**Smart water quality monitoring system**

One of the most valuable resources on earth is water, which all living things need to survive. However, a number of factors like pollution, climatic change, and human activity affect the quality of the water miserably. Therefore, it becomes extremely essential to monitor water quality to maintain its security and quality and avoid any negative health impacts on living beings. Traditional water quality monitoring techniques are expensive and time-consuming, and they frequently need qualified specialists and experts for their look-after. Fortunately, as technology has developed, smart water quality monitoring devices have appeared, providing a quicker, more affordable, and more effective approach to check water quality. In this tutorial, we'll look at the fundamentals of smart water quality monitoring systems, including how they operate, what they are beneficial for us, and how to use them.

**What is smart water quality monitoring system?**

These systems are advanced technology-based solutions that incessantly monitor water quality parameters in real-time and provide insights into the quality of the water. They use various sensors and equipment to collect data on various parameters such as pH, temperature, dissolved oxygen, turbidity, and conductivity. This collected data is then transmitted to a central database for analysis cum evaluation.

A network of sensors that are placed throughout the water distribution system, including reservoirs, treatment facilities, and distribution networks, make up the system. These sensors are made to automatically collect data on water quality, and the data is then sent to a central database over wired or wireless networks.

To be able to provide insights and identify anomalies, the data collected by these sensors is analysed using machine learning and artificial intelligence algorithms. A quick response and intervention are made possible by the system's ability to alert water management teams in real-time if any parameters deviate from their acceptable range.

**Advantages of the system**

When compared with older monitoring systems, the smart monitor for water quality systems has plenty of benefits to offer. It delivers unchanged and immediate water quality monitoring, establishing quicker problem detection and restoration. On top of that, the system negates the need for time-consuming manual data collection and analysis, minimising the possibility of human error. Data on water quality are far more precise and accurate because the system automatically collects and analyses data

**Requirements to make smart water quality monitoring system**

1.Sensors: They are needed to measure parameters like pH, temperature, dissolved oxygen, turbidity, and conductivity. pH sensors, temperature sensors, dissolved oxygen sensors, and turbidity sensors are mainly used in most of the systems.

2.Microcontroller: A microcontroller such as the ESP32 or Arduino is used to harness and process the data from the sensors. The ESP32 has built-in WiFi facilities, making it easier to transmit data wirelessly to the central database in a short period of time without causing any delay.

3.Wireless Communication: The system needs wireless communication capabilities to transmit the data collected from the sensors to the central database without any delay. The ESP32 microcontroller can be used to transmit data wirelessly via WiFi or Bluetooth.

4.Central Database: The data collected by the sensors is transmitted to a central database for storage and analysis purposes. Tools like Apache Hadoop or Apache Spark can be used for big data storage and analysis.

5.Machine Learning Algorithms: Advanced machine learning algorithms such as Decision Trees, Random Forest, and Support Vector Machines are majorly used to analyze the data collected by the sensors and detect anomalies and limitations of the system. This step is mandatory to decide the next actions that can be taken to improve the quality of water.

6.Data Analysis Tools: Tools like Python, R, and MATLAB are used to analyse and visualize the data collected by the sensors.

Sensors, microcontrollers, wireless communication, a central database, machine learning algorithms, and data analysis tools are just a few of the components that smart water quality monitoring systems need to work properly. These elements enable the system to continuously track water quality parameters and send out instant alerts, ensuring the security and calibre of our water resources.

**Detailed working of the system**

A smart water quality monitoring system is a cutting-edge technologically-based solution that checks water quality indicators in real-time. A network of sensors placed throughout the water distribution system, including reservoirs, treatment facilities, and distribution networks, make up the system.

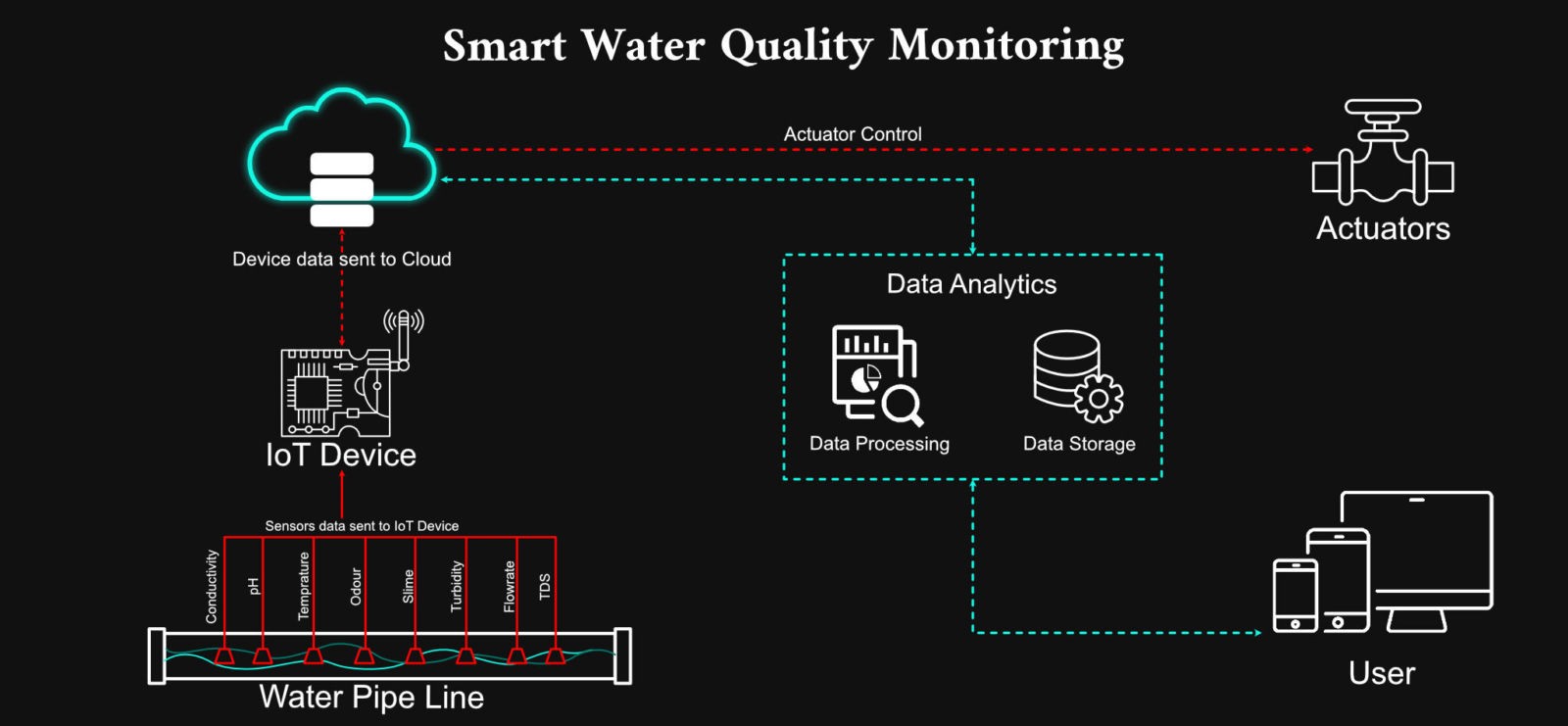
The sensors used in the system measure various water quality parameters such as pH, temperature, dissolved oxygen, turbidity, and conductivity. Each sensor is connected to a microcontroller, such as the ESP32, which collects and processes the data from the sensors. The microcontroller then transmits the data wirelessly to a central database using WiFi or Bluetooth.

The central database stores and analyses the data collected by the sensors. Tools like Apache Hadoop or Apache Spark can be used for big data storage and analysis. Machine learning algorithms such as Decision Trees, Random Forest, and Support Vector Machines can be used to analyse the data and detect anomalies.

The system can provide real-time alerts to water management teams if any parameters go out of their acceptable range, enabling rapid response and intervention. For example, if the pH level in the water drops below a specific level, the system will send an alert to the water management team. The team can then take immediate action to correct the issue before it becomes a significant problem.

In addition to providing real-time alerts, the system can also generate reports and visualizations of the data collected by the sensors. Data analysis tools such as Python, R, and MATLAB can be used to analyse and visualize the data.

On the whole, a smart water quality monitoring system provides several benefits over traditional monitoring systems. It provides continuous and real-time monitoring of water quality, enabling faster identification and resolution of issues. The system also reduces the need for manual data collection and analysis, which saves time and reduces the risk of human error. The system's automatic data collection and analysis ensure that water quality data is more accurate and reliable.



Here's a table comparing the features of traditional water quality monitoring system and the smart IoT based system

|  |  |  |
| --- | --- | --- |
| **Feature** | **Traditional System** | **Smart System** |
| Data Collection | Manual data collection | Automatic data collection |
| Data Frequency | Infrequent | Continuous |
| Data Accuracy | Limited accuracy | High accuracy |
| Data Analysis | Manual analysis | Automated analysis |
| Data Visualization | Limited | Advanced |
| Alerting | Manual alerting | Real-time alerting |
| Response Time | Slow | Rapid |
| System Maintenance | Labor-intensive | Automated |
| Cost | Lower upfront cost | Higher upfront cost |

**Prototype Building**

**Components required:**

1 ESP32 Board 1

2 Analog TDS Sensor 1

3 Temperature Sensor 1

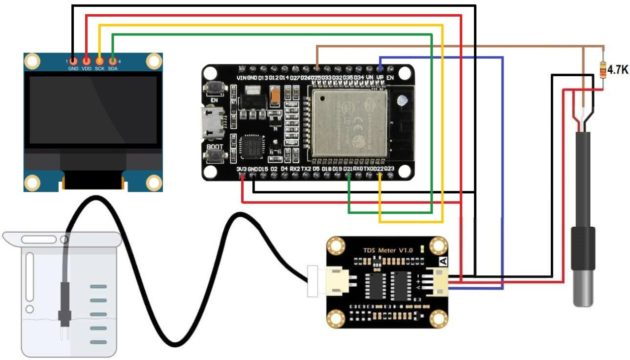
4 Resistor 4.7K 1

5 Jumper Wires 10

6 Breadboard 1

Let's now connect the ESP32 Board to the Analog TDS Sensor and DS18B20 Temperature Sensor. We are using a temperature sensor because the temperature parameter is required for the Electrical Conductivity EC Value adjustment. When the temperature rises and falls, the EC values change significantly.

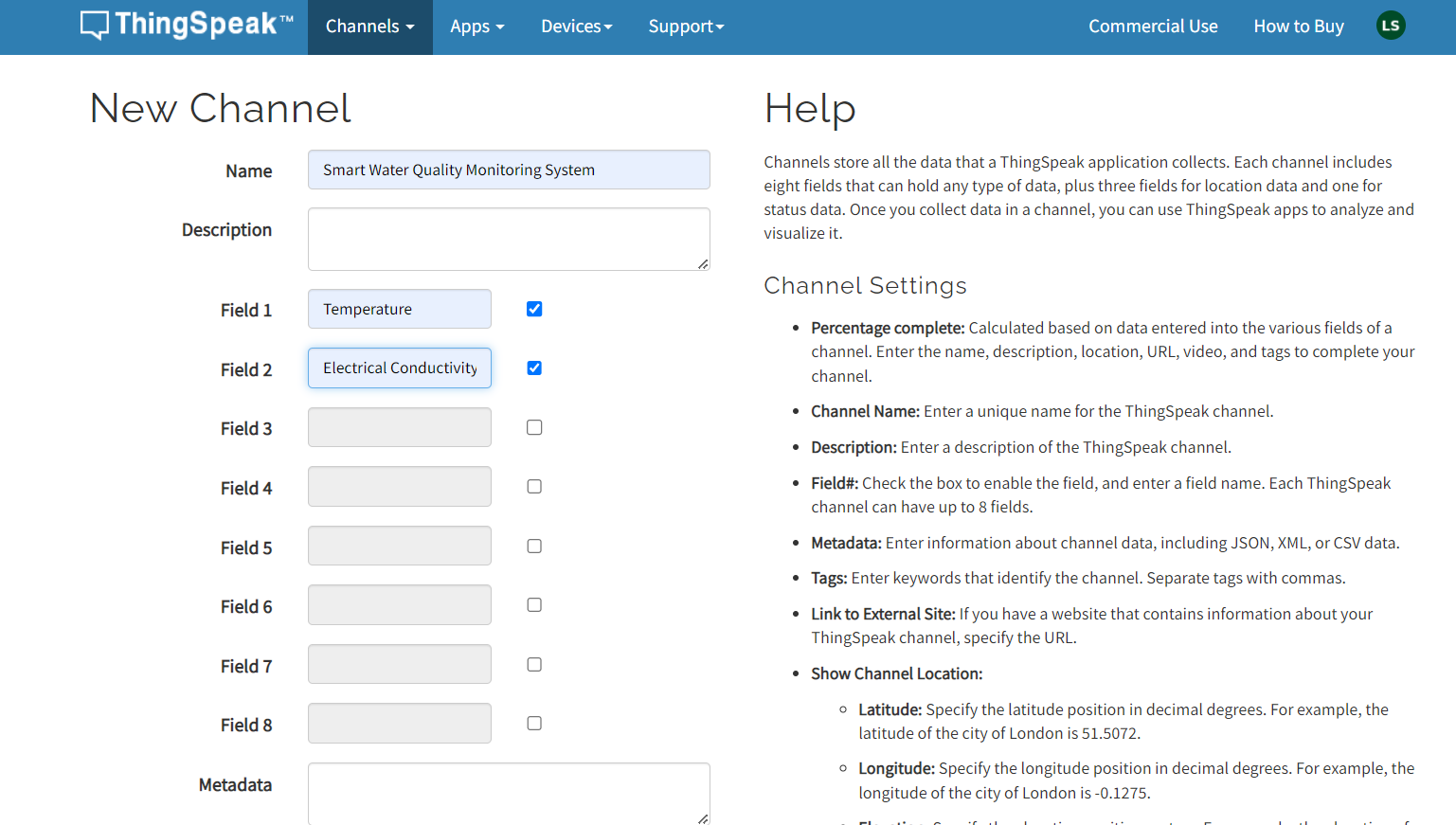
**Circuit diagram:**



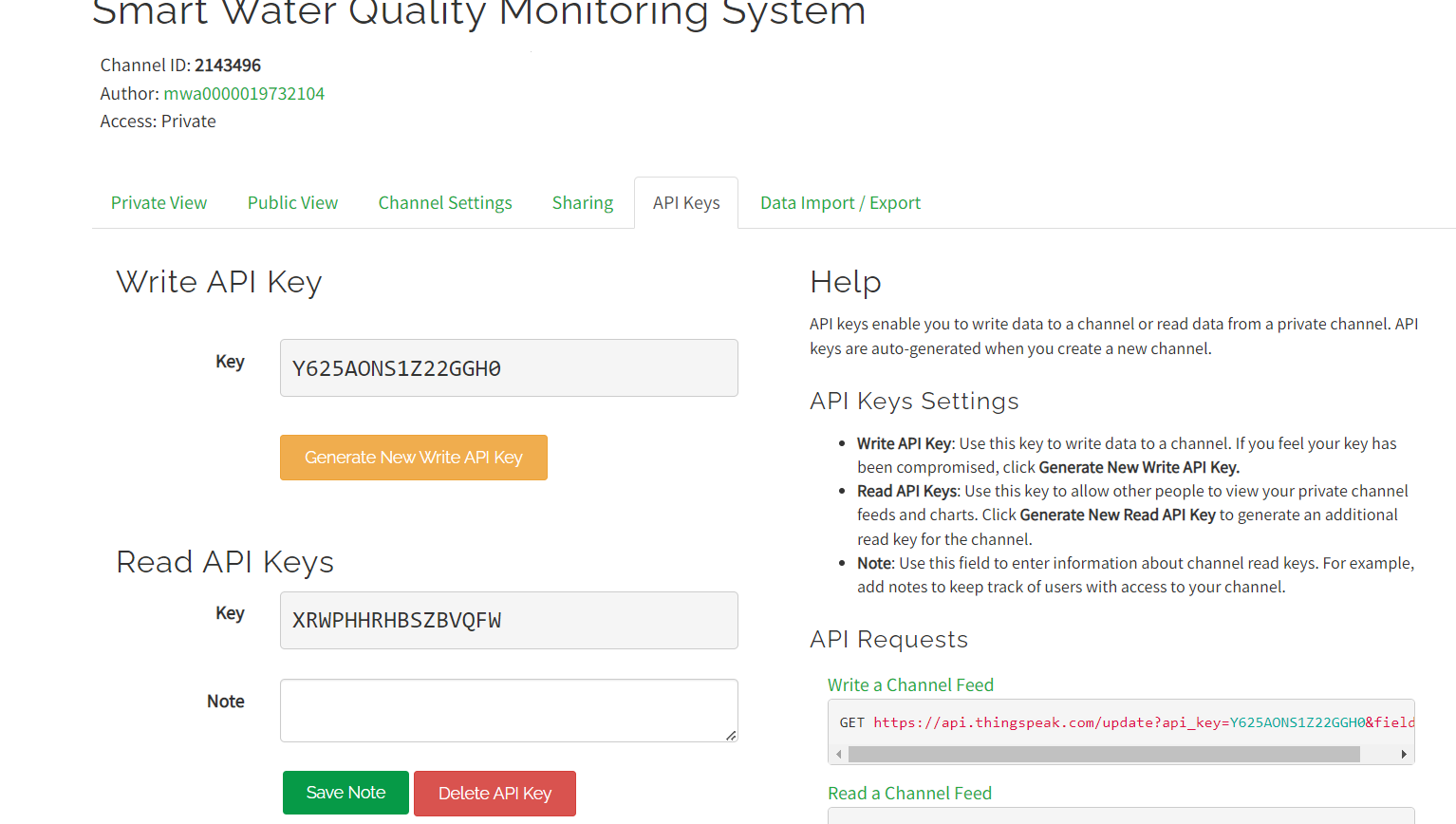
Connect the 3.3V and GND pins of the ESP32 to the VCC and GND pins of the temperature and TDS sensors, respectively. The output analogue pin of the TDS Sensor needs to be connected to the ESP32 A0 pin or the GPIO36 pin. In similar fashion, connect the DS18B20's output to the ESP32's IO14 Pin. It is crucial to connect the DS18B20 output pin and 3.3V VCC as a dependent power source and pull a 4.7K resistor.

**Putting Thingspeak Server together**

To keep track of Electrical Conductivity (EC) and temperature data on the Thingspeak server, you must first set up Thingspeak account. Check out https://thingspeak.com/ for setting up the Thingspeak Server. Create a new account or sign in with an existing one.



After that, navigate to the dashboard's API section and copy the API Key. The programming component will make use of this Key.



**Source Code/Program**

The C/C++ code for the ESP32-based Smart Water Quality Monitoring programme is created using the Arduino IDE. In this project, the code makes extensive use of libraries.

Below is a list of some of the libraries needed for this project, along with a download link. Install all of these libraries in the Arduino library folder after downloading them.

1. OneWire Library: <https://github.com/PaulStoffregen/OneWire>

2. Dallas Temperature Library: <https://github.com/milesburton/Arduino-Temperature-Control-Library>

3. ADS1015 Library: <https://github.com/adafruit/Adafruit_ADS1X15>

4. DFRobot ESP EC Library: <https://github.com/GreenPonik/DFRobot_ESP_EC_BY_GREENPONIK>

5. Adafruit GFX Library: <https://github.com/adafruit/Adafruit-GFX-Library>

6. Adafruit SSD1306 Library: <https://github.com/adafruit/Adafruit_SSD1306>

***Make sure that you add the following details before uploading the code.***

String APIKey = "\*\*\*\*\*\*"; // Enter your Write API key from ThingSpeak

const char \*SSID = "\*\*\*\*\*\*"; // replace with your wifi ssid

const char \*pass = "\*\*\*\*\*\*\*";

**Code:**

#include <Arduino.h>

#include <Wire.h>

#include <EEPROM.h>

#include <WiFi.h>

#include <OneWire.h>

#include <DallasTemperature.h>

#include <Adafruit\_ADS1015.h>

#include <DFRobot\_ESP\_EC.h>

#include <Adafruit\_GFX.h>

#include <Adafruit\_SSD1306.h>

#define SCREEN\_WIDTH 128

#define SCREEN\_HEIGHT 64

Adafruit\_SSD1306 display(SCREEN\_WIDTH, SCREEN\_HEIGHT, &Wire, -1);

#define ONE\_WIRE\_BUS 14        .

OneWire oneWire(ONE\_WIRE\_BUS);

DallasTemperature sensors(&oneWire);

DFRobot\_ESP\_EC ec;

Adafruit\_ADS1115 ads;

**float** voltage, ecValue, temperature = 25;

**String** APIKey = "\*\*\*\*\*\*\*\*\*\*\*\*\*\*";     *//  Enter your Write API key from ThingSpeak*

const **char** \*SSID =  "\*\*\*\*\*\*\*\*\*\*\*\*\*\*";     *// replace with your wifi ssid and wpa2 key*

const **char** \*pass =  "\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

const **char**\* server = "api.thingspeak.com";

WiFiClient client;

**void** setup()

{

  Serial.begin(115200);

  EEPROM.begin(32);*//needed EEPROM.begin to store calibration k in eeprom*

  ec.begin();

  sensors.begin();

**if** (!display.begin(SSD1306\_SWITCHCAPVCC, 0x3C)) { *// Address 0x3D for 128x64*

    Serial.println(F("SSD1306 allocation failed"));

**for** (;;);

  }

  delay(2000);

  display.clearDisplay();

  Serial.println("Connecting to…. ");

  Serial.println(SSID);

  WiFi.begin(SSID, pass);

**while** (WiFi.status() != WL\_CONNECTED)

  {

    delay(500);

    Serial.print(".");

  }

  Serial.println("");

  Serial.println("Success! WiFi connected");

}

**void** loop()

{

  voltage = analogRead(A0); *// A0 is the gpio 36*

  sensors.requestTemperatures();

  temperature = sensors.getTempCByIndex(0);  *// read your temperature sensor to execute temperature compensation*

  ecValue = ec.readEC(voltage, temperature); *// convert voltage to EC with temperature compensation*

  Serial.print("Temperature:");

  Serial.print(temperature, 2);

  Serial.println("ºC");

  Serial.print("EC:");

  Serial.println(ecValue, 2);

  display.setTextSize(2);

  display.setTextColor(WHITE);

  display.setCursor(0, 10);

  display.print("T:");

  display.print(temperature, 2);

  display.drawCircle(85, 10, 2, WHITE); *// put degree symbol ( ° )*

  display.setCursor(90, 10);

  display.print("C");

  display.setCursor(0, 40);

  display.print("EC:");

  display.print(ecValue, 2);

  display.display();

  delay(1500);

  display.clearDisplay();

  ec.calibration(voltage, temperature); *// calibration process by Serail CMD*

**if** (client.connect(server, 80))  *//   "184.106.153.149" or api.thingspeak.com*

  {

**String** postStr = APIKey;

    postStr += "&field1=";

    postStr += **String**(temperature, 2);

    postStr += "&field2=";

    postStr += **String**(ecValue, 2);

    postStr += "\r\n\r\n";

    delay(500);

    client.print("POST /update HTTP/1.1\n");

    client.print("Host: api.thingspeak.com\n");

    client.print("Connection: close\n");

    client.print("X-THINGSPEAKAPIKEY: " + APIKey + "\n");

    client.print("Content-Type: application/x-www-form-urlencoded\n");

    client.print("Content-Length: ");

    client.print(postStr.length());

    client.print("\n\n");

    client.print(postStr);

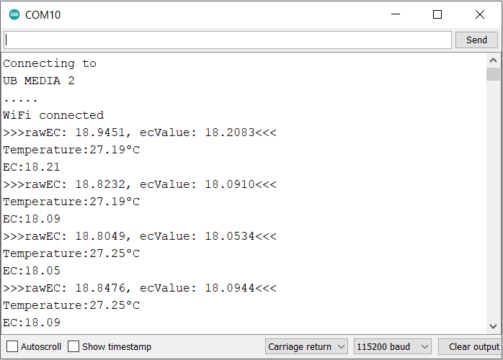
    delay(500);

  }

  client.stop();

}

Run the code following uploading it to Esp32 to see the results. Once the code has been uploaded to the ESP32 Board, the ESP32 will attempt to link to the WiFi network. As soon as it is connected to the WiFi network, it will start reading the EC and Temperature data from the sensor. To read those values, open the Serial Monitor.



**Conclusion**

In the end, an innovative development in the field of water management is a smart water quality monitoring system. With the aid of this modern technology, water quality parameters can be constantly tracked in an effective, affordable, and precise manner. The system's automated data collection, analysis, and visualisation skills give teams in charge of water management invaluable information about trends in water quality.

Smart water quality monitoring systems are definitely a crucial step towards sustainable water management because of the ever-rising demand for clean and healthy water and the expanding environmental concerns. The advantages of this technology cannot be overstated because it provides a means of preventing contamination, ensuring adherence to legal requirements, and encouraging environmental preservation.

The need for novel solutions, like smart water quality monitoring systems, will only rise as the population continues to exponentially increase and water resources become more scarce. These systems assure us to completely alter the way we manage our water resources because of their capacity to gather and analyse enormous amounts of data in real-time.