iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** X **Month of publication:** October 2026

DOI: https://doi.org/10.22214/ijraset.2026.85422

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue X Oct 2026- Available at www.ijraset.com

An IOT-Enabled Water-Level Monitoring and Pump-Control System Using Arduino and Ultrasonic Sensing

Mahi Kumari[1], Khushbu Kumari[2], Rakhi Kumari[3], Priti Kumari[4]
[1, 2, 3, 4]*Student, [Department of Electronics Engineering, Government Polytechnic Sitamarhi, Sitamarhi, India*

Abstract: ***Overhead water tanks in homes, hostels and institutions are commonly filled by pumps that are switched by hand, which leads to overflow, wasted electricity and pumps running dry when the sump is empty. This paper presents the design and evaluation methodology of a low-cost, IoT-enabled water-level monitoring and pump-control system built on an Arduino Uno. A waterproof JSN-SR04T ultrasonic sensor measures the distance to the water surface without contact, and a DS18B20 probe measures air temperature so that the speed of sound, and hence the level, is temperature-compensated. The controller converts distance to level and stored volume, switches a single-phase pump through an opto-isolated relay and contactor using hysteresis thresholds, and enforces three safety interlocks: sump dry-run protection, a no-rise timer that detects a pump running without delivering water, and rejection of implausible sensor readings. An ESP8266 module publishes telemetry and alerts over MQTT to a dashboard and accepts remote commands, while all switching decisions remain local so the tank is protected when the network fails. A review of 19 related studies shows that most existing systems provide threshold switching and remote display but rarely address temperature error or pump protection. Analysis shows that an uncompensated sensor calibrated at 20 °C over-reads distance by 4.2 cm at 1.0 m when the tank air reaches 45 °C, about 33 litres in a 1000-litre tank. A test protocol with defined metrics for level accuracy, switching accuracy, fault response and telemetry reliability is provided for prototype validation.***

IndexTerms: ***Arduino Uno, Internet of Things, ultrasonic sensor, water-level monitoring, pump control, MQTT, ESP8266, dry-run protection.***

I. INTRODUCTION

Most buildings in India and other developing regions store water in an overhead tank that is filled from an underground sump or a borewell by an electric pump. In many installations the pump is switched on and off by hand. The tank then overflows when the operator forgets to switch the pump off, runs empty when nobody notices the level falling, and the pump is damaged when it runs with no water in the sump. Each overflow wastes both water and the electrical energy used to lift it.

Float switches and electrode-type level controllers automate the basic on/off function, but they have moving parts or wetted electrodes that foul and corrode, and they give no information about how much water is stored. Ultrasonic distance sensors measure the level without touching the water, and low-cost microcontrollers and Wi-Fi modules now allow the level to be reported to a phone. The Internet of Things (IoT) provides the protocols and cloud services that make such remote monitoring practical [1].

A recent systematic review of IoT-based water-tank solutions found that most published systems monitor level and control the motor, but few detect leakage, and many depend on commercial cloud platforms with weak security [2]. Ultrasonic measurement also has a physical weakness that low-cost designs usually ignore: the speed of sound changes with air temperature, and the air inside a rooftop tank can be much warmer than the room in which the sensor was calibrated [4], [10], [21].

This paper designs an Arduino-based system that addresses these gaps. Its contributions are:

1) A three-layer architecture (field node, cloud, user applications) in which every pump decision is made locally, so the tank and pump are protected even when the network is unavailable.
2) Temperature-compensated ultrasonic level measurement with median filtering and plausibility checks.
3) A pump-control algorithm with hysteresis, minimum off-time, sump dry-run protection and a no-rise timer that detects a pump running without delivering water.
4) A structured review of 19 related studies, an analytical evaluation of the temperature error, and an experimental protocol with defined metrics for validating a prototype.

©IJRASET: All Rights are Reserved

Section II reviews related work. Sections III to VI describe the architecture, hardware, measurement and control algorithm, and IoT layer. Section VII sets out the experimental method, Section VIII presents the results and discussion, and Section IX concludes the paper.

II. LITERATURE REVIEW

The reviewed studies fall into three groups: systems that monitor a storage tank and switch its pump, systems that monitor water level in rivers and irrigation tanks over long distances, and studies that assess the accuracy of the sensors themselves. Table 1 summarises 19 studies published between 2013 and 2026.

A. Tank-level monitoring and pump control

Perumal et al. presented an early IoT-enabled water monitoring system and argued that consumer-level IoT projects could support water conservation [3]. Bello et al. used two HC-SR04 ultrasonic sensors and an ATmega328 microcontroller to measure the level in overhead and underground household tanks and display the stored volume on an LCD, and reported reliable operation with both calm and turbulent water [5].

Sachio et al. built an ESP8266-based controller that switches a pump or valve to prevent overflow, with remote monitoring through Blynk and PHP, and reported a control error of 2 cm on a 64 cm container [6]. Lestariningsih et al. combined a NodeMCU, an HC-SR04 sensor and a relay-driven pump with an Android interface through the Cayenne platform [7].

Praveen et al. measured tank and sump levels with an ultrasonic sensor and an ESP module, plotted the readings through Adafruit IO, and used red and green LEDs to tell the user when to switch the motor [9]. Harrischandra et al. added SMS alerts through a SIM800L GSM module to an Arduino Uno, relay and LCD design and calibrated the sensor against a ruler with 1 mm divisions [11]. Petru et al. used an Arduino Uno to stop a pump at a programmed maximum level, a NodeMCU to send readings to ThingSpeak, and a Bluetooth link for manual pump control from a phone [13].

More recent work extends this basic design. Rajanbabu et al. used a Raspberry Pi, an ultrasonic sensor and fuzzy-logic pump scheduling with demand-side management, and reported electricity-cost reductions of up to 6% [15]. Yap and Ali combined ultrasonic sensing, an Arduino, a relay-driven pump, a buzzer for abnormal conditions and Blynk remote control, and observed consistent switching within a 10–13 cm threshold band [16]. Mohamed Jumaiedy and Suratkon added a turbidity sensor to monitor water cleanliness as well as volume [17]. Kabir et al. replaced the ultrasonic sensor with a non-contact capacitive level sensor on an Arduino Uno R4 with relay pump control [20]. Jan et al. reviewed 19 such systems and recommended 32-bit wireless development boards and secure cloud servers [2].

B. Remote water-level monitoring in the field

Krishnaveni et al. installed a solar-powered NodeMCU node with an ultrasonic sensor at an irrigation tank in Tamil Nadu, sent data to Google Sheets, converted level to stored volume with a stage–volume curve, and achieved an overall accuracy of ±2.5 cm, better than a LiDAR sensor in their calibration [8]. Kabi et al. developed an open-hardware LoRa river-level node as a cheaper alternative to conventional gauging equipment [12].

Baratella et al. built an ultra-low-power LoRaWAN transmitter for a 4–20 mA level sensor that draws about 3 mA in sleep and can run about four months on two 10,000 mAh batteries [14]. Yapa et al. combined LoRa and MQTT for flood alerts in rural areas [18], and Sadi and Mulyati used an Arduino Mega, an HC-SR04 sensor and a PLC to operate a floodgate, with SMS backup and ThingSpeak logging [19].

C. Accuracy of ultrasonic level measurement

Fisher and Sui built an open-source ultrasonic logger for evaporation pans and added an air-temperature sensor specifically to correct the speed of sound [4]. Panagopoulos et al. compared an ultrasonic stage sensor with a pressure transducer in an urban stream over almost a year; the two agreed within 7%, and daily air-temperature swings caused visible fluctuations in the ultrasonic readings [10]. These studies show that temperature compensation matters in practice, yet none of the tank-control systems in Section 2.1 describes it.

©IJRASET: All Rights are Reserved

Table 1 Summary of reviewed literature

Ref.	Year	Sensing	Controller / link	Pump control	Key result or limitation
[2]	2022	Review of 19 systems	Various	Reviewed	Few systems detect leakage; cloud security weak
[3]	2015	Water monitoring	IoT node	—	Early consumer IoT water-monitoring concept
[4]	2013	Ultrasonic + air temp.	Microcontroller logger	No	Temperature correction of speed of sound
[5]	2018	2 × HC-SR04	ATmega328, LCD	Yes	Volume on LCD; works with turbulent water
[6]	2018	Ultrasonic	ESP8266, Blynk, PHP	Yes (pump/valve)	2 cm control error on 64 cm container
[7]	2019	HC-SR04	NodeMCU, Cayenne	Yes (relay)	Android HMI for remote control
[8]	2020	Ultrasonic, DHT11	NodeMCU, solar, Google Sheets	No	±2.5 cm; stage–volume conversion
[9]	2021	Ultrasonic (tank + sump)	ESP, Adafruit IO	Indicated by LEDs	Real-time graphs; motor switched by user
[10]	2021	Ultrasonic vs pressure	Field station	No	Agreement within 7%; temperature effects seen
[11]	2022	Ultrasonic	Arduino Uno, SIM800L	Yes (relay)	SMS alerts; ruler calibration
[12]	2023	Ultrasonic	LoRa node	No	Low-cost open-hardware river gauge
[13]	2023	HC-SR04	Uno + NodeMCU, ThingSpeak	Yes (relay)	Bluetooth manual control (preprint)
[14]	2023	4–20 mA level sensor	LoRaWAN, Grafana	No	3 mA sleep; ~4 months on batteries
[15]	2025	Ultrasonic	Raspberry Pi, fuzzy logic	Yes (scheduled)	Up to 6% lower electricity cost
[16]	2025	Ultrasonic	Arduino, Blynk	Yes (relay)	Buzzer alarm; 10–13 cm switching band
[17]	2025	Ultrasonic + turbidity	Microcontroller, Blynk	—	Adds water-quality indication
[18]	2025	Level sensor	LoRa + MQTT	No	Threshold flood alerts for rural rivers
[19]	2026	HC-SR04	Arduino Mega, PLC, SMS	Floodgate motor	ThingSpeak logging; SMS backup
[20]	2026	Capacitive (non-contact)	Arduino Uno R4	Yes (relay)	Alternative non-contact sensing

D. Research gap

The review shows a mature basic design — an ultrasonic sensor, a microcontroller, a relay and a cloud dashboard — but three recurring gaps, summarised in Table 2. First, temperature compensation is used in field-measurement studies [4], [8], [10] but is not described in the tank-control systems [5]–[7], [9], [11], [13], [16]. Second, pump protection is limited mostly to level thresholds; protection against a dry sump or a pump that runs without delivering water is rarely described [2]. Third, many systems route control through a cloud platform [6], [7], so their behaviour when the network fails is not defined. The proposed system addresses all three.

Table 2 Gaps in existing work and how the proposed system addresses them

Gap	Evidence in reviewed work	Approach in this paper
Temperature error in ultrasonic level	Corrected in field studies [4], [10]; not described in tank controllers	DS18B20 probe; speed of sound computed each cycle (Eq. 2)
Limited pump protection	Mostly level thresholds only [2], [6], [7], [13]	Sump float switch, no-rise timer, minimum off-time, maximum run time
Dependence on cloud for control	Control through Blynk or Cayenne [6], [7]	All switching decided locally; cloud only for monitoring and requests
Sensor faults and spurious echoes	Rarely discussed	Median filter, range and rate-of-change checks, fail-safe pump OFF

III. PROPOSED SYSTEM ARCHITECTURE

The system is organised in three layers (Fig. 1). The field node at the tank senses the water level and air temperature, decides whether the pump should run, switches the pump and shows the status locally. The cloud layer receives telemetry over MQTT, stores it and raises alerts. The user layer provides a dashboard, notifications and a way to send commands. Commands from the cloud are treated as requests: the field node checks them against its own interlocks before acting.

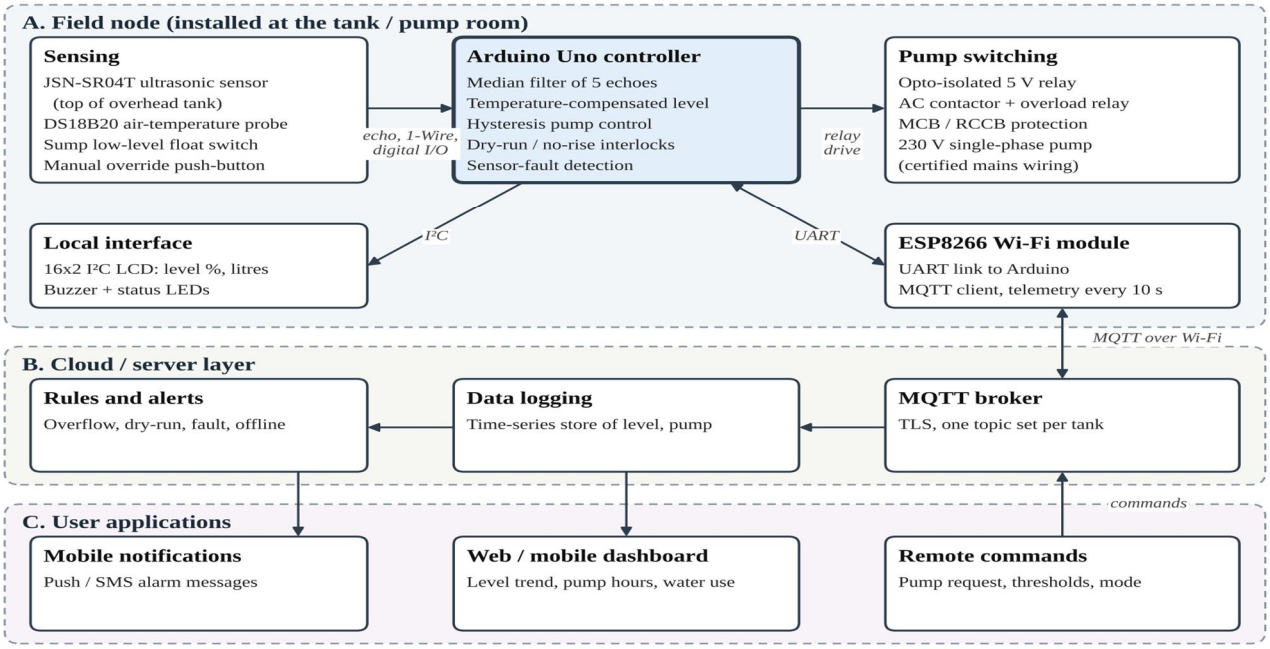


Fig. 1 Three-layer architecture of the proposed water-level monitoring and pump-control system

IV. HARDWARE DESIGN

The field node uses widely available modules, listed in Table 3. The JSN-SR04T sensor was chosen instead of the HC-SR04 used in most reviewed systems [5], [7], [13] because its transducer is sealed against the humid air inside a tank [23].

Table 3 Hardware components of the field node

Component	Function	Interface / note
Arduino Uno (ATmega328P, 16 MHz) [22]	Measurement, control logic, interlocks	5 V; watchdog timer enabled
JSN-SR04T waterproof ultrasonic sensor [23]	Distance to water surface	Trigger D7, echo D6; mount ≥ 25 cm above full level (blind zone)
DS18B20 temperature probe	Air temperature inside tank for speed of sound	1-Wire on D4 with 4.7 kΩ pull-up
Float switch in sump	Detects low sump level (dry-run risk)	D3 with internal pull-up
Opto-isolated 5 V relay module	Drives contactor coil	D8; relay does not switch motor current directly
AC contactor + overload relay, MCB/RCCB	Switches and protects the 230 V pump	Rated for motor current; installed by a qualified electrician
ESP8266 (ESP-01) Wi-Fi module [24]	MQTT telemetry and commands	UART via level shifter; 3.3 V regulator
16×2 LCD with I²C adapter, buzzer, LEDs	Local display and alarms	A4/A5 (I²C); buzzer D9
Push-button	Manual pump request / alarm reset	D2
5 V / 2 A isolated SMPS	Powers electronics	Separate from mains switching circuit

Because the pump runs from the 230 V mains, the design keeps the low-voltage electronics electrically separate from the motor circuit. The Arduino drives an opto-isolated relay, the relay energises a contactor coil, and the contactor switches the motor through an overload relay and a miniature circuit breaker. All mains wiring must be enclosed, earthed and installed by a qualified electrician. Laboratory testing can use a small 12 V DC pump in place of the mains pump without changing the firmware.

V. MEASUREMENT MODEL AND CONTROL ALGORITHM

A. Temperature-compensated level measurement

The sensor emits an ultrasonic burst and returns an echo pulse whose width t is the round-trip travel time. The distance d to the water surface is given by (1), and the speed of sound c in air is approximated from the air temperature T in °C by (2) [21]:

$$d = c \cdot t / 2 \quad (1)$$

$$c(T) = 331.3 + 0.606\ T \ \ (m/s) \quad (2)$$

If H is the distance from the sensor face to the tank bottom, the water height h, the fill level L and the stored volume V of a cylindrical tank of internal diameter D are:

$$h = H - d, \quad L = 100\ h / h_{max} \ \ (\%) \quad (3)$$

$$V = (\pi\ D^2 / 4) \cdot h \quad (4)$$

When a fixed speed of sound c_{ref}, valid at the calibration temperature, is used instead of (2), the measured distance is wrong by e_T, given by (5). The water level is wrong by the same amount with the opposite sign.

$$e_T = d\ (1 - c_{ref} / c(T)) \quad (5)$$

Each measurement cycle takes five echoes 60 ms apart and uses their median, which rejects single spurious echoes from ripples, condensation drops or the tank wall. A reading is accepted only if it lies within the sensor's valid range and differs from the previous accepted value by less than 10 cm; a larger jump is accepted only if it is repeated in three consecutive cycles. Three consecutive invalid cycles declare a sensor fault.

B. Pump-control Algorithm

The control logic is shown in Fig. 2 and its parameters in Table 4. Hysteresis between the switch-on level L_{on} and the switch-off level L_{off} prevents rapid cycling. A minimum off-time protects the motor from repeated starts, and a maximum run time limits damage if every other check fails.

The no-rise timer covers faults that a level threshold cannot detect, such as an air-locked pump, a closed valve, a burst delivery pipe or a borewell that has run dry. While the pump runs, the level must rise by at least Δh within T_{rise}; if it does not, the pump is switched off and an alarm is raised. With a 20 L/min pump and a tank of 0.785 m^2 cross-section, the level should rise about 2.5 cm per minute, so a requirement of 2 cm in 5 minutes gives a wide safety margin against false alarms.

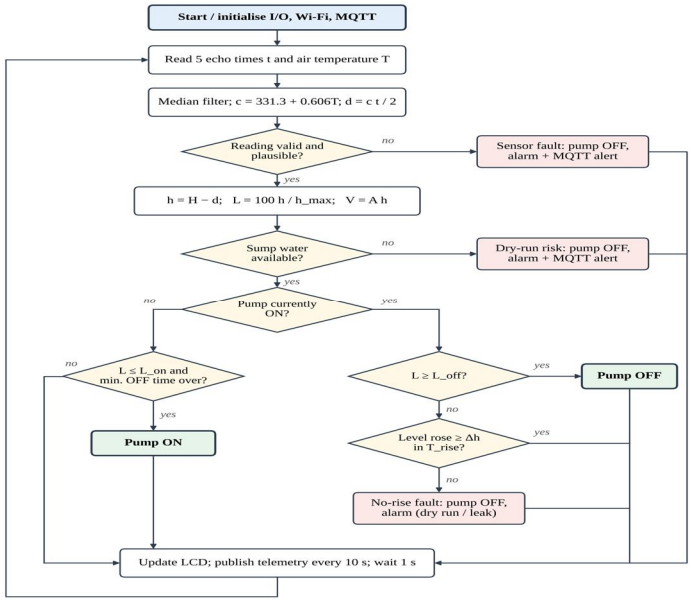


Fig. 2 Flowchart of the measurement and pump-control algorithm

Table 4 Control parameters (example values for a 1000-litre tank)

Parameter	Symbol	Value
Pump switch-on level	L_{on}	25%
Pump switch-off level	L_{off}	95%
Minimum level rise while pumping	Δh	2 cm
Window for level rise	T_{rise}	5 min
Minimum pump off-time	—	2 min
Maximum continuous run time	—	60 min
Echoes per measurement / spacing	—	5 / 60 ms
Maximum accepted change between readings	—	10 cm
Telemetry interval	—	10 s

At power-up the pump stays off until a valid level reading has been obtained, so a reset during a power cut can never start the pump blindly. The sump float switch overrides all other conditions: when it reports low water, the pump is switched off and cannot be restarted until the sump has refilled.

VI. IOT COMMUNICATION AND DASHBOARD

The ESP8266 receives a status line from the Arduino every 10 s and publishes it to an MQTT broker [25], [24]. MQTT was chosen for its small packet overhead and publish/subscribe model, which suit many low-power nodes on an unreliable link [1]. Table 5 lists the topics. The broker uses TLS and per-device credentials, which addresses the security concern raised in [2].

Table 5 MQTT topics used by each tank node

Topic	Direction	QoS	Content
tank/<id>/telemetry	Node → cloud, every 10 s	0	Level (%), volume (L), distance, air temperature, pump state, run time
tank/<id>/alert	Node → cloud, on event	1	Overflow risk, dry sump, no-rise fault, sensor fault
tank/<id>/cmd	Cloud → node	1	Pump request, threshold update, alarm reset
tank/<id>/status	Broker (last will)	1	online / offline

The dashboard shows the current level and volume, a level trend, pump running hours and the number of fill cycles, and pushes notifications for alerts and for a node going offline. If Wi-Fi or the broker is unavailable, the Arduino continues to control the pump and the ESP8266 buffers recent messages for later upload.

VII. EXPERIMENTAL METHODOLOGY

A. Test set-up

The prototype is tested on a transparent cylindrical tank fitted with a measuring tape along its wall, which serves as the reference for water height. A 12 V DC pump fed from a lower reservoir stands in for the mains pump during laboratory tests; the final installation uses a 1000-litre overhead tank and the contactor stage described in Section IV. Air temperature is logged by the DS18B20 probe and checked with a reference thermometer.

B. Test Scenarios

1) Static accuracy: set the water height to 10 known values across the range and record 30 readings at each, with and without temperature compensation.
2) Temperature effect: repeat selected static points with the tank air warmed (for example, by sunlight or a heater outside the tank) and cooled.
3) Switching accuracy: run complete fill and draw-down cycles and record the reference level at each pump ON and OFF event.
4) Dry-run protection: lower the sump float switch while pumping and record the time to pump OFF and alarm.
5) No-rise detection: close the delivery valve while pumping and record the detection time.
6) Sensor fault: disconnect the sensor or cover it, and confirm that the pump switches off and an alert is sent.

7) Network loss and power cycling: switch off Wi-Fi and the controller supply during operation and confirm safe local behaviour and recovery.

C. Evaluation Metrics

Level accuracy is expressed as the mean absolute error (MAE) and root-mean-square error (RMSE) between the measured height h_i and the reference height $h_{ref,i}$ over N readings:

$$MAE = (1/N)\ \Sigma\ |h_i - h_{ref,i}|, \quad RMSE = \sqrt{[(1/N)\ \Sigma\ (h_i - h_{ref,i})^2]} \tag{6}$$

Table 6 Performance metrics and targets

Metric	How it is measured	Target
Level MAE / RMSE	Eq. (6) against tape reference, 10 heights × 30 readings	≤ 1 cm / ≤ 1.5 cm
Repeatability	Standard deviation of 30 readings at fixed level	≤ 0.5 cm
Switching error	Reference level at ON/OFF events minus set threshold	≤ 2% of h_{max}
Dry-run response time	Float switch low → pump OFF	≤ 2 s
No-rise detection time	Valve closed → pump OFF	≤ T_{rise} + 10 s
Telemetry delivery ratio	Messages received ÷ messages sent over 24 h	≥ 98%
Local operation without network	Fill cycle completed with Wi-Fi off	Pass

VIII. RESULTS AND DISCUSSION

A. Effect of temperature compensation

Fig. 3 plots the distance error of an uncompensated sensor, calculated from (2) and (5), for a calibration temperature of 20 °C. At a true distance of 1.0 m the error is +2.6 cm at 35 °C, +4.2 cm at 45 °C and −2.7 cm at 5 °C; at 1.5 m the error at 45 °C reaches 6.3 cm. Because the error grows with distance, it is largest when the tank is nearly empty, which is exactly when the pump switch-on decision is made.

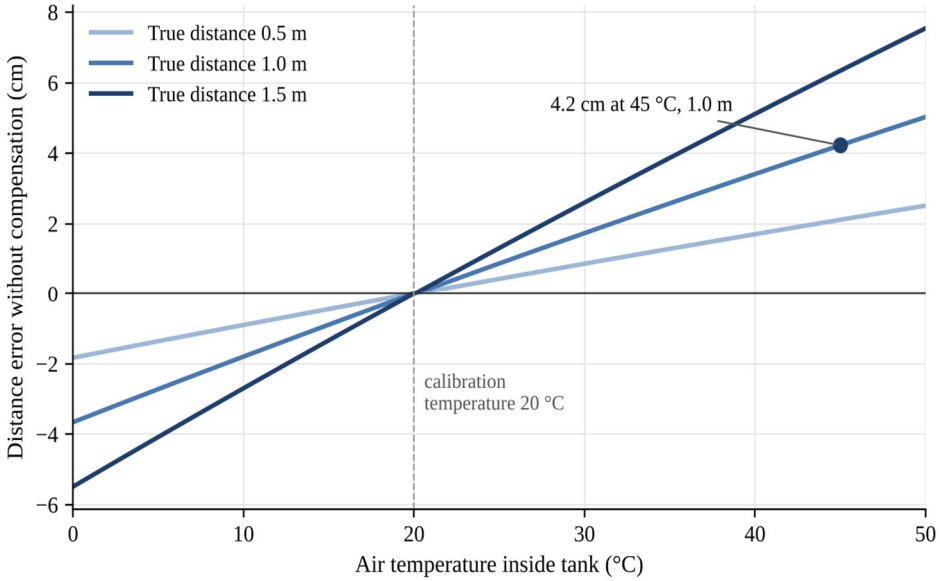


Fig. 3 Calculated distance error of an uncompensated ultrasonic sensor calibrated at 20 °C

In the example tank of Table 4 (D = 1.0 m, cross-section 0.785 m^2), each centimetre of level corresponds to 7.85 litres, so a 4.2 cm error is about 33 litres, or 3.3% of capacity. Since the air in a rooftop tank in summer can be far warmer than the calibration temperature, an uncompensated system would systematically under-report the stored water and switch the pump late. Computing c from the measured temperature removes this source of error; the remaining error is then set by the sensor's own accuracy and by surface ripples, which the median filter reduces.

B. Resolution and timing

The Arduino measures the echo pulse with a resolution of about 1 µs, which corresponds to 0.17 mm of distance, so timer resolution is negligible compared with the sensor's own accuracy of a few millimetres. A round trip over 2 m takes about 12 ms, so five echoes at 60 ms spacing complete in 0.3 s and the control loop easily runs once per second. A pump with a 20 L/min delivery fills the 700 litres between L_{on} and L_{off} in about 35 minutes, well inside the 60-minute maximum run time.

C. Prototype measurements

Measured results are to be entered in Table 7 after the tests in Section VII are completed. Report the number of trials and readings for each metric, and record failed and false events as well as successful ones.

Table 7 Measured performance of the prototype (to be completed after testing)

Metric	Without compensation	With compensation	Notes
Level MAE (cm)	[Insert]	[Insert]	N = 300 readings
Level RMSE (cm)	[Insert]	[Insert]	
Repeatability, SD (cm)	[Insert]	[Insert]	
Switching error at L_{on} / L_{off} (%)	—	[Insert]	Number of cycles: [ ]
Dry-run response time (s)	—	[Insert]	
No-rise detection time (s)	—	[Insert]	
Telemetry delivery ratio (%)	—	[Insert]	24 h test

D. Discussion

Compared with the tank controllers in Table 1, the proposed design adds three safeguards — temperature compensation, protection against dry running and no-delivery faults, and fully local decision-making — for the cost of a temperature probe, a float switch and additional firmware. The no-rise timer is particularly useful because it detects faults such as a closed valve or a dry borewell that no level threshold can catch.

The design has limitations. The JSN-SR04T has a blind zone of about 20–25 cm, so the sensor must be mounted well above the overflow level. Foam, heavy condensation or an inlet pipe in the beam path can cause false echoes, and the sensor must be aimed at an unobstructed area of the water surface. Wi-Fi coverage at a rooftop tank may be poor; LoRa or GSM links, as used in [11], [12], [14], are alternatives. Finally, the analytical results in Sections 8.1 and 8.2 must be confirmed by the measurements in Table 7.

IX. CONCLUSION AND FUTURE SCOPE

This paper presented an IoT-enabled water-level monitoring and pump-control system based on an Arduino Uno, a waterproof ultrasonic sensor and an ESP8266 module. A review of 19 related studies showed that existing systems seldom compensate for temperature or protect the pump beyond simple level thresholds. The proposed system computes the speed of sound from measured air temperature, which removes an error of up to 4.2 cm at 1 m for a 25 °C rise above the calibration temperature, and protects the pump with sump, no-rise, minimum off-time and maximum run-time interlocks while keeping every switching decision local.

Future work will complete the prototype measurements, add leakage detection by monitoring the level while the pump is off, extend the node to several tanks with an ESP32, use LoRaWAN for sites beyond Wi-Fi range, and add solar power for remote installations.

X. ACKNOWLEDGMENT

[Insert acknowledgment of sponsors, institutions or individuals who supported this work.]

REFERENCES

[1] 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 & Tutorials, vol. 17, no. 4, pp. 2347–2376, 2015, doi: 10.1109/COMST.2015.2444095.

[2] 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. no. 309, 2022, doi: 10.3390/w14030309.

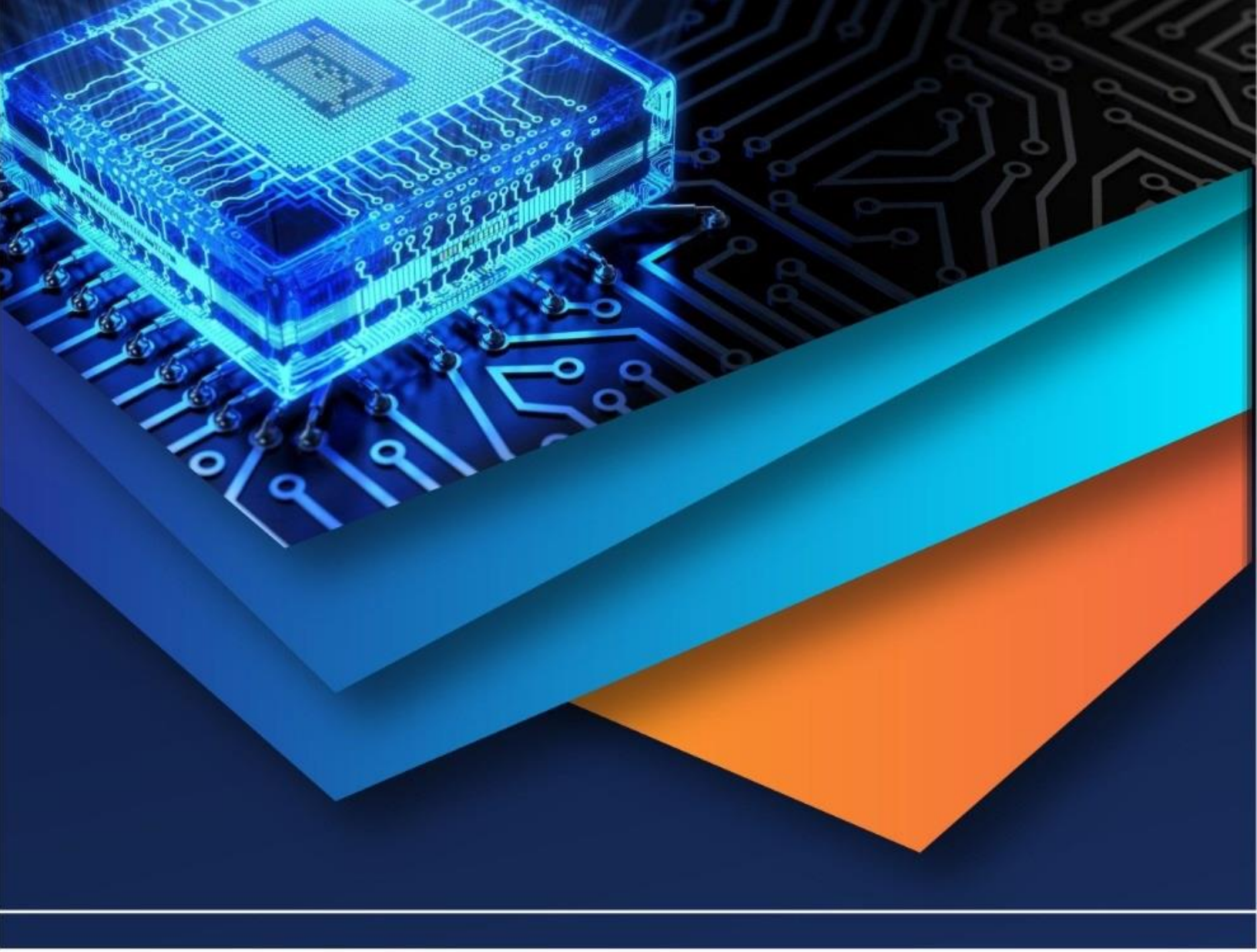
[3] T. Perumal, M. N. Sulaiman, and C. Y. Leong, “Internet of Things (IoT) enabled water monitoring system,” in Proc. IEEE 4th Global Conference on Consumer Electronics (GCCE), Osaka, Japan, 2015, doi: 10.1109/GCCE.2015.7398710.

[4] D. K. Fisher and R. Sui, “An inexpensive open-source ultrasonic sensing system for monitoring liquid levels,” Agricultural Engineering International: CIGR Journal, vol. 15, no. 4, pp. 328–334, 2013.

[5] M. I. Bello, S. M. Gana, M. I. Faruk, and M. J. Umar, "Autonomous ultrasonic based water level detection and control system," Nigerian Journal of Technology, vol. 37, no. 2, pp. 508–513, 2018, doi: 10.4314/njt.372.1690.
[6] S. Sachio, A. Noertjahyana, and R. Lim, "IoT based water level control system," in Proc. 3rd Technology Innovation Management and Engineering Science International Conference (TIMES-iCON), Bangkok, Thailand, 2018.
[7] T. Lestariningsih, B. Artono, N. Hidayatullah, and H. Kusbandono, "Microcontroller and Android HMI based water level and control system," EAI Endorsed Transactions on Internet of Things, vol. 5, no. 17, 2019, doi: 10.4108/eai.28-1-2019.162807.
[8] M. Krishnaveni, S. K. P. Kumar, E. A. Muthusamy, J. Kowshick, and K. G. Arunya, "Real-time monitoring of water level and storage dynamics of irrigation tank using IoT," H2Open Journal, vol. 3, no. 1, pp. 392–400, 2020, doi: 10.2166/h2oj.2020.123.
[9] A. Praveen, R. Radhika, D. Sidharth, S. Ambat, and T. Anjali, "Smart water level monitoring and management system using IoT," in Proc. 6th International Conference on Communication and Electronics Systems (ICCES), 2021, pp. 482–487.
[10] Y. Panagopoulos, A. Papadopoulos, G. Poulis, E. Nikiforakis, and E. Dimitriou, "Assessment of an ultrasonic water stage monitoring sensor operating in an urban stream," Sensors, vol. 21, no. 14, Art. no. 4689, 2021, doi: 10.3390/s21144689.
[11] I. P. E. A. Harrischandra, I. M. S. Wibawa, and N. Wendri, "Designing water level control equipment using SMS-based ultrasonic sensors," Buletin Fisika, vol. 23, no. 2, pp. 121–129, 2022, doi: 10.24843/BF.2022.v23.i02.p07.
[12] J. Kabi, C. Maina, S. Wamae, and E. Mharakurwa, "Low cost, LoRa based river water level data acquisition system," HardwareX, 2023 (open access, PMC10050633).
[13] L. T. Petru, M. V. Tivlea, and S. E. Popa, "Water level control and monitoring system in a tank made with Arduino Uno and NodeMCU ESP8266 development boards," Preprints, 2023, doi: 10.20944/preprints202304.0735.v1.
[14] V. J. Baratella, C. T. B. da Costa, D. Spadoti, A. G. Almeida, and B. C. da Silva, "Ultra low power water level transmission system using LoRaWAN technology," in Proc. XLI Simpósio Brasileiro de Telecomunicações e Processamento de Sinais (SBrT), 2023, doi: 10.14209/sbrt.2023.1570917497.
[15] J. Rajanbabu, G. R. Venkatakrishnan, R. Rengaraj, M. Rajalakshmi, and N. Jayaprakash, "An integrated smart water management system for efficient water conservation," International Journal of Electrical and Computer Engineering, vol. 15, no. 1, pp. 635–644, 2025, doi: 10.11591/ijece.v15i1.pp635-644.
[16] W. X. Yap and N. H. Ali, "IoT-based smart water management system for efficient consumption and conservation," in Proc. Multimedia University Engineering Conference (MECON), Cyberjaya, Malaysia, 2025, pp. 1–5, doi: 10.1109/MECON67253.2025.11277091.
[17] I. S. Mohamed Jumaiedy and A. Suratkon, "IoT based water tank level and turbidity detection and monitoring system," Recent Trends in Civil Engineering and Built Environment, vol. 6, no. 1, pp. 174–185, 2025.
[18] Y. M. R. N. T. Yapa, S. J. Sooriyaarachchi, and C. R. De Silva, "IoT-based river water level monitoring system using LoRa technology," in Proc. Engineering Research Unit (ERU) Symposium, University of Moratuwa, 2025, doi: 10.31705/ERU.2025.8.
[19] S. Sadi and S. Mulyati, "IoT-enabled smart water level monitoring and automatic floodgate control using Arduino and cloud integration," Journal of Electrical Engineering and Computer (JEECOM), vol. 8, no. 3, pp. 136–144, 2026, doi: 10.33650/jeecom.v8i3.17102.
[20] E. Kabir, M. R. Ali, M. R. A. Mamun, N. S. Ahmed, and M. S. Alam, "IoT-enabled automatic water level controller for remote monitoring and management," Journal of the Bangladesh Agricultural University, vol. 24, no. 3, pp. 275–283, 2026, doi: 10.3329/jbau.v24i3.93900.
[21] O. Cramer, "The variation of the specific heat ratio and the speed of sound in air with temperature, pressure, humidity, and CO2 concentration," Journal of the Acoustical Society of America, vol. 93, no. 5, pp. 2510–2525, 1993, doi: 10.1121/1.405827.
[22] Microchip Technology Inc., "ATmega328P 8-bit AVR microcontroller with 32K bytes in-system programmable flash," datasheet.
[23] JSN-SR04T waterproof ultrasonic ranging module, product datasheet.
[24] Espressif Systems, "ESP8266EX datasheet," technical documentation.
[25] OASIS, "MQTT version 3.1.1," OASIS Standard, A. Banks and R. Gupta, Eds., Oct. 2014.

Copyright & License:

© Authors retain the copyright of this article. This work is published under the Creative Commons Attribution 4.0 International License (CC BY 4.0), permitting unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)