# *PROJECT REPORT*

***Submitted in partial fulfillment of the requirements of the***

## IoT Based Smart Water Level Monitoring System using ESP32

***Submitted by***



| | |
|---|---|
| ***Jaya Vigneshwari G*** | ***20231BCA0113*** |
| ***Singh Priya Subodh*** | ***20231BCA0116*** |
| ***Srishti Mishra*** | ***20231BCA0131*** |
| ***Snigdha Mullick*** | ***20231BCA0136*** |



***Under the guidance of***


**Dr. Riyazulla Rahman J,**

**Assistant Professor – Senior Scale, PSIS**

**Department of Bachelors of Computer Applications**

**Presidency School of Information Science**

GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS

**PRESIDENCY UNIVERSITY**

**BENGALURU**


2025-2026

# DEPARTMENT OF BACHELOR OF COMPUTER APPLICATIONS

## PRESIDENCY SCHOOL OF INFORMATION SCIENCE

## PRESIDENCY UNIVERSITY

GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS

## CERTIFICATE

This is to certified that the Project report **"IoT based smart water level monitoring system using ESP32"** being submitted by ***Jaya Vigneshwari G, Singh Priya Subodh, Srishti Mishra, Snigdha Mullick***, in partial fulfillment of requirement for the award of degree of **Bachelor of Computer Applications** is a bonafide work carried out under my supervision.

**Dr. Riyazulla Rahman J,**
Assistant Professor – Senior Scale, PSIS

**Dr. Vetrimani Elangovan**
Assistant Professor –
Senior Scale & HoD, PSIS

| ***GROUP MEMBERS' DETAILS*** | |
|---|---|
| ***Jaya Vigneshwari G*** | ***20231BCA0113*** |
| ***Singh Priya Subodh*** | ***20231BCA0116*** |
| ***Srishti Mishra*** | ***20231BCA0131*** |
| ***Snigdha Mullick*** | ***20231BCA0136*** |

# ABSTRACT

This project presents the design and implementation of a low-cost IoT-based Smart Water Level Monitoring System using the ESP32 microcontroller. The system continuously measures the water level in an overhead tank using an HC-SR04 ultrasonic sensor, computes the level as an absolute value and percentage, and transmits these measurements to a cloud dashboard (Blynk) over Wi-Fi for real-time visualization and alerts. Optionally, the system can control the supply motor through a relay to prevent overflow and conserve energy.

The primary objectives are to:

(1) provide remote live monitoring of tank level,

(2) enable automated motor control based on level thresholds, and

(3) notify users with alerts when tank levels are critical (full / low).

The ESP32 was chosen for its integrated Wi-Fi capability and sufficient GPIO for sensor interfacing. Data sampling is performed every second; server updates and mobile notifications are handled through the Blynk IoT platform. The final implementation demonstrates a practical, scalable, and affordable solution to reduce water waste, save electricity, and avoid motor burnouts. Test results show the system reliably detects level changes and triggers alerts and motor control as intended. Future work includes multi-tank support, SMS fallback notifications, and machine-learning-based prediction of water usage patterns.

# ACKNOWLEDGEMENT

I would like to express my sincere gratitude to the Department of B.C.A for providing the opportunity to undertake this project as a part of the academic curriculum. This project has been a valuable learning experience, allowing me to explore practical applications of IoT and enhance my technical skills.

I extend my heartfelt thanks to my guide, Dr. Riyazulla Rahman J, for their constant support, timely guidance, and encouragement throughout the project. I am also grateful to our Head of the Department for their continuous motivation and for creating an environment that fosters learning and innovation.

I would like to thank my friends and family for their encouragement, patience, and understanding during the course of this work. Finally, I acknowledge the help of online communities and resources, including documentation and tutorials, which played a significant role in shaping my understanding and implementation of the project.

# TABLE OF CONTENTS

## Section – 1

# Introduction

## Problem Statement

In many households and buildings, water tanks are still monitored manually, which often leads to water overflow, wastage of electricity, and even damage to the motor due to human negligence. The lack of an automated system to track and control the water level in real time creates inefficiency and inconvenience. This recurring issue highlights the growing need for smarter and more reliable solutions to manage water resources effectively.

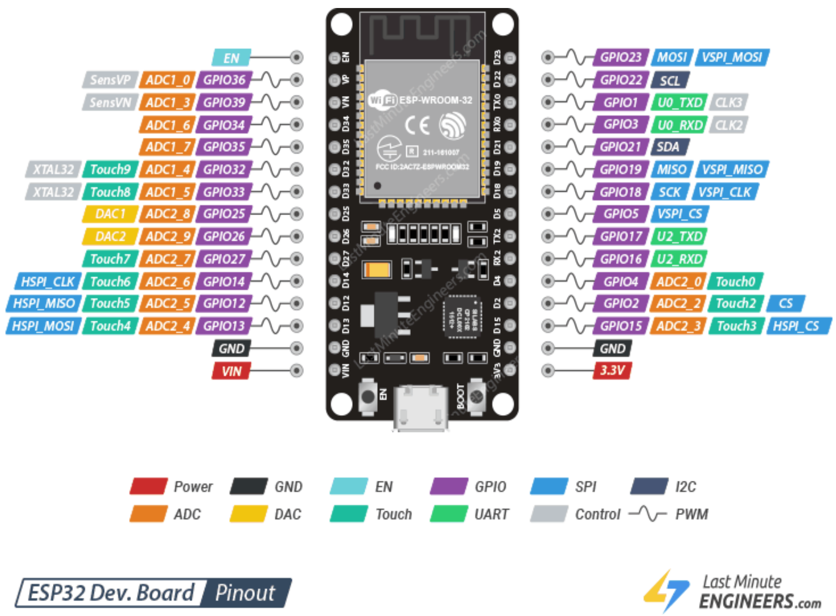
## Problem Solution

- Develop an IoT-based Smart Water Level Monitoring System using ESP32 and ultrasonic sensors.
- Continuously measure and display the water level in real time through a mobile app (like Blynk).
- Send automatic alerts when the tank is full or nearly empty to prevent overflow or shortage.
- Integrate a relay module to control the motor automatically based on water level readings.
- Enable remote monitoring and control via Wi-Fi for user convenience and water conservation.

# Components used:

## 1. Hardware components

### a. ESP 32




The ESP32 is a low-cost, low-power microcontroller with integrated Wi-Fi and Bluetooth, developed by Espressif Systems. It features a dual-core 32-bit processor, a variety of input/output options, and is popular for use in Internet of Things (IoT) applications due to its connectivity and versatility. It can be programmed in languages like C/C++ and MicroPython.

### b. Ultra Sonic Sensor

An ultrasonic sensor is a device that uses high-frequency sound waves to measure distance or detect objects without physical contact. It works by emitting a sound pulse, and then measuring the time it takes for the echo to bounce off an object and return to the sensor.

### c. Jumper wires

Jumper wires are short, insulated electrical conductors with connectors on each end, used to link components in electronic circuits without soldering.

## 2. Software components

### a. Arduino IDE

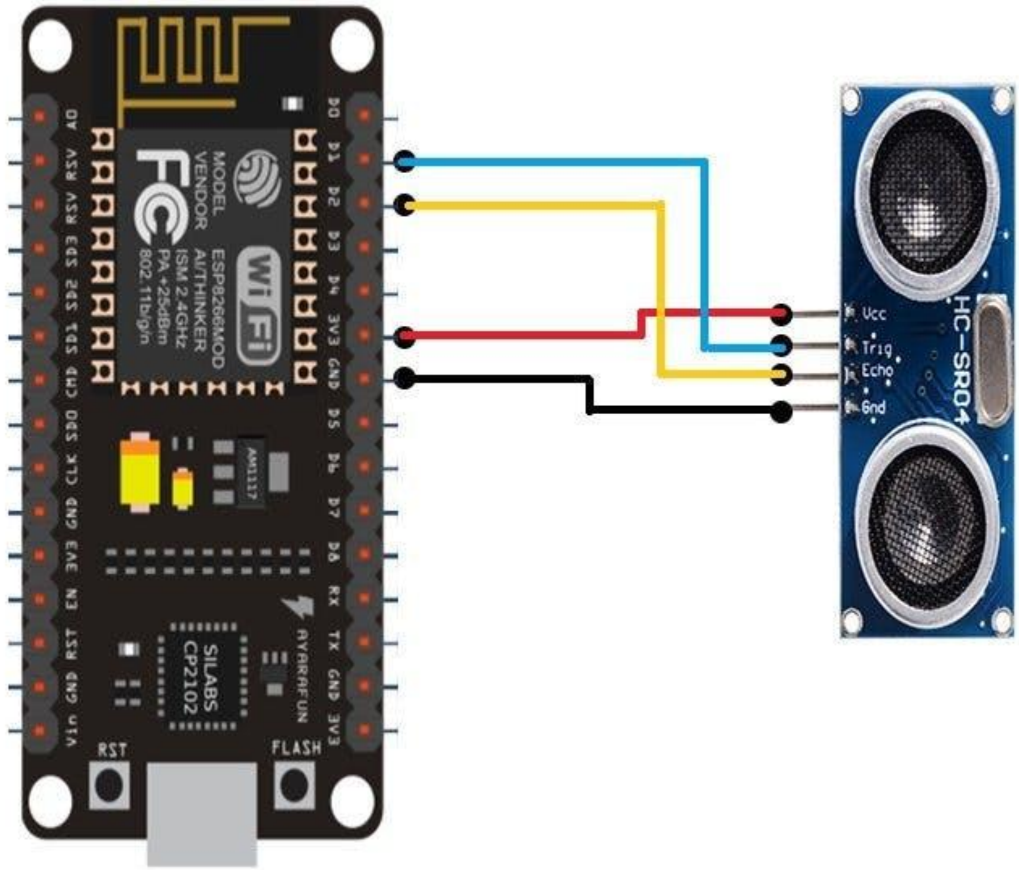
The Arduino IDE is an open-source software application for programming Arduino microcontrollers, allowing users to write and upload code (called "sketches") to create interactive projects. It combines a code editor, compiler, and serial monitor into one environment, making it easy for beginners and advanced users alike to develop hardware projects

**b. Blynk [cloud-based application]**


Blynk is a low-code IoT platform that lets users build and manage connected device applications without writing code. It includes a drag-and-drop app builder for creating customized mobile apps, a web-based console for managing devices, and a secure cloud for device connectivity and data handling. The platform supports a wide range of hardware and connectivity options, enabling users to build everything from personal prototypes to large-scale commercial products.

## Section – 2

# Circuit Diagram




# Connections:

- The Trig pin of the sensor is connected to the D1 pin of the ESP32
- The echo pin is connected to pin D2 of ESP.
- Vcc is connected to the power supply of 5V.
- Gnd is connected to the ground of supply.

## Section – 3

# Source Code & Algorithm

## SOURCE CODE:

```
#define BLYNK_TEMPLATE_ID "Your_Template_ID"
#define BLYNK_TEMPLATE_NAME "Water Level Monitor"
#define BLYNK_AUTH_TOKEN "Your_Blynk_Auth_Token"
#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>

// Wi-Fi Credentials
char ssid[] = "Your_WiFi_Name";
char pass[] = "Your_WiFi_Password";

// Ultrasonic Sensor Pins
#define TRIG_PIN  5
#define ECHO_PIN  18

// Tank Dimensions (in cm)
const int TANK_DEPTH = 100; // Tank height in centimeters
long duration;
```

```
float distance, waterLevel, waterPercent;
BlynkTimer timer;

/***************** Function to Measure Water Level *****************/
void measureWaterLevel() {
  // Send ultrasonic pulse
  digitalWrite(TRIG_PIN, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIG_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);

  // Read echo time
  duration = pulseIn(ECHO_PIN, HIGH);

  // Calculate Distance (cm)
  distance = duration * 0.034 / 2;

  // Calculate water level
  waterLevel = TANK_DEPTH - distance;

  // Calculate percentage
  waterPercent = (waterLevel / TANK_DEPTH) * 100.0;
  if (waterPercent < 0) waterPercent = 0;
  if (waterPercent > 100) waterPercent = 100;
```

```
 // Send data to Blynk
  Blynk.virtualWrite(V0, waterPercent);  // Water Level in %
  Blynk.virtualWrite(V1, waterLevel);    // Water Level in cm

  // Print on Serial Monitor
  Serial.print("Distance: ");
  Serial.print(distance);
  Serial.print(" cm | Water Level: ");
  Serial.print(waterLevel);
  Serial.print(" cm | Tank Filled: ");
  Serial.print(waterPercent);
  Serial.println(" %");
}
/***************** Setup Function *****************/
void setup() {
  Serial.begin(115200);
  pinMode(TRIG_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);

  // Connect to Blynk
  Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);
  timer.setInterval(1000L, measureWaterLevel);
```

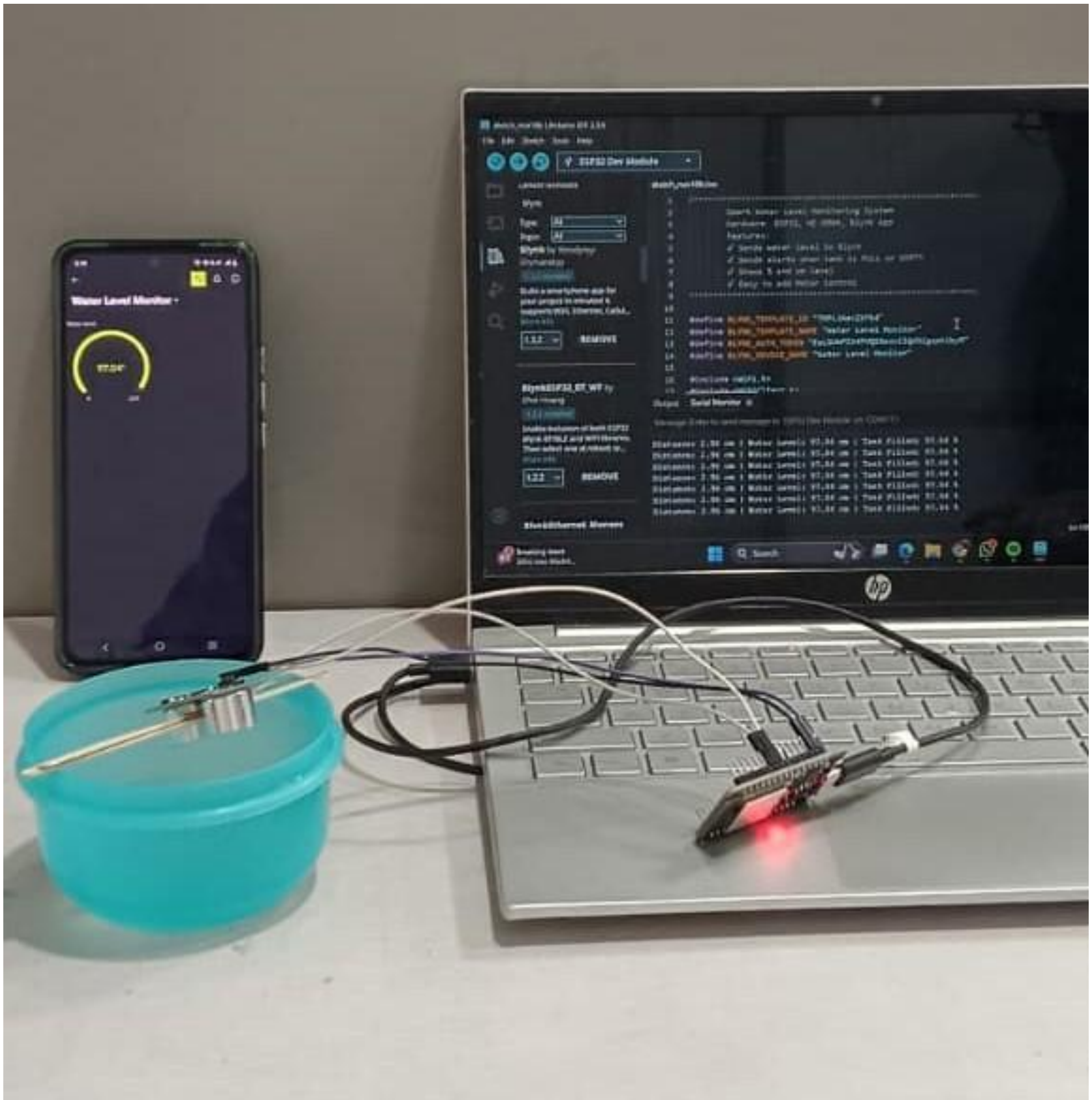
```
}



/**************** Main Loop ****************/
void loop() {
  Blynk.run();
  timer.run();}
```

## ALGORITHM:

1. Initialize serial, Wi-Fi, and Blynk connection. Set TRIG_PIN as OUTPUT and ECHO_PIN as INPUT.
2. Read ultrasonic: send 10µs pulse on TRIG, measure duration = pulseIn(ECHO, HIGH).
3. Compute *distance_cm = duration * 0.034 / 2*.
4. Compute *waterLevel_cm = TANK_DEPTH_CM - distance_cm.*
5. Compute *waterPercent = (waterLevel_cm / TANK_DEPTH_CM) * 100* and clamp between 0–100.
6. Send waterPercent and waterLevel_cm to Blynk virtual pins.
7. If waterPercent >= FULL_THRESHOLD (e.g., 95%) and relay is ON → switch relay OFF and send notification "Tank Full — Motor turned off."
8. If waterPercent <= LOW_THRESHOLD (e.g., 10%) and relay is OFF → (optional) send "Water Low — Please turn on motor" notification.
9. Use a timer (1s interval) to repeat measurements to avoid blocking main loop.
10. Implement debounce/hysteresis around thresholds to avoid relay chattering (e.g., motor off at ≥95%, motor re-enable only at ≤60%).

## Section – 4

# Results & Output




The Smart Water Level Monitoring System was successfully implemented and tested under different tank conditions to validate its performance. The ultrasonic sensor accurately measured the water level, and the ESP32 processed and transmitted the data to the IoT dashboard in real time. The Blynk application correctly displayed the tank percentage, numerical level readings, and triggered notifications when the water reached the predefined thresholds. The relay control also functioned as expected during simulated full- and low-level conditions. The following images show the live dashboard readings, sensor outputs, and system behavior during testing, demonstrating the reliability and effectiveness of the proposed IoT solution.

## Section – 5

# Conclusion

## Limitations:

The system performs adequately for household use. Limitations include sensitivity to ultrasonic reflections (foam, turbulent surfaces), requirement for stable Wi-Fi connectivity for cloud features, and the need for caution while switching high-power motors (use proper relay drivers and isolation). For remote rural deployment, GSM fallback for notifications is recommended.

## Future Scope:

• Add persistent data logging (CSV / cloud DB) for historical trends.
• Integrate SMS or voice call fallback using GSM module for no-Wi-Fi areas.
• Use multiple sensors for multi-tank support or to eliminate false readings.
• Apply ML models to predict water usage and optimize pump scheduling.
• Add battery backup and low-power modes for resilience.

To summarize,

The **Smart Water Level Monitoring System using ESP32** provides a practical and efficient solution for monitoring water levels in real time. By combining the HC-SR04 ultrasonic sensor with the Wi-Fi-enabled ESP32, the system accurately measures the tank level and updates the user through the IoT dashboard. This helps prevent overflow, reduces water wastage, and minimizes the need for manual checking.

The use of IoT technology makes the system accessible and user-friendly, allowing remote monitoring and notifications through the Blynk platform. Throughout the development, important concepts such as sensor calibration, cloud communication, and safe relay control were understood and applied, making the project both educational and functional.

Overall, the project meets its objectives and demonstrates how IoT can simplify daily tasks and improve resource management. With minor enhancements like data logging, GSM alerts, or multi-tank support, the system can be further expanded for broader real-world applications.

## Section – 6

# References


1. Article on "IoT Based Water Level Indicator using Ultrasonic Sensor" by "IoT Circuit Hub" [https://iotcircuithub.com/iot-based-water-level-indicator/]
2. Article on "IoT based Water monitoring system" by "huckster.io" [https://www.hackster.io/yaranaiotguru/iot-based-smart-water-level-monitoring-system-using-esp32-de3de2]
3. Paper published on IEEE by Dr. Safina Parveen, Dr. Syed Muhammed Nabeel Mufasa on "Smart Water Level Monitoring System Using Internet of Things (IoT)" [https://ieeexplore.ieee.org/document/10064661]
4. Espressif Systems. *"ESP32 Technical Reference Manual."* Espressif Official Documentation.
5. Espressif Systems. *"ESP32 Datasheet — Wi-Fi & Bluetooth Microcontroller SoC."* Espressif Developer Resources.
6. Blynk IoT Platform. *"Getting Started with Blynk IoT."* Blynk Documentation.
7. HC-SR04 Ultrasonic Sensor Datasheet. *"Ultrasonic Ranging Module HC-SR04 Specifications."* Elecfreaks / RobotDyn.