LONDON METROPOLITAN UNIVERSITY

islington college
(इस्लिङ्टन कलेज)

**Module Code & Module Title**
**CS5068NI– Cloud Computing & IoT**

# Water Level Monitoring System

**Assessment Type 50% Group Report**

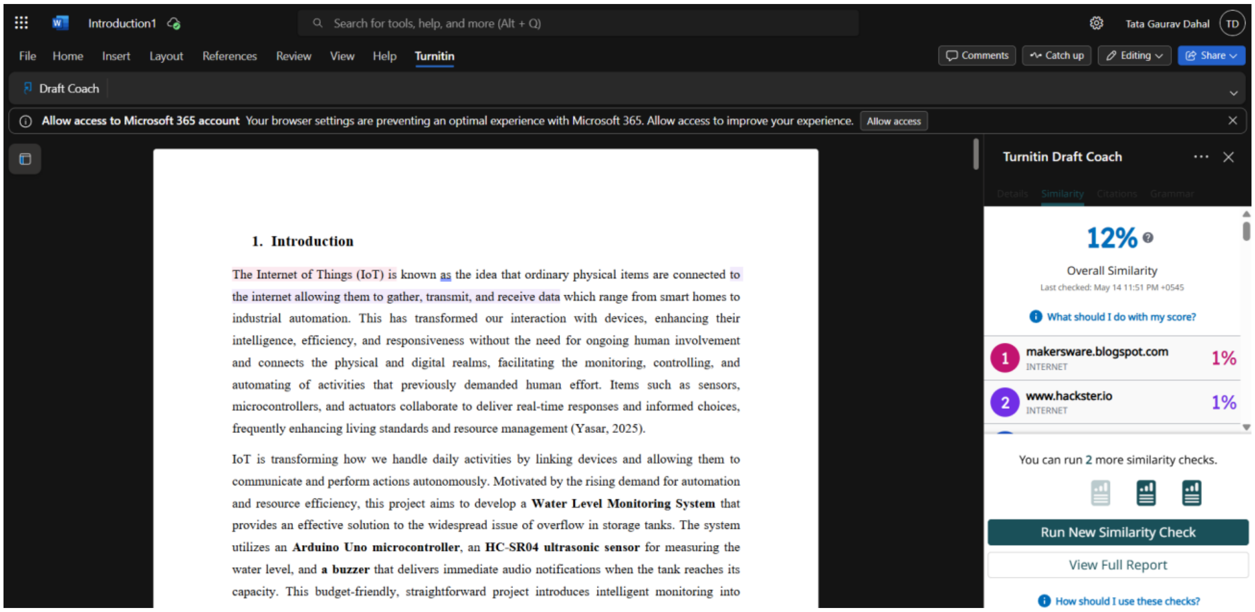
**Semester2024 Spring**

**Group Members**

| London Met ID | Student Name |
|---|---|
| 23047459 | Tata Gaurav Dahal |
| 23047449 | Piyush Shrestha |
| 23048520 | Rejina Phuyal |
| 23047411 | Gaurav Pratap Malla |
| 23047446 | Shrevika Khadka |

**Assignment Due Date: 15th May 2025**
**Assignment Submission Date:15th May 2025**
**Submitted to: Mr. Sugat Man Shakya**
**Word Count:5956**

*I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a mark of zero will be awarded.*


Introduction1
Search for tools, help, and more (Alt + Q)
Tata Gaurav Dahal
File Home Insert Layout References Review View Help Turnitin
Comments
Catch up
Editing
Share
Draft Coach
Allow access to Microsoft 365 account Your browser settings are preventing an optimal experience with Microsoft 365. Allow access to improve your experience.
Allow access
1. Introduction
The Internet of Things (IoT) is known as the idea that ordinary physical items are connected to the internet allowing them to gather, transmit, and receive data which range from smart homes to industrial automation. This has transformed our interaction with devices, enhancing their intelligence, efficiency, and responsiveness without the need for ongoing human involvement and connects the physical and digital realms, facilitating the monitoring, controlling, and automating of activities that previously demanded human effort. Items such as sensors, microcontrollers, and actuators collaborate to deliver real-time responses and informed choices, frequently enhancing living standards and resource management (Yasar, 2025).
IoT is transforming how we handle daily activities by linking devices and allowing them to communicate and perform actions autonomously. Motivated by the rising demand for automation and resource efficiency, this project aims to develop a Water Level Monitoring System that provides an effective solution to the widespread issue of overflow in storage tanks. The system utilizes an Arduino Uno microcontroller, an HC-SR04 ultrasonic sensor for measuring the water level, and a buzzer that delivers immediate audio notifications when the tank reaches its capacity. This budget-friendly, straightforward project introduces intelligent monitoring into
Turnitin Draft Coach
12%
Overall Similarity
Last checked: May 14 11:51 PM +0545
What should I do with my score?
1 makersware.blogspot.com INTERNET 1%
2 www.hackster.io INTERNET 1%
You can run 2 more similarity checks.
Run New Similarity Check
View Full Report
How should I use these checks?

## Acknowledgements

We would like to sincerely express our deepest gratitude to our module leader, Mr. Sugar Man Sakya, and our tutor, Mr. Bishnu Pandey, for providing us with this golden opportunity to undertake research and gain valuable hands-on experience through this group coursework project on the Water Level Monitoring System. Their guidance and encouragement have been instrumental in helping us understand how IoT functions in real-world applications.

We are also truly grateful to Islington College for offering the necessary resources and fostering an ideal learning environment that allows us to explore, experiment, and grow throughout the project.

Lastly, we extend our heartfelt thanks to our friends for their unwavering support, motivation, and encouragement, which continually inspired us to push forward and do our best.

## Table Of Contents

## Table of Figures

# Table of Tables

# 1. Introduction

The Internet of Things (IoT) is known as the idea that ordinary physical items are connected to the internet allowing them to gather, transmit, and receive data which range from smart homes to industrial automation. This has transformed our interaction with devices, enhancing their intelligence, efficiency, and responsiveness without the need for ongoing human involvement and connects the physical and digital realms, facilitating the monitoring, controlling, and automating of activities that previously demanded human effort. Items such as sensors, microcontrollers, and actuators collaborate to deliver real-time responses and informed choices, frequently enhancing living standards and resource management (Yasar, 2025).

IoT is transforming how we handle daily activities by linking devices and allowing them to communicate and perform actions autonomously. Motivated by the rising demand for automation and resource efficiency, this project aims to develop a **Water Level Monitoring System** that provides an effective solution to the widespread issue of overflow in storage tanks. The system utilizes an **Arduino Uno microcontroller**, an **HC-SR04 ultrasonic sensor** for measuring the water level, and **a buzzer** that delivers immediate audio notifications when the tank reaches its capacity. This budget-friendly, straightforward project introduces intelligent monitoring into homes, farms, and small businesses, eliminating the necessity for costly Wi-Fi-dependent systems.(Rowing, 2023).

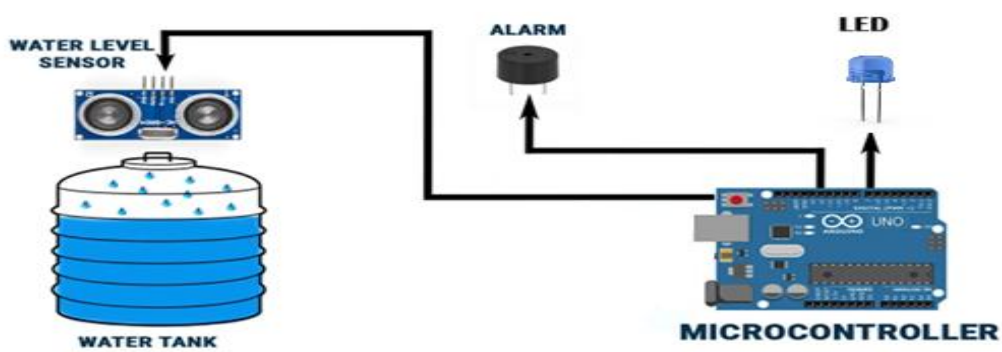



*Figure 1: Overview of water level monitor system*

The goal of this project is to tackle the common problems of water wastage, energy inefficiency, and equipment damage that stem from manually filling water tanks. In conventional scenarios, individuals frequently neglect to switch off water pumps, resulting in overflows and unnecessary resource expenditure. By employing the **ultrasonic sensor** to continuously assess the distance from the sensor to the water surface, the **Arduino** interprets this information and activates a **buzzer** alarm as soon as the designated water level threshold is exceeded. This prompt real-time notification enables users to react swiftly, thereby saving water and safeguarding pumping systems and the system functions entirely offline, which makes it suitable for regions lacking dependable internet connection.

In crafting the **Water Level Monitoring System**, readily available components like Arduino Uno, ultrasonic sensors, buzzers, jumper wires, resistors, and breadboards were utilized to ensure the system remains economical and straightforward to replicate. The primary objective is to enhance water management by making it smarter, more efficient, and accessible to all, thereby encouraging sustainable living practices. By automating a basic yet essential task such as monitoring tank water levels, this project illustrates how IoT concepts can be integrated into everyday life to effectively and affordably address real-world challenges.

## 1.1. Current Scenario

Water scarcity represents an increasing global issue that affects millions of lives daily. Despite advancements in technology, ineffective water management continues to be a prevalent problem, particularly in urban and developing regions. The United Nations World Water Development Report (2023) indicates that approximately 25-30% of the water supplied to cities is wasted due to inadequate monitoring, leaks, and human mistakes. Many households and industries still rely on manual filling of water tanks, and individuals frequently forget to switch off the motor after the tank is filled. This minor lapse results in a considerable amount of water being squandered each day, a waste that the environment can no longer tolerate.

*Figure 2 Water overflow*

In daily life, the act of manually checking water levels is not just inconvenient but also inconsistent. Overflowing tanks can cause property damage, raise water bills, and impose additional strain on energy resources by forcing pumps to operate unnecessarily. Over time, this situation also diminishes the lifespan of water pumps, creating further financial strain for users. In the absence of an intelligent, real-time alert system, managing resources becomes increasingly challenging, particularly for busy families and small enterprises. There is a distinct necessity for straightforward, cost-effective solutions that can automate water monitoring, assist in conserving water, lower expenses, and promote a more sustainable future (Peacock, 2018).

## 1.2. Problem Statement

Managing water storage manually has traditionally been a routine task in households, farms, and small industries. However, it is a method that is frequently unreliable, inconvenient, and susceptible to errors. In the fast pace of daily life, it is simple for individuals to forget to turn off the water pump once the tank is filled, resulting in overflowing tanks, wasted water, property damage, and unnecessarily elevated electricity bills. Over time, the continual overuse of water pumps not only wastes energy but also diminishes the lifespan of the equipment, leading to expensive repairs or replacements. Despite how prevalent these issues are, many individuals still lack access to a straightforward or cost-effective solution to monitor their water levels efficiently.

While advanced IoT-based water management systems are on the market today, they often entail high expenses, technical intricacies, and a significant dependence on internet connectivity rendering them inaccessible for many average users, particularly in rural regions. This fosters a genuine divide between the demand for smart water management and the provision of practical solutions. Therefore, there is a pressing requirement for a cost-efficient, simple-to-install, and offline water level monitoring system that can automatically identify when the tank is full and promptly alert users. A straightforward system like this could assist ordinary families, farmers, and small businesses in conserving water, lowering their electricity consumption, safeguarding their equipment, and contributing significantly to a more sustainable future.

## 1.3. Project Overview

To tackle these challenges, we have created and built a Water Level Monitoring System utilizing Arduino Uno, HC-SR04 Ultrasonic Sensor, and a Buzzer.

**Key Components Used:**

- **Arduino Uno:** Acts as the main controller, processing sensor data and managing the output based on water levels.
- **HC-SR04 Ultrasonic Sensor:** Measures the distance from the top of the tank to the water surface to detect the tank's fullness.
- **Buzzer:** Sounds an alarm when the water reaches the full level, immediately notifying users.
- **Breadboard and Jumper Wires:** Used for easy, temporary connections during circuit prototyping without the need for soldering.
- **Resistors and LEDs: Provide** additional visual indicators to support the buzzer alarm if needed. (Geeksforgeeks, 2023)

This initiative does not necessitate an internet connection, rendering it highly dependable, straightforward, and cost-effective for common usage.

## 1.4. Scope And Usability

The Water Level Monitoring System we have created is intended to be both functional and versatile, making it ideal for various applications:

- **Homes and Apartments:** It assists in overseeing household water tanks, guaranteeing you always possess the appropriate amount of water when required.
- **Farms and Agricultural Fields:** This system simplifies the tracking of irrigation water tanks, aiding farmers in optimizing water use for enhanced crop development.
- **Small Businesses and Factories**: It serves as an effective means to oversee and manage industrial water storage, ensuring operations run smoothly without unnecessary waste.
- **Educational Institutions and Hospitals**: The system advocates sustainable water practices, enabling these vital organizations to manage their water resources responsibly. (Ysi, 2022)

Due to its offline functionality, affordability, and ease of installation, the system is particularly suited for rural areas, developing regions, or smaller users who may lack access to costly smart monitoring technologies. It is a practical, budget-friendly solution that makes water management attainable for all.

## 1.5. Aim and Objectives

**Aim**

The main objective of this project is to create and execute an Arduino-driven Water Level Monitoring System that notifies users through a buzzer when the water tank has attained its maximum level, aiding in the prevention of overflow, conservation of water, saving of electricity, and safeguarding of equipment, all without depending on internet access.

**Objectives:**

- **Specific:** Create a functional prototype that effectively measures water levels utilizing Arduino Uno and an ultrasonic sensor and notifies users with a buzzer.
- **Measurable:** Obtain real-time detection precision within ±2 centimetres and guarantee buzzer activation at the maximum tank level.
- **Achievable:** Utilize cost-effective, readily accessible components and straightforward programming to finish the project by the course deadline.
- **Relevant:** Tackle the actual issue of water wastage, excessive energy usage, and ineffective manual monitoring.

- **Time-bound:** Finalize system design, testing, and ultimate implementation within the designated academic semester timeframe.

## 2. Background

The system operates as a monitoring solution utilizing Internet of Things technologies for measuring water levels in domestic tanks or general-use facilities. The system implements an Arduino Uno microcontroller together with an HC-SR04 ultrasonic sensor to monitor water height in real-time. This system exists for water conservation and overflow prevention and dry-run avoidance by generating real-time alerts and implementing control features. A vital resource such as water can benefit from IoT water monitoring systems which “reduce wastewater and stop water flow” through accurate tank level reporting (Sakinala, 2024).

The system maintains continuous water measurement which allows it to trigger predetermined alerts and responses dependent on the recorded level. The operation of the ultrasonic sensor enables it to send pulse signals to the microcontroller which determines the water height while triggering alarms and indicators upon detecting high or low levels. In real time the system shows low, medium and full tank status through colored LEDs paired with different buzzer tones. When set to control a pump the system enables it to turn the pump on during empty tank situations and turn it off during full tank conditions (Sakinala, 2024).

The primary system components consist of the HC-SR04 ultrasonic sensor together with the Arduino Uno. The HC-SR04 sensor generates ultrasonic sound at 40 kHz frequency which the system uses to determine echo return duration. By dividing the product of sensor sound speed with time by two the Arduino obtains water surface distance measurements. The practical implementation connects the Arduino’s 5V/GND pins to HC-SR04 VCC/GND and uses digital I/O pins for the TRIG and ECHO signals (Electronics, 2025) .

## 2.1. Designing Diagram

The diagram presents an authentic pictorial system design showing where components such as Arduino as well as sensor together with LED and buzzer will be placed on the breadboard. The wire connections between circuit components appear as colored lines which help builders to create the circuit in real-life applications. This design provides useful guidance for constructing the hardware equipment alongside troubleshooting operations during development.

**Block diagram:**

An IoT block diagram describes the organized relationship between physical system elements within an IoT framework. The design connects all devices alongside sensors and communication modules and hardware parts for collecting processing data transmission and action purposes with the help of simple pictorial design. (Evans, 2024).

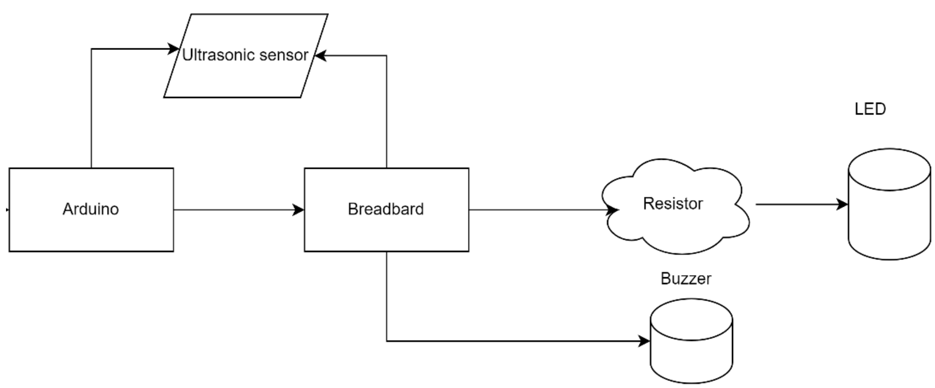



*Figure 3 Block Diagram*

The diagram shows the system's structure by making the Arduino perform as the main processing unit that manages information from the ultrasonic sensor and other system components. Connections through the breadboard allow execution while the diagram marks the incorporation of resistors for circuit regulation. The diagram represents the active relationship between primary hardware components of the system.

**Hardware articture:**

This is a water level monitoring system using Arduino microcontroller's hardware design. Aultrasonicsensor (HC-SR04) is fitted over the lid of the water tank in order toundetect the continuous water depth by determiningthe distance to the water surface. This data is transmitted by the sensor to the Arduino, which is used to process it in order to decide whether there is a low or high concentration of the water level. If a critical point is reached, the system triggers a buzzer alarm as well as an LED indicator to warns the user. This configuration is perfect for automation in water management systems in order to avoid overflow or dry tank status.

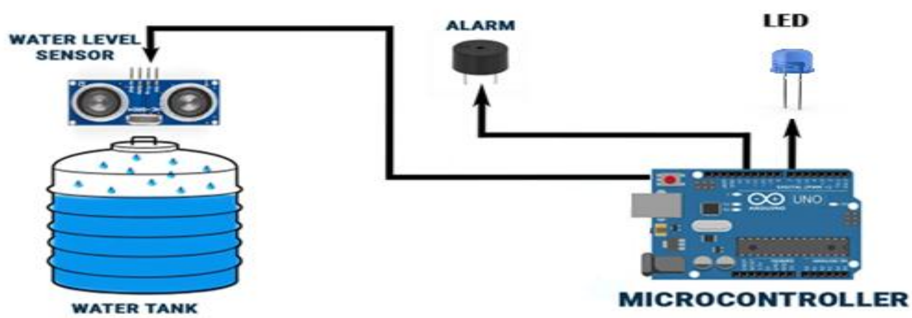



*Figure 4 Hardware articture*

**Circuit diagram**

A circuit diagram presents electrical circuits through two methods using either basic images or standardized industry symbols. The choice of symbols in a diagram relies on the persons who view it. Two major circuit notation forms exist as pictorial diagrams which display parts pictographically as well as schematic diagrams that follow standardized industry symbols. The electrical circuit presents its visual representation using schematic style diagrams which professionals need to understand. The pictorial style circuit diagram serves purposes for a wide audience which lacks technical expertise (Draw, 2025).

The diagram utilizes standard electronic symbols to show how devices connect with each other yet emphasizes only circuit operating principles. The graphical representation simplifies the system by showing fundamental electrical relations through which current and signals move between Arduino and sensors to output devices. A proper understanding of how the circuit functions electrically depends on this important graphical representation.

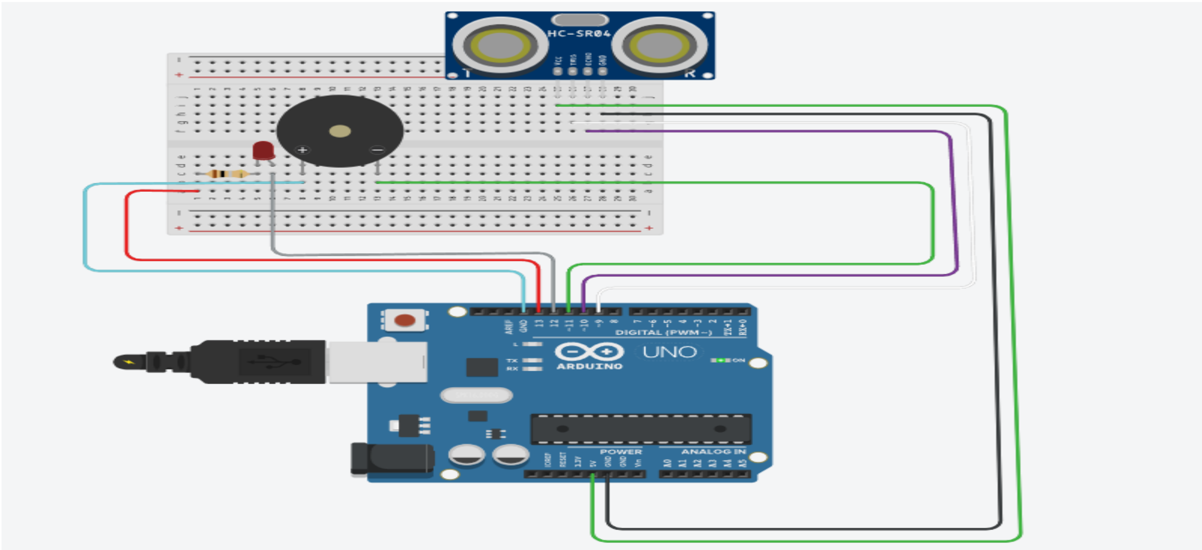



*Figure 5 Circuit diagram*

**Schematic Diagram**

A schematic diagram is a symbol picture of the components in a process, device or whatever with abstract, usually typical symbols and lines. A schematic diagram is a schematic picture of a system that has no more than the primary parts of that system and could include somewhat distorted or introduced details to help in understanding the system. (Hoffman, 2018).

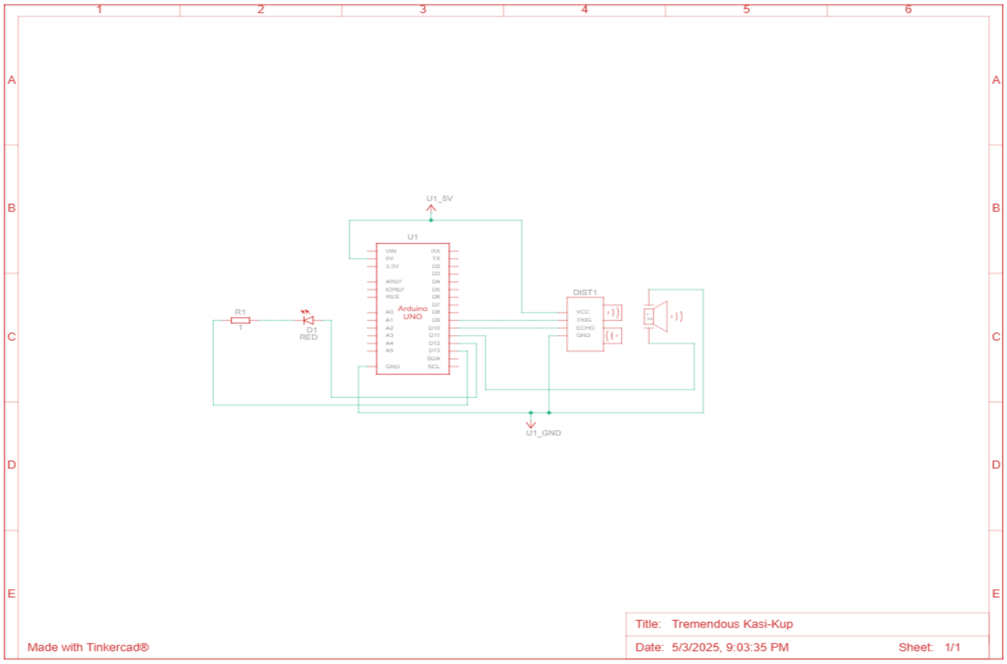



*Figure 6 Schematic Diagram*

**Flow chart:**

Flowcharts show how processes or workflows work through visual depictions of their dividing steps along with their critical decisions. Graphical representation serves as an invaluable resource to explain system operations which makes understanding complex processes accessible to people who do not need to examine code or written documentation. Each graphical representation consists of unique symbols which specifically illustrate different parts in the process. (Fadillah, 2023)

The flowchart shows how the system operates by first initializing itself before it continuously detects distances with the ultrasonic sensor. The system generates alerts through an LED and buzzer following distance measurements which determine critical or safety water levels for timely alert sounds. The automated system interfaces with water level checks in irregular intervalsthroughout detection periods.

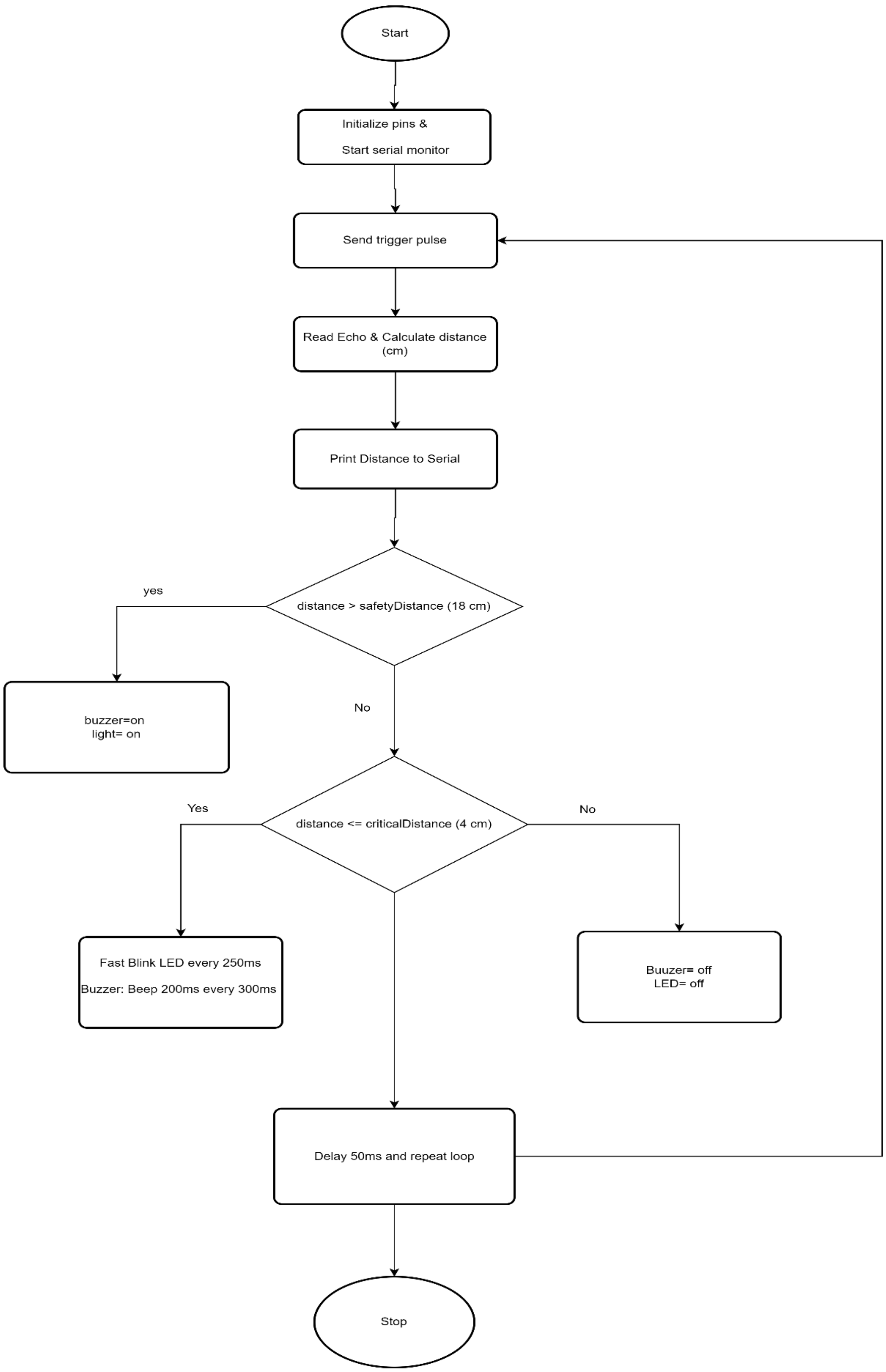



*Figure 7 Whole system flowchart diagram*

## 2.2. Required Components

- **Breadboard:** The construction base of electronics prototyping is known as a breadboard. The ordinary plastic board houses numerous small holes although it extends far beyond its initial appearance. Circuit building and electronics depend heavily on the breadboard for completion. A breadboard system first appeared in 1971 as the replacement for wire-wrap technology which engineers used before that time. Electrical wires needed to be wound multiple times over component leads or socket pins onto insulating boards during this method. The practice required great caution together with experience. The picture showcases how situations would rapidly deteriorate into disarray (Soldered, 2024).
- **Jump wire:** The basic function of jumpers is to operate as small metal closures that can begin or terminate circuit operations. Two or more connection points enable jumpers to regulate electrical circuit boards. Peripherals of a computer like the motherboard benefit from jumper configuration settings. With a motherboards capability to detect intrusions your computer system would become more secure. The jumper includes an on-off configuration option. Each end of jumper wires possesses electrical connector pins that link them to various circuit points. These connectors enable linkage between circuit points through non-soldered processes (Wiltronics, 2022).
- **Buzzer:** The piezo known as buzzer functions as a sound generating hardware component. Through digital interfaces this component enables an audio signal when the connected output reaches the HIGH state. Using an analog pulse-width modulation output will enable the buzzer to produce different tones and effects. The Grove Buzzer works at 3.3V along with 5V and creates sound at a level of 85 decibels. The module offers sound capabilities similar to digital watch button clicks during application feedback (Kit, 2025).
- **Ohm resistor 220:** To achieve current flow resistance the device called Resistor operates in electrical circuits. When adding a resistor in any electrical circuit it causes the voltage to decrease across every other circuit element. A resistor converts part of electrical energy into heat while the process occurs. The relationship between voltage and current and resistance follows Ohm's law formula which equals V=IR. The measured unit for resistance is the Ohm (symbol: Ω). The selection of a resistor only requires choosing a resistance value that matches the requirement approximately in simple circuits. Analyzing precise resistors requires Ohm's law (Davison, 2025).

- **Led bulb:** The wireless switches installed in lights allow users to eliminate the need for direct wiring between fixtures and switches. Internet-connected smart lights and bulbs function as parts of networks because this connection makes cloud-based control and monitoring possible. The management of lights through the web interface and mobile applications represents the main process of control for users. The operating management of lights at both individual and group levels happens by keeping track of external light intensities as well as daily times and room occupancy rates. Depending on the available features users can adjust their smart lights by changing colors and using dimming functions (Daisyenergy, 2024).
- **Arduino uno**: The Arduino.cc organization created the open-source Arduino Uno microcontroller board which operates with Microchip ATmega328P. The board operates through the Microchip ATmega328P microcontroller. The Arduino Uno stands as the leading development board among Arduino products which receives its stock name. With a total size of 2.7 inches by 2.1 inches it constitutes a compact development board. The platform displays near-complete software configuration as well as system hardware support for different applications. The board features digital and analog input/output pins which help users execute small developmental projects and create prototypes but the pin number remains fewer than Arduino mega. Engineering students together with professionals choose this board for its affordable price and applicable features to develop projects (GeeksforGeeks, 2022).
- **Ultrasonic sensor**: The ultrasonic sensor operates as an electronic instrument. The sensor releases ultrasonic waves that it changes into electrical data to measure the distance of a target object. Ultrasonic waves establish faster speeds compared to the human-perceptible sound range. The primary parts of ultrasonic sensors combine a transmitter to produce sounds with piezoelectric crystals and a receiver to receive sound after it reflects from the target (Renkeer, 2024).

# 3. Development

This section of the report explains how the project was developed, step by step. It covers everything from the planning stage to the final completion. It describes what was done during each part of the project, such as researching, designing, building, and testing. It also mentions any problems faced and how they were solved, giving a full picture of how the coursework was carried out.

## 3.1. Planning and Design

The main aim of this coursework was to design and build an IoT-based water tank monitoring and management system. This system was developed to help solve common water-related issues in households, such as tank overflow and water shortage. These problems are faced by many people in their daily lives, and this project is a step toward addressing them in a smart and efficient way.

As with any successful project, proper planning was the first step. It was important to have a clear vision of what the final system should look like and how it would function. At first, the idea was to create a simple system that could only monitor the water level in the tank. However, after some discussion and research, the plan was improved by adding management features. This meant the system would not just monitor the tank but also help control it, for instance, by automatically turning the motor on or off when needed.

The decision to expand the project was made because just monitoring the tank wasn't enough to solve the real problem. By adding management features, the system became much more useful and practical for everyday use.

To build this system, several electronic components and devices were used. These include sensors, microcontrollers, and communication modules, which help the system collect data and take action.

## 3.2. Resource Collection

To build and successfully demonstrate this coursework, several electronic components and tools were needed. Since not all the items were available in one place, they had to be collected from different sources. A good number of these components were provided by the resource department of Islington College, while the some were managed by the group members.

The team worked together to gather all the materials. Each member contributed by helping to search for the required components, either from the college or from outside sources. This teamwork made the process of collecting resources faster and easier.

From the resource department of Islington College, we were able to get the following essential items:

- Arduino Board - the main controller that runs the whole system.
- Jumper Wires - for making electrical connections between components.
- Ultrasonic Sensor - to measure the water level inside the tank.
- Breadboard - to connect and test the components without soldering.
- Ohm Resistor 220 - to limit the current flowing to components like LEDs, protecting them from damage.
- LED Bulb - to visually indicate the water level status, such as low or full tank alert. (MertArduino , 2022)

Besides the items provided by the college, we also collected extra components from outside that is buzzer. Buzzer makes sound to alert the users.

In short, collecting the right components was an important part of the project, and thanks to both the college's support and the efforts of the team, all the required materials were successfully arranged.

## 3.3. Hardware Assembly

This section highlights the specific hardware used in the system and where they were connected.The project focuses on developing a smart water management system using IoT technology to prevent issues like water overflow and shortage. The system works by continuously monitoring water levels and making automatic adjustments to avoid these problems. Several devices are involved in the project, with the Arduino being one of the main components. We have **used LED and buzzer as Actuators** for this project. Actuators are hardware components that acts based on sensor input. (Progressive , 2024)

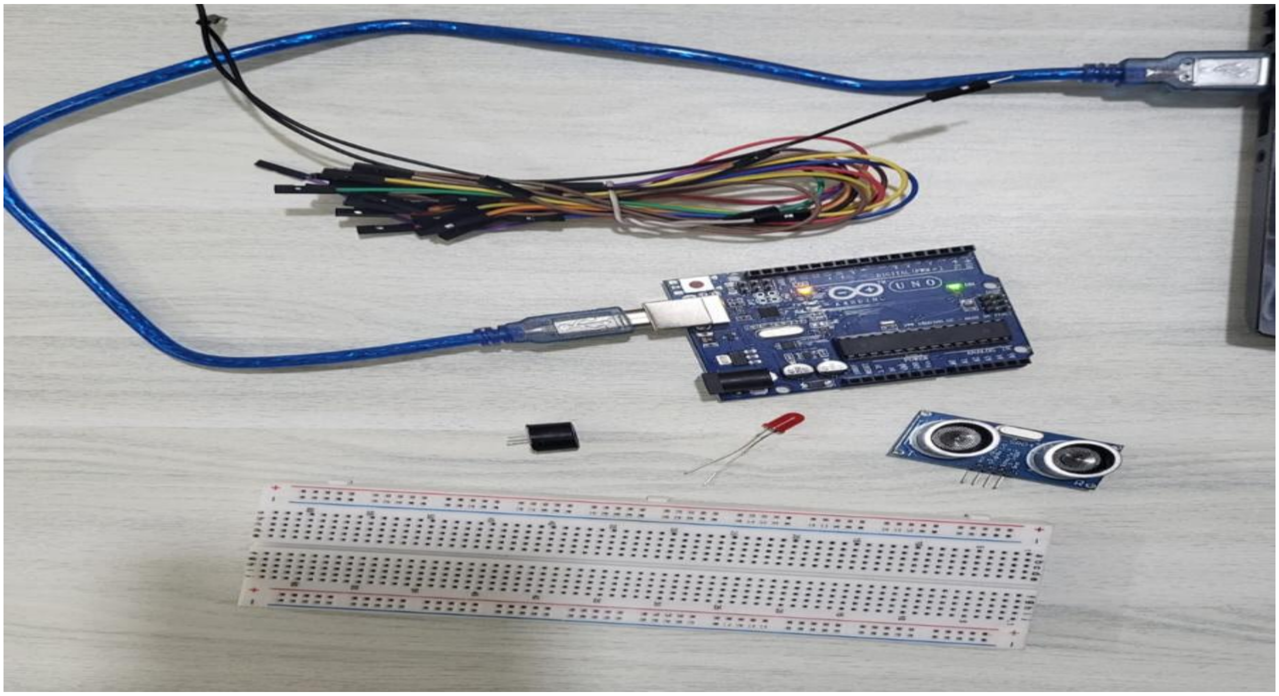
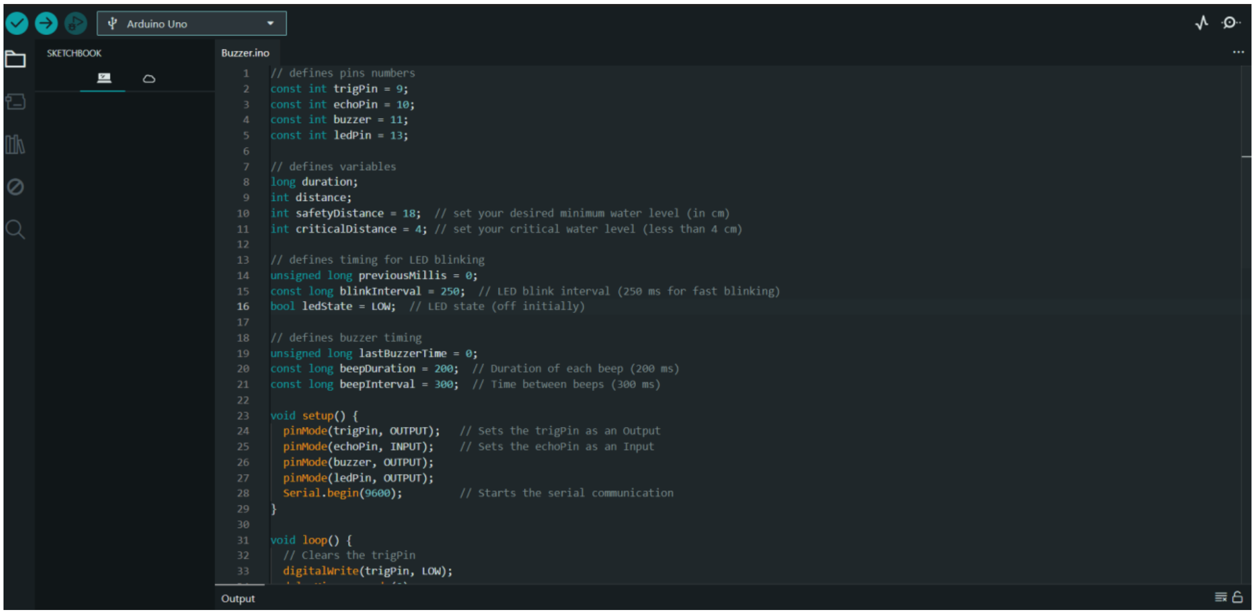


*Figure 8 Hardware collection*

- **Arduino Uno**: The central microcontroller unit. It connects to the ultrasonic sensor via digital pins and receives power from the PC via USB.
- **HC-SR04 Ultrasonic Sensor**: Positioned at the top of the water tank. The *Trig* and *Echo* pins are connected to the Arduino digital pins. It measures the water level by calculating the distance from the sensor to the water surface.
- **Breadboard**: Used to mount the components without soldering. It acts as a platform to connect LEDs, resistors, buzzer, and jumper wires.
- **LED Bulbs**: Connected to digital pins on the Arduino through 220-ohm resistors. They turn ON/OFF based on water levels.
- **Buzzer**: Connected to a digital pin and ground. It produces alerts based on the sensor readings.
- **Jumper Wires and Resistors**: Used to make the circuit stable and to prevent short circuits and overcurrent issues.

Each component plays a specific role—sensing, processing, and alerting—ensuring a fully integrated water level monitoring system.

## 3.4. System Programming




```
// defines pins numbers
const int trigPin = 9;
const int echoPin = 10;
const int buzzer = 11;
const int ledPin = 13;

// defines variables
long duration;
int distance;
int safetyDistance = 18;  // set your desired minimum water level (in cm)
int criticalDistance = 4; // set your critical water level (less than 4 cm)

// defines timing for LED blinking
unsigned long previousMillis = 0;
const long blinkInterval = 250;  // LED blink interval (250 ms for fast blinking)
bool ledState = LOW;  // LED state (off initially)

// defines buzzer timing
unsigned long lastBuzzerTime = 0;
const long beepDuration = 200;  // Duration of each beep (200 ms)
const long beepInterval = 300;  // Time between beeps (300 ms)

void setup() {
  pinMode(trigPin, OUTPUT);   // Sets the trigPin as an Output
  pinMode(echoPin, INPUT);    // Sets the echoPin as an Input
  pinMode(buzzer, OUTPUT);
  pinMode(ledPin, OUTPUT);
  Serial.begin(9600);         // Starts the serial communication
}

void loop() {
  // Clears the trigPin
  digitalWrite(trigPin, LOW);
```

*Figure 9 Code*

The system is programmed using the Arduino IDE. The code:

- Sends a trigger pulse from the Arduino to the HC-SR04 sensor.
- Measures the time it takes for the echo to return, which is then used to calculate the distance to the water.
- Based on the distance, it activates:

**LEDs** for visual indication and **buzzer** for sound alerts (when critically full or empty).

The uploaded program was successfully compiled and uploaded to the Arduino Uno. It runs without errors and controls the hardware components as intended.

## 3.5. Distance Measurement

The breadboard was turned on because all the devices and their connections were set up on it using wires. It was important to power the breadboard so that it could connect and run all the different components properly, and then the code was uploaded to arduino uno.

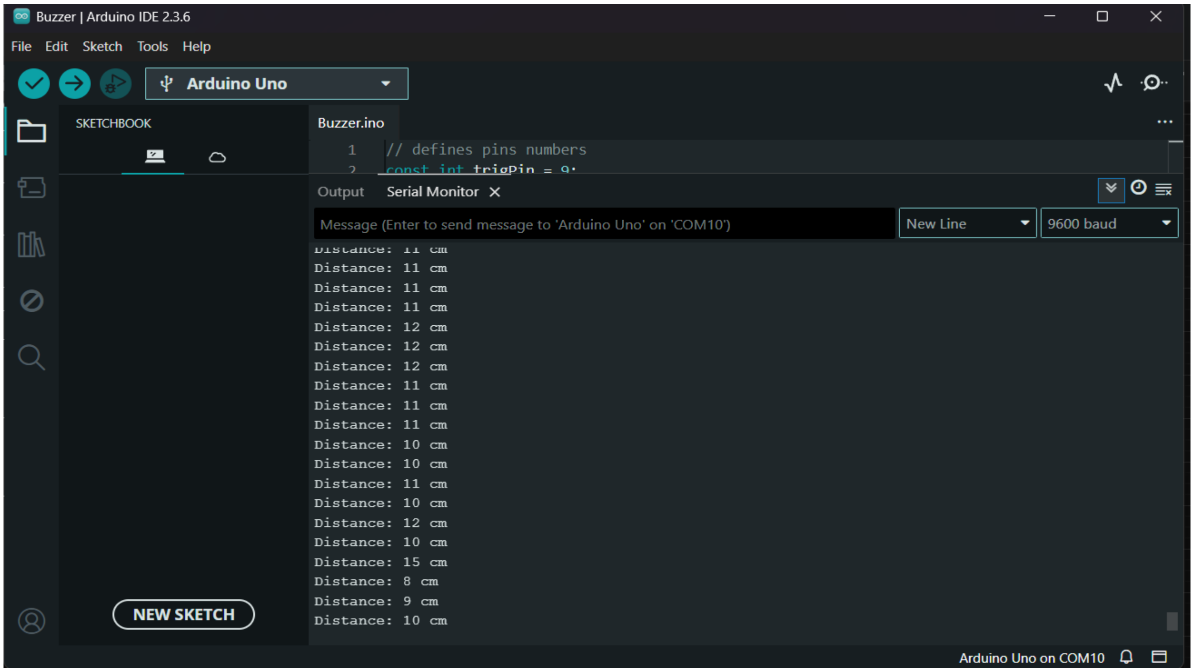



*Figure 10 Measuring distance*

The ultrasonic sensor accurately measures the distance from the top of the tank to the water surface. When tested, the sensor displayed real-time distance values in the serial monitor of the Arduino IDE. The values changed as the water level was adjusted.

This confirms that the sensor is functioning correctly as per the programmed logic and is capable of real-time monitoring.

## 3.6. Testing

This section outlines the testing of each major component separately with evidence.

**Sensor Testing**

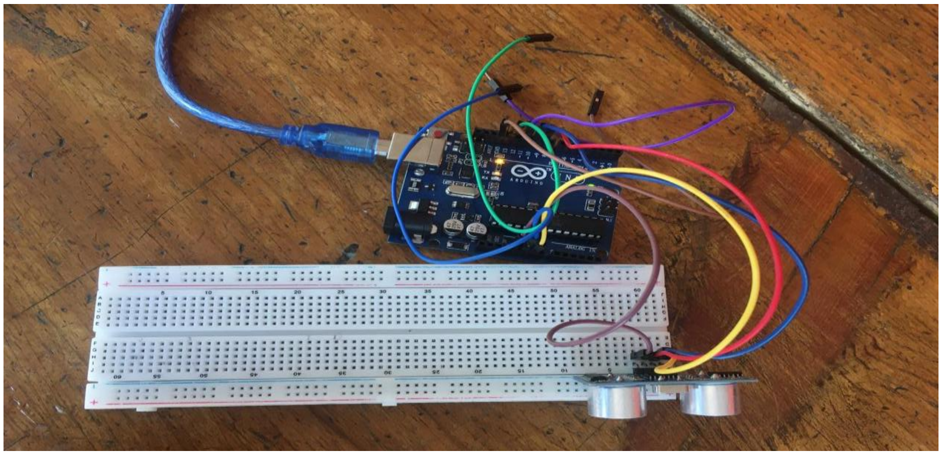


*Figure 11 Sensor connection*

The ultrasonic sensor was tested by placing objects at various distances. The serial monitor displayed accurate readings. It triggered the buzzer and LED appropriately based on preset distance values.

**Result**: Sensor is working perfectly as coded.

**Buzzer Testing**

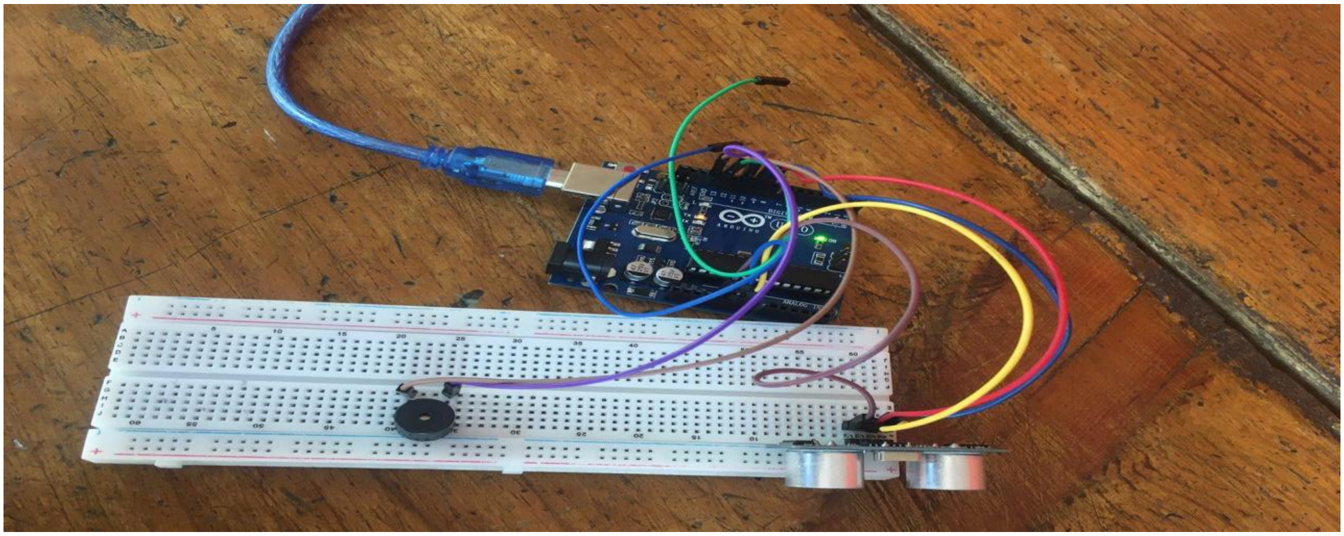


*Figure 12 Buzzer connection*

The buzzer produced audible alerts when the tank was nearly full or nearly empty. It remained silent at normal levels, just as intended in the program.

**Result**: Buzzer behaves exactly according to code logic.

**Led Testing**

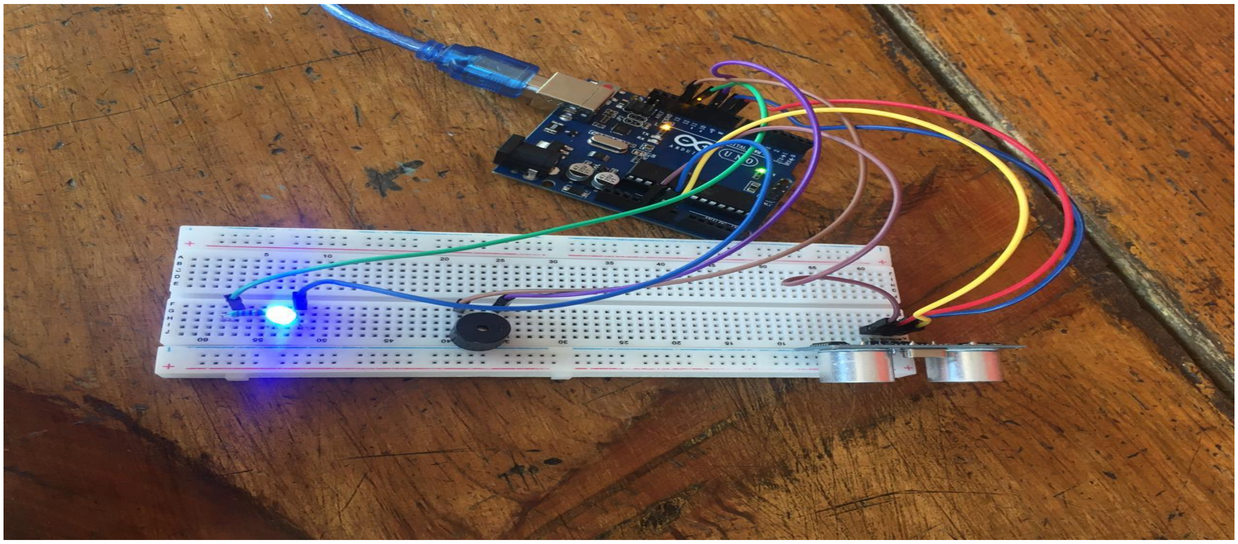


*Figure 13 LED connection*

The LED bulbs were tested for both low-level and high-level alerts. When the water reached critical levels, the LEDs turned ON or blinked as coded.

**Result**: LED indicators function as per code instructions.

When the water reaches top level i.e. critical high level which is 4cm in our project buzzer will beep server time and LED will also blink respectively. When the water level is neither too high nor too low buzzer and LED will be at rest and when the tank is almost empty that is 18cm, the buzzer will beep and LED will be turned on without any delay or pause.

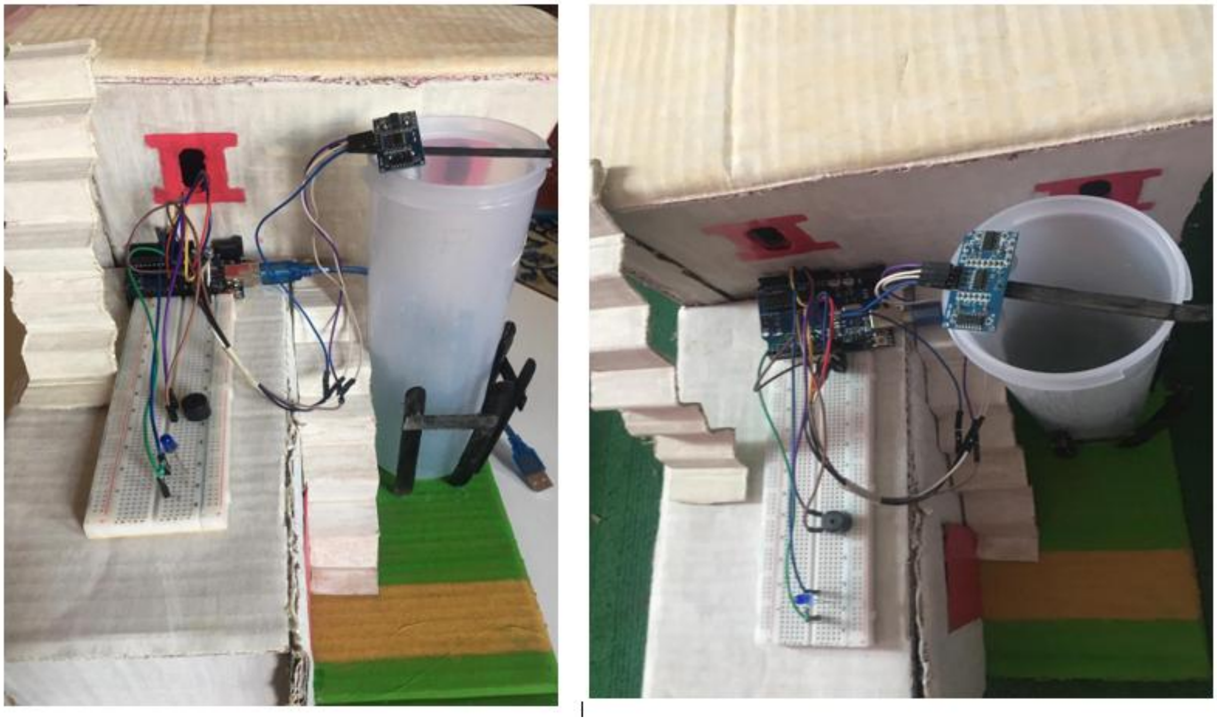


*Figure 14 Project final design*

After designing everything accordingly our prototype was ready and designing was also done,

# 4. Result and Testing:

The IoT-based Water Level Monitoring System fulfilled all project objectives that our team established at its initial planning stage. The project started with the purpose of creating practical real-time water level monitoring hardware which would resolve problems regarding tank overflow together with unexpected shortages. The project reached its final stage by engineering and testing a complete operating prototype.

The mounted ultrasonic sensor accurately read the water tank's levels through a combination of LED indicators and buzzer alerts for users observing minimal or maximum water levels. Through low water level detection, the system caused rapid LED blinking and buzzer warning sounds to notify users. Adequate notifications activated automatically during conditions of a nearly empty tank or complete water tank filling. Users received adequate notifications from the system that enabled them to maintain an appropriate water supply level thus avoiding both supplies running dry and unnecessary waste.

The system testing results indicated good reliability because it delivered consistent quick reactions to water level variations. IoT principles along with automated process demonstrated that basic components could be deployed for constructing an economical solution addressing ordinary household functions.

The implemented system established an intelligent solution that managed water tank usage efficiently and effectively while adopting user convenience features to tackle present-day water supply problems in residential areas.

## 4.1. Testing

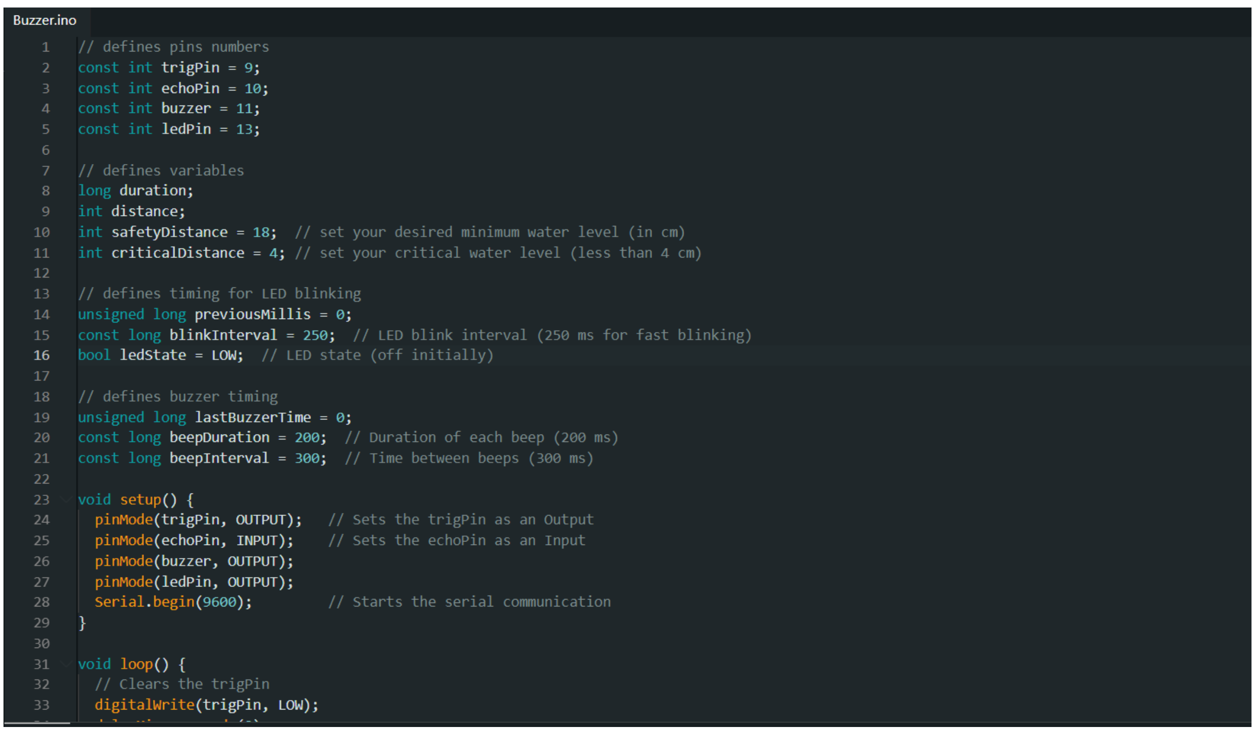
Testing is the process of determining whether or not a system, component or project operates as planned. It implies the process of the running of the hardware or the software to identify any errors, bugs or performance flaws.

Testing makes the system to be reliable, works properly, and adheres to the requirements. It allows problems to be found and addressed early, increases quality and user confidence.

### 4.1.1. Test-1

*Table 1 Test 1*

| | |
|---|---|
| Objective | To show the execution of code. |
| Action | The code was written, verified and uploaded into the Arduino using Arduino IDE software. |
| Expected Result | The code should be compiled and uploaded into Arduino. |
| Actual Result | The code was compiled and ran successfully. |
| Conclusion | The test was successfully. |



Buzzer.ino

```
// defines pins numbers
const int trigPin = 9;
const int echoPin = 10;
const int buzzer = 11;
const int ledPin = 13;

// defines variables
long duration;
int distance;
int safetyDistance = 18;  // set your desired minimum water level (in cm)
int criticalDistance = 4; // set your critical water level (less than 4 cm)

// defines timing for LED blinking
unsigned long previousMillis = 0;
const long blinkInterval = 250;  // LED blink interval (250 ms for fast blinking)
bool ledState = LOW;  // LED state (off initially)

// defines buzzer timing
unsigned long lastBuzzerTime = 0;
const long beepDuration = 200;  // Duration of each beep (200 ms)
const long beepInterval = 300;  // Time between beeps (300 ms)

void setup() {
  pinMode(trigPin, OUTPUT);   // Sets the trigPin as an Output
  pinMode(echoPin, INPUT);    // Sets the echoPin as an Input
  pinMode(buzzer, OUTPUT);
  pinMode(ledPin, OUTPUT);
  Serial.begin(9600);         // Starts the serial communication
}

void loop() {
  // Clears the trigPin
  digitalWrite(trigPin, LOW);
```

*Figure 15 Compiling the code*

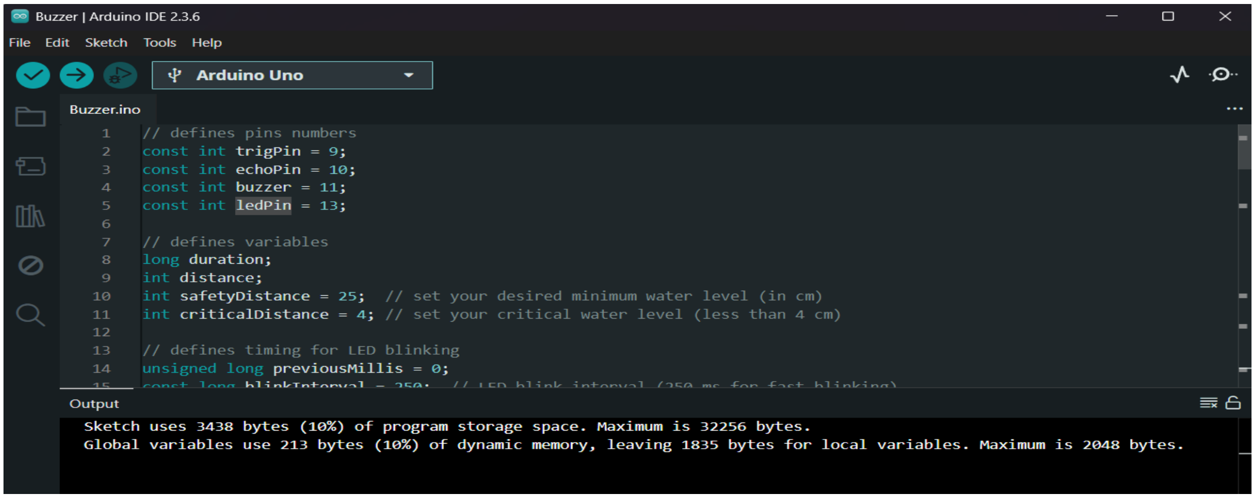



*Figure 16 Uploading the code*

### 4.1.2. Test-2

*Table 2 Test 2*

| | |
|---|---|
| Objective | To check the working of LED bulb and buzzer. |
| Action | The code to turn LED the bulb and buzzer were executed. |
| Expected Result | The LED bulb should be turned on and the buzzer should make noise. |
| Actual Result | The LED bub turned on and the buzzer made noise. |
| Conclusion | The test was successful. |

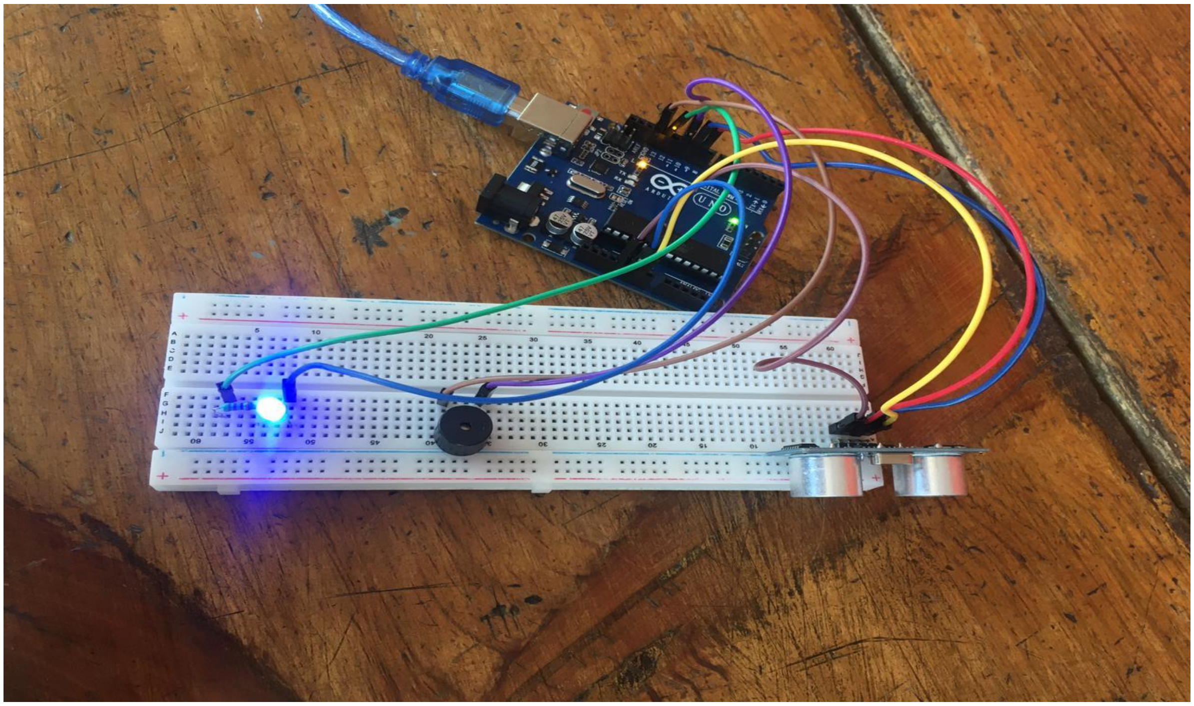


*Figure 17 Working of LED and buzzer*

### 4.1.3. Test-3

*Table 3 Test 3*

| | |
|---|---|
| Objective | To check if ultrasonic sensor is correctly working. |
| Action | An object was placed in front of an ultrasonic sensor. |
| Expected Result | When the object is placed in front the sensor bulb it would blink, the buzzer would make sound with Arduino serial monitor displaying the distance between object and sensor. |
| Actual Result | When the object was placed near the sensor bulb blinked and buzzer made sound with Arduino serial monitor displaying the distance between object and sensor. |
| Conclusion | The test was successful. |

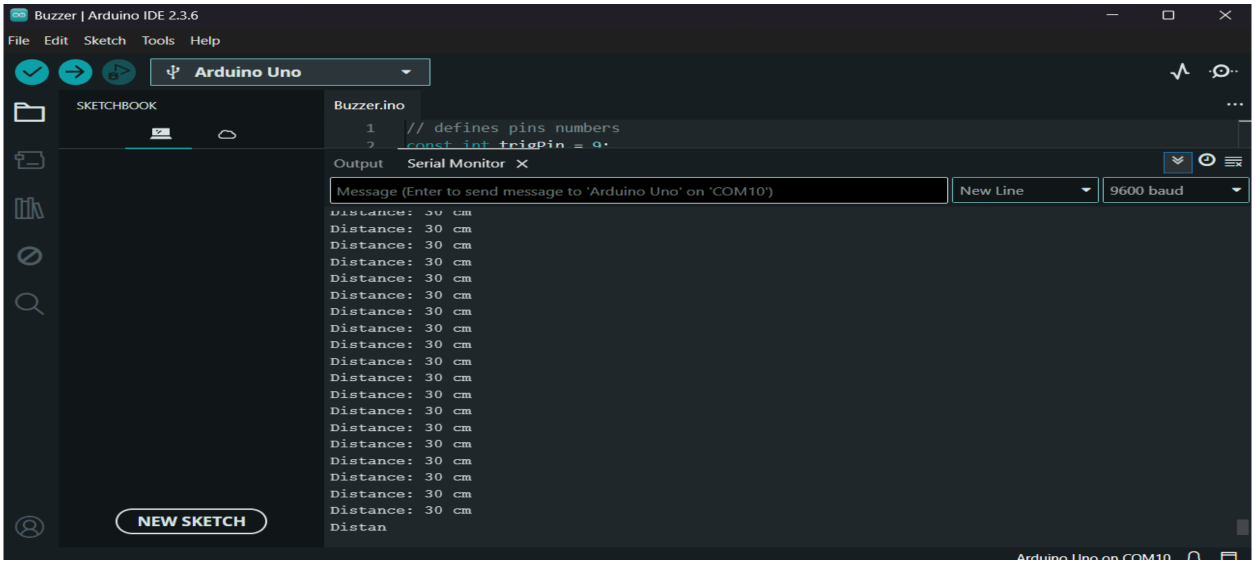



*Figure 18 Ultrasonic sensor distance measurement in serial monitor*

*Figure 19 Sensor usage*

#### 4.1.4. Test 4

*Table 4 Test 4*

| | |
|---|---|
| Objective | To check if lights and buzzzer both work when water tank is empty |
| Action | The water was removed from the bottle. |
| Expected Result | When the bottle is empty the bulb turns on and the buzzer beeps continuously. |
| Actual Result | When the bottle was empty the bulb did not turned on and the buzzer did not beeped continuously. |
| Conclusion | The test was unsuccessfully. |

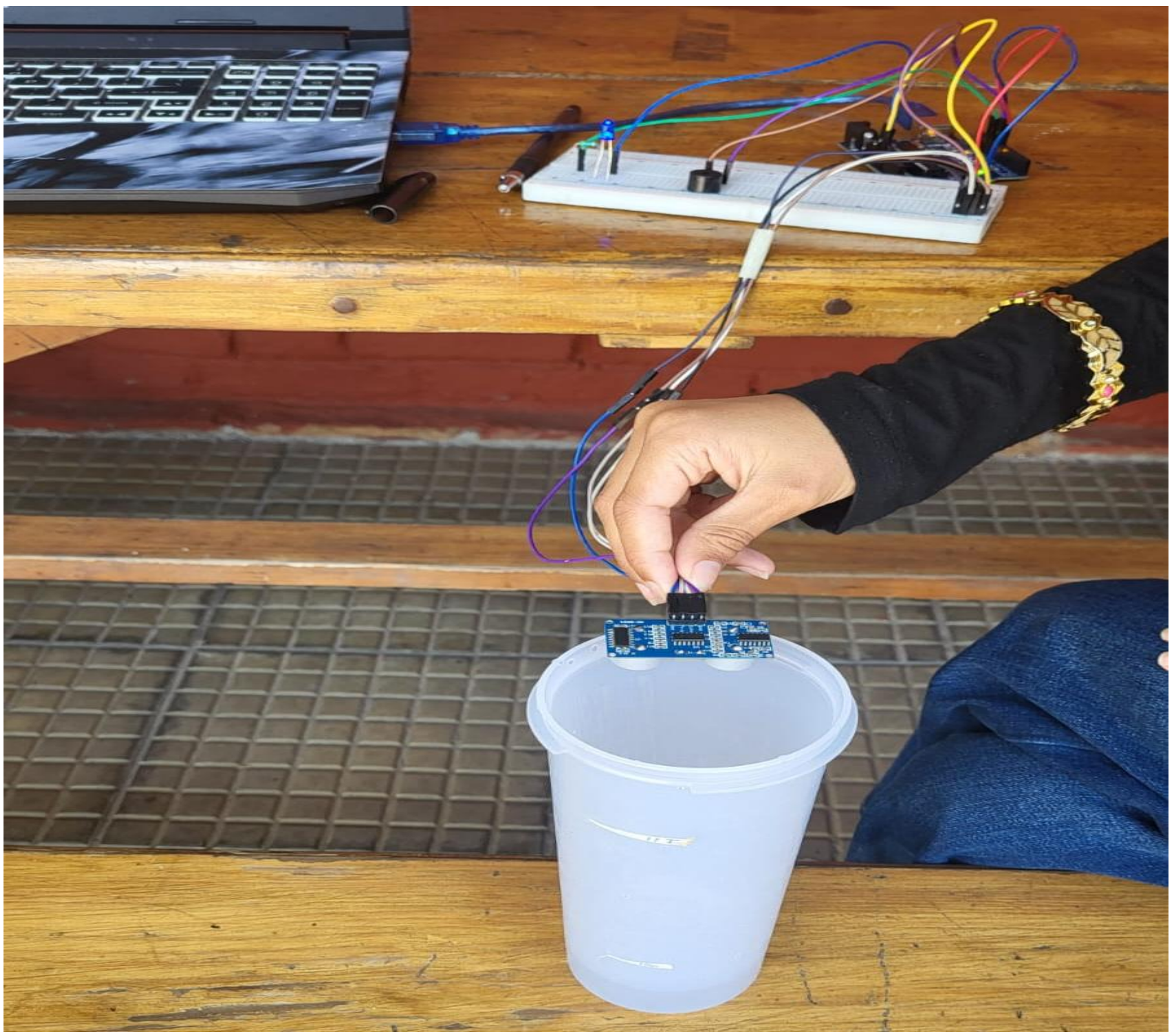


*Figure 20 LED error*

### 4.1.5. Test-5

*Table 5 Test 5*

| | |
|---|---|
| Objective | To check the working of the whole system. |
| Action | The water was filled full, half and again water was removed from the bottle. |
| Expected Result | When the water is full the bulb should blink multiple times, and the buzzer should emit quick beeps, when water level is in between critically low level and critically high bulb and buzzer both should turn off and when the bottle is empty the bulb turns on and the buzzer beeps continuously. |
| Actual Result | When the water was full the bulb blinked multiple times, when water level was in between critically low level and critically high bulb, and buzzer went turn off and the buzzer emitted quick beeps and when the bottle was empty the bulb turned on and the buzzer beeped continuously. |
| Conclusion | The test was successfully. |

*Figure 21 water placed near the sensor*

*Figure 22 Object in medium level distance*

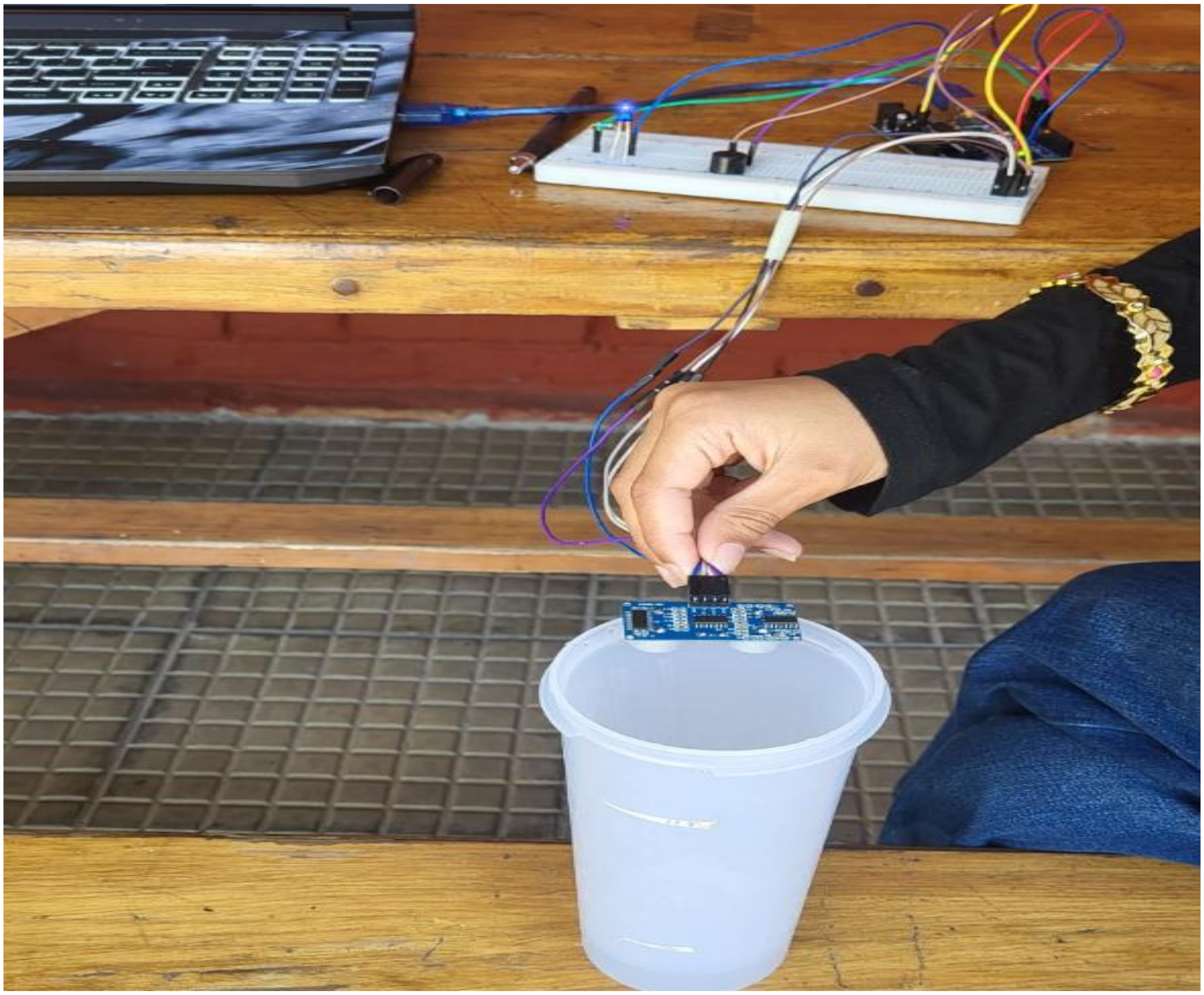


*Figure 23 Water is in more that 18cm distance down*

### 4.1.6. Test-6

*Table 6 Test 6*

| | |
|---|---|
| Objective | To check how the system behaves after restarting due to power loss. |
| Action | Turn Arduino power off and on again. |
| Expected Result | The system should resume normal operation and correct behaviour. |
| Actual Result | The system resumed normal operation and correct behaviour. |
| Conclusion | The test was successful. |

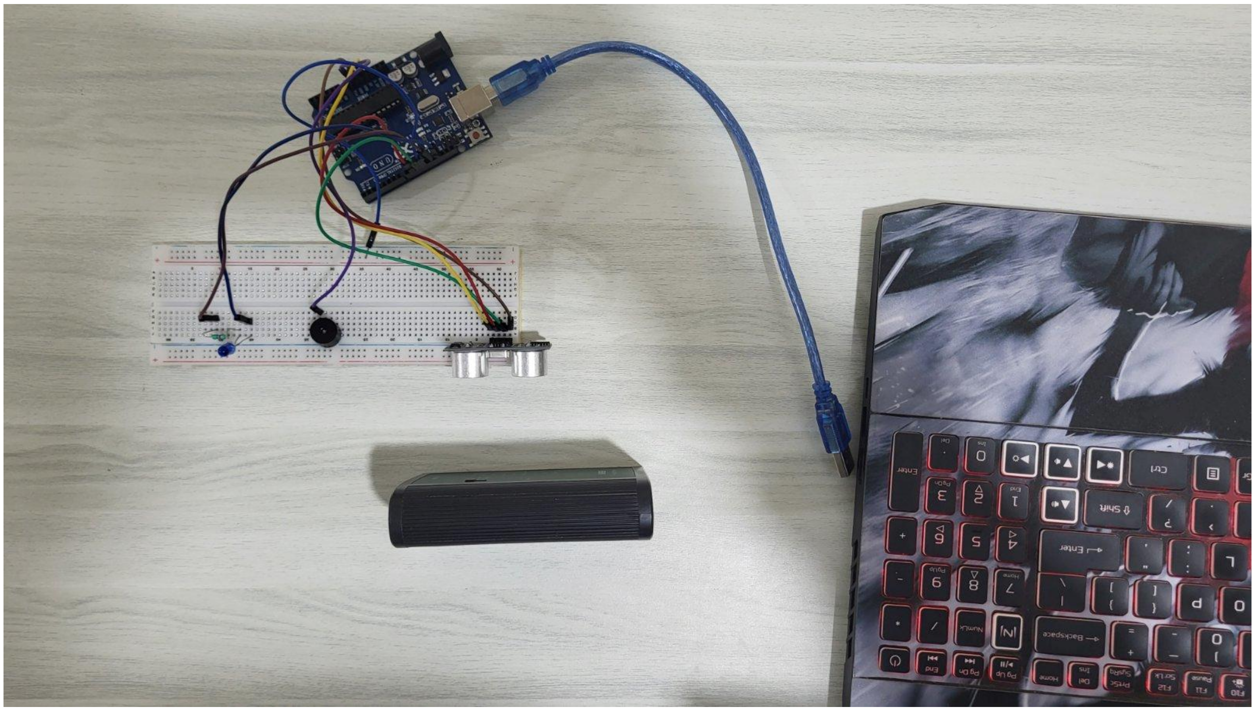


*Figure 24 Power cutoff*

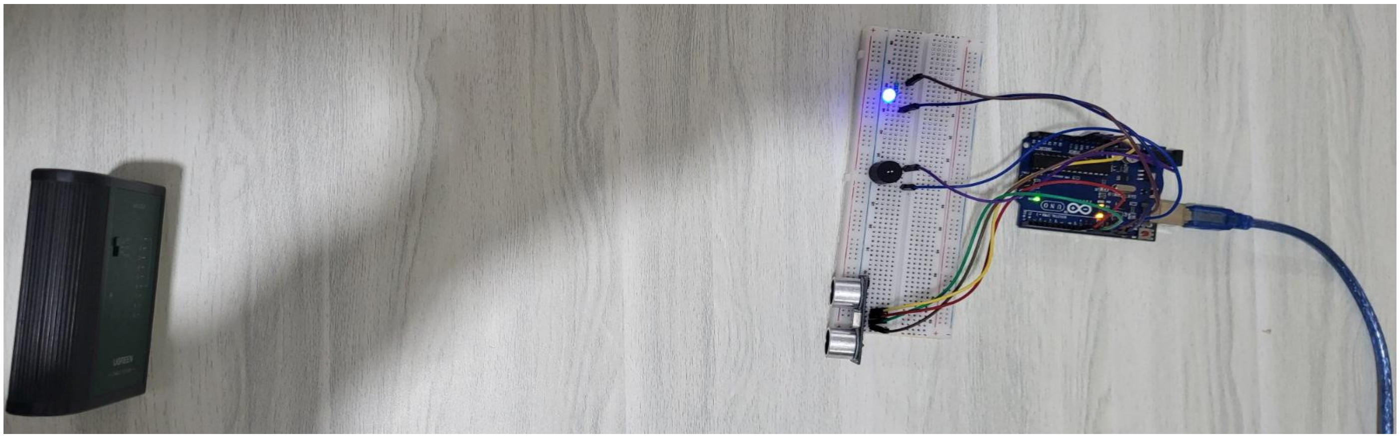


*Figure 25 Reconnected system*

## 5. Future works.

Our current Water Level Monitoring System does everything it should do to detect and monitor water levels accurately and provides helpful alerts. But as with anything good, there's always room for improvement, so in the future we can imagine several improvements and features that would make the system smarter, easier to use, and useful for daily life.

The Water Level Monitoring and Management System demonstrates strong potential to advance in future development. The existing version shows hpw it responds when ever water level increases or decreases. So, it can also be used in **flood detection** process as it can aleart when water level rises.

The system needs improvement through the addition of real-time water level alerts including mobile notification services and simple message alerts for user notifications. An interactive system would enhance management of water consumption while offering better efficiency for water utilization. The system becomes more user-friendly when connected to **Wi-Fi and IoT platforms Blynk or ThinkSpeak** which lets users manage their system through smartphones or any other connected device. (Packt, 2020)

Enhancement of the system design would enable automatic control of liquid flow through the **motor and pump** according to water position changes by incorporating future automation capabilities. The system would become user-friendly and intelligent when further enhanced which would advance water conservation goals while solving common water management requirements.

One of the most exciting upgrades would be to integrate the system with the Internet of Things (IoT). In this project we have used Arduino but using IoT this would allow users to check water levels from anywhere in the world in real time. By connecting the system to cloud services such as Blynk, ThingSpeak, or Firebase and utilizing microcontrollers like **the ESP8266 or ESP32**, we can archive and examine historical water data, issue automated notifications, and even manage segments of the system from a distance. (Techstudycell, 2022)

## 6. Conclusion

Creating the Water Level Detection System has been a fulfilling experience that extends beyond merely putting together components and writing code. This project has shown us the significant potential of straightforward technologies like Arduino, ultrasonic sensors, LCDs, and buzzers when utilized effectively. By integrating these tools, we have created a system that reliably tracks water levels and provides timely notifications, which can simplify and enhance daily water management.

In everyday life, this system can assist households in preventing tank overflows, avert motor damage caused by dry runs, and decrease unnecessary water consumption, all with minimal manual input. For agricultural purposes, it can guarantee timely irrigation, and in areas prone to flooding, it can offer early alerts to safeguard lives and property. These small but meaningful advantages render water management more intelligent, convenient, and less wasteful.

However, this is only the beginning. We perceive tremendous opportunity in enhancing this system's intelligence and flexibility. Features by replacing Arduino with IoT integration devices like NodeMCU ESP38 we can achieve cloud data storage, SMS/call notifications, and even mobile app capabilities could elevate this project into a real-time, remote-controlled solution. Introducing solar panels for off-grid sustainability or even implementing machine learning for flood forecasting could broaden its function from a basic monitor to a lifesaving device. With further enhancements like multi-source monitoring, voice alerts, and durable waterproof enclosures, the system could benefit users in households, farms, and flood-sensitive areas alike.

Aside from domestic and farming requirements, this system has immense prospects of flood detection. By situating ultrasonic sensors next to rivers, drains, or low-lying areas, it can detect levels of the rising water and give early warnings of the impending downpour or monsoon seasons. In the right application, it can help achieve community safety and disaster readiness particularly in flood-sensitive or deprived locations.

On a personal level, this project has enhanced our practical skills, ranging from electronics and programming to innovative problem solving. It has also deepened our understanding of the real-world significance of our work in technology, showing that even minor innovations can create substantial impacts when designed purposefully with people in mind.

While our current version has its constraints, it lays a robust groundwork for future development. With the appropriate upgrades and community participation, this system can evolve into a meaningful, scalable solution and perhaps even motivate others to create their own versions tailored to their specific requirements. This project is not solely about detecting water levels it is about harnessing technology to address everyday challenges, promote sustainable living, and make life a bit easier, smarter, and safer for everyone.

## 7. Bibliography

Daisyenergy, 2024. *DaisyEnergy*. [Online] Available at: https://daisyenergy.ca/introduction-and-benefits-of-iot-in-lighting/
[Accessed 3 May 2025].

Davison, B., 2025. *Resistor*. [Online] Available at: https://bdavison.napier.ac.uk/iot/Notes/components/resistor/
[Accessed 3 May 2025].

Draw, S., 2025. *Smart Draw*. [Online] Available at: https://www.smartdraw.com/circuit-diagram/?srsltid=AfmBOoroyLcaP0nH8SSNgJC5kFKRaD9d-7Mqn9mHGcTAeauc30C26sgo
[Accessed 5 May 2025].

Electronics, S., 2025. *SparkFun Electronics.* [Online] Available at: https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf#:~:text=of%20work%3A%20,high%20output%20IO%20duration%20is
[Accessed 3 May 2025].

Evans, H., 2024. *Velvetech.* [Online] Available at: https://www.velvetech.com/blog/iot-hardware-development/#:~:text=IoT%20hardware%20architecture%20refers%20to,transmit%2C%20and%20act%20on%20data.
[Accessed 5 May 2025].

Fadillah, R., 2023. *cmlabs.* [Online] Available at: https://cmlabs.co/en-id/seo-guidelines/flowchart
[Accessed 5 May 2025].

GeeksforGeeks, 2022. *GeeksforGeeks.* [Online] Available at: https://www.geeksforgeeks.org/points-to-remember-about-arduino-uno-or-mega-in-iot/
[Accessed 3 May 2025].

Geeksforgeeks, 2023. *geeksforgeeks.* [Online] Available at: https://www.geeksforgeeks.org/components-of-iot-and-relation-with-cloud-computing/
[Accessed 4 May 2025].

Hoffman, J., 2018. *packtpub.* [Online] Available at: https://www.packtpub.com/en-PT/product/mastering-arduino-9781788830584/chapter/circuit-diagrams-3/section/fritzing-diagrams-ch03lvl1sec24?srsltid=AfmBOooNdpq1dsygV83QayPDgFVwbV9YRWb5ncloP8UnfErxuPdSxaM1
[Accessed 5 May 2025].

Ismailov, A. S., 2022. *Researchgate.* [Online] Available at: https://www.researchgate.net/publication/359502443_Study_of_arduino_microcontroller_board
[Accessed 2 March 2025].

Kit, S., 2025. *Sensor Kit.* [Online] Available at: https://sensorkit.arduino.cc/sensorkit/module/lessons/lesson/04-the-buzzer
[Accessed 3 May 2025].

MertArduino , 2022. *Instructables.* [Online] Available at: https://www.instructables.com/Arduino-Ultrasonic-Sensor-HC-SR04-With-Buzzer/
[Accessed 2 May 2025].

Packt, 2020. *hands-on-internet-of-things.* [Online] Available at: https://www.packtpub.com/fr-ch/product/hands-on-internet-of-things-with-blynk-9781788995061?srsltid=AfmBOoo1lO2bZJxVO-lhYEdUw9OCxi1JXWxEvQyRgKBK4Iwwm3BL73qe
[Accessed 5 May 2025].

Peacock, C., 2018. *Why Is Water Level Monitoring Important,* China: Aquaread.

Progressive , 2024. *Progressiveautomations.* [Online] Available at: https://www.progressiveautomations.com/pages/actuators [Accessed 2 May 2025].

Renkeer, 2024. *Renkeer.* [Online] Available at: https://www.renkeer.com/what-is-ultrasonic-sensor/#:~:text=An%20ultrasonic%20sensor%20is%20an,sound%20that%20humans%20can%20hear).
[Accessed 3 May 2025].

Rowing, J., 2023. *Water Tank level controller using arduino,* s.l.: Electro Gadget T.

Sakinala, V., 2024. IoT Based Water Level Monitoring & Controlling System Using Arduino Uno ESP8266 Module. *nternational Research Journal of Modernization in Engineering Technology and Science (IRJMETS),* 6(5), p. 5.

Soldered, 2024. *Soldered.* [Online] Available at: https://soldered.com/learn/what-is-a-breadboard-and-how-to-use-it/?srsltid=AfmBOoodinMD2i8H6A4h1_6VXr8SWIDUHmY5-LkL3mTZS_9lqTfZSAnq [Accessed 3 May 2025].

Techstudycell, 2022. *instructables.* [Online] Available at: https://www.instructables.com/IoT-Based-Water-Level-Controller-Using-ESP32-

Wiltronics, 2022. *Wiltronics.* [Online] Available at: https://www.wiltronics.com.au/wiltronics-knowledge-base/what-are-jumper-wires/ [Accessed 3 May 2025].

Yasar, K., 2025. *TechTarget.* [Online] Available at: https://www.techtarget.com/iotagenda/definition/Internet-of-Things-IoT [Accessed 1 May 2025].

Ysi, 2022. *ysi.* [Online] Available at: https://www.ysi.com/parameters/level [Accessed 4 May 2025].

# 8. Appendix

## 8.1. Team contribution:

This project **Water Level Monitoring System** was done effectively with the collaboration between all members of the group and by division of work among members. In my capacity as the group leader, I provided overall coordination and backed up the timely progress, and I actually directly managed essential development stints. The individual contributions are as follows below:

**Tata Gaurav Dahal**: As the leader of the group, I was the one responsible for handling the whole project, the devices, verifying all wires, and writing some crucial parts of the code. I made sure that every step of the development remained intact and collaborated closely with all the members of the team to complete the project as achieved.

**Rejina Phuyal**: Rejina was involved in writing the code and automating the scripts for operations on a system. She also developed the presentation in such a way that she could convey our project's concept and implementation successfully.

**Gaurav Malla**: Gaurav worked on the testing and debugging of hardware parts. He ensured that every device was working as expected and aided to make sure that the whole system had reliable performance during various test cases.

**Shrevika Khadka**: Shrevika was part of designing the layout as well as the visuals for the presentation. She also contributed to the distribution and arrangement of the physical components when it came to hardware setup.

**Piyush Shrestha**: Piyush was the one who took care of documentation and formatting. He collated the written contents, organized the final report, and made sure it was in an acceptable order that was ready for submission.

Each member's effort was vital to the successful completion of the project, and the collaborative environment allowed us to deliver a functional and well-documented prototype on time.

## 8.2. Code

```
// defines pins numbers

const int trigPin = 9;

const int echoPin = 10;

const int buzzer = 11;

const int ledPin = 13;


// defines variables

long duration;

int distance;

int safetyDistance = 18;  // set your desired minimum water level (in cm)

int criticalDistance = 4; // set your critical water level (less than 4 cm)


// defines timing for LED blinking

unsigned long previousMillis = 0;

const long blinkInterval = 250;  // LED blink interval (250 ms for fast blinking)

bool ledState = LOW;  // LED state (off initially)


// defines buzzer timing

unsigned long lastBuzzerTime = 0;

const long beepDuration = 200;  // Duration of each beep (200 ms)

const long beepInterval = 300;  // Time between beeps (300 ms)
```

```
void setup() {
  pinMode(trigPin, OUTPUT);   // Sets the trigPin as an Output
  pinMode(echoPin, INPUT);    // Sets the echoPin as an Input
  pinMode(buzzer, OUTPUT);
  pinMode(ledPin, OUTPUT);
  Serial.begin(9600);         // Starts the serial communication
}

void loop() {
  // Clears the trigPin
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);

  // Sets the trigPin on HIGH state for 10 micro seconds
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  // Reads the echoPin, returns the sound wave travel time in microseconds
  duration = pulseIn(echoPin, HIGH);

  // Calculating the distance (in cm)
```

```
distance = duration * 0.034 / 2;


// Print the distance to Serial Monitor
Serial.print("Distance: ");
Serial.print(distance);
Serial.println(" cm");


// Check water level
if (distance > safetyDistance) {
  // Water level is LOW — alert
  digitalWrite(buzzer, HIGH);
  digitalWrite(ledPin, HIGH);
}
else if (distance <= criticalDistance) {
  // Water level is critically low (less than 4 cm) — fast blink LED and beep buzzer
  unsigned long currentMillis = millis();


  // Blink LED rapidly (every 250 ms)
  if (currentMillis - previousMillis >= blinkInterval) {
    previousMillis = currentMillis;
    ledState = !ledState;  // Toggle the LED state
    digitalWrite(ledPin, ledState);  // Apply the new LED state
  }
```

```
    // Buzzer beeps in bursts ("bep bep bep") with intervals
    if (currentMillis - lastBuzzerTime >= beepInterval) {
      lastBuzzerTime = currentMillis;
      digitalWrite(buzzer, HIGH);  // Turn on buzzer
      delay(beepDuration);         // Buzzer on for 200 ms
      digitalWrite(buzzer, LOW);   // Turn off buzzer
    }
  }
  else {
    // Water level is OK
    digitalWrite(buzzer, LOW);
    digitalWrite(ledPin, LOW);
  }
  delay(50); // short delay before next reading
}
```