Smart Water Level Monitoring System

IOT
FINAL REPORT

GROUP 02

Smart Water Level Monitoring System

IOT Final Project:

BY:
Madduri Dashwanth (Team Leader)
Bozza Prakash Abhinay
K Author Abhikhyath Gundi
Kothith Pappala

Summary

Our project presents a smart water level monitoring system, designed and simulated using Tinkercad, an online prototyping tool for Arduino-based electronics. The system demonstrates how IoT can help monitor and alert users about water level changes in real-time. Using an ultrasonic sensor, the Arduino calculates distance from the water surface, displays the percentage on an LCD, and activates a buzzer and RGB LED for low/high-level alerts.

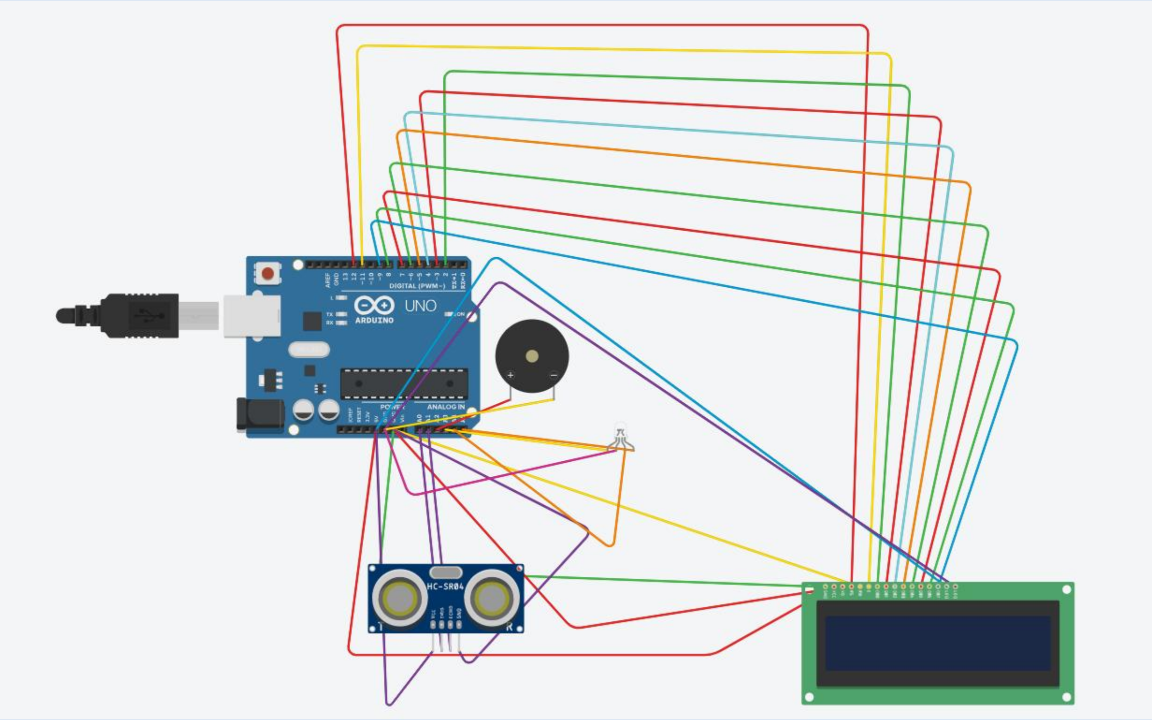
Project Overview

This project aims to simulate a real-time water level monitoring solution using Arduino Uno, ultrasonic sensor, LCD, buzzer, and RGB LED. The system shows the water level in percentage, alerts the user through colored lights and sound when predefined thresholds are crossed. The goal is to reduce manual checking and avoid overflow or dry conditions in tanks or reservoirs. It was entirely developed and tested in Tinkercad.

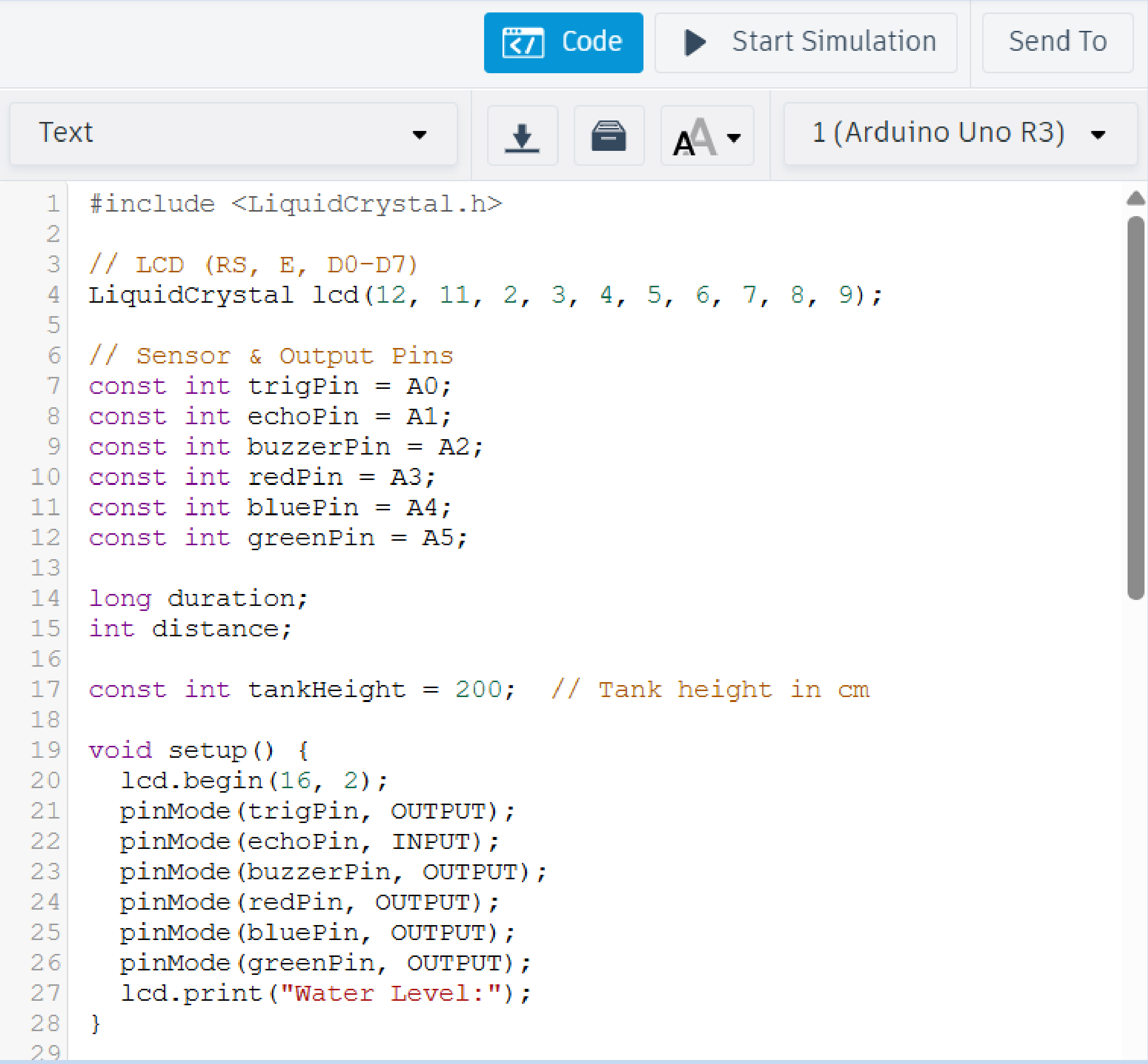
Implementation Details

Using Tinkercad, we designed and simulated a water level monitor. The ultrasonic sensor measures the distance between the sensor and water surface. This value is converted to a percentage to reflect the water level. An I2C LCD shows this percentage live. If the level falls below or rises above thresholds, an RGB LED changes color and a buzzer activates accordingly.

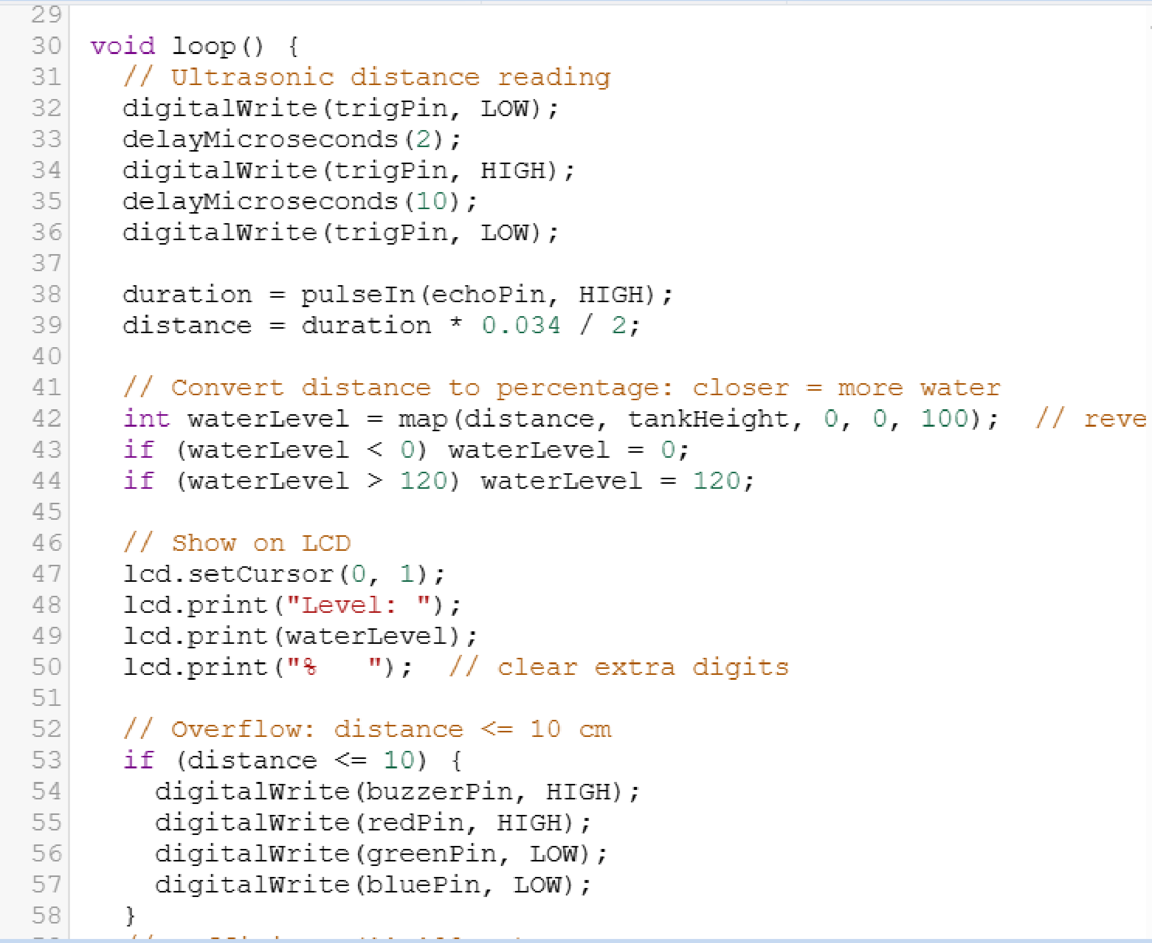
Circuit Diagram:



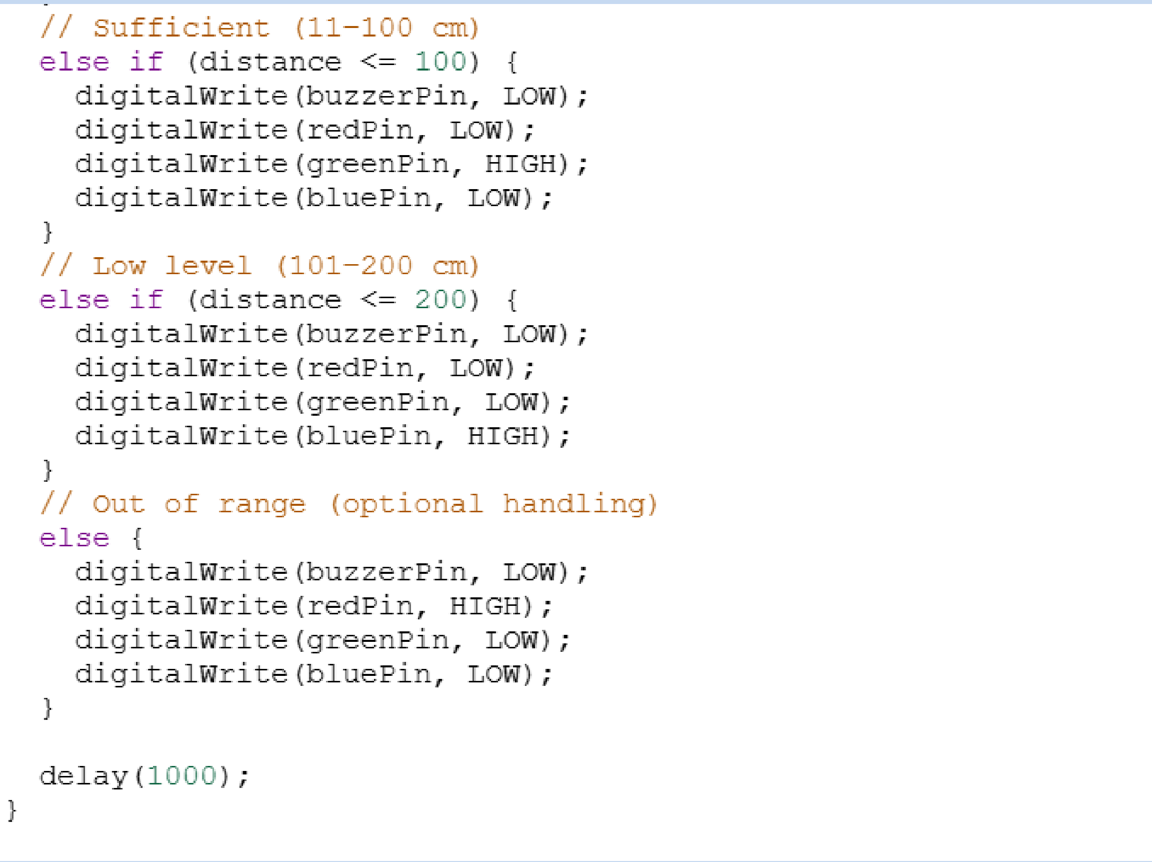
Arduino Code:



```
#include <LiquidCrystal.h>

// LCD (RS, E, D0-D7)
LiquidCrystal lcd(12, 11, 2, 3, 4, 5, 6, 7, 8, 9);

// Sensor & Output Pins
const int trigPin = A0;
const int echoPin = A1;
const int buzzerPin = A2;
const int redPin = A3;
const int bluePin = A4;
const int greenPin = A5;

long duration;
int distance;

const int tankHeight = 200;  // Tank height in cm

void setup() {
  lcd.begin(16, 2);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(buzzerPin, OUTPUT);
  pinMode(redPin, OUTPUT);
  pinMode(bluePin, OUTPUT);
  pinMode(greenPin, OUTPUT);
  lcd.print("Water Level:");
}
```

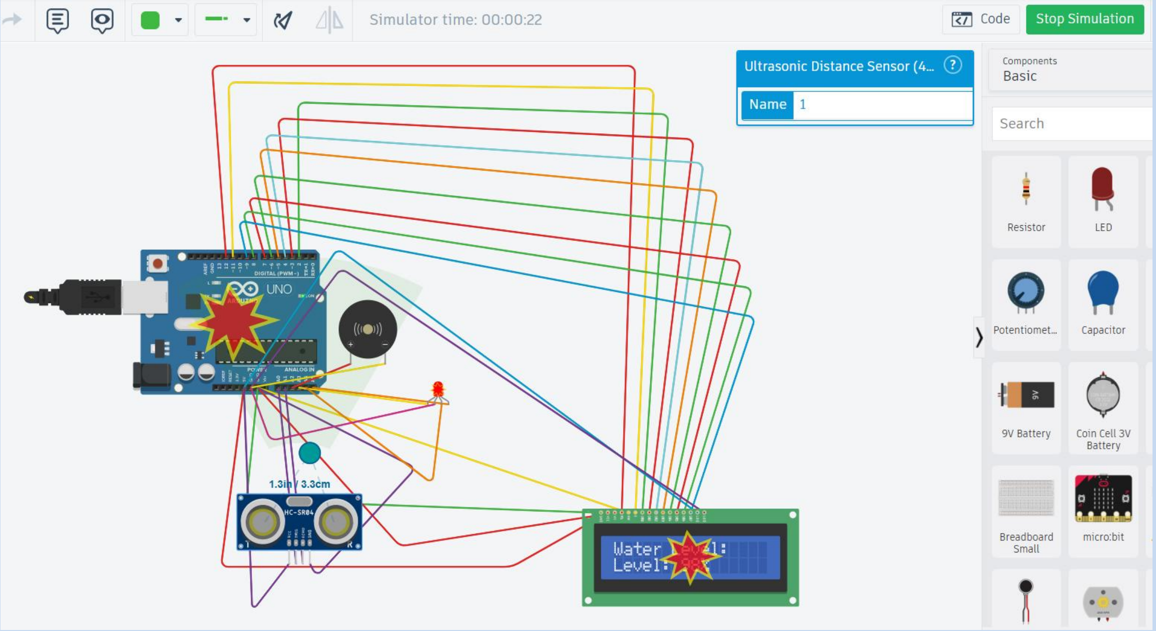


```cpp

void loop() {
  // Ultrasonic distance reading
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

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

  // Convert distance to percentage: closer = more water
  int waterLevel = map(distance, tankHeight, 0, 0, 100);  // reve
  if (waterLevel < 0) waterLevel = 0;
  if (waterLevel > 120) waterLevel = 120;

  // Show on LCD
  lcd.setCursor(0, 1);
  lcd.print("Level: ");
  lcd.print(waterLevel);
  lcd.print("%   ");  // clear extra digits

  // Overflow: distance <= 10 cm
  if (distance <= 10) {
    digitalWrite(buzzerPin, HIGH);
    digitalWrite(redPin, HIGH);
    digitalWrite(greenPin, LOW);
    digitalWrite(bluePin, LOW);
  }
```



```
  // Sufficient (11-100 cm)
  else if (distance <= 100) {
    digitalWrite(buzzerPin, LOW);
    digitalWrite(redPin, LOW);
    digitalWrite(greenPin, HIGH);
    digitalWrite(bluePin, LOW);
  }
  // Low level (101-200 cm)
  else if (distance <= 200) {
    digitalWrite(buzzerPin, LOW);
    digitalWrite(redPin, LOW);
    digitalWrite(greenPin, LOW);
    digitalWrite(bluePin, HIGH);
  }
  // Out of range (optional handling)
  else {
    digitalWrite(buzzerPin, LOW);
    digitalWrite(redPin, HIGH);
    digitalWrite(greenPin, LOW);
    digitalWrite(bluePin, LOW);
  }

  delay(1000);
}
```

SIMULATION OF THE CODE:



Simulator time: 00:00:22

Ultrasonic Distance Sensor (4...

Name 1

Reflection

Madduri Dashwanth – Arduino Programming & Logic
Dashwanth developed the Arduino code for accurately monitoring water levels and triggering buzzer alerts. He also calibrated sensor readings for different tank sizes.

Bozza Prakash Abhinay – Hardware & Sensor Design
Abhinay handled the physical circuit setup in Tinkercad, connecting the ultrasonic sensor, RGB LED, and LCD to the Arduino board. He ensured all hardware inputs and outputs functioned correctly.

K Author Abhikhyath Gundi – Project Support & Integration

Abhikhyath assisted in integrating various modules and ensured the display, buzzer, and RGB LED were synchronized. He also supported the team in debugging issues during simulation and testing.

Kothith Pappala – System Tester

Kothith thoroughly tested the simulation, including checking sensor values and verifying buzzer and LED behavior at different water levels. His testing ensured the system responded reliably to all expected scenarios.

Overall Learning Experience

This IoT project helped us learn how embedded electronics and Arduino programming can solve real-world problems. It taught us the value of alert systems in automation and introduced us to simulation tools like Tinkercad. Through hands-on experimentation, we gained confidence in working with sensors, displays, and logic flows in IoT environments.

-Thank You-