Automated Water Level Monitoring and Control System with OLED Display

Prepared by:

Mohar Chatterjee

Rohan Chatterjee

Electronics and Communication Engineering

Heritage Institute of Technology

Project Start : 22/08/2024
Project End : 06/09/2024

Preface

Water is an essential and finite resource, vital for life and critical for a wide array of industrial and domestic applications. The increasing demand for water conservation and efficient resource management has spurred innovation in monitoring and control systems. This project, *Automated Water Level Monitoring and Control System with OLED Display*, is a reflection of our commitment to harnessing technology for sustainable solutions.

The project integrates advanced components like an ultrasonic sensor, relay-controlled water pump, and OLED display to create a robust and versatile system. By addressing challenges such as overflow prevention, critical water level alerts, and efficient manual override mechanisms, this system aspires to redefine the standards of water management.

Through this report, we aim to provide a detailed account of the design, functionality, and testing of the system, demonstrating its relevance and effectiveness in real-world applications.

Acknowledgement

We would like to express our heartfelt gratitude to all who contributed to the successful completion of this project.

First and foremost, we thank our project guide and mentor, whose guidance, expertise, and encouragement were invaluable throughout the development of this system. Your insights and feedback shaped the foundation and fine-tuned the details of this project.

We extend our appreciation to the faculty members and the Department of Electronics and Communication Engineering at the Heritage Institute of Technology for providing us with the necessary infrastructure, resources, and knowledge to undertake this endeavour.

Special thanks to our peers and family members, whose unwavering support and motivation were instrumental in overcoming challenges and achieving our goals.

Finally, we acknowledge the collective efforts of every individual involved in the process, directly or indirectly, whose contributions made this project a reality. This work stands as a testament to teamwork, perseverance, and the spirit of innovation.

Thank you all for believing in us and for your continued encouragement.

Table of Contents

1. Introduction to Water Level Monitoring and Control Systems:

Water is a critical resource in both residential and industrial environments, making its monitoring and control essential. The need for automated systems to manage water levels is paramount to ensure efficiency, prevent overflow, and avoid shortages. Traditional manual methods are prone to errors and inefficiencies. Therefore, automation provides a reliable, consistent, and scalable solution to water management.

2. Project Overview:

The 'Automated Water Level Monitoring and Control System with OLED Display' is designed to monitor the water level in a tank, provide real-time feedback via an OLED display, and automate the control of water flow using a relay-controlled water pump. The system alerts users with a buzzer when the water level is critically low or at risk of overflowing. The automation can operate in both manual and automatic modes, offering flexibility in operation. LEDs are used to visually indicate the water level, and a mute function is provided to temporarily silence the buzzer.

3. System Architecture:

This project integrates an ultrasonic sensor to measure the water level, an OLED display to visualize the data, a relay to control a water pump, and a buzzer to signal critical conditions. The system can be operated in two modes: manual and automatic.

- **Automatic Mode:** The system automatically turns the pump on or off based on predefined water level thresholds.
- **Manual Mode:** The user manually controls the pump, but the system still monitors and provides alerts.

The mute function ensures that the buzzer can be silenced temporarily when the water level is within critical ranges. The system is built on an Arduino microcontroller, which processes inputs from the sensors and manages outputs to the display, relay, buzzer, and LEDs.

4. Features and Functionalities:

- **Real-time Water Level Monitoring:** The ultrasonic sensor continuously measures the water level, and the data is displayed on an OLED screen.
- **Visual Indicators:** An LED array provides a quick visual representation of the water level.
- **Automated Control:** The relay is automatically controlled based on water level thresholds.
- **Critical Alerts:** The buzzer sounds an alert when the water level is too low or too high.
- **Manual Override:** The system allows manual control of the pump with a slide switch.
- **Mute Function:** Users can mute the buzzer during critical water levels, with automatic reactivation if the level changes.

Bill of Materials (BOM):

Component	Quantity	Description	Data Sheet
Arduino Uno R3	1	Microcontroller for processing and control	https://docs.arduino.cc/resources/datasheets/A000066-datasheet.pdf
Ultrasonic Sensor (HC-SR04)	1	Measures the distance to the water surface	https://docs.google.com/document/d/1Y-yZnNhMYy7rwhAgyL_pfa39RsB-x2qR4vP8saG73rE/edit?tab=t.0
OLED Display (128x64, SSD1306), 0.96 inch	1	Displays water level and system status	https://www.electronicscomp.com/datasheet/ssd1306-datasheet.pdf

Single Channel Relay Module	1	Controls the water pump based on water level	https://handsontec.com/dataspecs/relay/1Ch-relay.pdf
Buzzer	1	Alerts user during critical water levels	https://drive.google.com/file/d/1_CeNp-s6odc2z37oX1qhuX2k9EIKJZSZ/view
LEDs (Blue, Green, Yellow, Orange, Red)	5	Visual indication of water levels	https://www.tme.eu/Document/e17ac5a5d91f1e843e5681cfa4a5ba4b/HLMP-HG64-VY0DD.pdf
Resistors (220Ω, 1kΩ)	10	Current limiting resistors for LEDs and pull-up resistors	https://www.farnell.com/datasheets/1807347.pdf
Push Button	1	Used for mute functionality	https://components101.com/sites/default/files/component_datasheet/Push-Button.pdf
6 pin Toggle Switch	1	Used for mode selection	https://www.farnell.com/datasheets/32931.pdf
Slide Switch	1	Used for manual mode control	https://www.farnell.com/datasheets/32947.pdf
Jumper Wires	40+	For connecting components	https://tubesandmore.com/sites/default/files/associated_files/s-w604_spec.pdf
Breadboard	2	For prototyping and testing	https://components101.com/sites/default/files/component_datasheet/Breadboard%20Datasheet.pdf
12V Power Supply	1	Powers the Arduino and peripherals	https://www.farnell.com/datasheets/2720924.pdf
Water Pump with pipe	1	For refilling purposes in automation	https://www.lenntech.com/Data-sheets/Danfoss-APP-11-13-L.pdf
9V external battery	1	For supplying power to the pump.	https://www.farnell.com/datasheets/1842389.pdf

5. Circuit Connections:

- **Ultrasonic Sensor:**
  - Trig Pin: Connected to Arduino digital pin 9
  - Echo Pin: Connected to Arduino digital pin 10
  - VCC: Connected to 5V
  - GND: Connected to GND

- **OLED Display:**
  - SCL: Connected to Arduino A5 (SCL)
  - SDA: Connected to Arduino A4 (SDA)
  - VCC: Connected to 3.3V
  - GND: Connected to GND

- **Relay Module:**
  - IN: Connected to Arduino digital pin 2
  - VCC: Connected to 5V
  - GND: Connected to GND
  - NOC: Connected to GND

- **Buzzer:**
  - Positive: Connected to Arduino digital pin 4
  - Negative: Connected to GND

- **LEDs:**
  - Red (RDP): Connected to Arduino digital pin 13
  - Yellow: Connected to Arduino digital pin 6
  - Orange: Connected to Arduino digital pin 5
  - Green: Connected to Arduino digital pin 8
  - Blue: Connected to Arduino digital pin 7

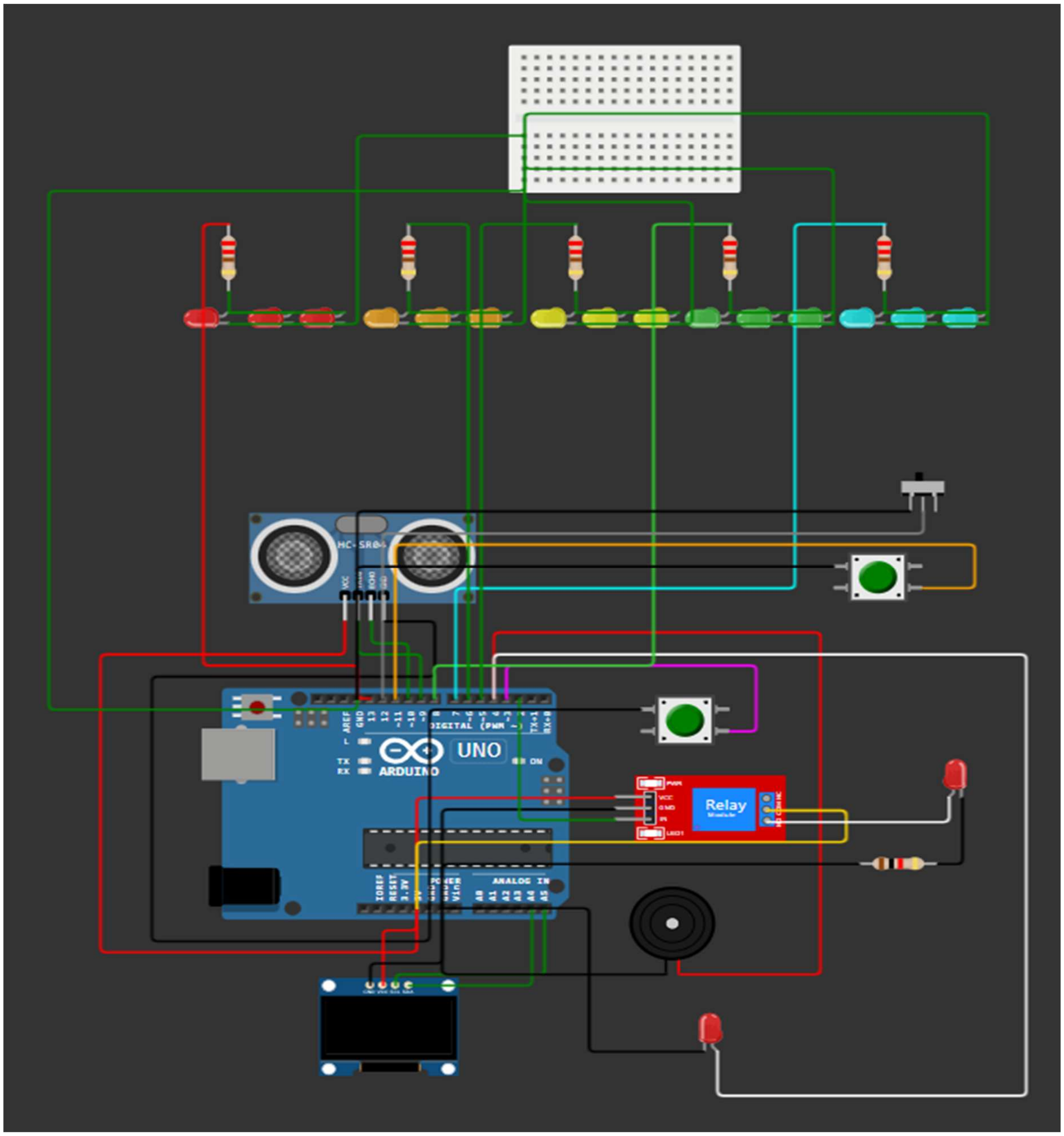
- **<u>Mode Toggle Button:</u>**
  - One Side: Connected to Arduino digital pin 11
  - Other Side: Connected to GND

- **<u>Mute Button:</u>**
  - One Side: Connected to Arduino digital pin 3
  - Other Side: Connected to GND
- **<u>Slide Switch:</u>**
  - One Side: Connected to Arduino digital pin 12
  - Other Side: Connected to GND

- **<u>Water Pump:</u>**
  - Positive Side: Connected to COM port of Relay
  - Negative Side: Connected to GND

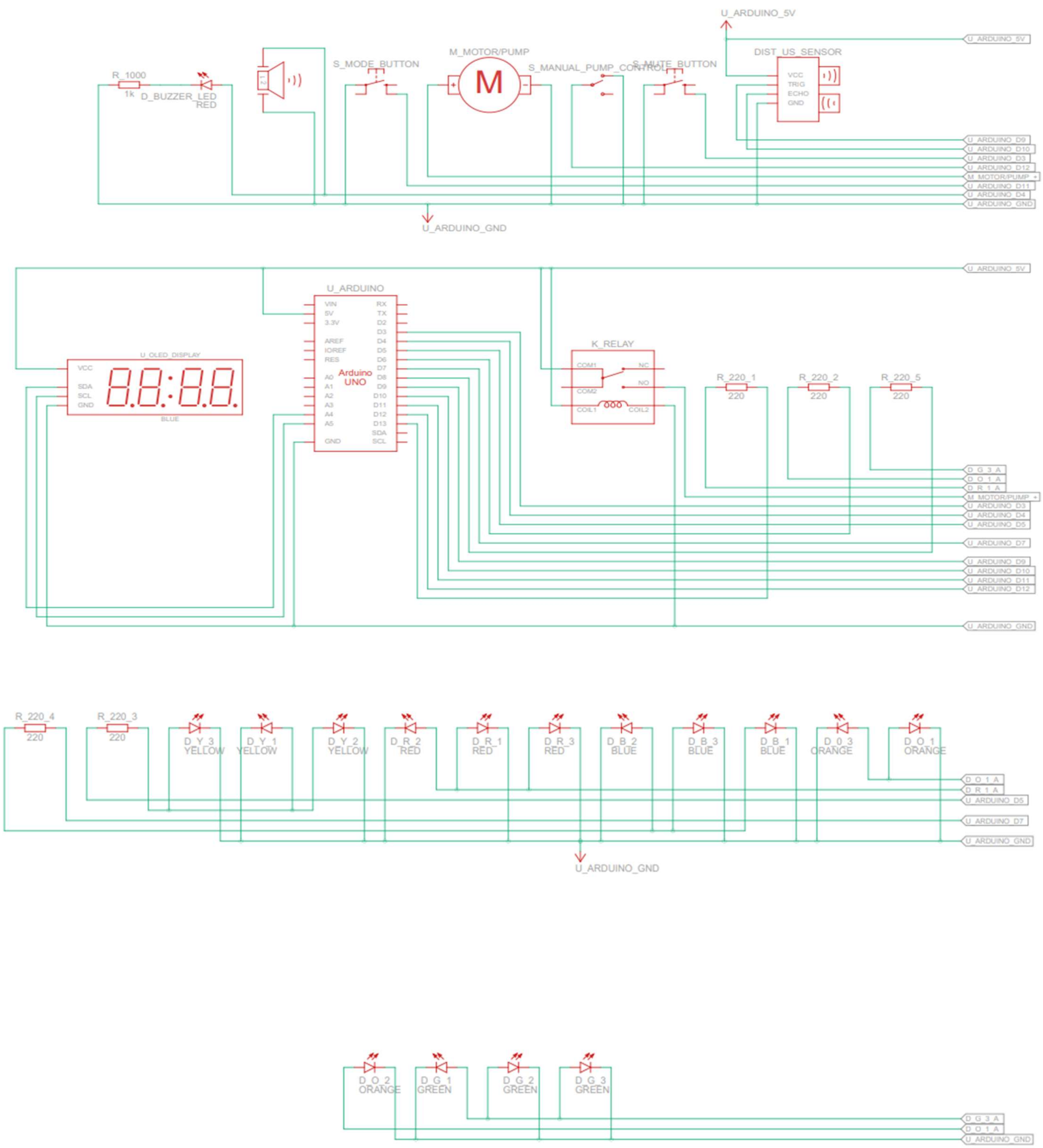
- **<u>Relay output connections for external 9V battery:</u>**
  - Battery Positive Terminal: COM port
  - Battery Negative Terminal: GND

6. Circuit Connections on Wokwi Simulator:



SIMULATION LINK: https://wokwi.com/projects/403604367575169025

7. Schematic Diagram:



8. Software Implementation:

The software is written in C++ using the Arduino IDE. It reads inputs from the ultrasonic sensor and mode switches, processes this data to determine the water level and system mode, and controls outputs to the OLED display, relay, buzzer, and LEDs accordingly. The system uses a debounce technique to ensure reliable button presses

CODE IMPLEMENTATION:

```
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
#define OLED_RESET -1
Adafruit_SSD1306 display (SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);

const int trigPin = 9;
const int echoPin = 10;
const int relayPin = 2;
const int mutePin = 3; // Define the pin for the mute switch
const int buzzerPin = 4; // Define the pin for the buzzer
const int RDP = 13;
const int darkYellowLEDPin = 6;
const int yellowLEDPin = 5;
const int greenLEDPin = 8;
const int blueLEDPin = 7;
const int modeSwitchPin = 11; // Define the pin for the mode toggle switch
const int slideSwitchPin = 12; // Define the pin for the manual mode slide switch
bool mute = false;
bool lastButtonState = HIGH; // Previous state of the mute button
unsigned long lastDebounceTime = 0; // Last time the mute button was toggled
unsigned long debounceDelay = 50; // The debounce time; increase if the output
flickers
bool autoMode = true; // Variable to track the current mode (default is auto)
bool lastModeButtonState = HIGH; // Previous state of the mode button
unsigned long lastModeDebounceTime = 0; // Last time the mode button was toggled

void setup () {
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  digitalWrite(relayPin, HIGH);
  pinMode(relayPin, OUTPUT);
  pinMode(mutePin, INPUT_PULLUP);
  pinMode(buzzerPin, OUTPUT); // Set the buzzer pin as output
  pinMode(RDP, OUTPUT);
```

```
  pinMode(darkYellowLEDPin, OUTPUT);
  pinMode(yellowLEDPin, OUTPUT);
  pinMode(greenLEDPin, OUTPUT);
  pinMode(blueLEDPin, OUTPUT);
  pinMode(modeSwitchPin, INPUT_PULLUP);
  pinMode(slideSwitchPin, INPUT_PULLUP);
  Serial.begin(9600);
  if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
        Serial.println(F ("SSD1306 allocation failed"));
        for (;;);
  }
  display.display();
  delay (2000);
  display.clearDisplay();
}

void loop () {
   long duration;
   int distance;
   int percent;
   int muteReading = digitalRead(mutePin);
   if (muteReading != lastButtonState) {
        lastDebounceTime = millis(); // reset the debouncing timer
   }
   if ((millis() - lastDebounceTime) > debounceDelay) {
        if (muteReading == LOW &&! mute) {
           mute = true; // Set the mute state
           noTone(buzzerPin); // Ensure no tone is playing
        }
   }
   lastButtonState = muteReading;
   autoMode = digitalRead(modeSwitchPin) == HIGH;
   digitalWrite(trigPin, LOW);
   delayMicroseconds(2);
   digitalWrite(trigPin, HIGH);
   delayMicroseconds(10);
   digitalWrite(trigPin, LOW);
   duration = pulseIn(echoPin, HIGH);
   distance = duration * 0.034 / 2;
   percent = 100 - map (distance, 0, 20, 0, 100);
   display.clearDisplay();

   // Print water level percentage
   display.setTextSize(1);
   display.setTextColor(SSD1306_WHITE);
   display.setCursor(9.5, 11);
   display.print("WATER LEVEL \n");
   display.print("--------------");
   display.print("\n");
```

```
 // Draw a simple bar graph representing the water level
 int barHeight = map (percent, 0, 100, 0, SCREEN_HEIGHT - 15); // Adjusted height
 int barWidth = 35;
 int barX = 90;
 int barY = 13; // Adjusted starting point
 // Calculate top of the bar
 int barTop = barY + (SCREEN_HEIGHT - 14.1 - barHeight);
 // Ensure bar height does not exceed screen height
 if (barHeight > (SCREEN_HEIGHT - 14.1)) {
   barHeight = SCREEN_HEIGHT - 14.1;
 }
 // Draw outer border rectangle
 display.drawRect(barX - 1, barY - 1, barWidth + 2, SCREEN_HEIGHT - 16 + 2,
SSD1306_WHITE);

 // Draw filled inner rectangle representing the bar
 display.fillRect(barX, barTop, barWidth, barHeight,SSD1306_WHITE);
 // Draw the dashed border box for percentage
 int boxWidth = 80;
 int boxHeight = 33;
 int boxX = 2;
 int boxY = SCREEN_HEIGHT - boxHeight - 3;
 // Draw top and bottom borders with dashes
 for (int i = boxX; i < boxX + boxWidth; i += 4) {
  display.drawPixel(i, boxY, SSD1306_WHITE);
  display.drawPixel(i + 1, boxY, SSD1306_WHITE);
  display.drawPixel(i, boxY + boxHeight, SSD1306_WHITE);
  display.drawPixel(i + 1, boxY + boxHeight, SSD1306_WHITE);
 }

 // Draw left and right borders
 for (int i = boxY; i < boxY + boxHeight; i += 4) {
  display.drawPixel(boxX, i, SSD1306_WHITE);
  display.drawPixel(boxX, i + 1, SSD1306_WHITE);
  display.drawPixel(boxX + boxWidth, i, SSD1306_WHITE);
  display.drawPixel(boxX + boxWidth, i + 1, SSD1306_WHITE);
 }

 // Display the percentage inside the box
 display.setCursor(boxX + 34, boxY + 8);
 display.setTextSize(1);
 display.print(percent);
 display.print("%");

 // Display mode
 display.setTextSize(1);
 display.setCursor(9.5, 0);
 if (autoMode) {
  display.print("MODE: AUTO");
```

```
 }
 else {
  display.print("MODE: MANUAL");
 }

 // Display refill status and warnings
 display.setTextSize(1);
 display.setCursor(17, SCREEN_HEIGHT - 16);

 if (digitalRead(relayPin) == LOW) {
   display.print("Refilling");
 }
 else if (!autoMode) {
       if (percent <= 15) {
          display.setCursor(20, SCREEN_HEIGHT - 16);
          display.print("LOW TANK");
       }
       else if (percent >= 95) {
          display.setCursor(20, SCREEN_HEIGHT - 16);
          display.print("OVERFLOW");
        }
  }

 // Buzzer logic
 unsigned int frequencies [] = {500, 400, 500, 400, 500, 400, 500, 400, 500, 400, 500,
400, 500, 400, 500, 400};
 unsigned int frequencies1[] = {1000, 2000, 1000, 2000, 1000, 2000, 1000, 2000, 1000,
2000, 1000, 2000, 1000, 2000, 1000, 2000};

 if (percent >= 95) {
  if (! mute) {
    digitalWrite(buzzerPin, HIGH);
    for (int i = 0; i < 16; i++) {
     tone (buzzerPin, frequencies1[i], 250);
     delay (100);
    }
  }
 }
 else if (percent <= 15) {
  if (! mute) {
     digitalWrite(buzzerPin, HIGH);
     for (int i = 0; i < 16; i++) {
          tone (buzzerPin, frequencies[i], 250);
          delay (100);
     }
  }
 }
else {
  mute = false; // Reset the mute state when outside the range
```

```
  noTone(buzzerPin);
  digitalWrite(buzzerPin, LOW); // Ensure the buzzer is off
 }

 // Relay control
 if (autoMode) {
    if (percent <= 16 && percent>=0) {
      digitalWrite(relayPin, LOW);
      noTone(buzzerPin);
    }
    else if (percent >= 92) {
      digitalWrite(relayPin, HIGH);
    }
  }
 else {
  // Manual mode: control the relay with the slide switch
  if (digitalRead(slideSwitchPin) == LOW) {
    digitalWrite(relayPin, LOW);
  }
  else {
    digitalWrite(relayPin, HIGH);
  }
 }

 // Update LED states progressively
 digitalWrite(RDP, percent > 0);
 digitalWrite(darkYellowLEDPin, percent > 25);
 digitalWrite(yellowLEDPin, percent > 48);
 digitalWrite(greenLEDPin, percent > 68);
 digitalWrite(blueLEDPin, percent > 90);

 display.display();
 delay(1000);
}
```

7. Testing and Validation:

- **Water Level Accuracy Testing:** The ultrasonic sensor's accuracy was verified by placing it at fixed heights and comparing the measured distances to actual water levels. After calibration, the sensor consistently provided accurate readings within a 2% margin of error, ensuring reliable water level monitoring.

- **OLED Display Validation**: The OLED screen was tested by simulating varying water levels and observing the display for clarity and alignment of graphical elements like bar graphs and percentages. The display functioned flawlessly, showing real-time updates with excellent visibility.

- **Relay Functionality Testing**: The relay's ability to control the water pump was tested by simulating conditions where water levels crossed high and low thresholds. The relay performed as expected, reliably turning the pump on and off according to predefined levels.

- **Buzzer Alert Testing**: The buzzer was tested for critical water level alerts by simulating low and high water levels and checking its response to the mute functionality. The buzzer produced distinct tones at critical levels, and the mute function operated without issues.

- **LED Indicators Validation**: LEDs were tested to ensure they accurately represented water levels by simulating various levels and observing the corresponding LEDs light up. The LEDs provided an intuitive and consistent visual representation of the water levels.

- **Manual and Automatic Mode Testing**: The system's manual and automatic modes were tested by toggling between them and observing pump control and alert responses. Both modes operated seamlessly, with smooth transitions and appropriate system behavior.

- **Mute Button Debounce Testing**: The mute button was tested for debouncing by rapidly pressing and releasing it multiple times. The debouncing implementation worked effectively, eliminating false triggers and ensuring accurate mute functionality.

Conclusion

The *Automated Water Level Monitoring and Control System with OLED Display* successfully achieves its goal of providing an efficient and reliable solution for water management. By integrating modern components such as an ultrasonic sensor for precise water level measurements, an OLED display for clear real-time visualization, and a relay-controlled pump for automation, the system ensures operational effectiveness in various scenarios.

The dual-mode operation—manual and automatic—offers flexibility to users, allowing them to adapt the system based on specific requirements. Critical water level alerts through both auditory (buzzer) and visual (LED) signals enhance safety and prevent potential damage caused by overflows or shortages. The addition of features like the mute function further increases user convenience, ensuring adaptability without compromising the system's responsiveness.

The comprehensive testing and validation process confirmed the system's reliability, accuracy, and durability. Whether in residential or industrial settings, the system can address water management challenges effectively, minimizing wastage and ensuring optimal resource utilization.
This project underscores the importance of automation in addressing critical resource management needs. Its robust design and functionality pave the way for potential future enhancements, such as wireless monitoring, mobile app integration, or IoT-based controls, which could further expand its utility. By combining technology with practical applications, this system exemplifies a step toward sustainable and smart resource management solutions.

Bibliography

Below are the references and resources utilized during the development and completion of the project:

- ***"Internet of Things: Principles and Paradigms"*** by Rajkumar Buyya and Amir Vahid Dastjerdi.
- ***"Designing the Internet of Things"*** by Adrian McEwen and Hakim Cassimally.
- ***"Building the Internet of Things"*** by Maciej Kranz.
- **Electronics Hub:** https://www.electronicshub.org/iot-project-ideas/
- **IoT Design and Project Ideas:** https://www.electronicsforu.com/iot-projects-ideas/
- **Water Management IoT Projects:** https://circuitdigest.com/
- **Wokwi Simulator:** https://wokwi.com/

These resources provided foundational knowledge and practical guidance in the design and implementation of the project, ensuring its success and functionality.