.

TABLE I. COMPONENTS OF THE DEVELOPED SYSTEM

S/N	Component	Description
1	Microcontroller	ESP32 (dual-core, built-in Wi-Fi)
2	Relay	JQX-16F (T91)-1Z; NO: 30 A, 240 VAC / 30 VDC; NC: 20 A
3	Diode	IN4007 (coil freewheeling)
4	Transistor	BC547 (NPN relay driver)
5	Buck converters	Stage 1: 12 VDC, 9 A; Stage 2: 5 VDC, 3 A
6	Sensors	Two float switches (minimum and maximum level)
7	Resistor	4.7 kΩ (base resistor)
8	Battery	24 V, 220 Ah (solar charged)
9	Capacitor	1000 µF, 50 V (supply filtering)
10	Pump	60 W DC submersible pump

B. Power Supply Unit

The power supply arrangement is shown in Fig. 1. A solar panel and charge controller maintain a 24 V battery, which is the primary source for the entire system. A first-stage buck converter steps the 24 V down to 12 V for the relay coil and for the input of the second stage, and a second-stage converter produces the 5 V rail that supplies the ESP32 and the float switches. Switch-mode conversion was preferred to linear regulation because it dissipates far less power and generates less heat, which suits continuously powered embedded equipment [16].

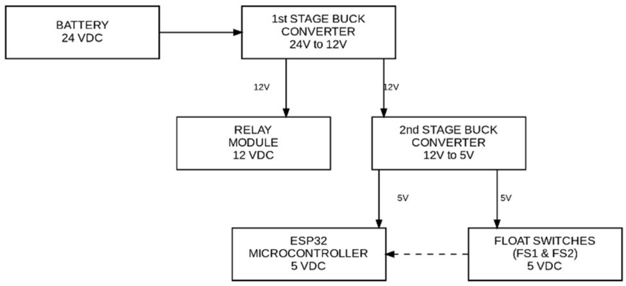

Fig. 1 Block diagram of the power supply unit

The required battery capacity was obtained from

$$C = (P \times t) / V \qquad (1)$$

where C is the battery capacity in ampere-hours, P the total connected load in watts, t the required backup time in hours and V the nominal battery voltage.

The estimated power demand of each load is given in Table II. For a total load of 62 W, a system voltage of 24 V and a required autonomy of four hours, (1) gives 10.33 Ah. Applying a 25 per cent design margin for conversion losses and battery ageing gives a design capacity of 12.91 Ah. The 24 V, 220 Ah battery available for the prototype greatly exceeds this figure and therefore provides a generous operating margin.

TABLE II. ESTIMATED LOAD POWER DEMAND

Component	Power rating
ESP32	1 W
Relay module	0.6 W
Float switches	0.2 W
LED indicator	0.2 W
DC submersible pump	60 W
Total load	62 W

C. Sensing Unit

The sensing unit detects the water level and supplies the corresponding logic inputs to the controller. Two float switches are used: float switch 1 is installed at the minimum level and float switch 2 at the maximum level. Each switch consists of a hollow float containing a permanent magnet and a fixed reed contact sealed inside the sensor body; as the water surface rises or falls the float moves with it, and at the actuation point the magnet closes or opens the reed contact [14, 17]. When the level falls below the minimum, float switch 1 changes state and signals that the tank requires refilling; when the level reaches the maximum, float switch 2 changes state and signals that the tank is full. The connection of the switches to the ESP32 is shown in Fig. 2.

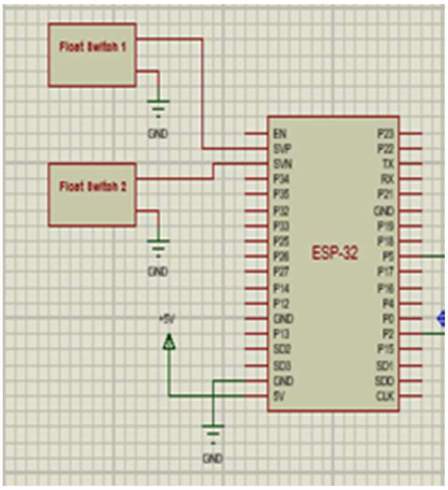

Fig. 2 Sensing unit circuit diagram

D. Control Unit

The ESP32 is the control element of the system. It is a low-power dual-core microcontroller with digital and analogue input/output, timers, several communication interfaces and integrated Wi-Fi and Bluetooth [13]. It was selected principally because the integrated Wi-Fi removes the need for an external communication module, reducing circuit complexity, cost and power consumption. An Arduino Uno would have required an additional network module; the ESP8266 has fewer input/output pins, less memory and lower processing capability; and a single-board computer such as the Raspberry Pi consumes considerably more power and requires an operating system that is unnecessary for a dedicated controller of this kind.

The controller polls the two float switches continuously, decides whether the pump should run, drives the relay accordingly and publishes the status to the cloud. The pin assignment is given in Table III.

TABLE III. ESP32 PIN ASSIGNMENT

Function	ESP32 pin
Float switch 1 (minimum level)	SVP (GPIO 36)
Float switch 2 (maximum level)	SVN (GPIO 39)
Relay drive	GPIO 5
Status LED	GPIO 2
Network interface	Built-in Wi-Fi
Supply	5 V

E. Switching Unit

The switching unit controls the power delivered to the pump. A relay is an electrically operated switch that allows a low-power control circuit to manage a circuit carrying a much larger current [18]. The ESP32 operates at 3.3 V and cannot supply the current required by the pump, so a BC547 transistor with a 4.7 kΩ base resistor energises the coil of a 12 V, 30 A relay whose contacts connect the battery to the pump [11]. An IN4007 diode across the coil suppresses the inductive voltage spike produced when the coil is de-energised. The relay also provides galvanic isolation between the controller and the pump circuit. Semiconductor alternatives were considered: MOSFETs switch faster and more efficiently but require careful gate drive and additional protection with inductive loads, while solid-state relays are silent and free of moving parts but are substantially more expensive and dissipate heat in conduction. For simple on-off control of a DC pump the electromechanical relay is the more economical and robust choice, and its 30 A rating leaves a wide margin over the current drawn by the 60 W pump. The interface is shown in Fig. 3.

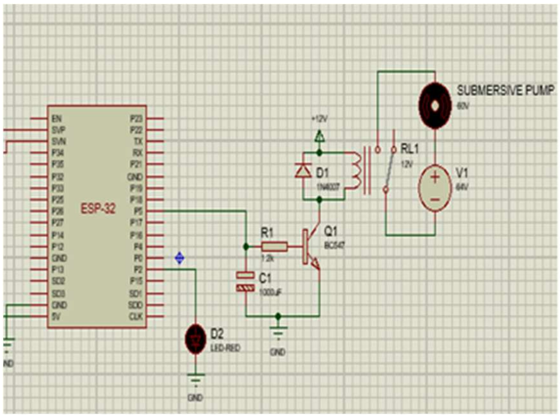

Fig. 3 Switching unit circuit diagram

F. IoT Communication Unit

Remote monitoring is provided by the ESP32 in conjunction with the Blynk Cloud platform, which links the controller to a smartphone or web dashboard, supports real-time visualisation, device control and notifications, and stores historical data [19]. Once the ESP32 has joined a Wi-Fi network it authenticates with the Blynk server and publishes the level and pump status, which the user can then view from any location with internet access. Blynk was preferred to alternatives such as ThingSpeak because of its interactive dashboard widgets and push notifications, and to a custom web server because it requires no server development or maintenance.

G. Hardware Implementation and Packaging

The complete hardware interconnection is shown in Fig. 4. The float switches were wired to the digital inputs of the ESP32, the relay driver to the designated output pin, and the ESP32 supply to the 5 V converter output, while the relay contacts switch the battery supply to the pump. A capacitor across the supply filters ripple, and an LED indicates pump operation and network status.

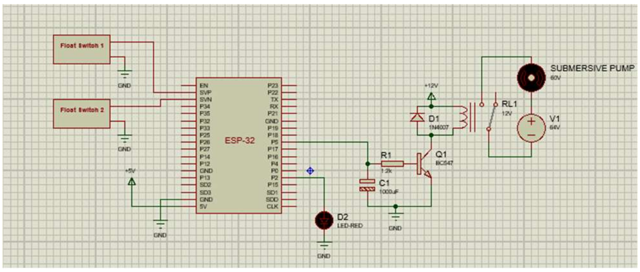

Fig. 4 Complete hardware connection diagram

After the circuit had been tested, the electronics were mounted in a protective enclosure with adequate spacing for heat dissipation and access for maintenance, and with openings for wiring and supply cables. The enclosure was installed adjacent to the storage tank and the float switches were fixed inside the tank at the minimum and maximum levels. Fig. 5 and Fig. 6 show the internal wiring and the completed prototype.

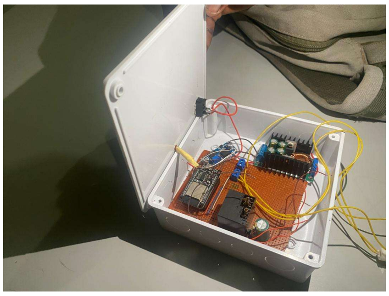

Fig. 5 Internal wiring of the packaged system

Fig. 6 Fully packaged prototype

H. Software Implementation

The firmware was developed in C/C++ in the Arduino IDE and uploaded to the ESP32 over USB. After initialising the input and output pins and the serial port, the program connects to the Wi-Fi network and authenticates with the Blynk server. It then enters a continuous loop in which both float switches are read: if the minimum-level switch indicates a low level the relay is energised and the pump starts; if the maximum-level switch indicates a full tank the relay is de-energised and the pump stops. The pump state is published to the dashboard on every iteration, and the Blynk service routine is executed to maintain the connection. The control flow is summarised in Fig. 7.

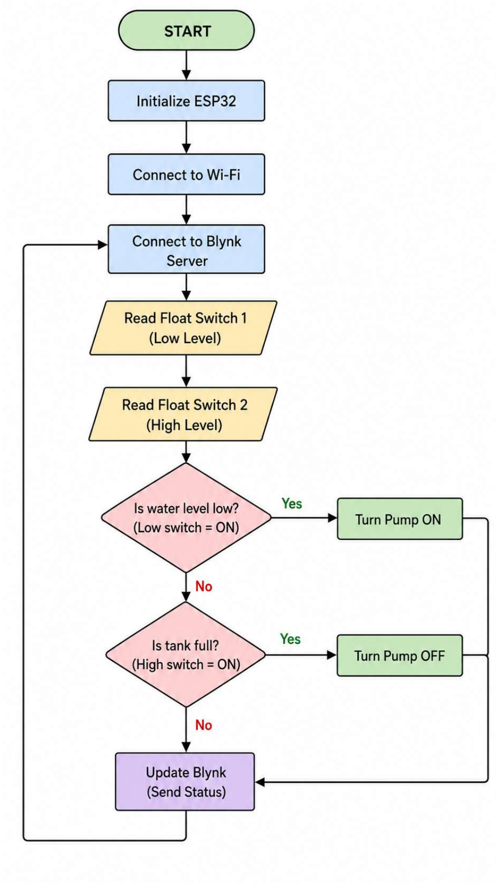

Fig. 7 Flowchart of the control program

IV. RESULTS AND DISCUSSION

The assembled prototype was tested to verify the operation of each unit and to characterise the performance of the complete system. Testing covered functional operation, level detection, automatic pump control, relay switching, cloud communication, response time and overall stability.

A. Functional Test

With the firmware loaded and the switches installed at their respective levels, the system was operated through complete fill and empty sequences. When the level fell below the minimum, float switch 1 produced the low-level signal, the ESP32 energised the relay and the 60 W pump started. When the level reached the maximum, float switch 2 produced the high-level signal and the controller de-energised the relay, stopping the pump. Status was transmitted to the Blynk Cloud throughout. The outcome of the functional test is summarised in Table IV.

TABLE IV. FUNCTIONAL TEST OF THE DEVELOPED SYSTEM

S/N	Test parameter	Expected	Observed	Status
1	Low-level detection	Pump ON	Pump ON	Pass
2	High-level detection	Pump OFF	Pump OFF	Pass
3	Relay operation	Switches correctly	Switched correctly	Pass
4	Wi-Fi connection	ESP32 connects	Connected	Pass
5	Cloud monitoring	Status updated	Updated	Pass
6	Solar power supply	Stable supply	Stable	Pass

B. Level Detection and Automatic Pump Control

Level detection was examined in five trials for each of three conditions: below the minimum, between the two thresholds, and at the maximum. Both float switches identified the corresponding condition correctly in every trial, giving a detection accuracy of 100% for all three conditions. The absence of false or missed detections reflects the direct mechanical action of the float switch, which does not depend on calibration, water conductivity or surface conditions.

Automatic control was verified over five complete pumping cycles. In each cycle the controller started the pump on the low-level signal, maintained pumping while the level rose between the two thresholds, and stopped the pump on the high-level signal, with the relay and pump states matching the controller decision in every case. The arrangement therefore removes the need for manual switching and prevents both overflow and dry running, which is the principal cause of premature pump failure.

C. Pump Response Time

Response time was measured as the interval between a change of state of a float switch and the corresponding change of state of the pump, over five trials in each direction. The results are given in Table V and plotted in Fig. 8. The mean switch-on delay was 0.79 s and the mean switch-off delay

0.76 s, with individual values ranging from 0.76 s to 0.82 s and from 0.74 s to 0.79 s respectively. These delays are governed mainly by input debouncing and relay coil actuation and are negligible in relation to the time required to fill or empty a domestic tank. The small spread between trials also indicates consistent controller behaviour.

TABLE V. MEASURED PUMP RESPONSE TIME

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

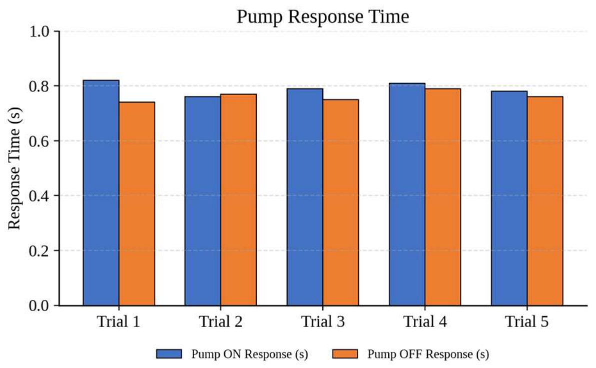

Fig. 8 Pump switching response time over five trials

D. Communication and System Reliability

Communication performance was assessed by repeated connection and update attempts. The ESP32 joined the Wi-Fi network and established a session with the Blynk server successfully in all ten attempts made for each, while 49 of 50 level updates and 49 of 50 pump-status updates were delivered, giving 98% reliability for data reporting. The two unsuccessful events observed across 100 general communication attempts are attributable to transient network interruption rather than to controller malfunction, and they did not affect pump control, which is executed locally and is therefore independent of the network.

Stability was assessed over an extended sequence of operations, summarised in Table VI. Taking the ratio of successful operations to total operations gives an overall reliability of 445/450, or approximately 98.89%. No persistent sensor failure, false triggering or controller malfunction was observed.

TABLE VI. RELIABILITY OF SYSTEM OPERATIONS

Operation	Tests	Successful	Reliability
Low-level detection	50	50	100%
High-level detection	50	50	100%
Pump ON operation	50	49	98%
Pump OFF operation	50	50	100%
Relay switching	100	99	99%
Wi-Fi connection	50	49	98%
Cloud communication	100	98	98%
Overall	450	445	98.89%

E. Comparison with Existing Systems

Table VII compares the developed system with two representative designs from the reviewed literature, on the basis of sensing method, controller, communication capability, power source and measured performance. The comparison shows that the present design attains automatic control and remote monitoring without an external communication module, and that switch-based detection matched or exceeded the accuracy reported for ultrasonic and conductive sensing while responding more quickly. Operation from a solar-charged battery is a further practical advantage in locations where the mains supply is unreliable, which is precisely where automatic tank management is most needed.

TABLE VII. COMPARISON WITH REPRESENTATIVE EXISTING SYSTEMS

Parameter	System 1	System 2	This work
Level sensing	Ultrasonic	Conductive	Float switches
Controller	Arduino Uno	NodeMCU	ESP32
Cloud platform	None	Blynk	Blynk
Pump switching	Relay	Relay	12 V / 30 A relay
Power source	AC mains	Solar/battery	Solar/battery
Remote monitoring	No	Yes	Yes
Detection accuracy	96%	98%	100%
Response time	1.20 s	0.95 s	0.79 s
Communication reliability	N/A	96%	98%

V. CONCLUSION

An IoT-based water level monitoring and control system has been designed, constructed and tested. Two float switches provide minimum and maximum level detection, an ESP32 microcontroller executes the control logic and publishes status to the Blynk Cloud over its integrated Wi-Fi interface, and a transistor-driven 12 V, 30 A relay switches a 60 W DC submersible pump supplied from a solar-charged 24 V battery through a two-stage buck converter. All four design objectives were met: the sensing, control and switching units operate as intended, and the cloud interface allows the tank and pump to be supervised remotely.

Testing returned correct level detection in every trial, mean switching delays of 0.79 s and 0.76 s, communication reliability of 98% and overall operational reliability of approximately 98.9%. The system prevents overflow and dry running, removes the need for routine manual inspection and operates without a mains supply, at a hardware cost well below that of ultrasonic or controller-based commercial alternatives.

For reliable service the float switches should be firmly mounted at the intended levels, the electronics protected from moisture, and the pump, wiring, relay and supply components inspected periodically. Future work will extend the design to several tanks served by a single controller, add flow measurement so that consumption as well as level can be reported, log historical data for consumption analysis, and introduce a local fallback control mode that maintains automatic pumping during prolonged loss of internet connectivity.

REFERENCES

[1] A. Alshami, M. Elsayed, E. Mallick, A. Mahmud, T. Zeide, and A. Kalyoncu, "IoT innovations in sustainable water and wastewater management and water quality monitoring: A comprehensive review of advancements, implications and future directions," *IEEE Access*, vol. 12, pp. 58427-58453, 2024.

[2] S. S. H. Rafin, R. Ahmed, M. A. Haque, M. K. Hossain, M. A. Haque, and O. A. Mohammed, "Power electronics revolutionized: A comprehensive analysis of emerging wide and ultrawide bandgap devices," *Micromachines*, vol. 14, no. 11, art. 2045, 2023.

[3] D. Balamurali *et al.*, "A solar-powered, Internet of Things (IoT)-controlled water irrigation system supported by rainfall forecasts utilizing aerosols: A review," *Environment, Development and Sustainability*, pp. 1-40, 2025.

[4] E. Bhattacharjee, S. Alam, M. Rokonuzzaman, S. M. Sultan, C. P. Tso, and K. S. Rahman, "Solar PV and IoT-integrated smart irrigation system with secure cloud control and water recycling," *Journal of Taibah University for Science*, vol. 20, no. 1, art. 2704294, 2026.

[5] S. A. H. AlMetwally, M. K. Hassan, and M. H. Mourad, "Real time Internet of Things (IoT) based water quality management system," *Procedia CIRP*, vol. 91, pp. 478-485, 2020.

[6] A. Ahmed, V. D. Chaudhari, I. S. Jadhav, M. N. Patil, and A. D. Vishwakarma, "Multilevel water level control and monitoring of multiple tanks," *Journal of Emerging Technologies and Innovative Research*, vol. 3, p. 66, 2018.

[7] S. S. Vellela, B. V. Babu, and Y. B. Mahendra, "IoT-based tank water monitoring systems: Enhancing efficiency and sustainability," *International Journal for Modern Trends in Science and Technology*, vol. 10, no. 2, pp. 291-298, 2024.

[8] A. J. Ramadhan, A. Ali, and H. Kareem, "Smart water-quality monitoring system based on enabled real-time Internet of Things," *Journal of Engineering Science and Technology*, vol. 15, no. 6, pp. 3514-3527, 2020.

[9] K. Ashton, "That 'Internet of Things' thing," *RFID Journal*, vol. 22, no. 7, pp. 97-114, 2009.

[10] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A survey on enabling technologies, protocols and applications," *IEEE Communications Surveys and Tutorials*, vol. 17, no. 4, pp. 2347-2376, 2015.

[11] S. K. Reza, S. A. M. Tariq, and S. M. Reza, "Microcontroller based automated water level sensing and controlling: Design and implementation issue," in *Proc. World Congr. Engineering and Computer Science*, vol. 1, pp. 20-22, 2010.

[12] G. P. Pereira, M. Z. Chaari, and F. Daroge, "IoT-enabled smart drip irrigation system using ESP32," *IoT*, vol. 4, no. 3, pp. 221-243, 2023.

[13] A. Morchid *et al.*, "IoT-enabled smart agriculture for improving water management: A smart irrigation control using embedded systems and server-sent events," *Scientific African*, vol. 27, art. e02527, 2025.

[14] 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, art. 309, 2022.

[15] M. A. Al-Shareeda, A. M. Ali, M. A. Hammoud, Z. H. M. Kazem, and M. A. Hussein, "Secure IoT-based real-time water level monitoring system using ESP32 for critical infrastructure," *Journal of Cyber Security and Risk Auditing*, no. 2, pp. 44-52, 2025.

[16] S. Kumaraguruparan and K. Elango, "Optimal control strategies for high-efficiency non-isolated DC-DC buck converters in IoT applications: A comparative study," *Heliyon*, vol. 10, no. 18, 2024.

[17] M. Whitford and A. P. Klimley, "An overview of behavioral, physiological and environmental sensors used in animal biotelemetry and biologging studies," *Animal Biotelemetry*, vol. 7, no. 1, art. 26, 2019.

[18] S. Perumal, R. Senthilkumar, S. Kishorekumar, V. Sanjay, and S. Prithiviraj, "Automatic motor control using Arduino and relay logic," *International Journal of Engineering and Extended Technologies Research*, vol. 8, no. 2, pp. 4705-4713, 2026.

[19] Blynk Inc., *Blynk Documentation: Getting Started with the Blynk IoT Platform*, 2024. [Online]. Available: https://docs.blynk.io

