Real Time Water Quality Monitoring & Analysis System

TEAM MEMBERS

- **ALDRICH MARIAN A**
- **BARATH M**
- **ASHISH C**
- **AKASH SAI**

AIM:

To design and implement a real-time water quality monitoring and analysis system using the ESP8266 microcontroller, turbidity sensor, OLED display, LED, and supporting components to detect and alert when the water turbidity exceeds a safe level.

COMPONENTS REQUIRED:

1. ESP-8266
2. Turbidity Sensor
3. OLED
4. LED
5. Resistor
6. Breadboard

ESP-8266:

A compact Wi-Fi-enabled microcontroller used for controlling sensors and processing data. It can send and receive information wirelessly, making it ideal for IoT applications like water quality monitoring.

Turbidity Sensor:

A sensor that detects the presence of suspended particles in water, giving an indication of water clarity. It outputs an analog signal that the microcontroller reads and processes to monitor water quality.

OLED:

An energy-efficient screen that visually shows information such as turbidity levels and alerts. It communicates with the microcontroller using the I2C protocol and requires minimal power.

LED:

A light-emitting component that signals changes in water quality. When turbidity crosses a set threshold, the LED glows to alert users of possible contamination.

Resistor:

A passive component used to limit the current flowing to the LED, ensuring that it operates safely without overheating or burning out.

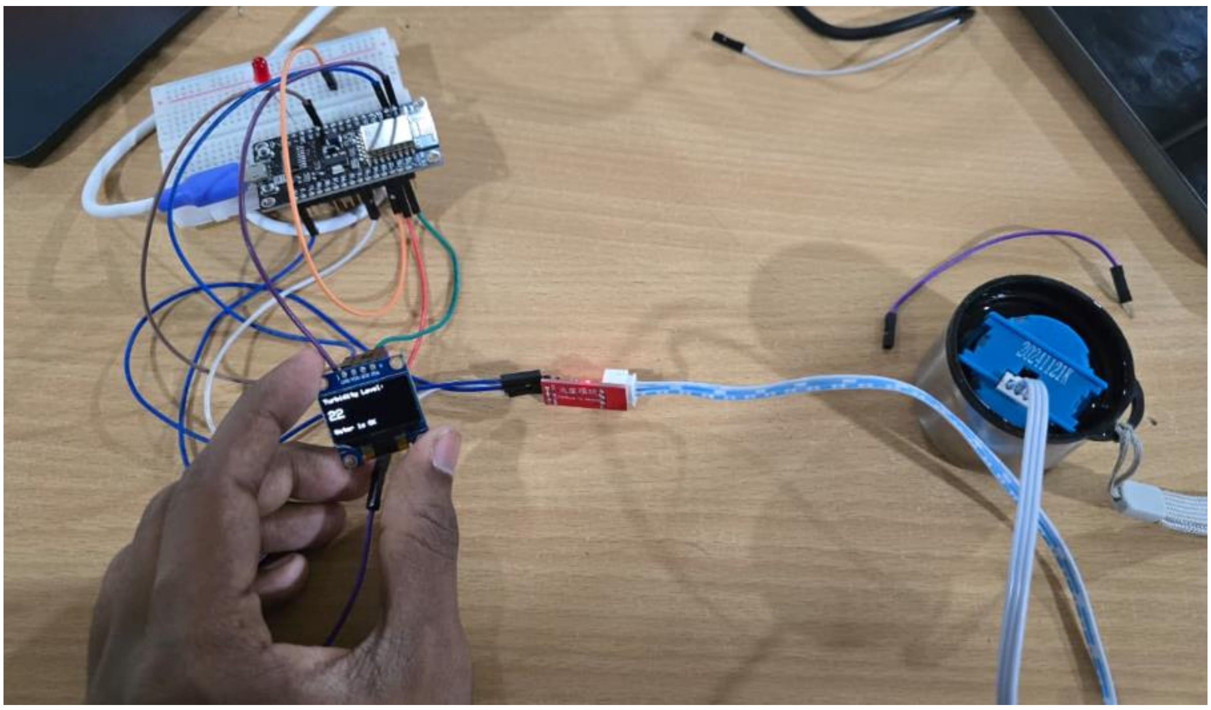
Breadboard:

A reusable prototyping board that allows electronic components to be connected without soldering. It is mainly used for testing, learning, and quick modification of circuits

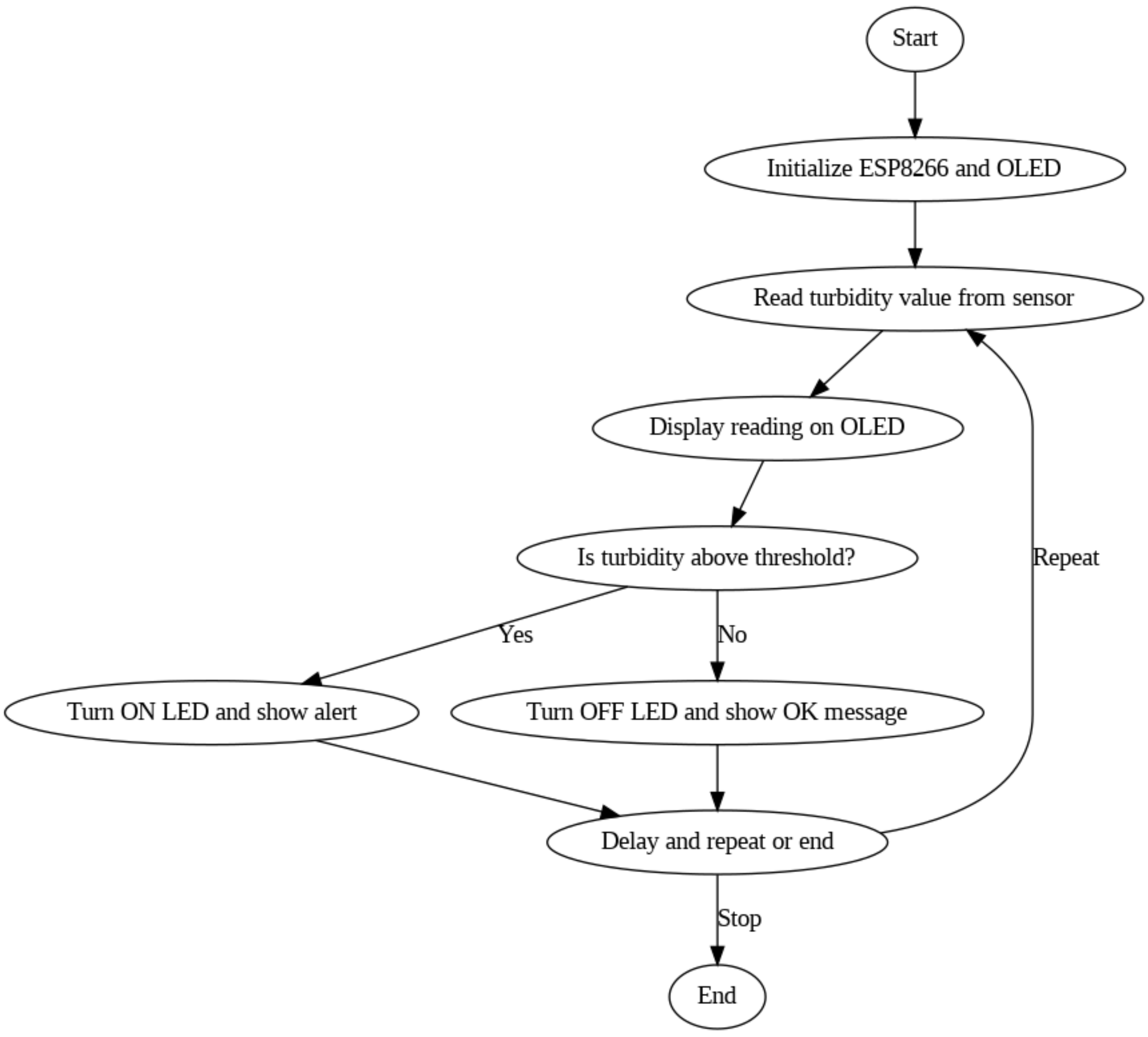
Pin Table:

Component	Pin on Component	ESP8266 Pin	Description
Turbidity Sensor VCC	VCC	3.3V	Power supply for the sensor
Turbidity Sensor GND	GND	GND	Ground connection
Turbidity Sensor A0	Analog output	A0	Reads turbidity data via ADC
OLED Display VCC	VCC	3.3V	Power supply for OLED
OLED Display GND	GND	GND	Ground connection
OLED Display SDA	Data line	GPIO4 (D2)	I2C data communication
OLED Display SCL	Clock line	GPIO5 (D1)	I2C clock communication
LED anode	+ lead	GPIO0 (D3)	Turns ON/OFF depending on turbidity alert
LED cathode	- lead	GND	Ground for LED
Resistor (330Ω)	Series between GPIO and LED anode	Between GPIO0 and LED anode	Limits current to protect the LED

Circuit Connection:



Flowchart:


Start
Initialize ESP8266 and OLED
Read turbidity value from sensor
Display reading on OLED
Is turbidity above threshold?
Yes
No
Turn ON LED and show alert
Turn OFF LED and show OK message
Delay and repeat or end
Repeat
Stop
End

Coding:

```
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

// OLED display settings
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
#define OLED_RESET    -1
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire,
OLED_RESET);

// Pin definitions
const int turbidityPin = A0;   // Analog input from turbidity sensor
const int ledPin = D3;         // Digital output to LED (through resistor)

// Threshold for turbidity alert
const int turbidityThreshold = 400; // Adjust this based on calibration

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

  // Initialize the LED pin
  pinMode(ledPin, OUTPUT);
  digitalWrite(ledPin, LOW);

  // Initialize the OLED
  if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
    Serial.println("OLED initialization failed!");
    while(true);
  }
  display.clearDisplay();
  display.setTextSize(1);
  display.setTextColor(SSD1306_WHITE);
  display.setCursor(0,0);
  display.println("Water Quality Monitor");
  display.display();
  delay(2000);
}

void loop() {
  // Read turbidity value from sensor
  int turbidityValue = analogRead(turbidityPin);

  // Display the turbidity value on OLED
  display.clearDisplay();
  display.setCursor(0,0);
  display.println("Turbidity Level:");
  display.setTextSize(2);
  display.setCursor(0,20);
  display.print(turbidityValue);
  display.setTextSize(1);

  // Check if turbidity exceeds threshold
  if(turbidityValue > turbidityThreshold) {
    digitalWrite(ledPin, HIGH); // Turn on LED
    display.setCursor(0,50);
    display.println("ALERT! High Turbidity");
  } else {
    digitalWrite(ledPin, LOW); // Turn off LED
    display.setCursor(0,50);
    display.println("Water is OK");
  }

  display.display();
  delay(1000); // Update every 1 second
}
```

Execution: