Naan Mudhalvan Phase 3 Assessment

Course Name : Internet of Things

Project Title : Smart Water Management

Team Name : Tech Enthusiast

Team Members :

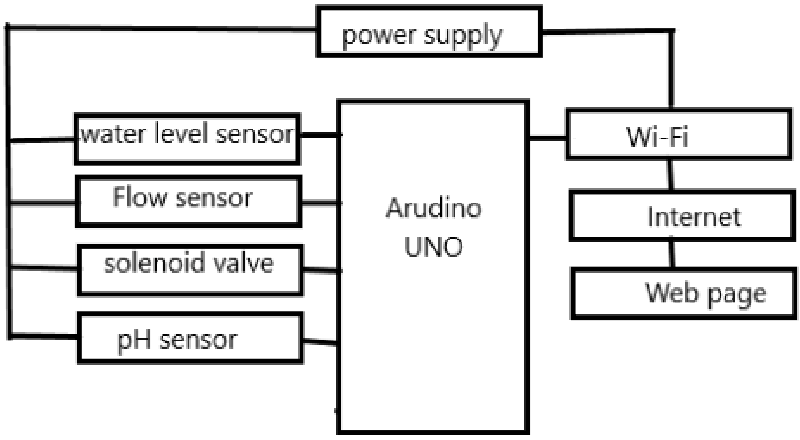
Name	Register number	Naan Mudhalvan ID
Senthilnathan J R	721221106093	au721221106093
Vimal N	721221106119	au721221106119
Soundar S	721221106098	au721221106098

Components Used:

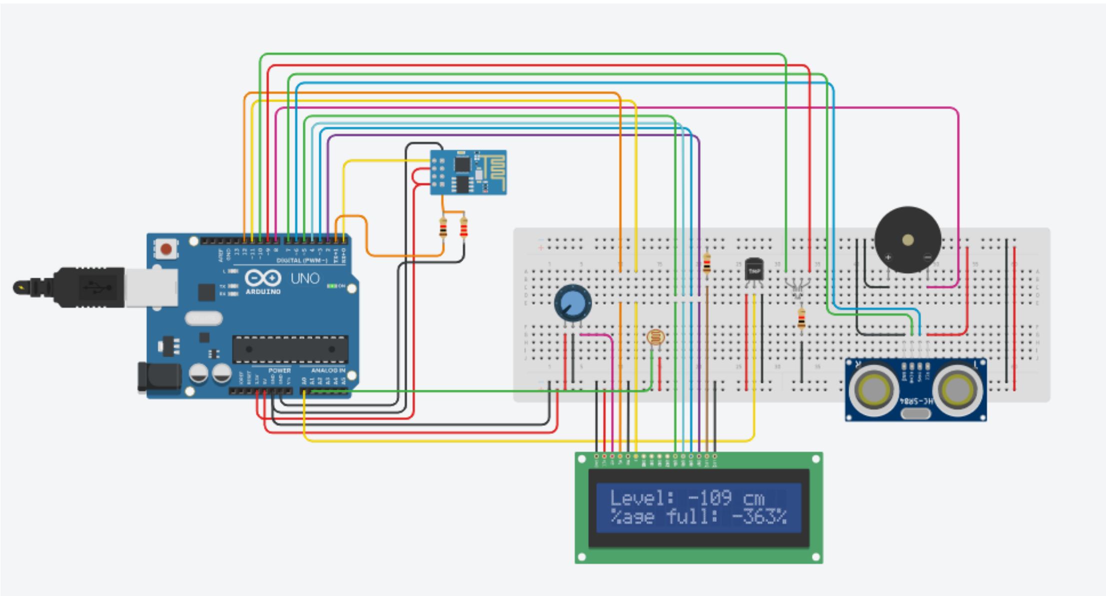
Component Name	Specifications	Quantity
Arduino Uno	-	1
Wi-Fi Module	ESP8266	1
Resistor	2.2 kΩ ,1kΩ,10kΩ	2,1,1
LED RGB	-	1
Ultrasonic sensor	-	1
Piezo	-	1
Temperature Sensor	TMP36	1
Potentiometer	250 Ω	1
Photoresistor	-	1
LCD display	-	1

Simulator used: Tinkercad

Block diagram:



Simulation:



Program:

```
#include <LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

int redLed = 9;
```

```
int greenLed = 10;
int trigPin = 6;
int echoPin = 7;
long duration;
int dist;
int percentfull;
int h = 30;
const int piezoPin = 8;
void setup() {
  pinMode(9, OUTPUT);
  pinMode(10, OUTPUT);
  lcd.begin(16, 2);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  Serial.begin(9600);
}
void loop() {
  lcd.setCursor(0, 0);
  int value = analogRead(A1);
  dist = findDist();
 percentfull = (h - dist) * 100 / h;
  lcd.setCursor(0, 0);
  lcd.print("Level: ");
  lcd.print(h - dist);
  lcd.print(" cm");
```

```
delay(10);
 lcd.setCursor(0, 1);
 lcd.print("%age full: ");
 lcd.print(percentfull);
 lcd.print("%");
 delay(1000);
if (percentfull >= 90) {
  digitalWrite(redLed, HIGH);
  digitalWrite(greenLed, LOW);
  tone(piezoPin, 500, 500);
  delay(200);
 } else {
  digitalWrite(redLed, LOW);
  digitalWrite(greenLed, HIGH);
  delay(500);
  digitalWrite(greenLed, LOW);
  delay(200);
 }
 delay(5000);
 lcd.clear();
}
int findDist() {
 digitalWrite(trigPin, LOW);
 delayMicroseconds(2);
 digitalWrite(trigPin, HIGH);
```

```
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  int distanceCm = duration * 0.034 / 2;
  return distanceCm;
}
```

Code working:

1. Library Inclusion - The code begins by including the "LiquidCrystal" library, which is used for interfacing with an LCD screen.

2. Global Variables and Pin Declarations - Several global variables are declared.LiquidCrystal lcd(12, 11, 5, 4, 3, 2)`: This initializes the LCD object with specific pin connections.These variables store the pin numbers for the ultrasonic distance sensor's trigger and echo pins.

3. Setup Function- The `setup()` function is executed once when the Arduino board is powered on or reset.It initializes various components.pinMode(9OUTPUT)pinMode(echoPin, INPUT)`: Sets `echoPin` as an input for receiving the ultrasonic sensor's echo signal**.**

4. Loop Function - The `loop()` function is the main part of the code, and it runs continuously.It performs the following tasks:

- Sets the cursor on the LCD to the top row.Reads an analog value from pin A1, which is commented as reading turbidity and temperature sensors (you should replace this with actual sensor reading code).If it's greater than or equal to 90%, it turns on the red LED, turns off the green LED, and plays a 500Hz tone on the piezo buzzer**.**

5. findDist() Function - This function measures the distance using an ultrasonic sensor. It follows these steps:

- Sets the trigger pin LOW, waits for 2 microseconds, and then sets it HIGH for 10 microseconds to trigger the sensor.- Sets the trigger pin LOW again.Measures the duration of the echo pulse using the `pulseIn()` function, which returns the time it takes for the echo to return.

Working of Smart Water Management:

Sensors and Data Collection:

The system relies on various types of sensors to collect data. These sensors can include flow meters, pressure sensors, level sensors, water quality sensors, and temperature sensors. The specific sensors used depend on the system's goals.

Data Transmission:

Collected data is transmitted to a central monitoring unit using various communication technologies. This can include wired connections (e.g., Ethernet or Modbus) or wireless technologies like Wi-Fi, cellular, LoRa, or Zigbee.

Data Processing and Analysis:

The central monitoring unit or cloud-based server processes the incoming data in real-time.Data analysis algorithms can detect anomalies, trends, and patterns in water usage, quality, and distribution. For example, the system might detect a sudden drop in water pressure that indicates a leak or an increase in water turbidity suggesting contamination.

User Interface and Alerts:

The processed data is presented through a user interface, which can be a web-based dashboard, desktop application.Users, such as water utility operators or homeowners, can monitor the status of the water system in real-time, view historical data, and receive alerts and notifications for important events.

Automation and Control:

In addition to monitoring, some smart water systems have control capabilities. They can remotely operate valves, pumps, and other components to optimize water distribution, reduce waste, and respond to detected issues.

Data Storage and Historical Analysis:

The system stores historical data, allowing for long-term analysis and trend identification. This data can be useful for planning, maintenance, and compliance reporting.

Reporting and Decision Support:

The system generates reports and provides decision support tools for operators and administrators. These reports can help in making informed decisions related to water management, maintenance, and resource allocation.

Conclusion:

In summary, this code is designed to monitor the water level in a container, display it on an LCD screen, and provide visual and audible alerts based on the water level. It uses an ultrasonic sensor to measure the distance to the water surface, calculates the percentage fullness, and provides feedback through LEDs and a piezo buzzer.