EIECC19-IOT WORKSHOP

Mini Project

Title: IoT Based Water TDS meter

Project Members:

ROHIT(2020UEI2807)

DEVANSHU YADAV(2020UEI2815)

Contents

1. Introduction. 3
2. Motivation. 5
3. Time distribution. 5
4. Hardware. 6
   - 4.1 Microcontroller. 6
   - 4.2 Sensors. 7
   - 4.3 Display. 8
   - 4.4 Digital Multimeter. 9
   - 4.5 Soldering Iron Kit. 10
   - 4.6 Power Supply. 11
   - 4.7 PCB Board. 11
5. Software. 12
   - 5.1 Arduino IDE. 12
   - 5.2 IOT Platform. 13
6. Schematic. 13
7. Working. 14
8. Programming. 14
9. Prototyping & testing. 18
10. PCB Manufacturing. 19
    - 10.1 PCB Layout. 19
    - 10.2 PCB Making Process. 19
    - 10.3 Soldering on PCB. 20
    - 10.4 Testing on PCB. 20
11. Final Product. 21
12. References. 22

1. Introduction

Water is a vital resource for life on Earth, and it covers approximately 71 of the planet's surface. It is a transparent, odourless, tasteless, and nearly colorless liquid that exists in three states: solid (ice), liquid, and gas (water vapour). Water is made up of two hydrogen atoms and one oxygen atom, and its chemical formula is H2O.

Water plays a crucial role in many natural processes, including the water cycle, which involves the evaporation of water from the Earth's surface, its condensation into clouds, and its precipitation back to the surface in the form of rain, snow, or hail. It is also essential for many living organisms, including humans, who require it for drinking, bathing, cooking, and hygiene.

Water is also used for various purposes, including agriculture, industry, and energy production. However, the availability of water is not uniform across the world, and many regions face water scarcity and droughts, which can have severe consequences for human and animal populations and the environment.

Water pollution is another critical issue, as human activities have led to the contamination of many water bodies, including rivers, lakes, and oceans, with chemicals, plastic, and other harmful substances. This pollution can harm aquatic life and also have negative impacts on human health and the economy.

In summary, water is a vital resource for life on Earth, and its availability, quality, and management are significant concerns for society and the environment.

- Importance of TDS in Drinking Water

TDS in drinking water originates from places like natural sources, sewage, urban run-offs, industrial wastewater, chemicals in the water treatment process, chemical fertilizers used in the garden and plumbing. Water is a universal solvent and easily picks up impurities and can absorb and dissolve these particles quickly. Although elevated levels of TDS in drinking water is not a health hazard, it does lend the water a bitter, salty, or brackish taste. Calcium and magnesium, two minerals commonly found in TDS, can also cause water hardness, scale formation, and staining.

- What are Different TDS levels

The TDS level helps indicate whether the drinking water is fit for consumption, requires filtration or is highly contaminated. Parts per million (PPM) is the measurement used for measuring TDS level in the water.

- TDS Level Chart for Drinking Water

TDS in Water (measured in PPM)	Suitability for Drinking Water
Between 50-150	Excellent for drinking
150-250	Good
250-300	Fair
300-500	300-500
Above 1200	Unacceptable

- Why Should You Measure TDs Levels

Naturally, mineral water has no smell or taste. A change in the TDS level changes the texture and taste, making the water unfit for consumption. Some of the reasons why you should measure the TDS level of your drinking water are:

*Taste (high TDS level can make the water salty and/or bitter).

*Health Concerns (water with high TDS level will not have a drastic impact on your health but the high level of lead or copper can make you fall sick).

*Cooking (TDS level above 1000 PPM can change the way the food tastes).

Not only drinking water TDS measurements are required in many other sectors also like:

- Spas & saloons
- Hydrophonics
- Fish aquarium
- Swimming Pools
- Various other industries

1. Motivation

1.Protection of public health: Monitoring the quality of drinking water sources, such as rivers, lakes, and groundwater, can help ensure that people are not exposed to harmful contaminants that could cause illness or disease.

2.Protection of aquatic life: Monitoring the quality of water bodies can help identify areas where aquatic life may be threatened by pollution, such as excessive nutrient loads, sedimentation, or toxic chemicals.

3.Compliance with regulations: Many jurisdictions require water quality monitoring to ensure compliance with environmental regulations related to water pollution, discharge of waste water, and other activities that can impact water quality.

4.Identification of pollution sources: Monitoring can help identify sources of pollution, such as industrial discharges or agricultural run off, so that appropriate remedial actions can be taken to prevent further contamination.

5.Scientific research: Water quality monitoring can also be motivated by a desire to better understand the factors that affect water quality, such as changes in land use, climate change, or natural variability. This research can inform future management decisions and help identify areas where further investigation is needed.

2. Time Distribution:

Week 1: Planning and research

Week 2: Hardware designing

Week 3: Software designing

Week 4: Prototyping & Testing

Week 5: PCB making & outer case

Week 6: Product making complete

3. Hardware

Microcontroller

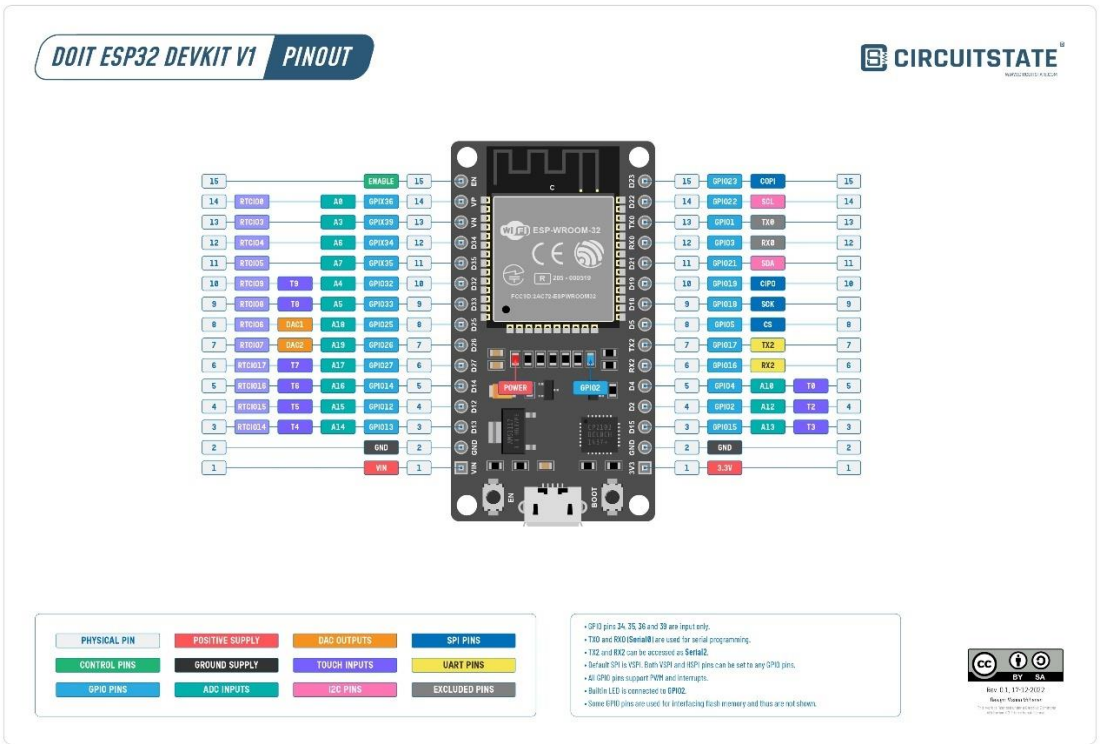
DOIT ESP 32 DEVKIT V1

The ESP32 DevKit V1 is a development board based on the ESP32 microcontroller. It is designed to help developers prototype their projects quickly and easily, and it comes with a variety of features to make this possible.

The ESP32 DevKit V1 features two built-in antennas, a power LED, a USB-to-serial converter, and a reset button. It also includes several GPIO pins for connecting to other components, as well as interfaces for SPI, I2C, and UART communication. Additionally, the board has support for Wi-Fi and Bluetooth connectivity, making it ideal for IoT and other wireless applications.

To start using the ESP32 DevKit V1, you will need to download and install the appropriate drivers and software. This includes the Arduino IDE, which you can use to write and upload code to the board. There are also many libraries and examples available online that can help you get started with programming the board.

Overall, the ESP32 DevKit V1 is a powerful and versatile development board that can be used for a wide range of projects. Whether you are a beginner or an experienced developer, it offers a convenient and flexible platform for prototyping and testing your ideas.



4.2 Sensors

4.2.1 Temperature Sensor

The DS18B20 is a digital temperature sensor that is widely used in various applications such as temperature monitoring, industrial control, and home automation. It is a 1-Wire interface sensor, which means that it only requires a single wire for communication with a microcontroller or other devices.

This is DS18B20 Water Proof Temperature Probe – Black (1m) Original Chip which is based on the DS18B20 sensor.

It is very handy for when you need to measure something far away, or in wet conditions. Because they are digital, you don't get any signal degradation even over a long distances.

These 1-wire digital temperature sensors are fairly precise (±0.5°C over much of the range) and can give up to 12 bits of precision from the onboard digital-to-analog converter.

They work great with any microcontroller using a single digital pin, and you can even connect multiple ones to the same pin, each one has a unique 64-bit ID burned in at the factory to differentiate them. Usable with 3.0-5.0V systems. When using with microcontroller put a 4.7k resistor to sensing pin, which is required as a pull-up from the DATA to VCC line.

Overall, the DS18B20 is a reliable and accurate temperature sensor that is easy to use and ideal for various temperature monitoring applications.

Wired connection

Red: VCC/VDD

Yellow/White: DATA

Black: GND

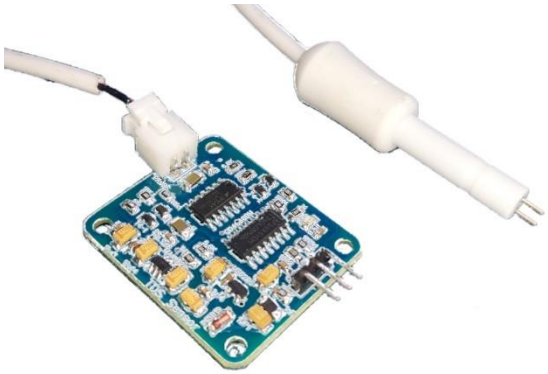
Specifications:

Model	**DS18B20**
Operating Voltage	3 – 5.5 volts
Operating Temperature(°C)	-55 to 125
Accuracy	±0.5°C
Cable Length	1 meter

4.2.2 TDS Sensor

Analog TDS Sensor for Arduino is an Arduino-compatible TDS Meter Kit for measuring TDS value of the water, to reflect the cleanliness of the water. TDS meter can be applied to domestic water, hydroponic and other fields of water quality testing.

What is TDS:



TDS (Total Dissolved Solids) indicates that how many milligrams of soluble solids dissolved in one liter of water. In general the higher the TDS value the more soluble solids dissolved in water and the less clean the water is. Therefore the TDS value can be used as one of the references for reflecting the cleanliness of water.

TDS pen is widely used equipment to measure TDS value. The price is affordable, and it is easy to use. The analog TDS sensor kit that is compatible with Arduino, plug, and play, easy to use. Matching with Arduino controller, you can build a TDS detector easily to measure the TDS value of liquid.

This TDS sensor supports 3.3 ~ 5.5V wide voltage input, and 0 ~ 2.3V analog voltage output, which makes it compatible with a 5V or 3.3V control system or board. The excitation source is an AC signal, which can effectively prevent the probe from polarization and prolong the life of the probe, meanwhile, increase the stability of the output signal. The TDS probe is waterproof, it can be immersed in water for a long time measurement.

FEATURES:

- Wide Voltage Input: 3.3~5.5V
- Good Compatibility Output: 0~2.3V analog signal output, compatible with 5V or 3.3V controller
- AC Excitation Source: effectively prevent the probe from polarization
- Waterproof Probe
- Easy to Use: Arduino compatible, simple connection, plug, and play without soldering.

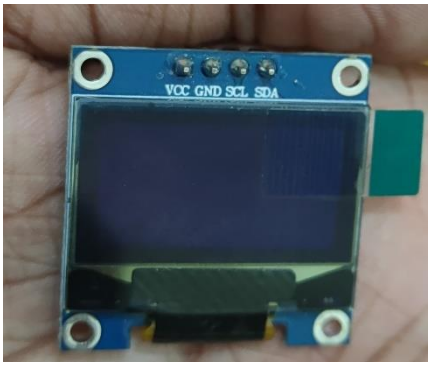
SPECIFICATION:

Input Voltage	**3.3-5.5V**
Output Voltage	0-2.3V
TDS Measurement Range	0-1000ppm
Accuracy	±10% F.S (25℃)
Cable Length	83 cm

4.3 Display

OLED (Organic Light-Emitting Diode) is a self light-emitting technology composed of a thin, multi-layered organic film placed between an anode and cathode. In contrast to LCD technology, OLED does not require a backlight. OLED possesses high application potential for virtually all types of displays and is regarded as the ultimate technology for the next generation of flat-panel display

This 0.96 Inch I2C/IIC 4pin OLED Display Module BLUE can be interfaced with any microcontroller using SPI/IIC/I2C protocols. It is having a resolution of 128×64. The package includes display board, display, 4 pin male header pre-soldered to board.



OLED monochrome 128×64 dot matrix display module. The characteristics of this display module are high brightness, self-emission, high contrast ratio, slim/thin outline, wide viewing angle, wide temperature range and low power consumption.

Pin Definition:

1.GND: Power ground

2.VCC: Power positive

3.SCL: Clock wire

4.SDA: Data wire

Specification:

Input Voltage	**3.3-5.5V**
Operating Temperature	-40 - 70 ℃
Interface type	IIC
Resolution	128X64 pixels

4.4 Digital Multimeter

Plusivo Digital Multimeter typical Low-Budget, High Accuracy, and best quality digital multimeter. The multimeter features backlight, hold function, premium probes, voltage, current, and resistance measurement and also can be used for conductivity measurement. This meter has all the basic functions that you may need and it really looks great and solid at its price.

This multimeter is mainly for measuring DC and AC voltage, AC current, resistance, and diode, with continuity buzzer, data hold, and insulation test function, the best device for home repair. It is a good assistant to test and troubleshoot faults that exist in an electronic device. This Digital LCD Multimeter can test Voltmeter Ohmmeter Ammeter.

The perfect all-around tool for working with anything electrical!

Very useful for debugging electrical problems including small power wires, car components, small electrical circuits, and more!

Features:

- Measure DC / AC voltages
- Measures the current intensity
- Measure resistance
- Test diodes
- Test transistors
- Test circuit continuity
- Measures the capacity of capacitors.

Specifications:

Measuring Range	Voltage: 0-600volts Current: 2mA-10A
Fuse Protection	F 200mA/250V
Display	With backlit lcd, 1999counts, Updates 2-3/sec
Measuring Method	Dual-slope integration A/D

4.5 Solder Iron Kit

4.6 Power Supply

Adapter

Battery

4.7 PCB

5. Software

5.1 Arduino IDE

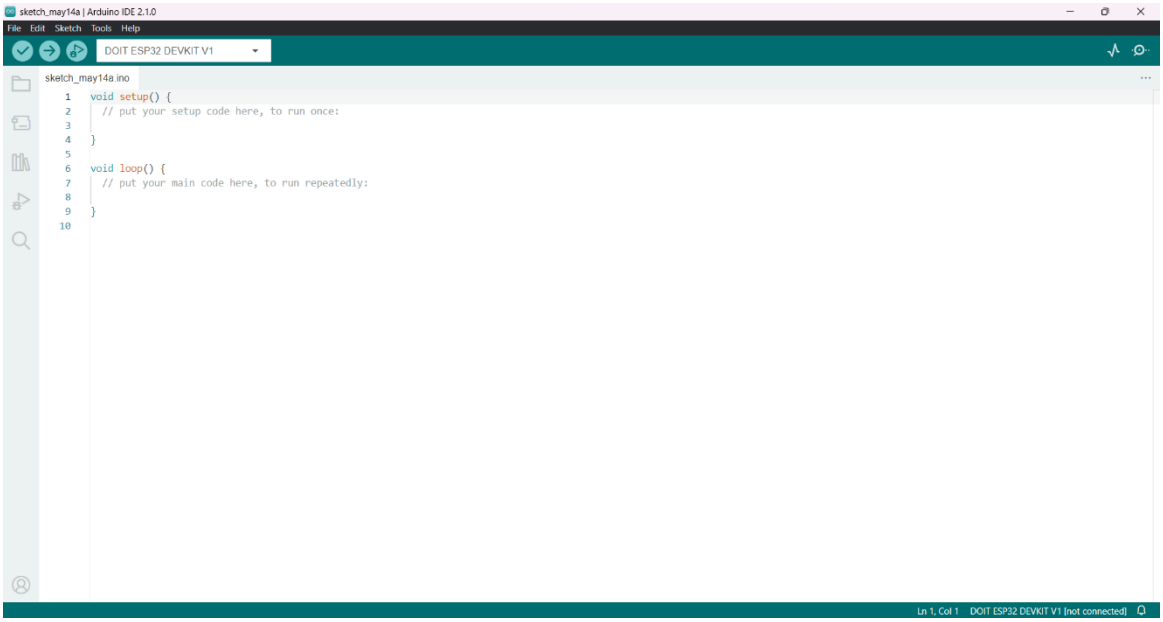
The Arduino Integrated Development Environment (IDE) is a software application that is used to program and develop software for Arduino boards. It provides a user-friendly interface for writing, compiling, and uploading code to an Arduino board.

The Arduino IDE is based on the Processing programming environment, which is a programming language and development environment designed for artists and designers. The IDE includes a code editor, a compiler, a library manager, and a serial monitor, among other features.

To use the Arduino IDE, you need to download and install it on your computer. Once installed, you can open the IDE and create a new sketch, which is a program written in the Arduino programming language. You can then write your code in the editor, compile it, and upload it to your Arduino board using a USB cable.

The Arduino IDE includes a large library of code examples and pre-written functions that can be used to simplify programming for beginners. Additionally, it supports many popular programming languages such as C++, which makes it easy for experienced programmers to use.

Overall, the Arduino IDE is an essential tool for anyone who wants to program and develop software for Arduino boards. With its user-friendly interface and comprehensive features, it makes it easy for beginners to get started and for experienced programmers to develop complex projects.



5.2 IOT Platform

ThingSpeak:

ThingSpeak is an open-source Internet of Things (IoT) platform that allows developers to collect, analyze, and visualize data from connected devices. It is a cloud-based platform that provides an easy-to-use interface for storing, analyzing, and sharing data generated by IoT devices.

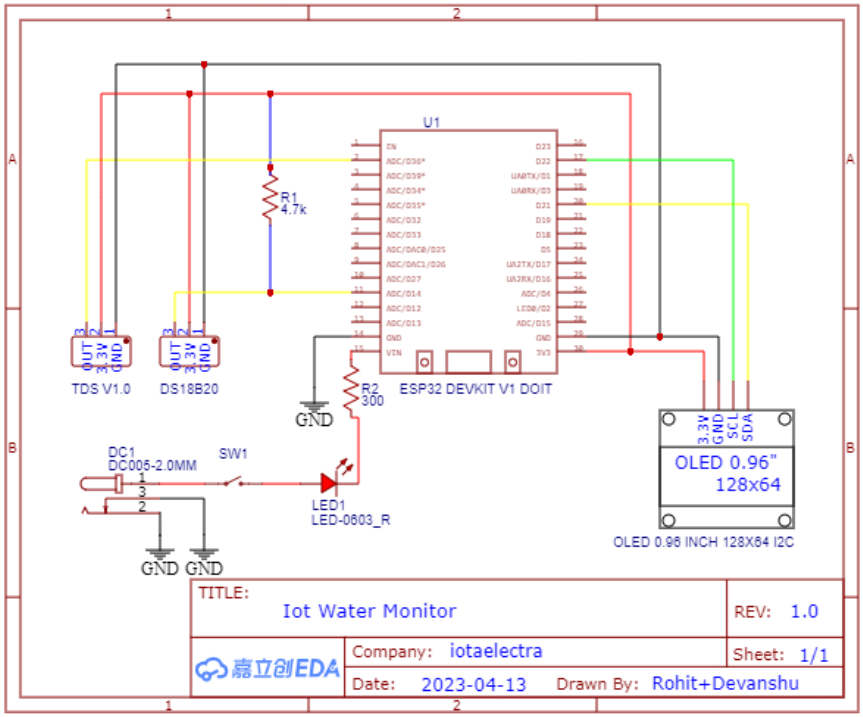
ThingSpeak is based on MATLAB, which is a widely used programming language and development environment for data analysis and visualization. It provides a wide range of features, including data visualization tools, support for real-time data streaming, and an API for integrating with other software and hardware platforms.

To use ThingSpeak, you need to create an account on the platform and create a channel, which is a virtual representation of your IoT device. You can then use the ThingSpeak API to send data to your channel from your connected device, and use the platform's tools to analyze and visualize the data.

ThingSpeak also includes support for third-party services such as Twitter, IFTTT, and MATLAB analysis functions, which allow you to create custom data processing and visualization workflows. Additionally, it includes a mobile app that lets you monitor your IoT devices and receive alerts when data exceeds certain thresholds.

Overall, ThingSpeak is a powerful and flexible IoT platform that is ideal for developers who want to collect, analyze, and visualize data from their connected devices. Its ease of use, robust feature set, and wide range of third-party integrations make it an excellent choice for a wide range of IoT applications.

6. Schematic:



7. Working

1. TDS sensor has inbuilt module for TDS calculation.
2. Temperature Sensor is used for measuring temperature.
3. As, we know TDS varies with temperature.
4. TDS and EC values are related as:

   TDS (mg/ L) = k x EC (µS /cm)

5. Also, we know that EC and temperature are related as

   $EC = [\ 1 + a\ (t - 25)\ ]\ EC_{25}$ with $a = 0.020$

6. This equation will hold true for TDS also.
7. So, we have to perform temperature compensation.
8. We have used ESP32 as microcontroller, it has inbuilt antena which connects with WiFi/mobile hotspot and will be used for sending data to ThingSpeak platform.
9. We can moniter the data on ThingSpeak server by making channel and creating an API.
10. We can moniter the TDS, temperature value on OLED offline also.

8. Programming:

Download the required libraries in advance.

```
#include <Wire.h>
#include <EEPROM.h>
#include <WiFi.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <OneWire.h>
#include <DallasTemperature.h>


String apiKey = "6JYRQ1Z8IYWPTMZD";     //  Enter your Write API key from ThingSpeak
const char *ssid =  "Realme 7";     // replace with your wifi ssid and wpa2 key
const char *pass =  "";
const char* server = "api.thingspeak.com";
WiFiClient client;

const int oneWireBus = 14; // GPIO where the DS18B20 is connected to

#define SCREEN_WIDTH 128 // OLED display width, in pixels
#define SCREEN_HEIGHT 64 // OLED display height, in pixels

// Declaration for an SSD1306 display connected to I2C (SDA, SCL pins)
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);
```

```
#define TdsSensorPin 36
#define VREF 3.3      // analog reference voltage(Volt) of the ADC
#define SCOUNT  30           // sum of sample point
int analogBuffer[SCOUNT];    // store the analog value in the array, read from ADC
int analogBufferTemp[SCOUNT];
int analogBufferIndex = 0;
int copyIndex = 0;
float averageVoltage = 0;
float tdsValue = 0;
float temperature = 0;


OneWire oneWire(oneWireBus);    // Setup a oneWire instance to communicate with any OneWire devices

DallasTemperature sensors(&oneWire);    // Pass our oneWire reference to Dallas Temperature sensor

int getMedianNum(int bArray[], int iFilterLen)
{
  int bTab[iFilterLen];
  for (byte i = 0; i < iFilterLen; i++)
    bTab[i] = bArray[i];
  int i, j, bTemp;
  for (j = 0; j < iFilterLen - 1; j++)
  {
    for (i = 0; i < iFilterLen - j - 1; i++)
    {
      if (bTab[i] > bTab[i + 1])
      {
        bTemp = bTab[i];
        bTab[i] = bTab[i + 1];
        bTab[i + 1] = bTemp;
      }
    }
  }
  if ((iFilterLen & 1) > 0)
    bTemp = bTab[(iFilterLen - 1) / 2];
  else
    bTemp = (bTab[iFilterLen / 2] + bTab[iFilterLen / 2 - 1]) / 2;
  return bTemp;
}


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

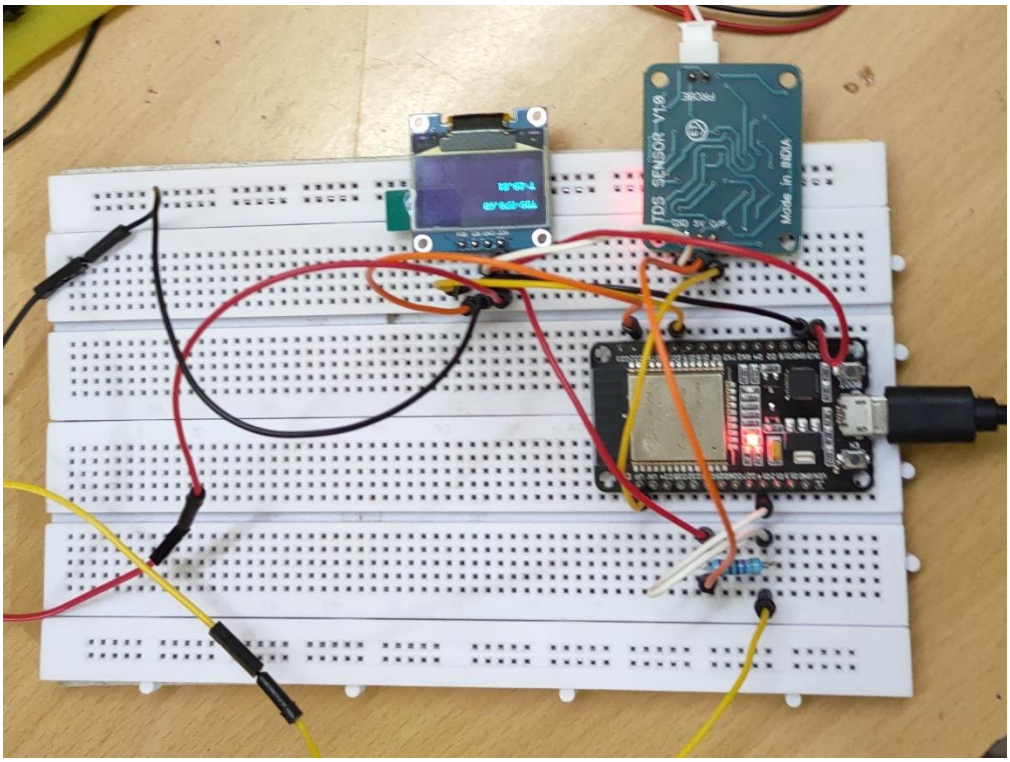
```
  pinMode(TdsSensorPin, INPUT);
  sensors.begin();

  Serial.println("Connecting to ");
  Serial.println(ssid);


  WiFi.begin(ssid, pass);

  while (WiFi.status() != WL_CONNECTED)
  {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");

  EEPROM.begin(32);//needed EEPROM.begin to store calibration k in eeprom
  if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) { // Address 0x3D for
128x64
    Serial.println(F("SSD1306 allocation failed"));
    for (;;);
  }
  delay(2000);
  display.clearDisplay();
}


void loop()
{
  sensors.requestTemperatures();
  float temperature = sensors.getTempCByIndex(0);

  static unsigned long analogSampleTimepoint = millis();
  if (millis() - analogSampleTimepoint > 40U)  //every 40 milliseconds,read
the analog value from the ADC
  {
    analogSampleTimepoint = millis();
    analogBuffer[analogBufferIndex] = analogRead(TdsSensorPin);    //read
the analog value and store into the buffer
    analogBufferIndex++;
    if (analogBufferIndex == SCOUNT)
      analogBufferIndex = 0;
  }
  static unsigned long printTimepoint = millis();
  if (millis() - printTimepoint > 800U)
  {
    printTimepoint = millis();
```

```
for (copyIndex = 0; copyIndex < SCOUNT; copyIndex++)
      analogBufferTemp[copyIndex] = analogBuffer[copyIndex];
    averageVoltage = getMedianNum(analogBufferTemp, SCOUNT) * (float)VREF / 1024.0; // read the analog value more stable by the median filtering algorithm, and convert to voltage value
    float compensationCoefficient = 1.0 + 0.02 * (temperature - 25.0); //temperature compensation formula: fFinalResult(25^C) = fFinalResult(current)/(1.0+0.02*(fTP-25.0));
    float compensationVolatge = averageVoltage / compensationCoefficient; //temperature compensation
    tdsValue = (133.42 * compensationVolatge * compensationVolatge * compensationVolatge - 255.86 * compensationVolatge * compensationVolatge + 857.39 * compensationVolatge) * 0.5; //convert voltage value to tds value

    Serial.print("TDS Value:");
    Serial.print(tdsValue, 0);
    Serial.println("ppm");

    Serial.print("Temperature:");
    Serial.print(temperature);
    Serial.println("ºC");
  }

  display.setTextSize(1);
  display.setTextColor(WHITE);

  display.setCursor(0, 30);
  display.print("T:");
  display.print(temperature, 2);

  display.setCursor(0, 10);
  display.print("TDS:");
  display.print(tdsValue, 2);
  display.display();
  delay(1500);
  display.clearDisplay();

   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(tdsValue, 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();
}
```

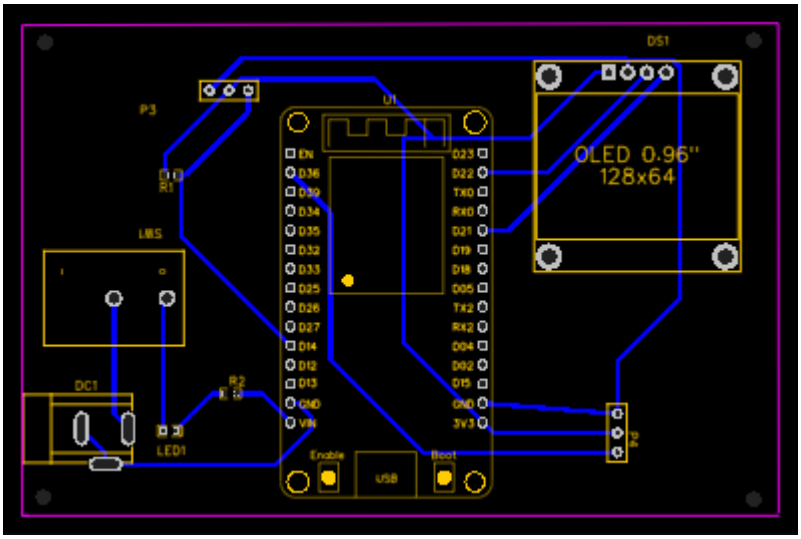
9. Prototyping & testing

On Breadboard



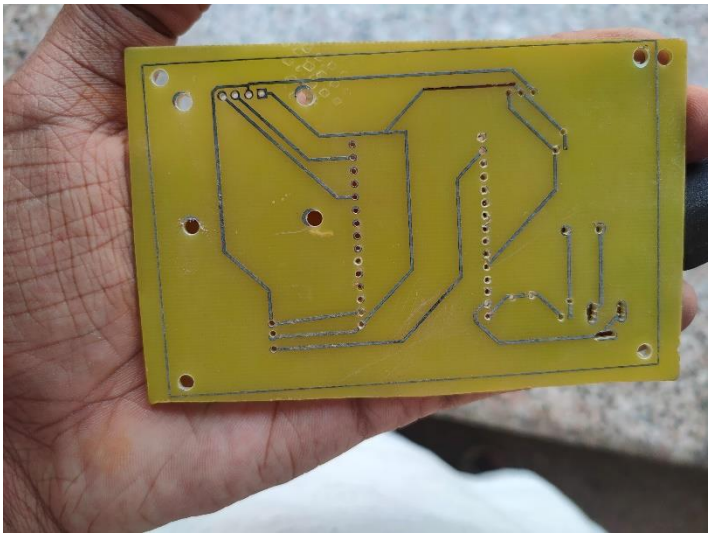
10. PCB Manufacturing

10.1 PCB Layout



10.2 PCB Making Process

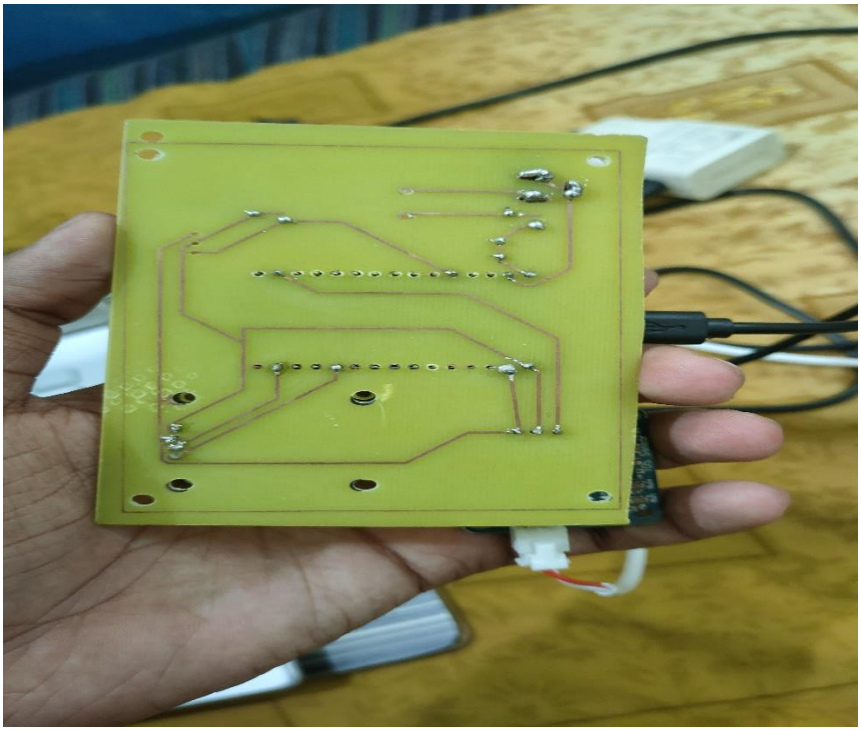
1. Cut PCB of your size from the Board
2. Polish the side of PCB which has copper, using sandpaper.
3. Print the PCB layout (only bottom layer) on glossy paper using printer.
4. Put the PCB print on the polished copper side of PCB.
5. Press on the print and PCB using iron for about 10-15 minutes.
6. Remove paper using water.
7. For etching, use FeCl3 solution (4:1).
8. Etching should be performed during daytime.
9. Now our should look like this one below.



10. Remove paper using sand paper.
11. Using drill machine make holes in the PCB to put header pins for components.

10.3 PCB soldering

1. Place all the header pins and perform soldering using soldering iron.
2. Now the PCB look like this one.



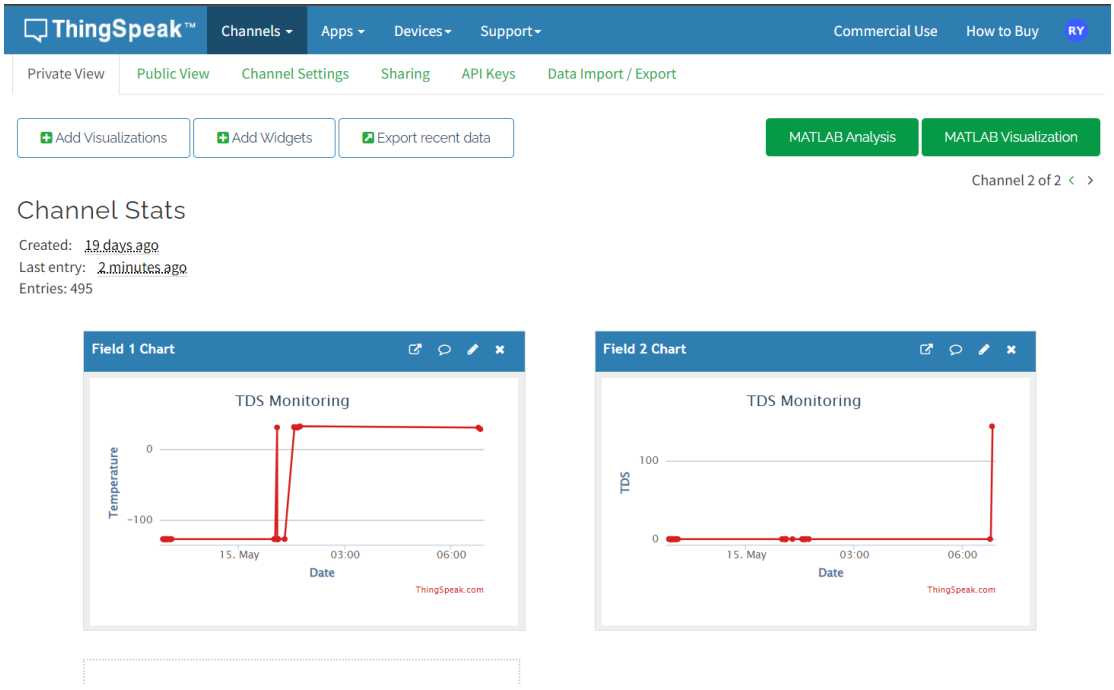
3. Place components on the header pins.

10.4 Testing On PCB

Test the PCB in various conditions.

11. Case and Final Product

Now using drill machine make holes for probes , power supply, and screws.



12. References

1. Google: www.google.com
2. Wikipedia: www.wikipedia.org
3. Aquion: https://www.aqion.de/site/112
4. IOP Conference: https://iopscience.iop.org/article/10.1088/1755-1315/118/1/012019/pdf#:~:text=Conductivity%20(EC)%20and%20total%20dissolved,EC%20(in%2025%200C).
5. Random Nerd Tutorials: www.randomnerdtutorials.com
6. Robu: www.robu.in
7. Various Electronics websites