WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Water Level Indicator using Ultrasonic Sensor and Arduino

Abhishek Chauhan[1], Aaditya Kumar Gupta[2], Abhay Gupta[3], Dipesh Sahu[4]

[1]Computer Science and Engineering Department, Shri Shankaracharya Technical Campus, Bhilai – 490020, India
[2]Computer Science and Engineering Department, Shri Shankaracharya Technical Campus, Bhilai – 490020, India
[3]Computer Science and Engineering Department, Shri Shankaracharya Technical Campus, Bhilai – 490020, India
[4]Computer Science and Engineering Department, Shri Shankaracharya Technical Campus, Bhilai – 490020, India
[1]abhishekchauhan9826@gmail.com , [2]aadityakumargupta03@gmail.com, [3]abha19gupt11@gmail.com, [4]dipeshsahu988@gmail.com

ABSTRACT —

Although the majority of the Earth's surface is covered with water, less than 5% is suitable for human use. Therefore, conserving water has become a critical global concern, necessitating improved water management strategies. Monitoring water levels is an essential aspect for both residential and governmental purposes. This paper explores an automated water level monitoring system using an ultrasonic sensor, which measures the distance from the sensor to the water surface and calculates the percentage of water present in the tank. The system consists of an Arduino microcontroller, a water tank, a buzzer, and an LCD display. All components are integrated with the Arduino and operate automatically based on the uploaded program. The tank is conceptually divided into levels such as 10%, 20%, up to 100%, where 10% represents a low water condition and 100% indicates a full tank. When the water level drops to around 10%, the buzzer activates to alert the user. This eliminates the need for constant manual supervision. The core working principle is based on ultrasonic sound wave reflection. The sensor emits ultrasonic waves through one module and receives the reflected waves through another. The time taken for the echo to return is used to calculate the water level accurately.

Keywords — Ultrasonic waves, echo, trig, Arduino, smart water, water level management, automation, water level, trig, echo.

I. Introduction

Water scarcity continues to be among India's biggest issues, with millions of families and agricultural networks struggling with reduced access to clean water. In most areas, manual checking of water storage tanks, including reservoirs and tanks, results in great inefficiencies [1]. Spillover from overflowing tanks and unused capacity due to inaccurate level checks lead to water wastage, aggravating the water-scarce situation in many areas. These inefficiencies are especially bad in rural homes and urban slums, where resources are scarce and access to advanced technology is poor. Automated water level monitoring systems provide an exciting solution by offering real-time water level data, allowing users to make smart decisions regarding water use and eliminating the need for continuous human intervention. Such systems save water while meeting global and domestic objectives of sustainable resource use.

In this paper, an affordable and user-friendly system of water level indicator is presented with the capability of overcoming such difficulties. The system utilizes an Arduino Uno R3 microcontroller, an HC-SR04 ultrasonic sensor, a 16x2 I2C LCD display, and an active buzzer for sensing water levels in vessels [2]. Through the measurement of water height, computation of empty space and percentage of water left, and the provision of audible signals upon critically low levels (≤3 cm), the system provides effective water management. Affordability is given preference in the design, with an overall cost of less than $20, so it can be afforded by low-income consumers and small-scale farm users in India. Moreover, the system is also scalable and adjustable, accommodating different container sizes and uses, ranging from household water tanks to communal reservoirs.

.

Fig. 1. Echo illustration

The development process employed TinkerCad for simulation of circuits and the Arduino Integrated Development Environment (IDE) for coding, which resulted in a strong and reliable system. The work was aimed by well-defined objectives: to come up with an affordable solution to monitor the water level, to display in real-time the height of the water, space when empty, and percentage, to alert about low water levels in a critical level, and to test and certify the system's performance via simulation and hardware verification. These goals are in line with the conference theme of sustainable national development since the system supports water conservation and effective use of resources, key elements of India's sustainable development agenda.

The paper is organized to give an overall view of the project. Section II presents the methodology, such as system requirements, components, and development methodology. Section III describes the system design, touching on architecture, circuit connections, and the algorithm. Section IV presents the implementation and testing procedures, while Section V gives the results and their implications. Finally, Section VI concludes the paper and suggests potential future upgrades to enhance the system's functionality and impact even further.

II. Methodology

The system development of the water level indicator system followed a formal methodology to ensure functionality, cost-effectiveness, and reliability. This section discusses system requirements, hardware and software elements, and the phased approach to development, which gives background to the design decisions and technical implications.

A. SYSTEM REQUIREMENTS

The system was built to satisfy certain functional and non-functional requirements in order to solve the issues of manual water monitoring. Functionally, the system had to measure water height in a container with an HC-SR04 ultrasonic sensor that uses sound waves to find distance. The values were to be read on a 16x2 I2C LCD with the water height, empty space in centimeters, and percentage of water in the container's capacity. To maximize usability, the system had to set off an audible alarm through a buzzer once the water level reached below 3 cm, indicating the need for refilling. The system also needed to refresh its measurements and show them every 0.5 seconds to reflect real-time feedback to the users.

Non-functional requirements dealt with usability and convenience. The overall system cost had to be limited to below $20 to provide an affordable solution for large-scale deployment in resource-limited environments. The response had to be fast, with less than 1 second response time, for the updates to be provided in a timely manner. Power efficiency was of utmost importance to allow for continuous operation, especially in regions where the availability of power was spotty. Lastly, the interface had to be user-friendly, providing clean and understandable output on the LCD, making it easy to use for people with minimal technical know-how, such as small farmers or rural households.

B. COMPONENTS

Hardware components were well chosen in a trade-off of cost, performance, and ease of integration. The Arduino Uno R3 acted as the microcontroller, picked due to its versatility, cost-effectiveness, and good community support. Equipped with 14 digital input/output pins, 6 analog inputs, and 32 KB flash memory, it had enough processing capacity for the system requirements [3]. The HC-SR04 ultrasonic sensor was chosen because it can measure between 2 to 400 cm with a tolerance of ±3 mm, suitable for sensing water levels in tanks of different capacities. The 16x2 I2C LCD display was used due to its small size and effective communication using the I2C protocol, which minimizes the number of pins needed and makes wiring simpler. An active buzzer powered at 5V was also added to allow audible notifications so that users were informed of low water levels even when they were not paying attention to the LCD. Prototyping was aided by a breadboard and jumper wires, with power and programming capability being delivered through a USB cable.

From a software perspective, the Arduino IDE was employed in writing and transferring the program onto the Arduino Uno. Web-based TinkerCad allowed circuit designing and simulation to ensure connections before actual assembly and enabled the team to check if connections were correctly made. Two of the most important libraries used were the Wire library for I2C communication between the LCD and Arduino, and LiquidCrystal_I2C library to manage the LCD display. These libraries helped save time while programming and provided dependable operation of the system components.

C. DEVELOPMENT APPROACH

Development was split into three different phases to maintain a systematic and productive workflow. During the first phase, the circuit was created and simulated through TinkerCad. This process enabled the team to check the interfacing of the Arduino, ultrasonic sensor, LCD, and buzzer and resolve any possible issues before assembling them physically. The simulation environment offered a safe platform to check the behavior of the system under different conditions, e.g., varying water levels or sensor distances.

In the second phase, the software was implemented using the Arduino IDE. The program, or "sketch," was developed to drive the ultrasonic sensor, calculate its measurements, and present the results on the LCD. The sketch also contained code to trigger the buzzer when needed. The Wire and LiquidCrystal_I2C libraries were used to simplify communication with the LCD, minimizing the code complexity. The software was tested and perfected in stages to provide accurate calculations and consistent operation.

The last stage consisted of hardware prototyping and testing. The parts were mounted on a breadboard, and the system was tested in a physical water container to ensure its functionality in actual conditions. This stage consisted of unit testing of individual components, integration testing of the entire system, and performance testing to measure response time, power usage, and reliability. By following this phased approach, the team ensured that the system met its functional and non-functional requirements while maintaining a focus on affordability and usability.

III. System Design

The system design process aimed to design a robust and efficient structure, set solid relationships between parts, and construct an algorithm that would control the operation of the system. The section presents in-depth details on the system architecture, circuit links, and the algorithm, and it provides a glimpse into the technical basis of the water level indicator.

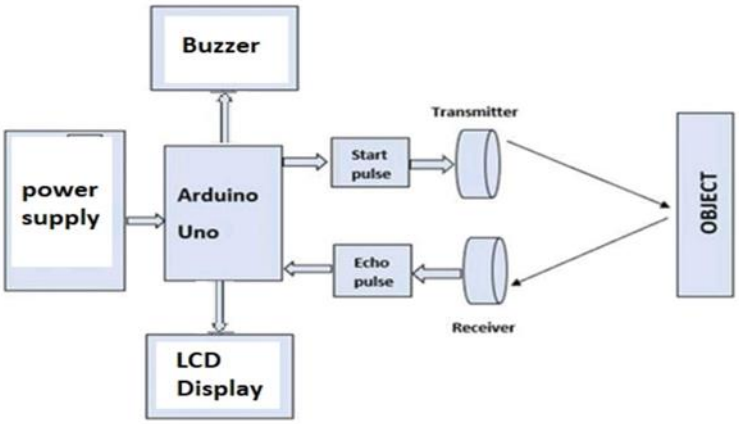


Fig. 2. System Design

A. SYSTEM ARCHITECTURE

The water level indicator system was implemented with a modular structure to facilitate ease of use and maintenance. The system had four main modules, each responsible for a function. The input module was the HC-SR04 ultrasonic sensor, which detected the distance between the sensor and the water level by sending ultrasonic waves and computing the time it took for the echo to get back [3,4]. This distance was utilized to compute the water height in the container.

The processing module was managed by the Arduino Uno R3, which acted as the brain of the system. The Arduino took raw data from the ultrasonic sensor, processed it to determine the water height, empty space, and percentage of water left, and issued commands to the output devices. The output module featured the 16x2 I2C LCD display, which displayed the calculated readings in readable form, and the buzzer, which gave an audible warning when the water level dipped to 3 cm or less. The power module, being a USB cable or a 9V battery, powered all the parts, ensuring round-the-clock usage.

This modular construction enabled simple troubleshooting and future modifications, like the inclusion of wireless communication for remote sensing or incorporation of substitute sensors for improved accuracy. By compartmentalizing the system into individual modules, the design was flexible and adaptable to various applications.

B. CIRCUIT DIAGRAM

The circuit was meticulously constructed to provide sound communication among elements while keeping it simple. The HC-SR04 ultrasonic sensor was hooked up to the Arduino in the following manner: the Trig pin to analog pin A0, the Echo pin to analog pin A1, the VCC pin to the 5V output of the Arduino, and the GND pin to the ground of the Arduino. This allowed the Arduino to provide the sensor with a trigger pulse and receive the echo to measure distance.

The 16x2 I2C LCD display was interfaced with the use of the I2C protocol, which needs only two data lines. The SDA (data) pin was interfaced to analog pin A4, the SCL (clock) pin to analog pin A5, the VCC pin to the 5V output of the Arduino, and the GND pin to ground. This setup minimized the number of pins used, which made the circuit simpler and left Arduino pins free for future expansions.

The buzzer was wired with its positive terminal to digital pin D11 and its negative terminal to ground. This enabled the Arduino to switch the buzzer on and off, creating audible beeps when activated by low water levels. The whole circuit was built on a breadboard, with jumper wires providing safe connections. A USB cable linked the Arduino to a power source, usually a computer or a 5V power adapter, for programming and use.

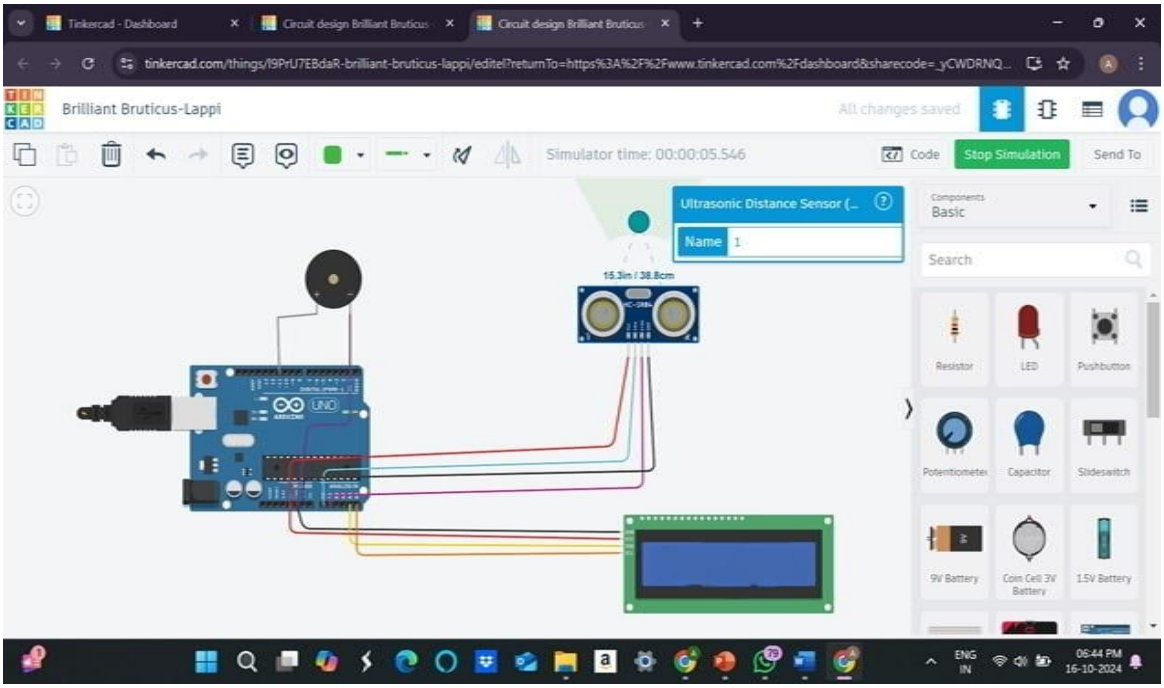


Fig. 3. Circuit diagram of the water level indicator system.

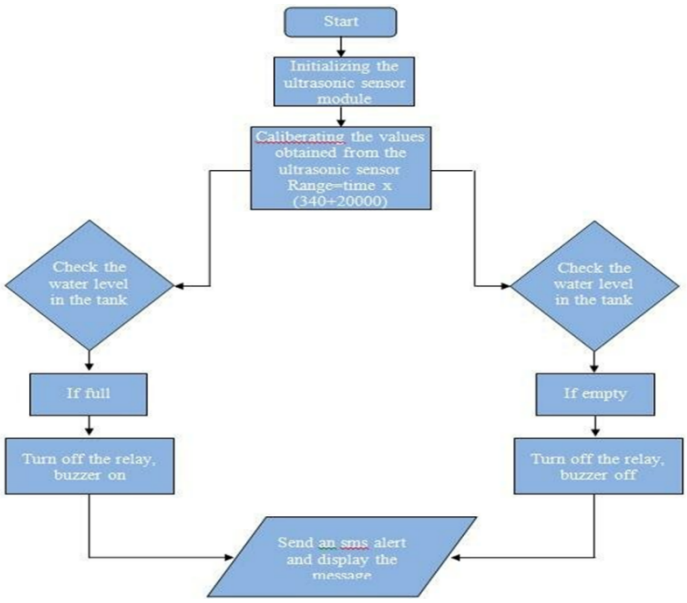
C. ALGORITHM

The operation of the system was controlled by a clearly defined algorithm that ensured the correct measurements and timely results. At the setup stage, the Arduino set up the pins of the ultrasonic sensor (Trig and Echo), the LCD, and the buzzer. It also recorded the total height of the void container by reading from the ultrasonic sensor initially, which was used as a reference point for future calculations.

Inside the main loop, the system carried out the following actions recursively. The Arduino first generated a 10-microsecond pulse on the HC-SR04 sensor's Trig pin to trigger a measurement. The sensor then sent ultrasonic waves and measured the time it took for the echo to reach it, captured as the pulse duration through the Echo pin. The height of water was calculated based on the formula: distance = (duration / 2) / 29.1, where duration was divided by 2 for the round trip of the sound wave and 29.1 is the velocity of sound in air (microseconds per centimeter). This gave the distance between the sensor and the surface of water and this was then subtracted from the overall height of the container to find the height of water.

Then, the space left empty was determined by subtracting the water height from the height of the container. The percentage of water left was calculated as: waterPercent = (waterHeight * 100.0) / totalHeight, to provide an accurate indication of the fill level of the container. These measurements—water height, space left empty, and percentage—were shown on the 16x2 I2C LCD, with proper formatting to make them easy to read. If the water level was 3 cm or below, the buzzer was triggered to emit five 200-millisecond beeps, warning the user of the low water level. The system refreshed its measurements and display every 0.5 seconds, giving real-time feedback without overloading the Arduino's processing capacity.

Fig. 4. Block diagram of water level measurement and control in tank.



This algorithm was optimized to be efficient, keeping computational overhead to a minimum while producing accurate and timely results. The addition of safety checks, like the handling of out-of-range sensor inputs, made the system more reliable in actual-world conditions.

IV. Implementation and Testing

The implementation and testing phases were critical to transforming the conceptual design into a functional system. This section provides a detailed account of how the hardware and software were assembled, programmed, and validated, ensuring the system met its objectives.

A. IMPLEMENTATION

1) Hardware Setup : The hardware was assembled on a breadboard to facilitate prototyping and modifications. The Arduino Uno R3 was placed at the center of the breadboard, serving as the hub for all connections. The HC-SR04 ultrasonic sensor was mounted above the test container, with its Trig and Echo pins connected to analog pins A0 and A1, respectively, and its power pins connected to the Arduino's 5V and ground lines [5]. The 16x2 I2C LCD was positioned for easy viewing, with its SDA and SCL pins wired to analog pins A4 and A5, and its power pins connected to 5V and ground. The buzzer was attached to digital pin D11 and ground, ensuring it could be activated by the Arduino's digital output. Jumper wires were used to secure all connections, and care was taken to avoid loose or incorrect wiring that could cause signal interference. The Arduino was powered via a USB cable connected to a computer, which also served as the interface for uploading the program.

The test container was a 20 cm deep glass vessel, chosen for its standard size and compatibility with the sensor's measurement range. The sensor was positioned at the top of the container, facing downward, to accurately measure the distance to the water surface. The hardware setup was designed to be portable and easy to replicate, making it suitable for deployment in various settings, from households to small farms.

2) Software Implementation :

The software was developed using the Arduino IDE, which provided a user-friendly environment for writing and uploading the program to the Arduino Uno [6,7]. The sketch incorporated the Wire and LiquidCrystal_I2C libraries to manage communication with the LCD. The code was structured to initialize the system, perform measurements, process data, and control the outputs in a continuous loop. Below is the complete code

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 16, 2);

#define trigPin A0       // Ultrasonic sensor Trig pin
#define echoPin A1       // Ultrasonic sensor Echo pin
#define buzzerPin 11     // Buzzer pin

long totalHeight = 0;    // Store total glass height

void setup() {
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(buzzerPin, OUTPUT);

  lcd.init();            // Initialize LCD
  lcd.backlight();
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Measuring Glass...");
  delay(2000);           // Wait for sensor to stabilize

  totalHeight = measureHeight();  // Measure the empty glass height

  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Glass Height:");
  lcd.setCursor(0, 1);
  lcd.print(totalHeight);
  lcd.print(" cm");

  delay(2000);           // Show glass height
```

```
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Empty:");        // Label for water level
  lcd.setCursor(0, 1);
  lcd.print("Water:");        // Label for empty space
}

void loop() {
  long waterHeight = measureHeight();               // Current water level
  if (waterHeight > totalHeight) {
    waterHeight = totalHeight;                      // Safety check
  }

  long emptySpace = totalHeight - waterHeight;      // Empty portion

  int waterPercent = (int)((waterHeight * 100.0) / totalHeight);   // Percentage calculation

  // Display water height
  lcd.setCursor(7, 0);
  lcd.print("     ");                               // Clear old data
  lcd.setCursor(7, 0);
  lcd.print(waterHeight);
  lcd.print("cm");

  // Display empty space
  lcd.setCursor(7, 1);
  lcd.print("     ");
  lcd.setCursor(7, 1);
  lcd.print(emptySpace);
  lcd.print("cm");

  // Display water level percentage (right-aligned)
  lcd.setCursor(13, 0);
  lcd.print("    ");                                // Clear old %
  lcd.setCursor(13, 0);
  lcd.print(waterPercent);
  lcd.print("%");

  // Buzzer alert: beep if water is ≤ 3 cm
  if (waterHeight <= 3) {
    for (int i = 0; i < 5; i++) {
      digitalWrite(buzzerPin, HIGH);
      delay(200);
      digitalWrite(buzzerPin, LOW);
      delay(200);
    }
  } else {
    digitalWrite(buzzerPin, LOW);
  }

  delay(500);  // Update every half second
}

// Function to measure height using ultrasonic sensor (returns cm)
long measureHeight() {
  long duration;

  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
```

```
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  duration = pulseIn(echoPin, HIGH, 30000);  // Timeout safety

  if (duration == 0) return 0;               // Out of range case

  long distance = (duration / 2) / 29.1;     // Convert to centimeters
  return distance;
}
```

Fig. 5. Project Images

B. TESTING

The testing procedure was done in three phases—unit testing, integration testing, and performance testing—to guarantee the reliability and correctness of the system.

1) Unit Testing : Each unit was tested separately to check its function. The HC-SR04 ultrasonic sensor was tested in distances of 5 cm, 10 cm, and 20 cm and verified its precision within ±3 mm as the datasheet demands. The 16x2 I2C LCD was tested to provide correct display of water height, empty space,

and percentage without flicker, and proper positioning [8]. The buzzer was also tested to validate that it was producing five beeps of duration 200-millisecond when actuated, confirming the alert's audibility as well as being appropriately timed. These tests established that each hardware worked as specified prior to integrating them together.

2) Integration Testing : The complete system was then tested in a glass container of 20 cm depth for imitation of actual use. The water level was varied in steps, and the outputs of the system were measured. The outcomes, tabulated in Table II, proved that the system could measure and show the water height, empty space, and percentage correctly and also trigger the buzzer when the water level dropped to 3 cm or less. For instance, when the water level was at 2 cm, the system accurately exhibited 18 cm of free space and a fill level of 10%, and the buzzer made five beeps. At 10 cm, the system exhibited 10 cm of free space and a fill level of 50%, with the buzzer off [9,10]. All these confirmed that the system functioned together as a cohesive entity and was satisfactory in fulfilling its functional demands.

TABLE I. TEST RESULTS

Water Height (cm)	Empty Space (cm)	Percentage (%)	Buzzer Status
02	18	10	ON
10	10	50	OFF
18	02	90	OFF

3) Performance Testing : This evaluates the efficiency of the system to function. The response time remained at 0.5 seconds, fulfilling the requirement for real-time update[11]. Power usage was around 50 mA at 5V, showing it to be fit for continuous use. The system also coped with out-of-range sensor reading well via a timeout mechanism, providing robustness in adverse conditions.

V. Results and Discussions

A. RESULTS

The water level indicator system provided outstanding performance on all design targets, showing its capability as a useful instrument for water management. The system provided a measurement accuracy of ±3 mm for water height, which was very accurate data that can be trusted by users in making decisions. Real-time data were provided continuously every 0.5 seconds, with the 16x2 I2C LCD display showing water height, free space, and percentage in a neat, well-organized, and easily readable manner. The active buzzer consistently sounded for water heights of 3 cm or below, providing five different 200-millisecond beeps that effectively notified users of critical situations. The overall price of the system was $18, and component prices were as listed: Arduino Uno R3 for $10, HC-SR04 ultrasonic sensor for $3, 16x2 I2C LCD display for $4, and active buzzer for $1. This cost-effectiveness further increases the affordability of the system, especially by low-income families and communities in water-poor areas. Simulation in TinkerCad highly consistent with hardware performance confirms the design process and indicates the system readiness for real-world application.

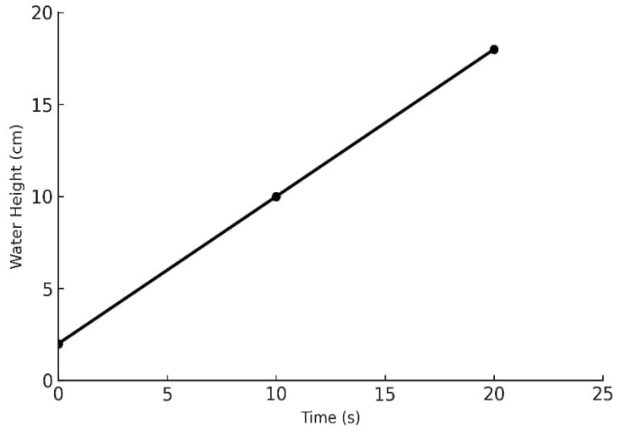


Fig. 6. Water level trend during testing.

B. DISCUSSIONS

The water level indicator system provides a valuable contribution to sustainable water management through the timely and precise supply of information that assists the user in optimizing water consumption. By warning users about low water levels, the system avoids underfilling, which may cause shortages, and prompts on-time refills, avoiding wastage through overflow or mismanagement. Being low-cost and open-source, it is extremely accessible, especially for rural and urban Indian households, where water conservation is a matter of utmost urgency. In contrast to commercial water level monitoring systems,

which typically include advanced features like remote monitoring or wireless connectivity, this system emphasizes simplicity and cost-effectiveness, making it a perfect solution for resource-limited environments. Although it does not have remote capabilities, its emphasis on core functionality guarantees it fulfills the requirements of users who need robust, low-cost monitoring without extensive infrastructure.

The process development was confronted by two significant hurdles, both of which were effectively overcome. In the first place, the HC-SR04 ultrasonic sensor sometimes returned noisy readings when in a high humidity environment, and this might impair measurement quality. This was averted through the introduction of a timeout safety function within the software, which invalidates invalid or out-of-range measurements to promote stable operation. Second, the 16x2 I2C LCD display had occasional address conflicts, which were fixed by checking and hard-coding the proper I2C address (0x27) upon system startup. Those remedies improved the system's reliability, allowing it to be deployed in various environmental conditions, ranging from the high humidity of coastal areas to the dryness of inland environments.

C. CONTRIBUTION TO SUSTAINABILITY

The water level indicator system directly assists India's goals towards sustainable development, especially those of water resource management and conservation of the environment. With its accurate monitoring and prompt warning, the system minimizes wastage of water, making every drop count. This is imperative in India, where millions of homes battle on a daily basis to access enough water for drinking, cooking, and hygiene. Low power consumption of the system, measured around 50 mA at 5V, keeps its energy footprint to a minimum, thereby qualifying for extended operation in electricity-starved environments. In addition, the system's architecture accommodates integration with solar power, further reducing dependence on fossil fuels for its operation and adding value to its ecological credentials. By enabling users to make informed choices regarding water consumption, the system is part of a larger endeavor to fight water scarcity, support resource equity, and create resilient communities in water-scarce areas.

VI. Conclusions

This paper has discussed in detail a water level indicator system that can aid sustainable water management in India. Constructed with an Arduino Uno R3, an HC-SR04 ultrasonic sensor, a 16x2 I2C LCD display, and an active buzzer, the system is able to measure water height, empty space, and percentage correctly and update these values every 0.5 seconds to offer real-time feedback. It consistently activates audible alarms at water levels of 3 cm or lower, allowing users to take immediate action to avoid shortages. Selling for only $18, the system is a budget-friendly, user-friendly solution for residential and small-scale applications, confirmed through extensive simulation in TinkerCad and hardware testing in an actual environment.

The system's contributions to water conservation are significant, providing a practical and scalable tool to reduce wastage and encourage efficient usage. Its affordability, simplicity, and open-source characteristics make it versatile for use in a broad array of applications, from residential water tanks to community reservoirs, specifically in India's water-lacking areas. The successful overcoming of challenges like sensor noise and LCD address conflicts highlights the system's robustness and reliability, making it able to function effectively under different conditions. In the future, future developments could greatly enhance the system's capabilities. Incorporating Internet of Things (IoT) technology would allow remote monitoring through smartphones or cloud platforms, enabling users to monitor water levels anywhere and enable integration with smart home systems. Moreover, the integration of solar power would minimize the system's ecological impact, making it an even more sustainable option for resource-poor communities. By filling a key gap in India's water management ecosystem, this system is a significant step toward a more sustainable and equitable future.

Acknowledgement

We hereby extend our heartfelt thanks to all those who helped us in the successful execution of our research project entitled "Water Level Indicator using Ultrasonic Sensor and Arduino." We are profoundly thankful to our project mentor, Prof. Somesh Dewangan, Associate Professor, Department of Computer Science and Engineering, Shri Shankaracharya Technical Campus, Bhilai, for his rich input, incessant support, and valuable feedback during the project. His knowledge in the area of embedded systems and interest in our progress were responsible for overcoming obstacles and ensuring our project goals.

We would like to express our sincere gratitude to the staff and faculty members of the Computer Science and Engineering Department at Shri Shankaracharya Technical Campus, Bhilai, for extending to us the resources, infrastructure, and support required to perform this work. Their support provided a learning- and innovation-friendly environment.

We also appreciate the open-source community for their contribution of tools like the Arduino IDE, TinkerCad, and libraries Wire and LiquidCrystal_I2C, which greatly simplified the development and simulation process. These tools were pivotal in the effective implementation of our project.

We express gratitude to our colleagues and friends for their co-operation, advice, and encouragement throughout the period of this research. Their offer of ideas and suggestions and providing critical feedback enhanced our work.

Finally, we acknowledge a particular debt of gratitude to our families for their consistent encouragement, tolerance, and patience that provided us with the fortitude to continue working on this project with diligence.

REFERENCES

[1] Kumar, R., et al. (2018). Design and Implementation of Water Level Monitoring System Using Ultrasonic Sensor. International Journal of Engineering Research and Applications.

[2]Sharma, P., & Roy, A. (2020). Smart Water Tank Monitoring System Using IoT. Journal of Advanced Research in Embedded Systems.

[3]Arduino Official Documentation. (n.d.). Getting Started with Arduino UNO. Retrieved from https://www.arduino.cc
[4]HC-SR04 Ultrasonic Sensor Datasheet. (n.d.). Retrieved from https://www.electronicwings.com
[5] http://arduino-info.wikispaces.com
[6] https://components101.com/ultrasonic-sensor-working-pinout-datasheet
[7] Beza Negash Getu, Hussian A. Attia, Automatic Water Level Sensor and Controller System, IEEE volume-1 issue-3 2016.
[8] https://circuitdigest.com/microcontroller-projects/arduino-ultrasonic-sensor-based-distance-measurement
[9] https://youtu.be/DC3n1K5aVfM?si=ccvm7z-zVXRU2SzU
[10] Kundu T, Placko D 2010 Advanced Ultrasonic Methods for Material and Structure Inspection,Wiley-ISTE, UK.
[11] http://arduino-info.wikispaces.com/Ultrasonic+Distance+Sensor