

# Water Level Indicator Using Arduino and ESP8266

## Project Overview

The **Water Level Indicator System** helps monitor the water level in a tank and provides real-time updates. This system uses **ultrasonic sensors** or **float sensors** to detect water levels and displays them on an **LCD screen**. The system can also send alerts via **Wi-Fi (ESP8266)** to a cloud platform like **Firebase** or **Blynk**, allowing remote monitoring.

## Objectives

- ✓ **Measure water level in a tank** using sensors.
- ✓ **Display water level** on an LCD screen.
- ✓ **Send real-time data to Firebase or Blynk** for remote monitoring.
- ✓ **Trigger alerts** (buzzer or LED) if the water level is too low or too high.
- ✓ **Automate water pumping system** (optional).

## Components Required

1. **Arduino Uno** – Main controller.
2. **ESP8266 Wi-Fi Module** – Sends data to the cloud.
3. **Ultrasonic Sensor (HC-SR04) or Float Sensor** – Measures water level.
4. **LCD Display (16x2 or OLED 0.96")** – Shows real-time water level.
5. **Buzzer** – Alerts when the water level is low or full.
6. **Relay Module (Optional)** – Controls a water pump automatically.
7. **LED Indicators (Optional)** – Shows different water levels.
8. **Jumper Wires & Breadboard** – For connections.
9. **5V Power Supply** – Powers the system.

## How the System Works

1. The **ultrasonic sensor (HC-SR04)** measures the distance to the water surface.
2. The **Arduino** calculates the **water level** based on the tank height.
3. The **LCD screen** displays the water level percentage.
4. If the water level is **too low or too high**, a **buzzer sounds** as an alert.
5. The **ESP8266** sends data to **Firebase or Blynk**, allowing **remote monitoring**.
6. (Optional) A **relay module** can **turn on/off a water pump** when needed.

---

## Circuit Diagram

### HC-SR04 to Arduino Uno

#### HC-SR04 Pin Arduino Uno Pin

|      |     |
|------|-----|
| VCC  | 5V  |
| GND  | GND |
| TRIG | D7  |
| ECHO | D6  |

### ESP8266 to Arduino Uno

#### ESP8266 Pin Arduino Uno Pin

|     |      |
|-----|------|
| VCC | 3.3V |
| GND | GND  |
| TX  | RX   |
| RX  | TX   |

### Buzzer & LED Indicators (Optional)

#### Component Arduino Pin

|            |     |
|------------|-----|
| Buzzer     | D9  |
| LED (Low)  | D10 |
| LED (Full) | D11 |

---

## Arduino Code for Water Level Indicator

This code reads water level using an **HC-SR04 sensor**, displays the value on an **LCD screen**, and sends data to **Firebase**.

```
#include <ESP8266WiFi.h>
#include <FirebaseESP8266.h>
#include <LiquidCrystal.h>

// Wi-Fi Credentials
const char* ssid = "Your_WiFi_Name";
const char* password = "Your_WiFi_Password";

// Firebase Credentials
#define FIREBASE_HOST "Your_Firebase_Host"
#define FIREBASE_AUTH "Your_Firebase_Auth-Token"
```

```
FirebaseData firebaseData;

// LCD Setup
LiquidCrystal lcd(7, 6, 5, 4, 3, 2);

// Ultrasonic Sensor Pins
#define TRIG 9
#define ECHO 8

// Tank Height in cm
#define TANK_HEIGHT 100

void setup() {
  Serial.begin(9600);
  lcd.begin(16, 2);
  pinMode(TRIG, OUTPUT);
  pinMode(ECHO, INPUT);

  // Wi-Fi Connection
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.print(".");
  }
  Serial.println("Connected to Wi-Fi!");

  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
}

void loop() {
  // Measure water level
  digitalWrite(TRIG, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIG, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG, LOW);

  long duration = pulseIn(ECHO, HIGH);
  int distance = duration * 0.034 / 2;
  int waterLevel = TANK_HEIGHT - distance; // Convert distance to water
  level percentage

  if (waterLevel < 0) waterLevel = 0; // Prevent negative values

  Serial.print("Water Level: ");
  Serial.print(waterLevel);
  Serial.println(" cm");

  // Display on LCD
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Water Level:");
  lcd.setCursor(0, 1);
  lcd.print(waterLevel);
  lcd.print(" cm");
}
```

```
// Send data to Firebase
Firebase.setInt(firebaseData, "/WaterLevel", waterLevel);

// Low water level alert
if (waterLevel < 20) {
    digitalWrite(10, HIGH); // Turn on low water LED
    tone(9, 1000);          // Sound buzzer
    delay(1000);
    noTone(9);
    digitalWrite(10, LOW);
}

// High water level alert
if (waterLevel > 80) {
    digitalWrite(11, HIGH); // Turn on full tank LED
} else {
    digitalWrite(11, LOW);
}

delay(5000); // Wait 5 seconds before next reading
}
```

---

## How to Use the Water Level Indicator

1. **Upload the code** to Arduino using Arduino IDE.
2. **Connect to Wi-Fi** and check the serial monitor.
3. **Monitor the water level** on the **LCD display**.
4. **Check Firebase** to view water levels remotely.
5. **Listen for buzzer alerts** when water is too low or full.
6. (Optional) **Automate a water pump** using a relay.

---

## Features & Benefits

- ✓ **Real-time water level monitoring** – Prevents tank overflows and shortages.
- ✓ **Remote monitoring** – View water level from anywhere using Firebase.
- ✓ **Alert system** – Buzzer and LED notify users when water is low or full.
- ✓ **Automated pump control** – Can turn on/off water pumps.
- ✓ **Low-cost & efficient** – Uses affordable components.

---

## Future Enhancements

- Mobile App Integration** – Create an app for monitoring.
- Solar-powered system** – Use solar panels for power efficiency.

**Multiple tank monitoring** – Connect more sensors for multiple tanks.

**Water quality monitoring** – Add **pH** and **TDS sensors** for water quality analysis.



**BINARY BRAINS**