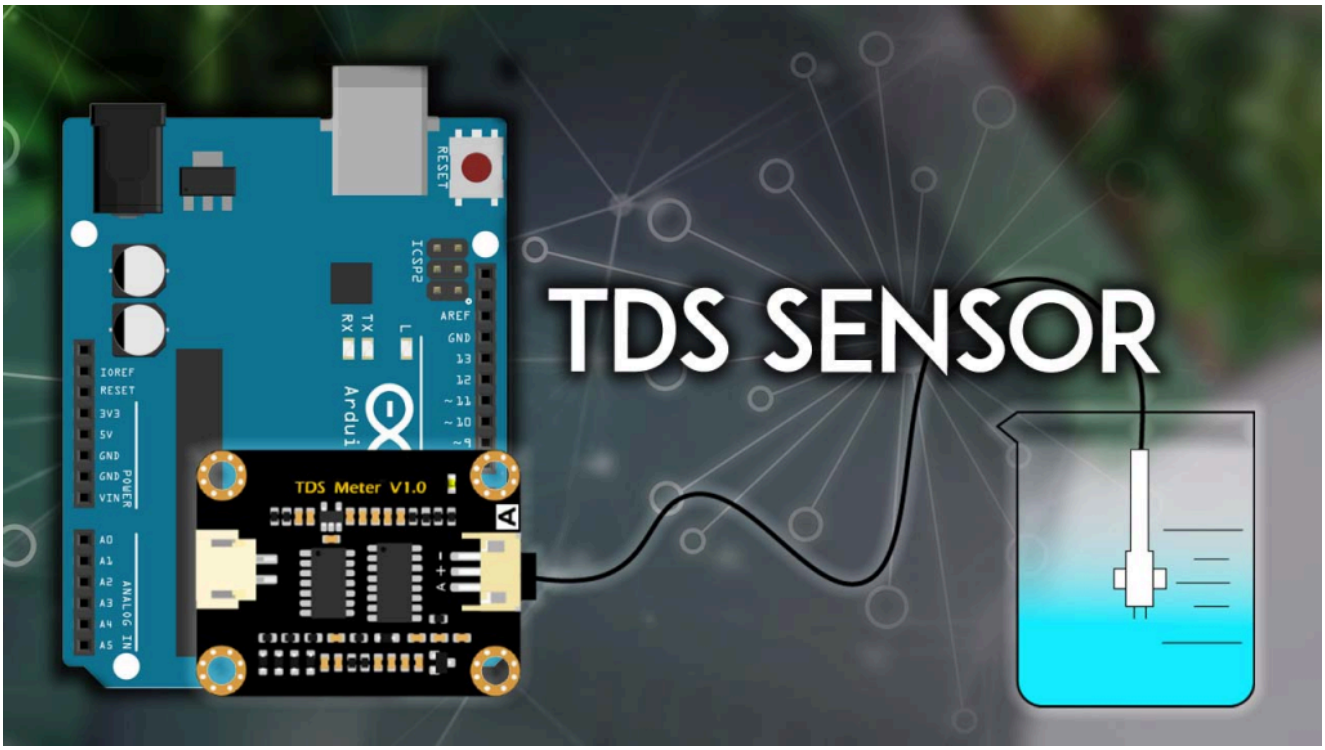
Arduino mit TDS-Sensor (Wasserqualitätssensor)

In dieser Anleitung erfahren Sie, wie Sie ein TDS-Messgerät (Total Dissolved Solids) mit einer Arduino-Platine verwenden. Ein TDS-Messgerät gibt die Gesamtzahl der gelösten Feststoffe wie Salze, Mineralien und Metalle in einer Lösung an. Mit diesem Parameter können Sie sich ein Bild von der Wasserqualität machen und Wasser aus verschiedenen Quellen vergleichen. Eine der Hauptanwendungen eines TDS-Messgeräts ist die Überwachung der Wasserqualität im Aquarium.

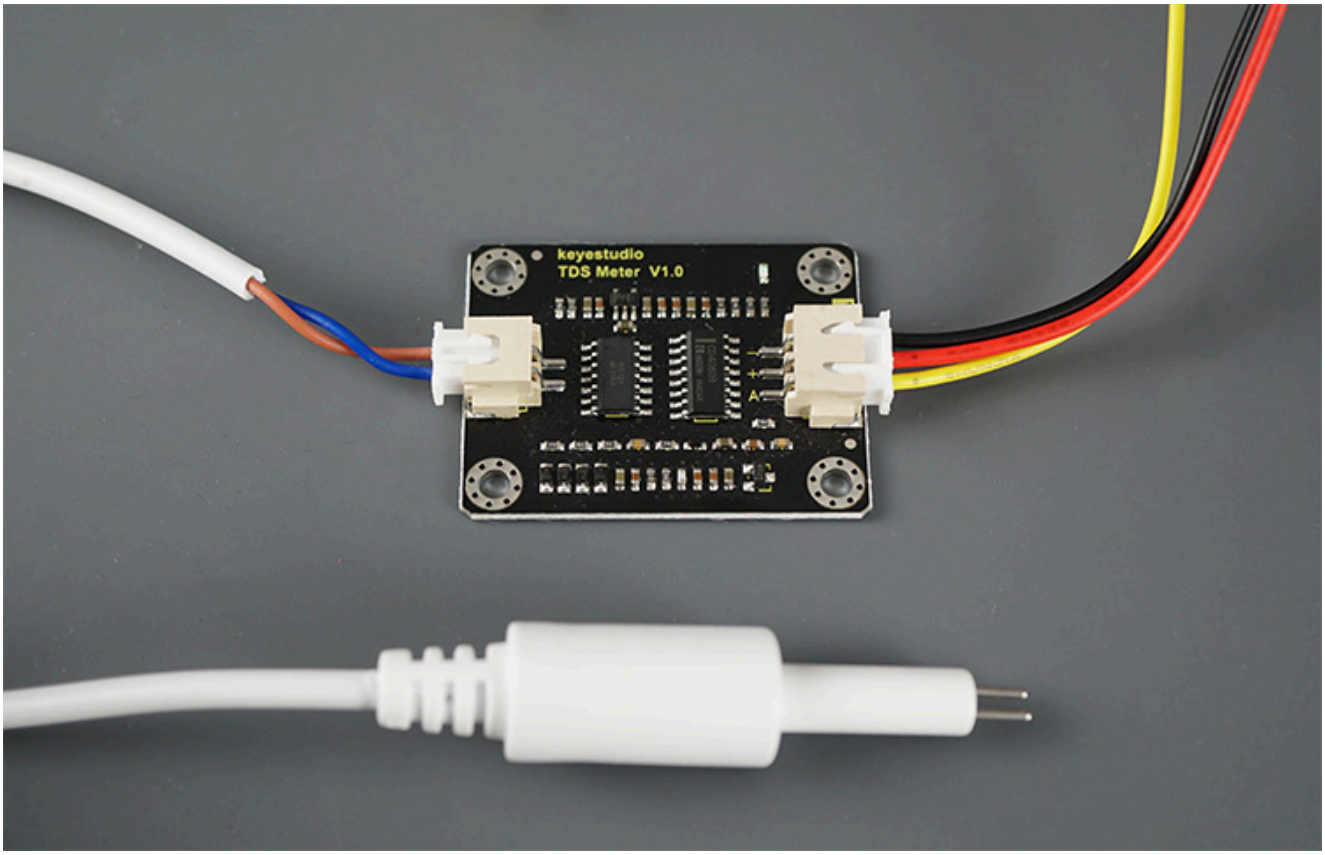


Wir verwenden das TDS-Messgerät von keystudio und zeigen Ihnen ein einfaches Beispiel zur Messung von TDS in ppm-Einheiten mit Arduino IDE.

Inhaltsverzeichnis

In diesem Tutorial behandeln wir die folgenden Themen

Ein TDS-Messgerät misst die Gesamtzahl der gelösten Feststoffe wie Salze, Mineralien und Metalle im Wasser. Mit zunehmender Anzahl gelöster Feststoffe im Wasser nimmt die Leitfähigkeit des Wassers zu, was es uns ermöglicht, die gesamten gelösten Feststoffe in ppm (mg/L) zu berechnen.



Obwohl dies ein guter Indikator zur Überwachung der Wasserqualität ist, **Beachten Sie, dass es keine Verunreinigungen im Wasser misst**. Daher können Sie sich nicht allein auf diesen Indikator verlassen, um festzustellen, ob das Wasser zum Verzehr geeignet ist oder nicht.

Ein TDS-Messgerät kann zur Überwachung der Wasserqualität in vielen Anwendungen wie Pools, Aquarien, Aquarien, Hydrokulturen, Wasserreinigern usw. nützlich sein.

In diesem Tutorial verwenden wir das TDS-Messgerät von keystudio, das mit einem Schnittstellenmodul und einer Elektrodensonde geliefert wird (siehe Bild oben).

For more information about the TDS meter, we recommend taking a look at the official documentation.

This tutorial refers to the TDS Meter V1.0 from keystudio. Here are the sensor parameters:

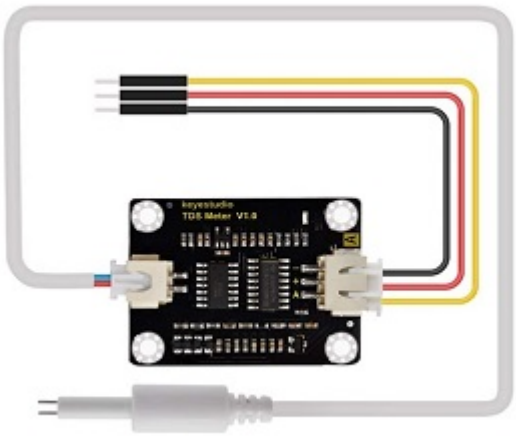
TDS Meter:

- Input Voltage: DC 3.3 ~ 5.5V
- Output Voltage: 0 ~ 2.3V
- Working Current: 3 ~ 6mA
- TDS Measurement Range: 0 ~ 1000ppm
- TDS Measurement Accuracy: ± 10% F.S. (25 ℃)
- Module Interface: XH2.54-3P
- Electrode Interface: XH2.54-2P

TDS Probe:

- Number of Needle: 2
- Total Length: 60cm
- Connection Interface: XH2.54-2P
- Color: White
- Waterproof Probe

Where to Buy TDS Sensor?



You can check the TDS sensor on Maker Advisor to find the best price:

- TDS Sensor

the parts for your projects at the best price!

Interfacing the TDS Meter with the Arduino

The TDS meter outputs an analog signal that can be measured using the Arduino analog pins (A0 to A5).

Wire the sensor as in the following table:

TDS Sensor	Arduino
GND	GND
VCC	3.3V
Data	A0 (or any other Arduino analog pin)

Reading TDS (water quality) with the Arduino-Code

As we mentioned previously, the sensor outputs an analog signal that can be converted to TDS in ppm. We're using the code provided by the sensor documentation with some modifications.

To get more accurate results, you'll probably need to calibrate your sensor against a solution with a known TDS value. However, it might not be needed if you are not concerned about specific values but about a qualitative value of TDS.

Upload the following code to your Arduino.

```
// Original source code: https://wiki.keyestudio.com/KS0429_keye
// Project details: https://RandomNerdTutorials.com/arduino-tds-

#define TdsSensorPin A0
#define VREF 5.0              // analog reference voltage(Volt)
#define SCOUNT  30            // sum of sample point

int analogBuffer[SCOUNT];     // store the analog value in the a
int analogBufferTemp[SCOUNT];
int analogBufferIndex = 0;
int copyIndex = 0;

float averageVoltage = 0;
float tdsValue = 0;
float temperature = 16;       // current temperature for compens

// median filtering algorithm
```

```
  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];
```

View raw code

How the Code Works

Let's take a quick look at the code. You can also skip right away to the Demonstration section.

The `TdsSensorPin` variable saves the GPIO where you want to get the readings. The ESP8266 only has one analog pin, A0.

```
#define TdsSensorPin A0
```

Then, insert the analog voltage reference for the ADC. For an Arduino, it is 5V.

```
#define VREF 5.0           // analog reference voltage(Volt) of
```

Before getting a measurement value, we'll apply a median filtering algorithm to get a more stable value. The `SCOUNT` variable refers to the number of samples we'll filter before getting an actual value.

```
#define SCOUNT  30           // sum of sample point
```

Then, we need some arrays to store the readings as well as some index variables that will allow us to go through the arrays.

```
int analogBufferTemp[SCOUNT];
int analogBufferIndex = 0;
int copyIndex = 0;
```

Initialize the `averageVoltage` variable and `tsdValue` as float variables.

```
float averageVoltage = 0;
float tdsValue = 0;
```

The temperature variable saves the current temperature value. The temperature influences the readings, so there is an algorithm that compensates for fluctuations in temperature. In this example, the reference temperature is 25ºC, but you can change it depending on your environment. For more accurate results, you can add a temperature sensor and get the actual temperature at the time of reading the sensor.

```
float temperature = 25;       // current temperature for compensa
```

The following function will be used to get a stable TDS value from an array of readings.

```
// median filtering algorithm
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];
```

```
      }
    }
  }
  if ((iFilterLen & 1) > 0){
    bTemp = bTab[(iFilterLen - 1) / 2];
  }
  else {
    bTemp = (bTab[iFilterLen / 2] + bTab[iFilterLen / 2 - 1]) / 2
  }
  return bTemp;
}
```

In the `setup()`, initialize the Serial Monitor at a baud rate of 115200.

```
Serial.begin(115200);
```

Set the TDS sensor pin as an input.

```
pinMode(TdsSensorPin,INPUT);
```

In the loop(), get new TDS readings every 40 milliseconds and save them in the buffer:

```
static unsigned long analogSampleTimepoint = millis();
if(millis()-analogSampleTimepoint > 40U){     //every 40 millisec
  analogSampleTimepoint = millis();
  analogBuffer[analogBufferIndex] = analogRead(TdsSensorPin);
  analogBufferIndex++;
  if(analogBufferIndex == SCOUNT){
    analogBufferIndex = 0;
  }
}
```

voltage by using the filtering algorithm created before:

```
static unsigned long printTimepoint = millis();
if(millis()-printTimepoint > 800U){
  printTimepoint = millis();
  for(copyIndex=0; copyIndex<SCOUNT; copyIndex++){
    analogBufferTemp[copyIndex] = analogBuffer[copyIndex];

    // read the analog value more stable by the median filtering
    averageVoltage = getMedianNum(analogBufferTemp,SCOUNT) * (flo
```

Then, it calculates a temperature compensation coefficient and calculates the TDS value taking that value into account:

```
//temperature compensation formula: fFinalResult(25^C) = fFinalRe
float compensationCoefficient = 1.0+0.02*(temperature-25.0);
//temperature compensation
float compensationVoltage=averageVoltage/compensationCoefficient;

//convert voltage value to tds value
tdsValue=(133.42*compensationVoltage*compensationVoltage*compensa
```

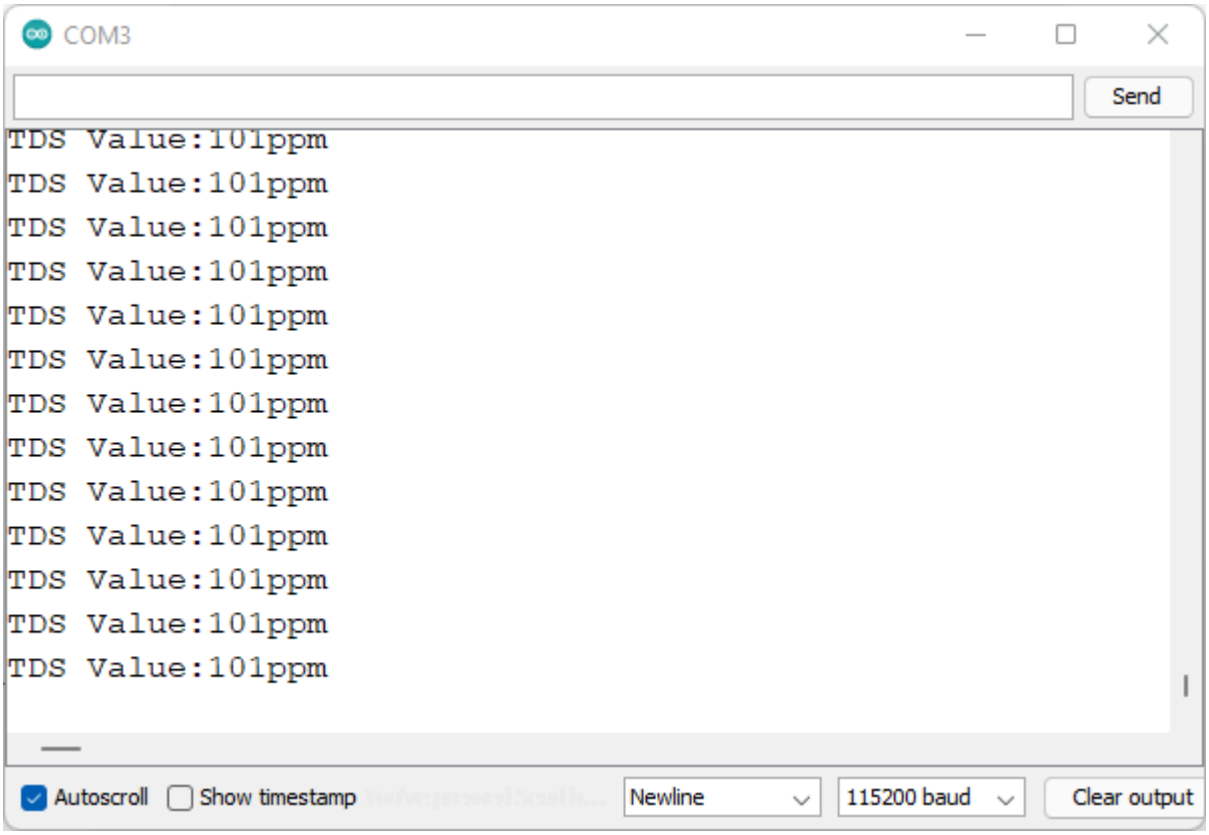
Finally, it prints the TDS value in ppm:

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

Demonstration

After copying the code to the Arduino IDE, upload the code to your board. Don't forget to select the right board in **Tools** > **Board** and the right COM port in **Tools** > **Port**.

It will show a value of approximately 0 if the probe is not submerged. Put the probe on a solution to check its TDS. You can try with tap water and add some salt to see the values increase.



I measured the TDS value for tap water in my house, and I got a value of around 100ppm, which is a good value for drinking water.

reasonable value.

Finally, I also measured the TDS value of bottled water and I got a value of around 25ppm (the same value when using an ESP8266).

Wrapping Up

A TDS meter can measure the total dissolved solids in a solution. It can be used as an indicator of water quality and allows you to characterize the water. The meter returns the TDS value in ppm (parts per million—mg/L). The TDS value has many applications but it cannot be used by itself to determine if the water is drinkable or not.

A great application of this type of sensor is an aquarium water quality monitor. You can use this sensor alongside a waterproof DS18B20 temperature sensor to monitor your fish tank, for example.

We have tutorials for other popular sensors with the Arduino board that you may like:

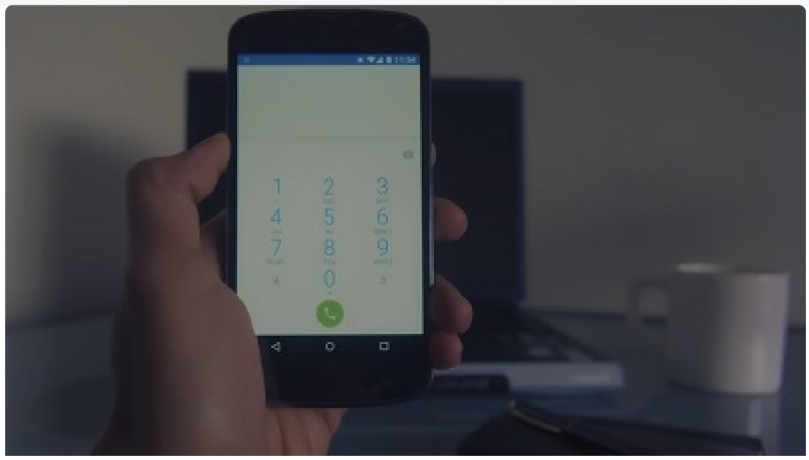
- **BME680** Environmental Sensor with Arduino (Gas, Temperature, Humidity, Pressure)
- **BME280** Sensor with Arduino (Pressure, Temperature, Humidity)
- **DS18B20** Temperature Sensor with Arduino
- **DHT11/DHT22** Humidity and Temperature Sensor With Arduino
- **LM35**, **LM335** and **LM34** Temperature Sensors with Arduino
- **HC-SR04** Ultrasonic Sensor with Arduino
- **BH1750** Ambient Light Sensor with Arduino
- **K-Type Thermocouple** and MAX6675 Amplifier with Arduino

Learn more about the Arduino with our resources:

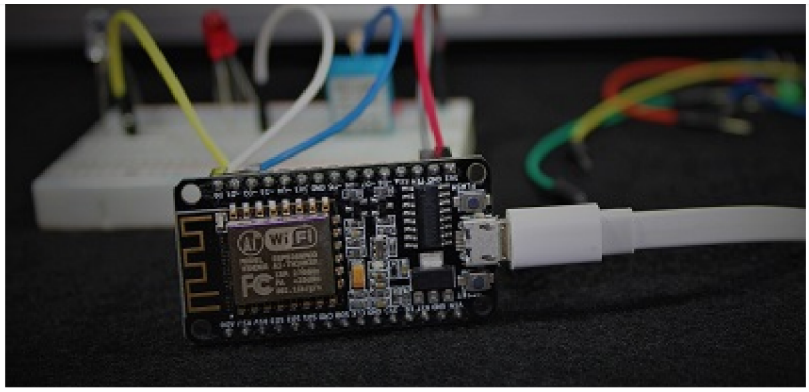
- Arduino Mini-Course (free)
- Arduino Step-by-step Projects course
- More Arduino tutorials and projects…

Thanks for reading.

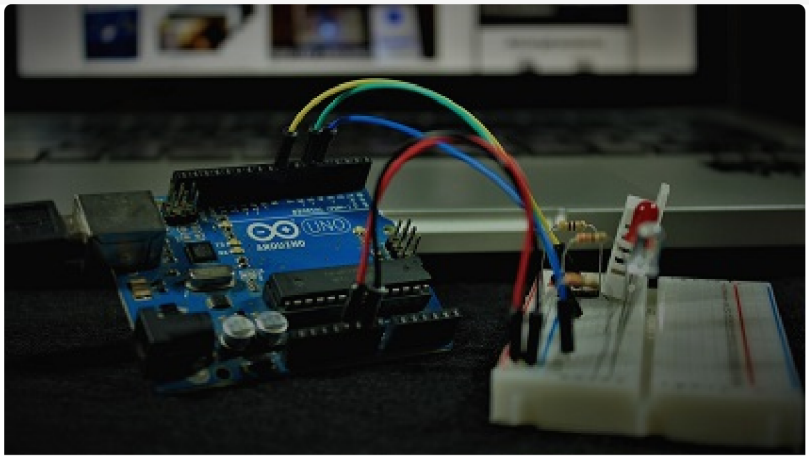
Recommended Resources



Build a Home Automation System from Scratch » With Raspberry Pi, ESP8266, Arduino, and Node-RED.



Home Automation using ESP8266 eBook and video course » Build IoT and home automation projects.



Arduino Step-by-Step Projects » Build 25 Arduino projects with our course, even with no prior experience!

What to Read Next…

ESP32/ESP8266 Thermostat Web Server – Control Output Based on Temperature

Getting Date and Time with ESP32 on Arduino IDE (NTP Client)

ESP8266 Publishing DHT22 Readings with MQTT to Raspberry Pi

14 thoughts on “Arduino with TDS Sensor (Water Quality Sensor)”

Tamojit

July 2, 2022 at 12:48 pm

```
tdsValue=
(133.42*compensationVoltage*compensationVoltage*compensationVoltage-
255.86*compensationVoltage*compensationVoltage +
857.39*compensationVoltage)*0.5;
```

used as factors for compensation voltage?

Reply

Sara Santos

July 4, 2022 at 1:41 pm

Hi.
The link is provided in the blog post: https://wiki.keyestudio.com/KS0429_keyestudio_TDS_Meter_V1.0#Test_Code
Regards,
Sara

Reply

Branko Almeira

November 12, 2022 at 9:41 pm

Buenas tardes! Estoy realizando un sistema de hidroponía automatizado, en el utilizo un sensor TDS y publico datos en firebase. Mi consulta es ¿Se puede dejar el sensor inmerso en la solución nutritiva y realizar lecturas sin sacarlo de dicha solución nutritiva?
Excelentes tutoriales!
Saludos desde Argentina.

Reply

Validio
February 7, 2023 at 3:25 am

Is it works for ESP32?

Reply

Sara Santos
February 11, 2023 at 8:33 am

Yes.
Take a look at this tutorial: https://randomnerdtutorials.com/esp32-tds-water-quality-sensor/
Regards,
Sara

Reply

Jjmnp
April 5, 2023 at 1:42 am

How many accuracy testing should be done? I just need a statement of an expert or user on how many trials/testing should be done.

Reply

tabor

August 2, 2023 at 12:06 pm

what have a library for proteus tds meter

Reply

Tifah

December 10, 2024 at 8:06 am

hi, please share the library too (nocontextzim@gmail.com)
i would greatly appreciate

Reply

johnmerry

September 28, 2023 at 11:28 am

Hi, can i change the TDS value from ppm to EC value like (microS/cm). If yes, what are the equation or constants i need to take note of?

Reply

umer ahmed

December 25, 2023 at 3:39 pm

My tds sensor not showing any constant result it start with 0 and increased upto 2106 value even it is not deep it in any solution, please guide me

Reply

Muhammad Misbachul munir
March 19, 2024 at 4:23 am

Is it works for STM32?

Reply

Sara Santos
March 19, 2024 at 10:53 am

Hi.
I'm not sure.
I didn't teste it.
Regards,
Sara

Reply

Miguel
January 18, 2025 at 4:10 pm

Greetings! I am a student researcher from Pacita Complex National High School, a high school located in the province of Laguna in the Philippines. I am conducting a study regarding analysis of pollution spread in water, specifically fluvii and limnai (rivers and lakes), and utilizing a TDS Sensor is crucial to my study, specifically measuring the ppm of my treated samples. I am somewhat new to robotics and arduino and I am looking for a TDS Sensor until I came across this website and saw your tutorial. I would be happy to cite the author but it can't be seen here. I would want to ask the author if I can cite him/her in my study and if s/he can elaborate this tutorial further! I endow my highest gratitude if you ever respond! Thanks!

Reply

Sara Santos
January 18, 2025 at 6:52 pm

Hallo.
Ich habe dieses Tutorial geschrieben. Mein Name ist Sara Santos, aber Sie können auf unseren Blog verweisen: Random Nerd Tutorials.
Vielen Dank, dass Sie unsere Arbeit verfolgt haben.
Grüße,
Sara

Antwort

Hinterlasse einen Kommentar

Name *

Email *

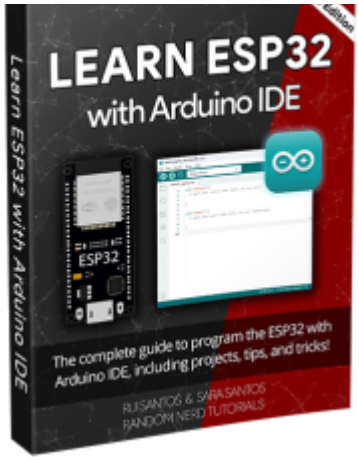
Website

☐ Benachrichtigen Sie mich über Folgekommentare per E-Mail.

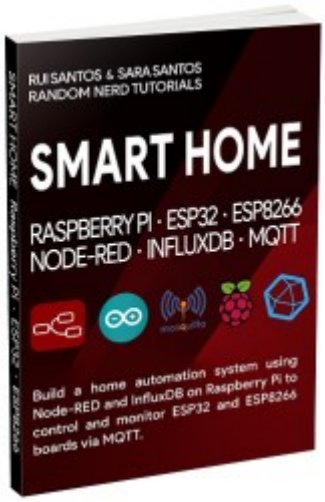
☐ Benachrichtigen Sie mich per E-Mail über neue Beiträge.

Kommentar posten

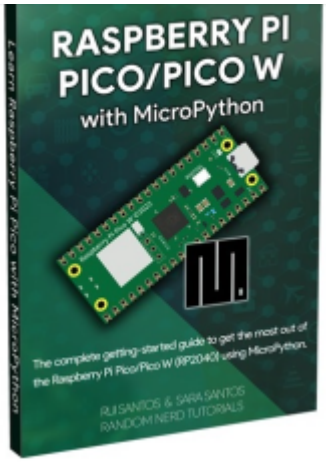
Offenlegung durch Partner: *Random Nerd Tutorials nimmt an Affiliate-Werbeprogrammen teil, die uns die Möglichkeit bieten sollen, durch Links zu Amazon, eBay, AliExpress und anderen Websites Gebühren zu verdienen. Wir könnten dafür entschädigt werden, dass wir Verkehr und Geschäfte an diese Unternehmen verweisen.*



ESP32 lernen mit Arduino IDE eBook »
Vollständige Anleitung zum Programmieren des ESP32 mit Arduino IDE!



SMART HOME mit Raspberry Pi, ESP32 und ESP8266 » Erfahren Sie, wie Sie ein komplettes Hausautomationssystem aufbauen.



Lernen Sie Raspberry Pi Pico/Pico W mit MicroPython » Die vollständige Anleitung für den Einstieg, um das Beste aus der Mikrocontrollerplatine Raspberry Pi Pico/Pico W (RP2040) mithilfe der Programmiersprache MicroPython herauszuholen.

🔥 **LVGL lernen: GUIs für ESP32-Projekte erstellen »** Erfahren Sie, wie Sie mit der Arduino IDE grafische Benutzeroberflächen (GUIs) für ESP32-Projekte mit LVGL (Light Versatile Graphics Library) erstellen.