Smart Fluid Monitoring System Using IoT and Multi-Sensor Integration for Real-Time Analysis

Submitted by

BHUVANESHWARI K 2116230701055

DEEPA S 2116230701065

In partial fulfillment for the award of the degree

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

RAJALAKSHMI ENGINEERING COLLEGE

ANNA UNIVERSITY: CHENNAI 600 025

MAY 2025

BONAFIDE CERTIFICATE

Certified that this project **“Smart Fluid Monitoring System Using IoT and Multi-Sensor Integration for Real-Time Analysis**” is the bonafide work of “**BHUVANESHWARI K (2116230701055) and DEEPA S (2116230701065)”** who carried out the project work under my supervision.

SIGNATURE
Dr.N.Duraimurugan, M.Tech., Ph.D.

Associate Professor,

Computer Science & Engineering

Rajalakshmi Engineering College (Autonomous)

Thandalam, Chennai -602105.

Submitted for the **ANNA UNIVERSITY** practical examination Mini-Project work viva voice held on___________________

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

Initially we thank the Almighty for being with us through every walk of our life and showering his blessings through the endeavor to put forth this report. Our sincere thanks to our Chairman **Mr. S.MEGANATHAN, B.E, F.I.E.,** our Vice Chairman **Mr. ABHAY SHANKAR MEGANATHAN, B.E., M.S.,** and our respected Chairperson **Dr. (Mrs.) THANGAM MEGANATHAN, Ph.D.,** for providing us with the requisite infrastructure and sincere endeavoring in educating us in their premier institution.

Our sincere thanks to **Dr. S.N. MURUGESAN, M.E., Ph.D.,** our beloved Principal for his kind support and facilities provided to complete our work in time.

We express our sincere thanks to **Dr. P. KUMAR, M.E., Ph.D.,** Professor and Head of the Department of Computer Science and Engineering for his guidance and encouragement throughout the project work.

We also extend our sincere and hearty thanks to our Internal Guide **Dr.N.Duraimurugan, M.Tech., Ph.D.** Associate Professor, Department of Computer Science and Engineering for his valuable guidance and motivation during the completion of this project. Our sincere thanks to our family members, friends and other staff members of information technology.

BHUVANESHWARI K 2116230701055

DEEPA S 2116230701065

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

This project explores the critical role of fluid level monitoring in various industries and examines the transformative potential of IoT technology in optimizing fluid management. By integrating IoT into fluid level monitoring systems, real-time data collection and intervention capabilities are enhanced, leading to better resource management, reduced waste, and improved operational efficiency. This system encompasses containers or tanks where fluid levels are monitored using IoT-enabled sensors like ultrasonic, float, and water leakage sensors. The data collected by these sensors is processed by an ESP32 microcontroller and sent to a remote interface for real-time monitoring. In the event of an overflow, low fluid levels, or leakage, the system triggers alerts to notify concerned parties for timely action, thereby preventing potential damage or water wastage. The system ensures the optimal maintenance of fluid levels, contributing to sustainability and resource conservation. Further development and integration of advanced technologies can expand the applicability and efficiency of IoT-based fluid level monitoring solutions across industries.

CHAPTER 1
INTRODUCTION

1.1 INTRODUCTION
In the dynamic world of resource management and industrial processes, maintaining optimal fluid levels is crucial for ensuring operational efficiency and safety. Deviations in fluid levels can lead to system failures, resource wastage, and increased operational costs. The integration of advanced technologies becomes essential to meet the increasing demand for accurate, real-time monitoring and efficient fluid management.

The advent of Internet of Things (IoT) technology has revolutionized fluid level monitoring systems, offering real-time insights and actionable intelligence to stakeholders across various industries. By leveraging IoT-enabled sensors and connectivity, industries can proactively monitor fluid levels, detect deviations, and trigger timely interventions to prevent overflows, leakage, or insufficient fluid supply. This leads to enhanced operational efficiency, resource conservation, and minimized downtime.

Through an in-depth exploration of existing research, industry practices, and technological advancements, we highlight the significant role of IoT-based fluid level monitoring systems in optimizing fluid management, ensuring sustainability, and improving system reliability

1.2 SCOPE OF THE WORK
This project aims to develop an advanced fluid level monitoring system for industrial applications, focusing on real-time monitoring and proactive management of fluid levels. The system will collect real-time data from various sensors, including ultrasonic sensors, float sensors, and water leakage sensors, to detect and monitor fluid levels accurately. It will analyze the data to identify

deviations from acceptable thresholds, such as overflows, low levels, or leaks, and trigger immediate alerts to the concerned authorities for prompt action.

1.3 PROBLEM STATEMENT

Limited research has been conducted on the causes and consequences of improper fluid management in industrial systems, yet fluid-related issues such as overflows, shortages, and leakage have significant economic and operational impacts. The inability to accurately monitor fluid levels and respond in real-time often results in resource wastage, operational inefficiencies, and system failures, especially in industries that rely heavily on precise fluid management. The lack of automated, real-time monitoring systems further exacerbates these problems. The proposed solution is a Smart Fluid Level Container Monitoring System that uses IoT-enabled sensors to continuously track fluid levels, detect deviations, and trigger immediate alerts to prevent overflows, leaks, or insufficient fluid levels, thus ensuring efficient resource use and minimizing operational risks.

1.4 AIM AND OBJECTIVES OF THE PROJECT

This project aims to create a Smart Fluid Level Container Monitoring System with a real-time web dashboard to manage fluid levels across various industrial containers. The objectives are to develop the necessary hardware and software for real-time data collection, analysis, and fluid level monitoring. Additionally, the project includes creating a user-friendly web interface for displaying fluid levels, as well as setting up automated alerts to notify administrators of any deviations such as overflows or shortages. This will ensure efficient resource management and prevent potential issues in industrial fluid storage.

CHAPTER 2

SYSTEM SPECIFICATIONS

2.1 IOT DEVICES

1. ESP32
2. Float Sensor
3. UltraSonic Sensor
4. Water Leakage Sensor
5. Buzzer

2.2 SYSTEM HARDWARE SPECIFICATIONS

PROCESSOR	Intel i5 11^{th} Gen
MEMORY SIZE	8 GB (Minimum)
HDD	40 GB (Minimum)

2.3 SOFTWARE SPECIFICATIONS

Operating System	Windows 11
Front – End	Html,Css,JS
Back – End	Firebase(Real time Data), Express and Node JS
Browser	Google Chrome
IDE	Visual Studio Code,Aurdino IDE

CHAPTER 3
SYSTEM DESIGN

3.1 ARCHITECTURE DIAGRAM

An architecture diagram is a graphical representation of a set of concepts, that are part of an architecture, including their principles, elements and components

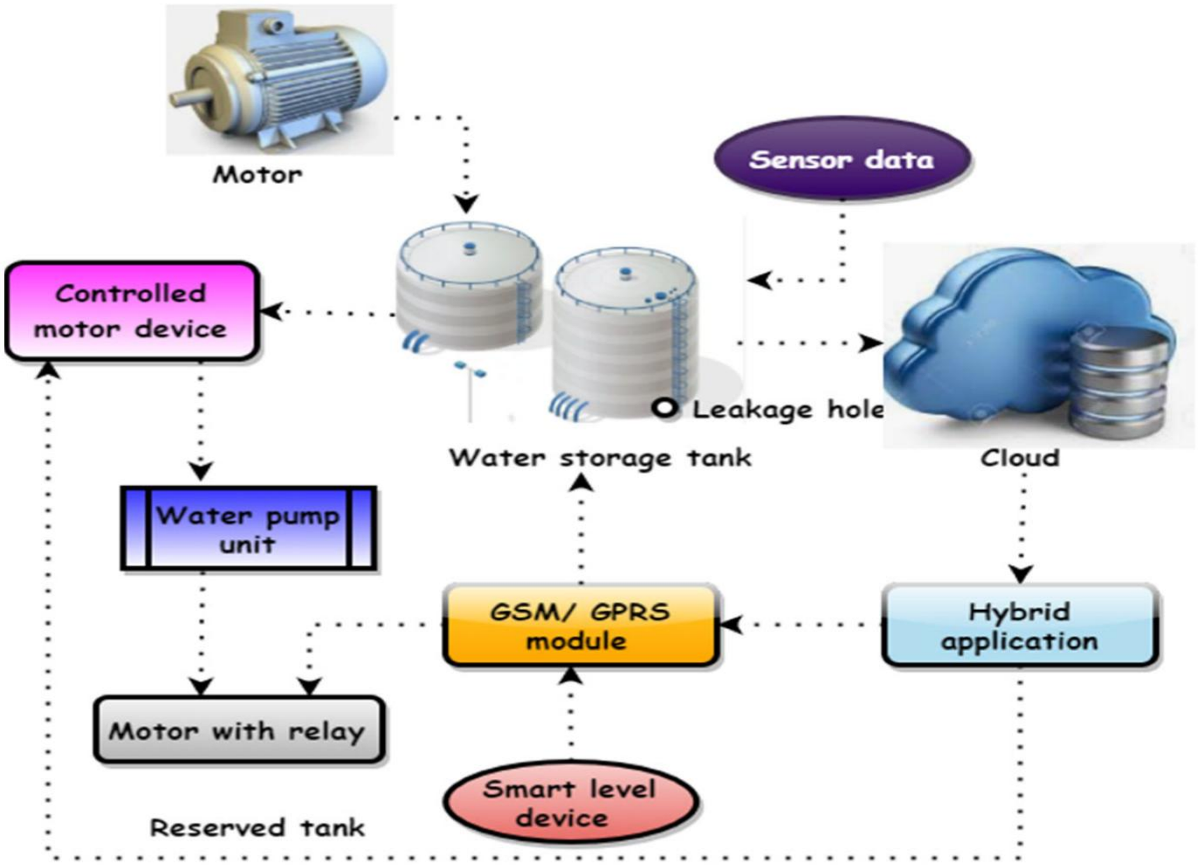

Figure 3.1 Architecture Diagram

From the above Figure 3.1, the architecture of the system is well understood.

3.2 USE CASE DIAGRAM

A use case is a list of actions or event steps typically defining the interactions between a role (known in the Unified Modelling Language as an actor) and a system to achieve a goal. The actor can be a human or other external system.

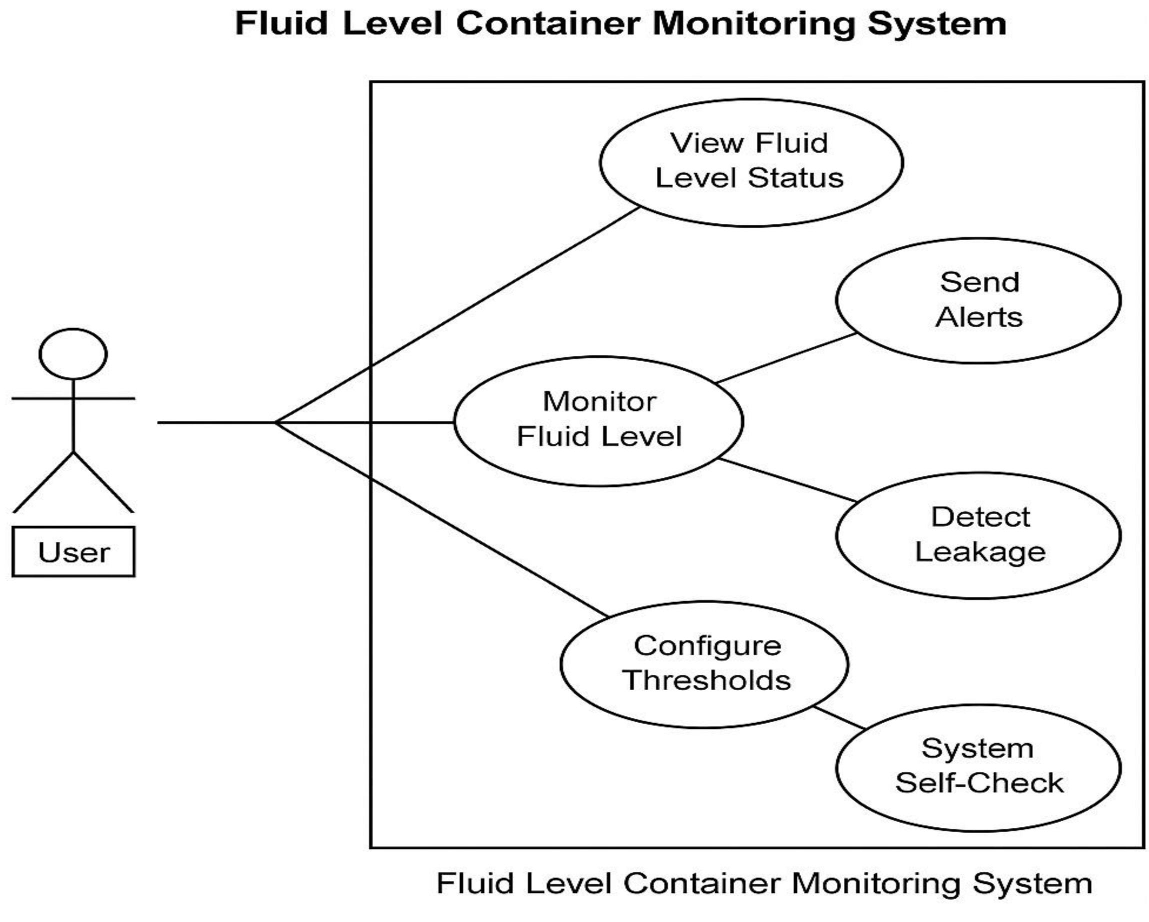

Figure 3.2 Use case diagram

From the above figure 3.2, the interactions between a role in the system is shown

3.3 ACTIVITY DIAGRAM

An activity in Unified Modelling Language (UML) is a major task that must take place in order to fulfill an operation contract. Activities can be represented inactivity diagrams. An activity can represent: The invocation of an operation. A step in a business process.

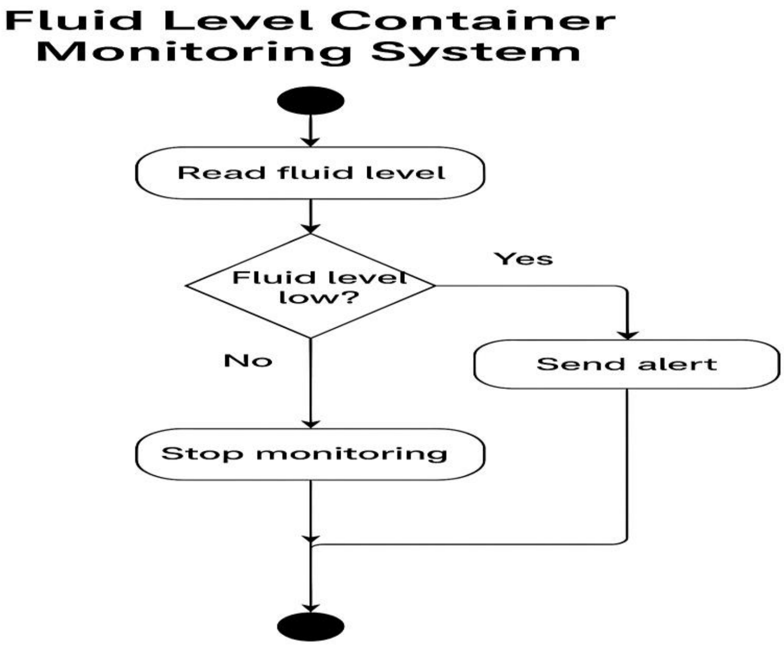

Figure 3.3 Activity Diagram

From the above figure 3.3, the activities of the system are shown

3.4 CLASS DIAGRAM

A class diagram is an illustration of the relationships and source code dependencies among classes in the Unified Modelling Language (UML). In this context, a class defines the methods and variables in an object, which is a specific entity in a program or the unit of code representing that entity.

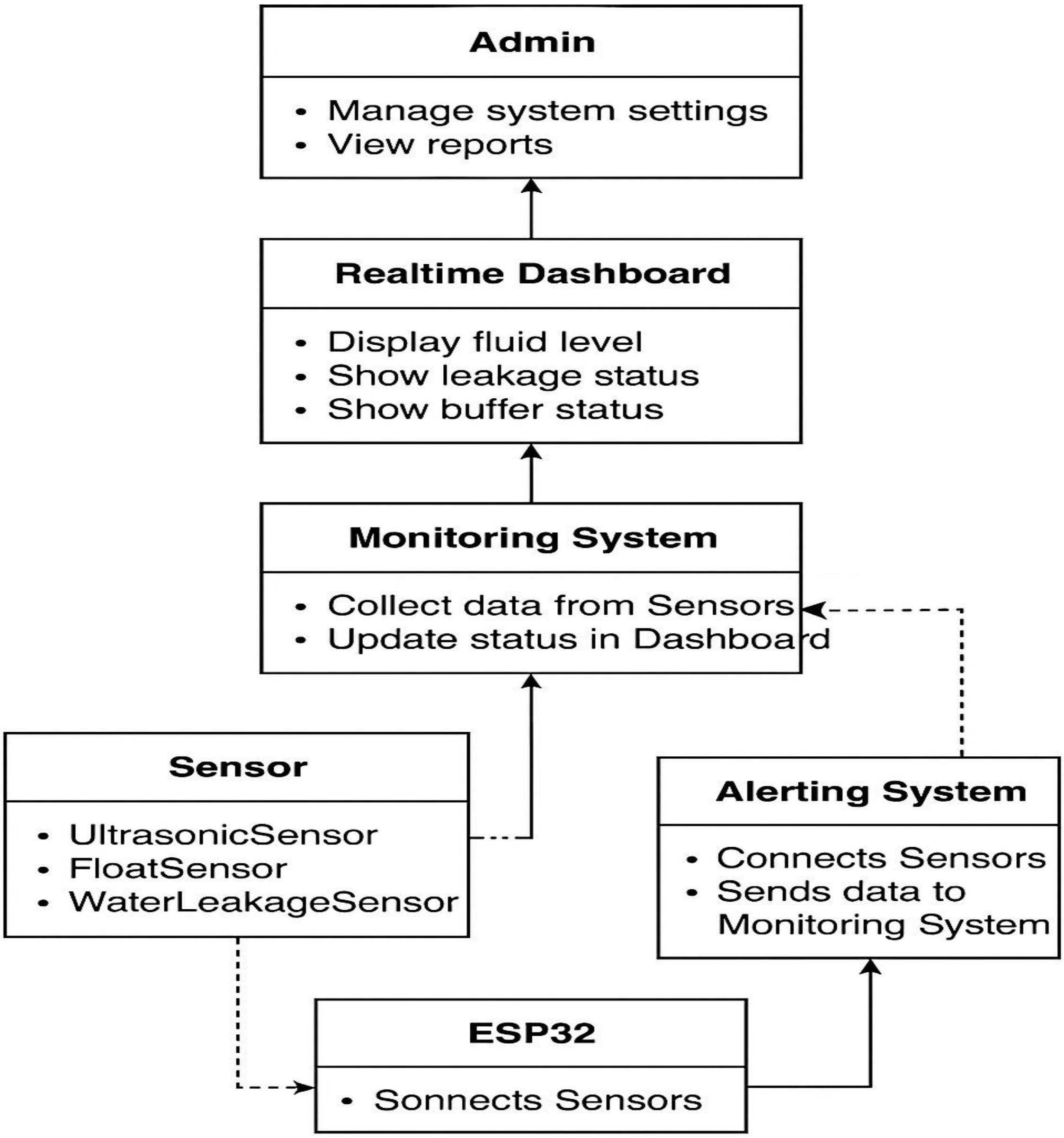

Figure 3.4 Class Diagram

The above Figure 3.4 is the class diagram for the system.

CHAPTER 4

MODULE DESCRIPTION

4.1 HARDWARE MODULE:

This module consists of an Ultrasonic Sensor for measuring fluid levels, a Float Sensor for liquid presence verification, a Water Leakage Sensor to detect spillage or leaks, and a Buffer to stabilize sensor signals. The ESP32 microcontroller is used for collecting sensor data and managing communication with the processing unit.

4.2 DATA COLLECTION AND PROCESSING MODULE:

This module interfaces with the ESP32 to gather real-time data from all connected sensors. It processes the inputs using embedded logic to convert raw signals into meaningful fluid level metrics and stores the information locally or in a connected database for further analysis.

4.3 ALERTING MODULE:

It continuously monitors sensor readings for abnormalities such as overflow, leakage, or low fluid levels. When thresholds are crossed, the system triggers alerts via buzzer or notification signals, enabling immediate user intervention.

4.4 WEB APPLICATION MODULE:

A responsive and user-friendly web dashboard is developed to display real-time fluid levels and alert statuses. It provides remote access to data, helping users monitor and manage container conditions anytime and anywhere.

4.5 INTEGRATION MODULE:

This module ensures smooth data flow between the hardware, processing logic, and the web dashboard. It integrates all subsystems to deliver synchronized and live monitoring updates, ensuring efficient operation of the entire monitoring system.

CHAPTER 6

SAMPLE CODING

ESP32 Program

```
#define TRIG_PIN 5    // Ultrasonic Trigger
#define ECHO_PIN 4    // Ultrasonic Echo
#define FLOAT_PIN 18  // Float Sensor
#define TEMP_PIN 19   // DS18B20 Temperature Sensor
#define LEAK_PIN 21   // Water Leakage Sensor
#define BUZZER_PIN 22 // Buzzer or LED

#include <OneWire.h>
#include <DallasTemperature.h>

OneWire oneWire(TEMP_PIN);
DallasTemperature sensors(&oneWire);

long duration;
float distance;
float temperature;

void setup() {
  Serial.begin(115200);

  pinMode(TRIG_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);
  pinMode(FLOAT_PIN, INPUT_PULLUP);
  pinMode(LEAK_PIN, INPUT_PULLUP);
  pinMode(BUZZER_PIN, OUTPUT);

  sensors.begin();

  Serial.println("Smart Fluid Container Monitoring System Initialized");
}

void loop() {
  Serial.println("--------------------------------------------------");

  // Ultrasonic Sensor - Distance Measurement
```

```
digitalWrite(TRIG_PIN, LOW);
delayMicroseconds(2);
digitalWrite(TRIG_PIN, HIGH);
delayMicroseconds(10);
digitalWrite(TRIG_PIN, LOW);

duration = pulseIn(ECHO_PIN, HIGH);
distance = duration * 0.034 / 2;

Serial.print("Ultrasonic Distance: ");
Serial.print(distance);
Serial.println(" cm");

// Float Sensor Reading
int floatStatus = digitalRead(FLOAT_PIN);
if (floatStatus == LOW) {
  Serial.println("Float Sensor: LOW Fluid Level");
} else {
  Serial.println("Float Sensor: HIGH Fluid Level");
}

// Temperature Sensor Reading
/*sensors.requestTemperatures();
temperature = sensors.getTempCByIndex(0);
Serial.print("Temperature: ");
Serial.print(temperature);
Serial.println(" °C");*/

// Water Leakage Sensor Reading
int leakStatus = digitalRead(LEAK_PIN);
if (leakStatus == HIGH) {
  Serial.println("Leakage Status: LEAK DETECTED!");
} else {
  Serial.println("Leakage Status: NO LEAK");
}

// Buzzer/LED Indication
if (floatStatus == LOW || leakStatus == HIGH) {
  digitalWrite(BUZZER_PIN, HIGH); // Alert ON
} else {
  digitalWrite(BUZZER_PIN, LOW);  // Alert OFF
}
```

```
  delay(2000); // Delay 2 sec
}
```

Web Application

1. index.js

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Smart Fluid Container Monitor</title>
   <script src="https://www.gstatic.com/firebasejs/8.10.0/firebase-app.js"></script>
   <script src="https://www.gstatic.com/firebasejs/8.10.0/firebase-database.js"></script>
   <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
   <link rel="stylesheet" href="styles.css">
</head>
<body>
   <div class="container">
      <header>
         <h1>Smart Fluid Container Monitor</h1>
         <div class="device-status">
            <span id="connection-status">⬤ Offline</span>
            <span id="last-update">Last Update: Never</span>
         </div>
      </header>

      <div class="dashboard-grid">
         <!-- Current Readings Section -->
         <div class="card" id="current-readings">
            <h2>Current Readings</h2>
            <div class="readings-grid">
               <div class="reading-box">
                  <h3>Fluid Level</h3>
                  <p id="fluid-level">-- cm</p>
               </div>
               <div class="reading-box">
                  <h3>Fluid Status</h3>
                  <p id="fluid-status">--</p>
               </div>
               <div class="reading-box" id="leak-box">
                  <h3>Leak Status</h3>
                  <p id="leak-status">No Leak</p>
               </div>
            </div>
```

```html
            </div>

            <!-- Charts Section -->
            <div class="card">
                <h2>Fluid Level Trend</h2>
                <canvas id="fluidLevelChart"></canvas>
            </div>

            <!-- Settings Section -->
            <div class="card" id="settings">
                <h2>Settings</h2>
                <div class="settings-grid">
                    <div class="setting-item">
                        <label for="min-fluid-level">Min Fluid Level (cm):</label>
                        <input type="number" id="min-fluid-level" value="10">
                    </div>
                    <div class="setting-item">
                        <label for="refresh-interval">Refresh Interval:</label>
                        <select id="refresh-interval">
                            <option value="2000">2 seconds</option>
                            <option value="5000">5 seconds</option>
                            <option value="10000">10 seconds</option>
                        </select>
                    </div>
                    <div class="setting-item">
                        <label for="real-time-updates">Real-time Updates:</label>
                        <input type="checkbox" id="real-time-updates" checked>
                    </div>
                </div>
            </div>

            <!-- Debug Section -->
            <div class="card" id="debug">
                <h2>Debug Controls</h2>
                <div class="debug-buttons">
                    <button id="test-buzzer">Test Buzzer</button>
                    <button id="export-data">Export Data (CSV)</button>
                </div>
            </div>
        </div>
    </div>
    <script src="app.js"></script>
</body>
</html>
```

Styles.css

```css
* {
   margin: 0;
   padding: 0;
   box-sizing: border-box;
}

:root {
   --primary-color: #2563eb;
   --secondary-color: #1e40af;
   --background-color: #f1f5f9;
   --card-background: #ffffff;
   --danger-color: #dc2626;
   --success-color: #16a34a;
   --text-color: #1e293b;
}

body {
   font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', Roboto, Oxygen, Ubuntu,
Cantarell, sans-serif;
   background-color: var(--background-color);
   color: var(--text-color);
   line-height: 1.6;
}

.container {
   max-width: 1200px;
   margin: 0 auto;
   padding: 20px;
}

header {
   display: flex;
   justify-content: space-between;
   align-items: center;
   margin-bottom: 30px;
}

h1 {
   color: var(--primary-color);
   font-size: 2rem;
}

.device-status {
   display: flex;
   gap: 20px;
   align-items: center;
}

.dashboard-grid {
   display: grid;
```

```
    grid-template-columns: repeat(auto-fit, minmax(300px, 1fr));
    gap: 20px;
}

.card {
    background: var(--card-background);
    border-radius: 10px;
    padding: 20px;
    box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);
}

.readings-grid {
    display: grid;
    grid-template-columns: repeat(2, 1fr);
    gap: 15px;
    margin-top: 15px;
}

.reading-box {
    background: var(--background-color);
    padding: 15px;
    border-radius: 8px;
    text-align: center;
}

.reading-box h3 {
    font-size: 0.9rem;
    color: var(--text-color);
    margin-bottom: 5px;
}

.reading-box p {
    font-size: 1.5rem;
    font-weight: bold;
    color: var(--primary-color);
}

.settings-grid {
    display: grid;
    gap: 15px;
    margin-top: 15px;
}

.setting-item {
    display: flex;
    justify-content: space-between;
    align-items: center;
}

.setting-item label {
```

```
    font-weight: 500;
}

input[type="number"],
select {
    padding: 8px;
    border: 1px solid #ddd;
    border-radius: 4px;
    width: 120px;
}

.debug-buttons {
    display: flex;
    gap: 10px;
    margin-top: 15px;
}

button {
    background-color: var(--primary-color);
    color: white;
    border: none;
    padding: 10px 20px;
    border-radius: 5px;
    cursor: pointer;
    transition: background-color 0.3s;
}

button:hover {
    background-color: var(--secondary-color);
}

#leak-box.alert {
    background-color: var(--danger-color);
    color: white;
    animation: blink 1s infinite;
}

@keyframes blink {
    0% { opacity: 1; }
    50% { opacity: 0.5; }
    100% { opacity: 1; }
}

/* Mobile Responsive Design */
@media (max-width: 768px) {
    header {
        flex-direction: column;
        text-align: center;
        gap: 15px;
    }
```

```
    .readings-grid {
        grid-template-columns: 1fr;
    }

    .setting-item {
        flex-direction: column;
        gap: 5px;
        align-items: flex-start;
    }

    input[type="number"],
    select {
        width: 100%;
    }
}

/* Chart Styles */
canvas {
    width: 100% !important;
    height: 200px !important;
    margin-top: 15px;
}
```

App.js

```
// Firebase Configuration
const firebaseConfig = {
    apiKey: "AIzaSyAsSBNvA-wG24a-L5i52DhcuGqC1QT8nX4",
    authDomain: "smart-fluid-monitoring-system.firebaseapp.com",
    projectId: "smart-fluid-monitoring-system",
    storageBucket: "smart-fluid-monitoring-system.firebasestorage.app",
    messagingSenderId: "25297853703",
    appId: "1:25297853703:web:8ccab211c73a20179179a2",
    measurementId: "G-DPF0V4LK45",
    databaseURL: "https://smart-fluid-monitoring-system-default-rtdb.firebaseio.com"
};

// Initialize Firebase
firebase.initializeApp(firebaseConfig);
const database = firebase.database();

// DOM Elements
const elements = {
    fluidLevel: document.getElementById('fluid-level'),
    fluidStatus: document.getElementById('fluid-status'),
    leakStatus: document.getElementById('leak-status'),
    leakBox: document.getElementById('leak-box'),
    connectionStatus: document.getElementById('connection-status'),
    lastUpdate: document.getElementById('last-update'),
    minFluidLevel: document.getElementById('min-fluid-level'),
```

```
    refreshInterval: document.getElementById('refresh-interval'),
    realTimeUpdates: document.getElementById('real-time-updates'),
    testBuzzer: document.getElementById('test-buzzer'),
    exportData: document.getElementById('export-data')
};

// Chart Configuration
const fluidLevelChart = new Chart(document.getElementById('fluidLevelChart'), {
    type: 'line',
    data: {
        labels: [],
        datasets: [{
            label: 'Fluid Level (cm)',
            data: [],
            borderColor: '#2563eb',
            tension: 0.1
        }]
    },
    options: {
        responsive: true,
        maintainAspectRatio: false
    }
});

// Data Storage
let sensorData = [];
let updateInterval;

// Update UI with sensor data
function updateUI(data) {
    if (!data) return;

    elements.fluidLevel.textContent = `${data.distance.toFixed(1)} cm`;
    elements.fluidStatus.textContent = data.floatLevel.includes("HIGH") ? 'HIGH' : 'LOW';

    // Update leak status
    const isLeaking = data.leakStatus.includes("LEAK");
    elements.leakStatus.textContent = isLeaking ? 'LEAK DETECTED!' : 'No Leak';
    elements.leakBox.classList.toggle('alert', isLeaking);

    // Update last update time
    elements.lastUpdate.textContent = `Last Update: ${new Date().toLocaleTimeString()}`;

    // Update charts
    const timestamp = new Date().toLocaleTimeString();

    // Update fluid level chart
    fluidLevelChart.data.labels.push(timestamp);
    fluidLevelChart.data.datasets[0].data.push(data.distance);
    if (fluidLevelChart.data.labels.length > 20) {
```

```
        fluidLevelChart.data.labels.shift();
        fluidLevelChart.data.datasets[0].data.shift();
    }
    fluidLevelChart.update();

    // Store data for export
    sensorData.push({
        timestamp: new Date().toISOString(),
        fluidLevel: data.distance,
        fluidStatus: data.floatLevel.includes("HIGH") ? 'HIGH' : 'LOW',
        leakStatus: data.leakStatus.includes("LEAK") ? 'LEAK' : 'NO LEAK'
    });
}

// Firebase Real-time Updates
function startRealtimeUpdates() {
    const sensorRef = database.ref('sensor_data');

    sensorRef.on('value', (snapshot) => {
        const data = snapshot.val();
        if (data) {
            elements.connectionStatus.textContent = '🟢 Online';
            updateUI(data);
        }
    }, (error) => {
        console.error('Error reading sensor data:', error);
        elements.connectionStatus.textContent = '🔴 Error';
    });
}

// Settings Event Listeners
elements.realTimeUpdates.addEventListener('change', (e) => {
    if (e.target.checked) {
        startRealtimeUpdates();
    } else {
        database.ref('sensor_data').off();
        elements.connectionStatus.textContent = '⚫ Updates Paused';
    }
});

elements.refreshInterval.addEventListener('change', (e) => {
    const interval = parseInt(e.target.value);
    clearInterval(updateInterval);
    if (!elements.realTimeUpdates.checked) {
        updateInterval = setInterval(() => {
            database.ref('sensor_data').once('value').then((snapshot) => {
                updateUI(snapshot.val());
            });
        }, interval);
    }
```

```
});

// Debug Controls
elements.testBuzzer.addEventListener('click', () => {
    database.ref('debug/buzzer').set(true)
        .then(() => setTimeout(() => database.ref('debug/buzzer').set(false), 1000));
});

// Export Data
elements.exportData.addEventListener('click', () => {
    const csv = [
        ['Timestamp', 'Fluid Level (cm)', 'Fluid Status', 'Leak Status'],
        ...sensorData.map(row => [
            row.timestamp,
            row.fluidLevel,
            row.fluidStatus,
            row.leakStatus
        ])
    ].map(row => row.join(',')).join('\n');

    const blob = new Blob([csv], { type: 'text/csv' });
    const url = window.URL.createObjectURL(blob);
    const a = document.createElement('a');
    a.setAttribute('hidden', '');
    a.setAttribute('href', url);
    a.setAttribute('download', 'sensor_data.csv');
    document.body.appendChild(a);
    a.click();
    document.body.removeChild(a);
});

// Initialize
document.addEventListener('DOMContentLoaded', () => {
    // Start real-time updates if enabled
    if (elements.realTimeUpdates.checked) {
        startRealtimeUpdates();
    }
});
```

CHAPTER 7

SCREEN SHOTS

1. Dashboard Page

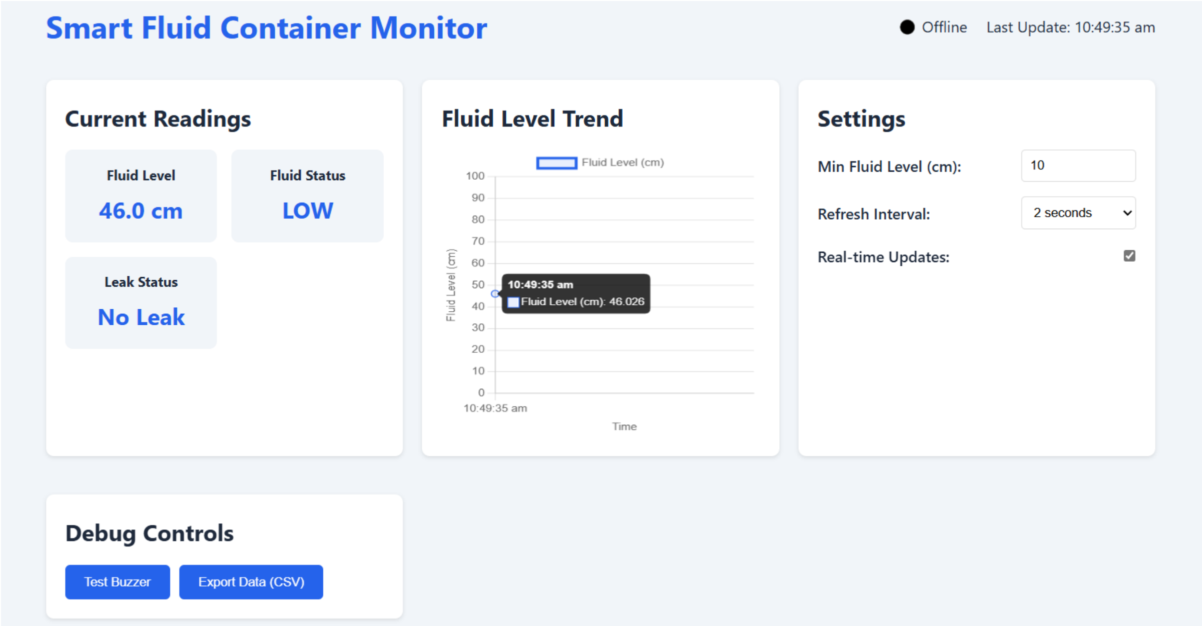

Figure 7.1 Responsive Dashboard

2. Data Sent from ESP32 to Server

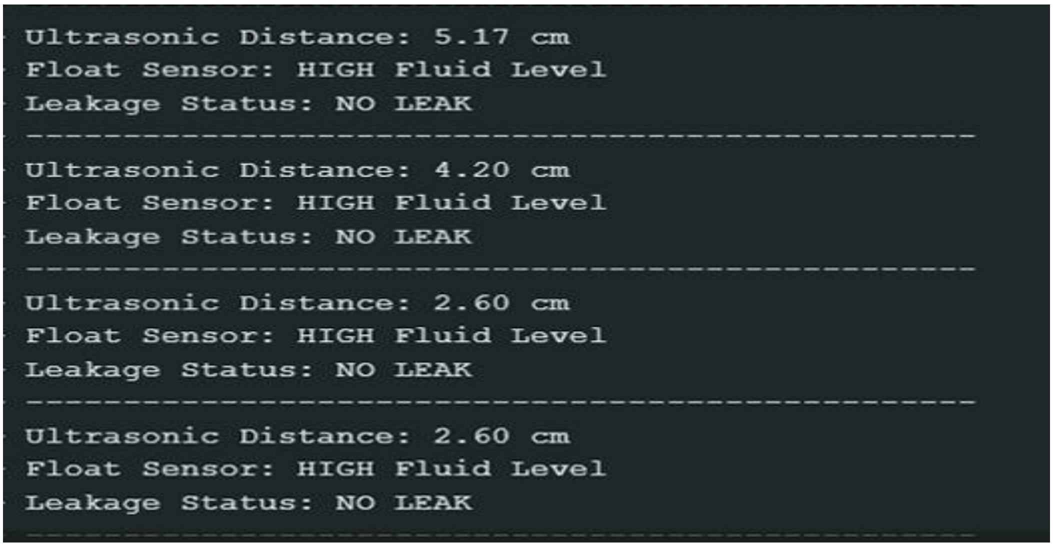

Figure 7.2 Data Received by Server from ESP32

CHAPTER 8

CONCLUSION AND FUTURE ENHANCEMENT

The Smart Fluid Level Container Monitoring System offers an innovative solution for managing liquid levels efficiently using IoT technology. By combining sensors such as ultrasonic, float, and leakage detectors with the ESP32 microcontroller and a real-time web dashboard, the system ensures precise monitoring, timely alerts, and easy access to data. This significantly improves the management of fluid resources, minimizes wastage, and enhances operational safety. The system's real-time capabilities promote preventive action, making it highly suitable for various industries where fluid monitoring is critical.

Future enhancements can focus on integrating more advanced data analytics to predict usage patterns and detect anomalies. Introducing AI and machine learning algorithms could automate decision-making and further reduce manual intervention. Cloud integration for large-scale data storage and analysis, mobile application support for remote access, and additional sensors for detecting fluid quality or contamination could extend the system's versatility. Finally, combining the system with automated refill mechanisms can lead to a fully autonomous fluid management system**.**

REFERENCES

1. [1] N. Saeidi, K. Selvam, F. Tortato, M. Wiemer, and H. Kuhn, "High Precision Liquid Level and Leak Detection Based on Capacitive Micromachined Ultrasound Transducer," *IEEE Sensors Symposium*, 2022, doi: 10.1109/SSI56489.2022.9901430.
2. [2] D. S. Bhandare, V. S. Jape, H. H. Kulkarni, S. M. Mahajan, P. Sable, Y. Pawar, "Automatic Liquid Level Control of Two Tank System using PLC," *IEEE International Conference on Intelligent Systems and Automation (ICISAA)*, 2024, doi: 10.1109/ICISAA62385.2024.10828622.
3. [3] X. He, Y. Ma, T. Li, Q. Sun, Q. Feng, "Real-Time Liquid Level

Monitoring With Low-Cost UHF RFID Sensors," *IEEE Sensors Journal*, 2025, doi: 10.1109/JSEN.2025.3528128.

4. [4] S. S. Gondkar, D. B. Pardeshi, P. W. William, "Innovative System for Water Level Management using IoT to prevent Water Wastage," *IEEE International Conference on Advances in Computing and Artificial Intelligence (ICAAIC)*, 2022, doi: 10.1109/ICAAIC53929.2022.9792746.
5. [5] N. R. Gayathri, S. Manjula, W. T. Chembian, V. Dhivya, K. R. Anirudh Dhanunjay, "Smart Water Quality Monitoring and Management System," *IEEE International Conference on Power Electronics and Communication Technology (ICPECTS)*, 2024, doi: 10.1109/ICPECTS62210.2024.10780355.