Gravity: Analog TDS Sensor / Meter For Arduino
SKU: SEN0244

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 a widely used equipment to measure TDS value. The price is affordable, and it is easy to use, but it is not able to transmit data to the control system for online monitoring to do some water quality analysis. The professional instrument has high accuracy and can send data to the control system, but the price is expensive for the ordinary people. To this end, we have launched an analog TDS sensor kit which 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 product supports 3.3 ~ 5.5V wide voltage input, and 0 ~ 2.3V analog voltage output, which makes it compatible with 5V or 3.3V control system or board. The excitation source is 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 long time measurement.

This product can be used in water quality application, such as domestic water, hydroponics. With this product, you can easily DIY a TDS detector to reflect the cleanliness of water to protect your health.
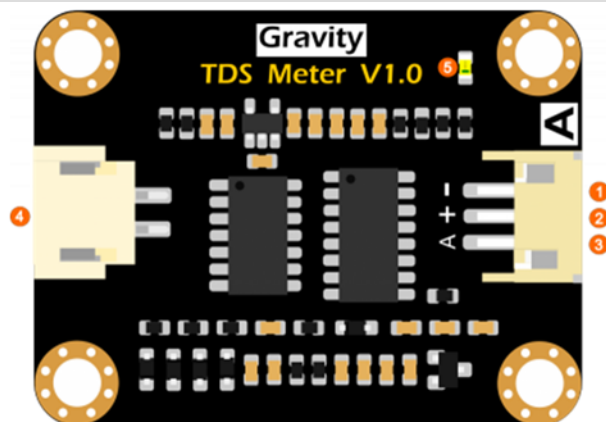
> ⚠️ **Attention:**
>
> 1.The probe can not be used in water above 55 degrees centigrade.
> 2.The probe can not be left too close to the edge of the container, otherwise it will affect the reading.
> 3.The head and the cable of the probe are waterproof, but the connector and the signal transmitter board are not waterproof. Please be careful.

Specification

- **Signal Transmitter Board**
  Input Voltage: 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 Size: 42 * 32mm
  Module Interface: PH2.0-3P
  Electrode Interface: XH2.54-2P
- **TDS probe**
  Number of Needle: 2
  Total Length: 83cm
  Connection Interface: XH2.54-2P
  Colour: Black
  Other: Waterproof Probe

Board Overview

Analog TDS Sensor / Meter For Arduino

Num	Label	Description
1	-	Power GND(0V)
2	+	Power VCC(3.3 ~ 5.5V)
3	A	Analog Signal Output(0 ~ 2.3V)
4	TDS	TDS Probe Connector
5	LED	Power Indicator

Tutorial

This tutorial will show you how to measure the TDS value of the water. Please read this tutorial carefully, and pay attention to the steps and details.

> ⚠ The probe can not to be used in water above 55 degrees centigrade.
> The probe can not be too close to the edge of the container, otherwise it will affect the reading.
> The head and the cable of the probe are waterproof, but the connector and the signal transmitter board are not waterproof.Please pay attention to use.
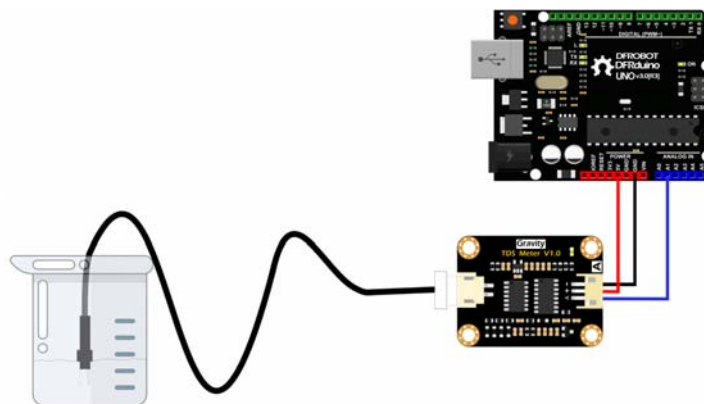
Requirements

- **Hardware**
  DFRduino UNO R3 (or similar) x 1
  Analog TDS Sensor / Meter Module x 1
  TDS Probe x1
  Jumper Wires x3
  tested liquid x1

- **Software**

Arduino IDE (Version requirements: V1.0.x or V1.8.x), Click to Download Arduino IDE from Arduino®

https://www.arduino.cc/en/Main/Software%7C

Connection Diagram

Sample Code

```
/**************************************************

 DFRobot Gravity: Analog TDS Sensor / Meter For Arduino

 <https://www.dfrobot.com/wiki/index.php/Gravity:_Analog_TDS_Sensor_/_Mete
r_For_Arduino_SKU:_SEN0244>


 Created 2017-8-22

 By Jason <jason.ling@dfrobot.com@dfrobot.com>


 GNU Lesser General Public License.

 See <http://www.gnu.org/licenses/> for details.

 All above must be included in any redistribution


 /***********Notice and Trouble shooting***************

 1. This code is tested on Arduino Uno and Leonardo with Arduino IDE 1.0.5
r2 and 1.8.2.

 2. More details, please click this link: <https://www.dfrobot.com/wiki/in
dex.php/Gravity:_Analog_TDS_Sensor_/_Meter_For_Arduino_SKU:_SEN0244>

 **************************************************/


#define TdsSensorPin A1

#define VREF 5.0       // 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,copyIndex = 0;

float averageVoltage = 0,tdsValue = 0,temperature = 25;


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

    pinMode(TdsSensorPin,INPUT);
}
```

```
void loop()
{
    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("voltage:");
      //Serial.print(averageVoltage,2);
      //Serial.print("V    ");
      Serial.print("TDS Value:");
      Serial.print(tdsValue,0);
      Serial.println("ppm");
    }
```

```
}
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;
}
```

Expected Results

After uploading the sample code,open the serial monitor of the Arduino IDE. Then insert the TDS probe into the water, and gently stir it. Waiting for the reading to be stable, and you will get the TDS value of the water.
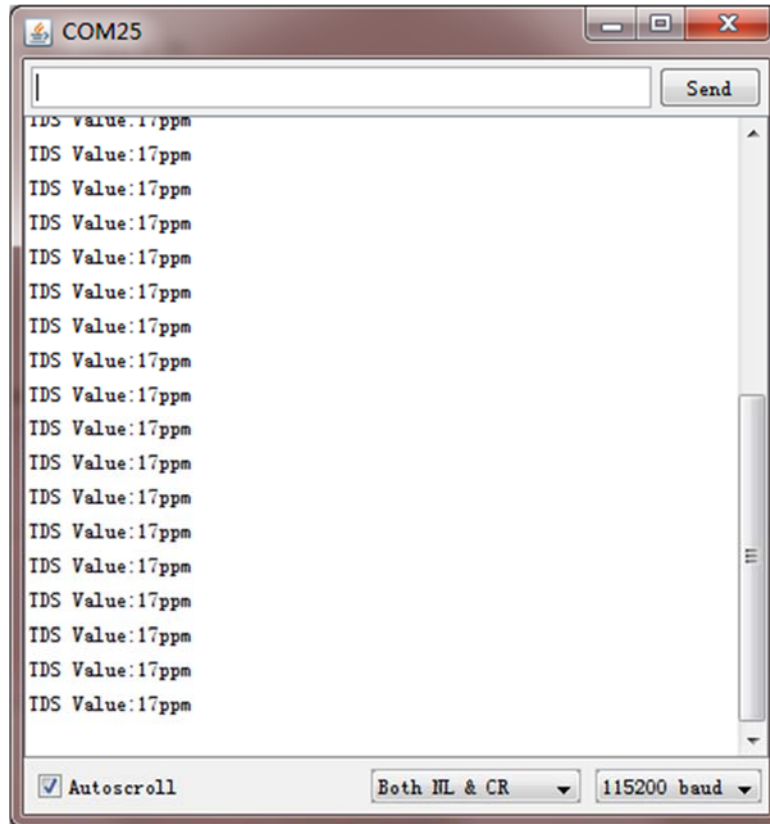
FAQ

A. This TDS probe has no temperature sensor, but the temperature compensation algorithm is reserved in the sample code. The temperature variable in the sample code will default to 25 °C without a temperature sensor. You can add a waterproof temperature sensor to read the temperature,then update the temperature variable, to make automatic temperature compensation.

For any questions, advice or cool ideas to share, please visit the **DFRobot Forum**.