Water Quality Monitoring System

Team Name: LogiK Jets

Task Force:

Lakshya Sharma (Ideator)
Geetansh Kumawat
Jayesh Mangal
Jasmeet Kaur (Leader)
Kritika Sharma

Affiliation:

JECRC Foundation (Department of ECE)

Section: H

Introduction

Water is essential for life, and ensuring its purity is crucial for human health, agriculture, and industrial applications. Traditional water quality testing methods are time-consuming and require laboratory analysis. To overcome this challenge, we have developed a Water Quality Monitoring System using a TDS (Total Dissolved Solids) sensor integrated with IoT technology.

This system provides real-time monitoring of water quality by measuring TDS levels, which indicate the presence of dissolved substances such as salts, minerals, and contaminants. High TDS levels can affect water taste, safety, and usability. By using IoT, our system can continuously track water quality and send alerts if unsafe conditions are detected. This project is designed to be used in households, industries, agriculture, and environmental monitoring, ensuring access to clean and safe water.

Hardware Required

- ESP32
- TDS Sensor
- I2C Converter
- 16×2 LCD Display
- Breadboard
- Jumper Wires
- Water Containers

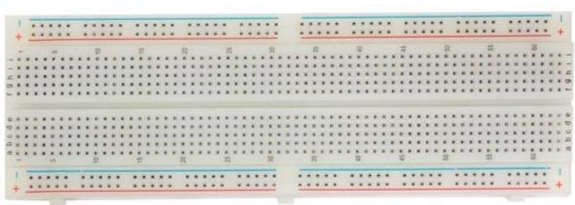
ESP 32

TDS Sensor

I2C Converter

LCD Display

Jumper Wires



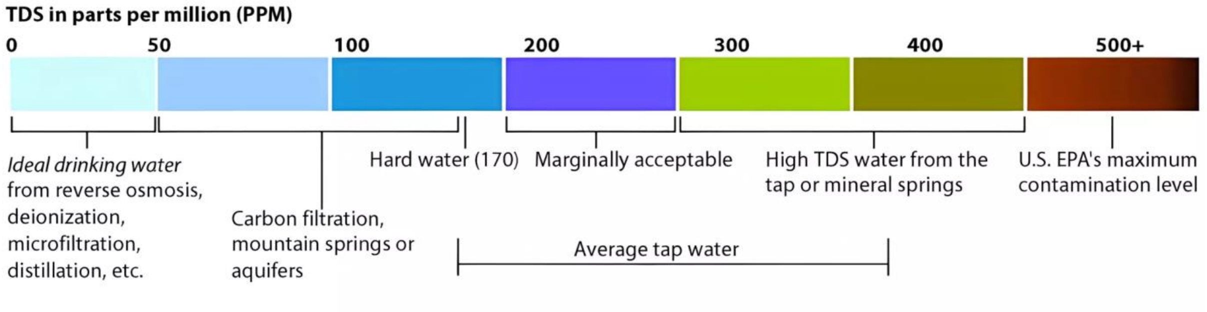
Breadboard

TDS Sensor

A TDS (Total Dissolved Solids) Sensor measures the total amount of dissolved solids in a liquid, typically water. Dissolved solids include inorganic salts and small organic substances. The measurement is expressed in parts per million (ppm) or milligrams per liter(mg/L).

TDS is an essential parameter for assessing water quality. A high TDS value may indicate contamination, while a very low value might suggest a lack of essential minerals.

Suitable for domestic water, hydroponics, and other water quality testing applications, it supports a wide input voltage range (3.3–5.5V) and outputs an analog signal (0–2.3V), making it compatible with both 3.3V and 5V control systems.



Key Features

- Uses an AC signal to prevent probe polarization, increasing probe lifespan and signal stability.
- Waterproof TDS probe supports extended immersion for consistent measurements.

Specifications

- Input Voltage: 3.3–5.5V
- Output Voltage: 0–2.3V
- Current: 3–6mA
- TDS Range: 0–1000 ppm
- Accuracy: ±10% FS (at 25°C)
- Probe: 2-needle, waterproof

Usage Tips

- Temperature Limit: Do not use in water above 55°C.
- Placement: Avoid placing the probe near the container's edge to maintain accuracy.
- Waterproofing: Only the probe head and cable are waterproof; the connector and transmitter board are not.

Components of TDS Sensor

- Probe/Electrodes: Submersible electrodes measure the conductivity of the solution.
- Signal Conditioning Circuit: Converts the electrical conductivity into a TDS value.
- Output: Generally, provides an analog voltage proportional to the TDS.

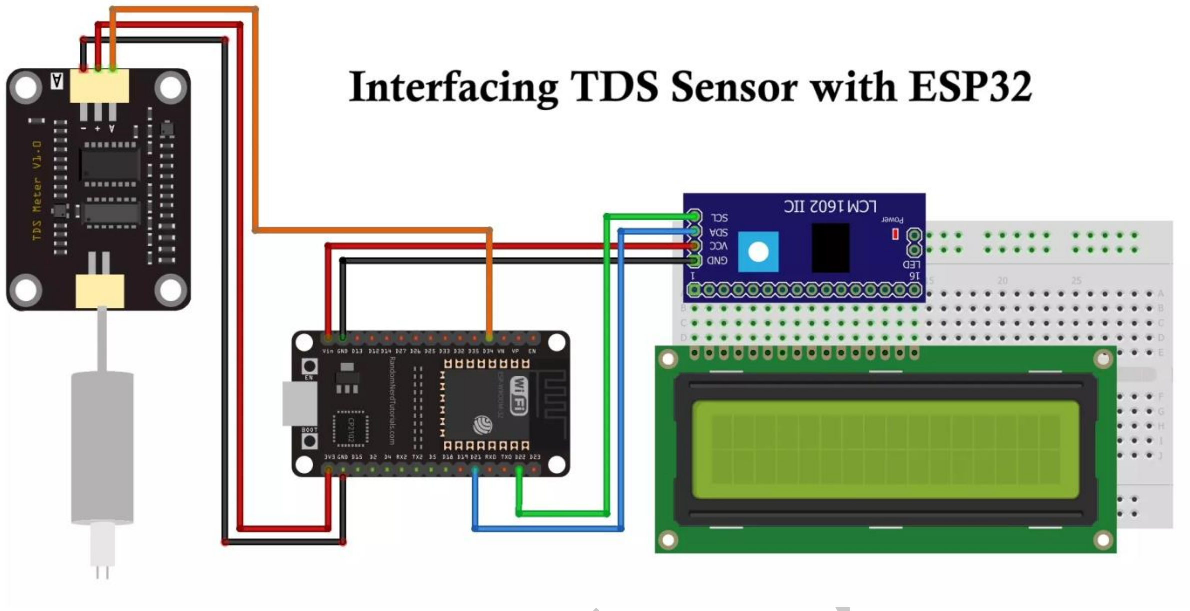
Working Principle of TDS Sensor

- Conductivity Measurement: TDS sensors measure the electrical conductivity (EC) of water. Dissolved solids like salts, minerals, and ions increase conductivity.
- Conversion to TDS: The measured EC value is converted into TDS using a conversion factor. The factor depends on the specific solution but is typically around 0.5–0.8.

Applications of TDS Sensors

- Water Purification Systems: Monitoring drinking water quality to ensure safety.
- Aquaculture: Ensuring suitable water conditions for aquatic life.
- Agriculture: Managing water quality for irrigation in hydroponics.
- Industrial Processes: Maintaining appropriate TDS levels for machinery and processes.

Circuit Diagram



Connections

- Connect the TDS probe to the TDS sensor module.
- Connect the module’s analog output to an analog pin on the ESP32 (e.g., GPIO34).
- Provide power to the TDS module from the ESP32’s 3.3V pin.
- Connect the ground of the TDS module to the ESP32’s GND.
- Connect 16×2 LCD module to ESP32 using I2C converter module as shown in the circuit diagram.

Features of Water Quality Monitoring System

- Display Module: OLED or LCD to show TDS values locally.
- IoT Integration: Sending TDS data to a cloud server or mobile app using ESP32’s Wi-Fi capabilities.
- Alerts: You can program the ESP32 to trigger alerts if TDS exceeds a specific threshold.

Software Implementation

Follow these steps to program the ESP32:

1. Install the Required Libraries

- Install the ESP32 board package in Arduino IDE.
- Install libraries for display modules if using.
- Got to blynk.io create you application as shown in the video.
- Add your credentials to code and upload to the ESP32 board.

2. Code for Reading TDS Sensor

```
#define BLYNK_TEMPLATE_ID "TMPL3pOAc0TpX"
#define BLYNK_TEMPLATE_NAME "Water Quality Monitoring"
#define BLYNK_AUTH_TOKEN "5i7WnkJmsuGEC52vcxIGAWw1CO7fEwKQ"

// Comment this out to disable prints and save space
#define BLYNK_PRINT Serial

#include <BlynkSimpleEsp32.h>
#include <WiFi.h>
#include <WiFiClient.h>
#include <Wire.h>
#include <LiquidCrystal_PCF8574.h>

// LCD configuration
LiquidCrystal_PCF8574 lcd(0x27); // Use correct I2C address (0x27 or 0x3F)

// WiFi credentials
char ssid[] = "Galaxy A03 Core0945";
char pass[] = "jsqe8129";

char auth[] = BLYNK_AUTH_TOKEN;

namespace pin {
    const byte tds_sensor = 34;
}

namespace device {
    float aref = 3.3; // Vref, this is for 3.3V compatible controller boards, for Arduino use 5.0V.
}

namespace sensor {
    float ec = 0;
    unsigned int tds = 0;
    float ecCalibration = 1;
```

```
}
void displayName(const char* firstName, const char* surname)
    {
      lcd.clear();
      lcd.setCursor(3, 0);
      lcd.print(firstName);
      lcd.setCursor(3, 1);
      lcd.print(surname);
      delay(3000);
    }

void setup() {
    Serial.begin(115200); // Debugging on hardware Serial 0
    Blynk.begin(auth, ssid, pass);

    // Initialize LCD
    Wire.begin();
    lcd.begin(16, 2); // Set as a 16x2 LCD
    lcd.setBacklight(255); // Set backlight brightness

    // Display initial message
    lcd.setCursor(3, 0);
    lcd.print("LogiK Jets");
    delay(5000);

    displayName("Lakshya", "Sharma");
    displayName("Geetansh", "Kumawat");
    displayName("Jayesh", "Mangal");
    displayName("Jasmeet", "Kaur");
    displayName("Kritika", "Sharma");

    lcd.clear();
    lcd.setCursor(1, 0);
    lcd.print("Water Quality");
    lcd.setCursor(3, 1);
    lcd.print("Monitoring   ");
    delay(2000);
    lcd.clear();
}

void loop() {
    Blynk.run();
    readTdsQuick();
    displayTdsEc();
    delay(1000);
}

void readTdsQuick() {
```

```
    // Read the raw analog value and convert to voltage
    float rawEc = analogRead(pin::tds_sensor) * device::aref / 1024.0;

    // Debugging: Print the raw analog value
    Serial.print(F("Raw Analog Value: "));
    Serial.println(rawEc);

    // Adjust this offset based on the sensor's dry reading (without immersion)
    float offset = 0.14; // Set this to the observed raw analog value in air

    // Apply calibration and offset compensation
    sensor::ec = (rawEc * sensor::ecCalibration) - offset;

    // If the EC is below zero after adjustment, set it to zero
    if (sensor::ec < 0) sensor::ec = 0;

    // Convert voltage value to TDS value using a cubic equation
    sensor::tds = (133.42 * pow(sensor::ec, 3) - 255.86 * sensor::ec * sensor::ec + 857.39 *
sensor::ec) * 0.5 / 10;

    // Debugging: Print the TDS and EC values
    Serial.print(F("TDS: "));
    Serial.println(sensor::tds);
    Serial.print(F("EC: "));
    Serial.println(sensor::ec, 2);

    // Display values on LCD
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("TDS: ");
    lcd.setCursor(5, 0);
    lcd.print(sensor::tds);

    lcd.setCursor(0, 1);
    lcd.print("EC: ");
    lcd.setCursor(5, 1);
    lcd.print(sensor::ec, 2);

    // Send data to Blynk virtual pins
    Blynk.virtualWrite(V0, sensor::tds);
    Blynk.virtualWrite(V1, sensor::ec);
}
void displayTdsEc()
{
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("TDS: ");
  lcd.setCursor(5, 0);
```

```
  lcd.print(sensor::tds);

  lcd.setCursor(0, 1);
  lcd.print("EC: ");
  lcd.setCursor(5, 1);
  lcd.print(sensor::ec, 2);
}
```

Steps to run the program

- Connect the ESP32 to your computer via USB.
- Open Arduino IDE and load the above code.
- Select the correct board (ESP32 Dev Module) and port in Tools.
- Upload the code.
- Open the blynk.io application to see the TDS readings in ppm.

Calibration

To ensure accurate readings:

- Use a TDS calibration solution of known ppm (e.g., 200 ppm).
- Adjust the calibration factor in the code based on the observed readings until it matches the known TDS value.

Tips for accurate readings

- Ensure the probe is clean and free from debris.
- Do not immerse the probe's entire module—only the sensor tip should be in water.
- Avoid using the sensor in high-temperature or corrosive environments.
- Stabilize the water sample before taking readings for better accuracy.
- Use a proper power supply to avoid noise in the readings.

Conclusion

A TDS sensor, when interfaced with an ESP32 or Arduino, provides a versatile and efficient solution for real-time water quality monitoring. It simplifies the measurement and monitoring of TDS levels for environmental, agricultural, and industrial purposes. It is ideal for applications such as home aquariums and industrial water treatment systems. TDS sensor is useful in advanced and reliable water quality monitoring systems for a wide range of needs.