Technical Logics

Step 1: Include Libraries and Initialize LCD:

#include <Wire.h>

#include <LiquidCrystal_I2C.h>

- The Wire.h library is used to communicate with I2C devices like the LCD.
- The LiquidCrystal_I2C.h library controls the I2C-based LCD.

Step 2: Create LCD Object:

LiquidCrystal_I2C lcd(0x27, 16, 2);

- Creates an LCD object named lcd with the address 0x27 (common for I2C LCDs), with 16 columns and 2 rows.

Step 3: Define Pins and Tank Heigh:

const int trigPin = 9;

const int echoPin = 10;

const int greenLED = 3;

const int yellowLED = 4;

const int redLED = 5;

const int buzzer = 6;

const int tankHeight = 100;

- **Ultrasonic Sensor Pins:**
  - trigPin (Trigger pin) is connected to Pin 9.
  - echoPin (Echo pin) is connected to Pin 10.
- **LED Pins:**

    - greenLED is connected to Pin 3.
    - yellowLED is connected to Pin 4.
    - redLED is connected to Pin 5.
- **Buzzer Pin:**
    - buzzer is connected to Pin 6.
- **Tank Height:**
    - Tank Height is set to 100 cm, representing the height of the water tank.

Step 4: Setup Function (Executed Once)

```
void setup() {
  Serial.begin(9600);  // Start Serial Communication
  pinMode(trigPin, OUTPUT);  // Set trigPin as OUTPUT
  pinMode(echoPin, INPUT);   // Set echoPin as INPUT
  pinMode(greenLED, OUTPUT); // Set greenLED as OUTPUT
  pinMode(yellowLED, OUTPUT); // Set yellowLED as OUTPUT
  pinMode(redLED, OUTPUT);    // Set redLED as OUTPUT
  pinMode(buzzer, OUTPUT);    // Set buzzer as OUTPUT
  lcd.init();          // Initialize the LCD
  lcd.backlight();     // Turn on LCD backlight
  lcd.setCursor(0, 0);  // Set cursor to the first row, first column
  lcd.print("Water Level:"); // Print initial message on the LCD
}
```

Initialization Tasks:

1. Serial Communication is started for debugging.

2. Pins are configured as input/output.
3. LCD is initialized, and a message "Water Level:" is displayed.

Step 5: Loop Function (Executed Repeatedly)

The loop() function continuously measures the water level and updates the display, LEDs, and buzzer based on conditions.

Step 5.1: Send Trigger Pulse to Ultrasonic Sensor

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

- Sends a trigger pulse to the ultrasonic sensor to start distance measurement.

Step 5.2: Measure Echo Pulse Duration

```
long duration = pulseIn(echoPin, HIGH);
```

- pulseIn() measures the time (in microseconds) that the echo pin stays HIGH, which is proportional to the distance.

Step 5.3: Calculate Distance and Water Level

```
long distance = duration * 0.034 / 2;
long waterLevel = tankHeight - distance;
```

- Formula: distance = (duration * 0.034) / 2
    - Converts duration to distance in cm (speed of sound = 343 m/s).
- Water Level Calculation:

- Subtracts the measured distance from the tankHeight to calculate the current water level.

Step 5.4: Display Water Level on LCD

```
lcd.setCursor(0, 1);
if (waterLevel > 0 && waterLevel <= tankHeight) {
  lcd.print("Level: ");
  lcd.print(waterLevel);
  lcd.print(" cm   ");
} else if (waterLevel <= 0) {
  lcd.print("Level: Empty   ");
} else {
  lcd.print("Out of Range   ");
}
```

- The LCD displays the current water level in centimeters.
  - If the water level is between 0 and 100 cm, it shows the level.
  - If the water level is less than 0 cm, it shows "Empty".
  - If the value exceeds the tank's height, it shows "Out of Range".

Step 5.5: Control LEDs and Buzzer Based on Water Level

Condition 1: Low Water Level (< 30 cm)

```
if (waterLevel < 30) {
  digitalWrite(greenLED, LOW);
  digitalWrite(yellowLED, LOW);
  digitalWrite(redLED, HIGH);
  tone(buzzer, 1000);  // Activate buzzer with 1000 Hz tone
```

```
}
```

- Red LED turns ON, and the buzzer sounds to indicate low water level.

Condition 2: Medium Water Level (30-70 cm)

```
else if (waterLevel >= 30 && waterLevel <= 70) {
  digitalWrite(greenLED, LOW);
  digitalWrite(yellowLED, HIGH);
  digitalWrite(redLED, LOW);
  noTone(buzzer);  // Turn off buzzer
}
```

- Yellow LED turns ON, and the buzzer is OFF.

Condition 3: High Water Level (> 70 cm)

```
else if (waterLevel > 70) {
  digitalWrite(greenLED, HIGH);
  digitalWrite(yellowLED, LOW);
  digitalWrite(redLED, LOW);
  tone(buzzer, 500);  // Activate buzzer with 500 Hz tone
}
```

- Green LED turns ON, and the buzzer sounds with a lower frequency tone.

Step 5.6: Print Water Level to Serial Monitor

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

- Prints the water level to the Serial Monitor for debugging purposes.

Step 5.7: Delay for Stability

delay(500);

- Adds a 500 ms delay to stabilize the sensor readings.