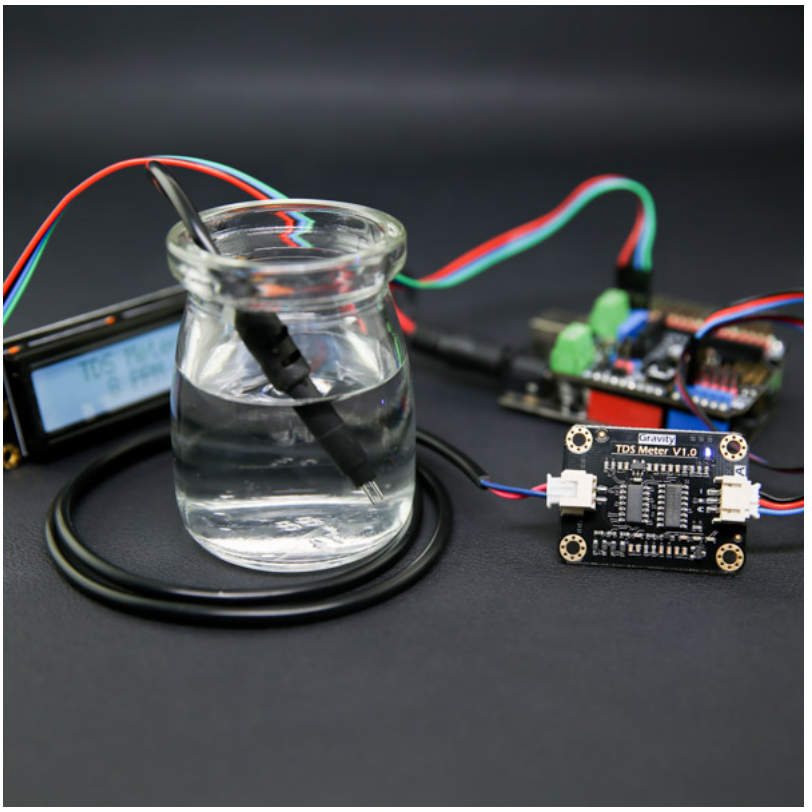
Gravity__Analog_TDS_Sensor___Meter_For_Arduino_SKU__SEN0244-DFRobot

Web Clip



Introduction

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

A TDS pen is a widely used peice of equipment to measure TDS value. The price is affordable, and it is easy to use, however commonly it is not able to transmit data to a control system for online monitoring of water quality. In general professional

analog TDS sensor kit which is compatible with Arduino, plug and play, and is easy to use. Matching with Arduino controller, you can build a TDS detector easily to measure the TDS value of liquid without needing to purchase expensive equipment.

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 systems or boards. The excitation source is AC signal, which can effectively prevent the probe from polarization and prolong the life of the probe, meanwhile can help 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 analysis and hydroponics. With this product, you can easily DIY a TDS detector to reflect the cleanliness of water to protect your health!

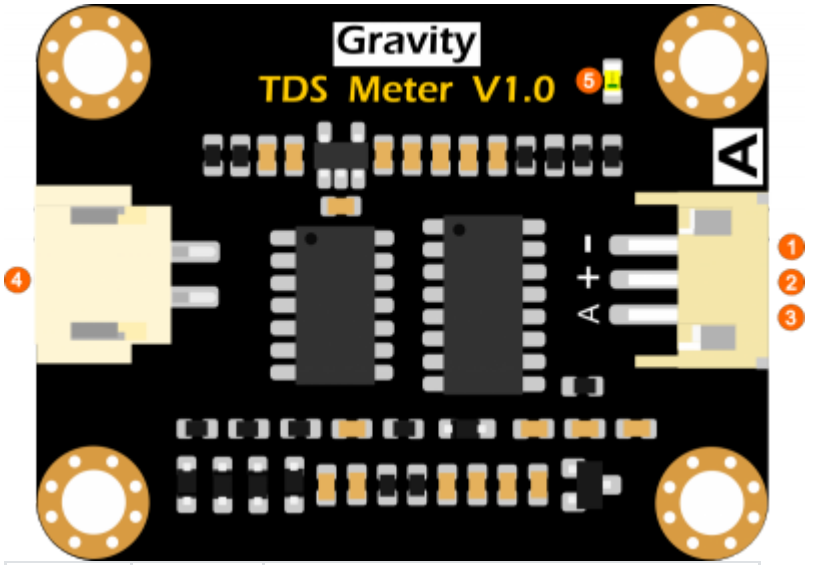
> **Attention:**
> **1.The probe can not be used in water above 55 degrees centigrade.**
> **2.The probe can not be left too close to the edge of the container, otherwise it will affect the reading.**
> **3.The head and the cable of the probe are waterproof, but the connector and the signal transmitter 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 °C)
  - 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



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

Basic Tutorial

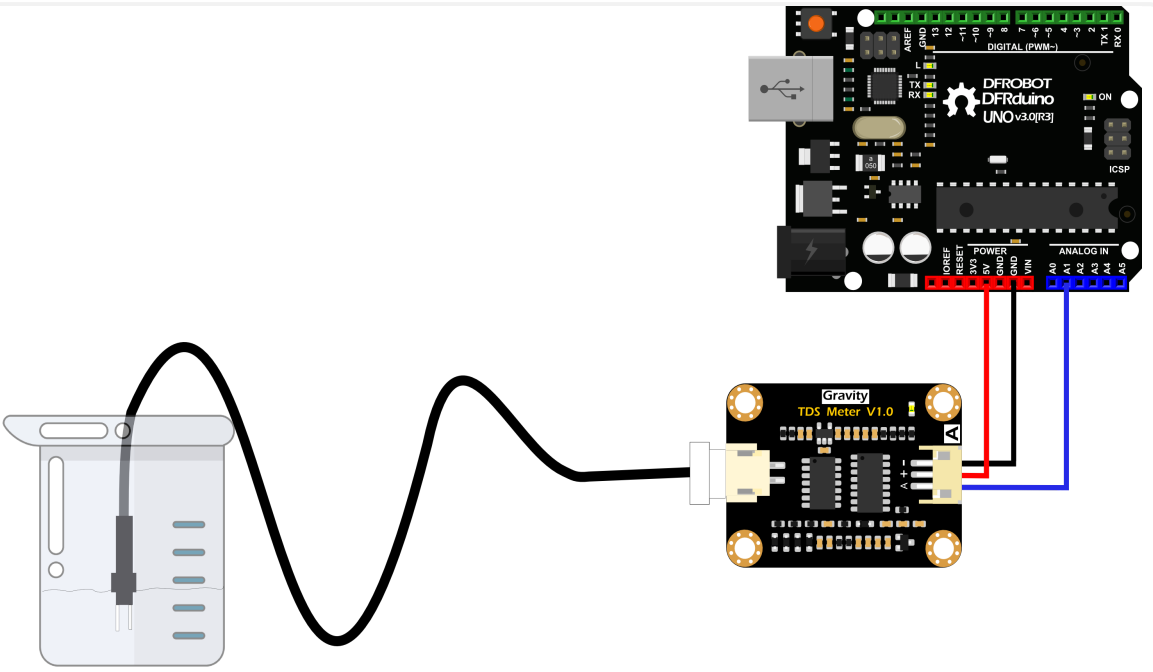
This tutorial will show you how to measure the TDS value of the water. Please read this tutorial carefully, and pay attention to the steps and details.

> **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®

Connection Diagram



Sample Code

```
/*************************************************
 DFRobot Gravity: Analog TDS Sensor / Meter For Arduino
 <https://www.dfrobot.com/wiki/index.php/Gravity:_Analog_TDS_Sensor_/_Meter_

 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 r
 2. More details, please click this link: <https://www.dfrobot.com/wiki/inde
 **************************************************/

#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 fr
int analogBufferTemp[SCOUNT];
int analogBufferIndex = 0,copyIndex = 0;
float averageVoltage = 0,tdsValue = 0,temperature = 25;
```

```
voidsetup(){Serial.begin(115200);pinMode(TdsSensorPin,INPUT);}voidloop(){
   static unsigned long analogSampleTimepoint =millis();if(millis()-analogSa
```

```
>40U)//every 40 milliseconds,read the analog value from the ADC{
     analogSampleTimepoint =millis();
     analogBuffer[analogBufferIndex]=analogRead(TdsSensorPin);//read the ana
     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 /102
```

```
;// read the analog value more stable by the median filtering algorithm, and
      float compensationCoefficient=1.0+0.02*(temperature-25.0);//temperatur
      float compensationVolatge=averageVoltage/compensationCoefficient;//ten
      tdsValue=(133.42*compensationVolatge*compensationVolatge*compensationV
```

```
-255.86*compensationVolatge*compensationVolatge +857.39*compensationVolatge)
```

```
0.5;//convert voltage value to tds value//Serial.print("voltage:");//Serial.
```

```
//Serial.print("V   ");Serial.print("TDS Value:");Serial.print(tdsValue,0);S
```

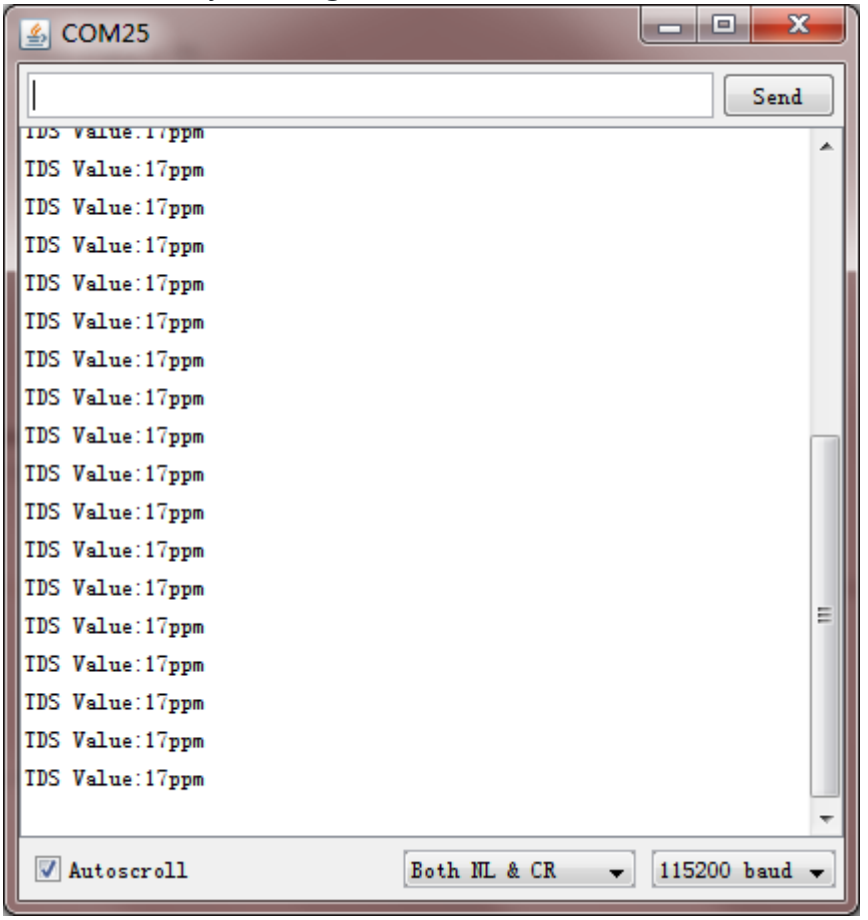
```
.println("ppm");}}intgetMedianNum(int bArray[],int iFilterLen){int bTab[iFil
```

```
];for(byte i =0; i<iFilterLen; i++)
      bTab[i]= bArray[i];int i, j, bTemp;for(j =0; j < iFilterLen -1; j++){
```

```
(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. Then wait for the reading to be stable, and you will get the TDS value of the water.



Advanced Tutorial

Through the basic tutorial the TDS value of the liquid can be easily measured. However, due to the individual differences of different TDS probe, differences of the main control board, and no onboard temperature compensation, the measured value can have some errors.Therefore, to obtain a more accurate TDS

value, calibration is required before measurement. In addition, it is recommended to connect a temperature sensor for temperature compensation to improve accuracy. Normally, the TDS value is half of the electrical conductivity value, that is: TDS = EC / 2. The wiring diagram is same as the basic tutorial. During the calibration, a liquid solution of known electrical conductivity or TDS value is needed, such as 1413us/cm standard buffer slution. If converted to a TDS value, it is about 707 ppm. The TDS value can also be measured using a TDS pen if you do not have a standard buffer solution. The following will demonstrate how to calibrate.

Download and install the **DFRobot Gravity TDS Sensor Library**. How to install Libraries in Arduino IDE?

Sample Code

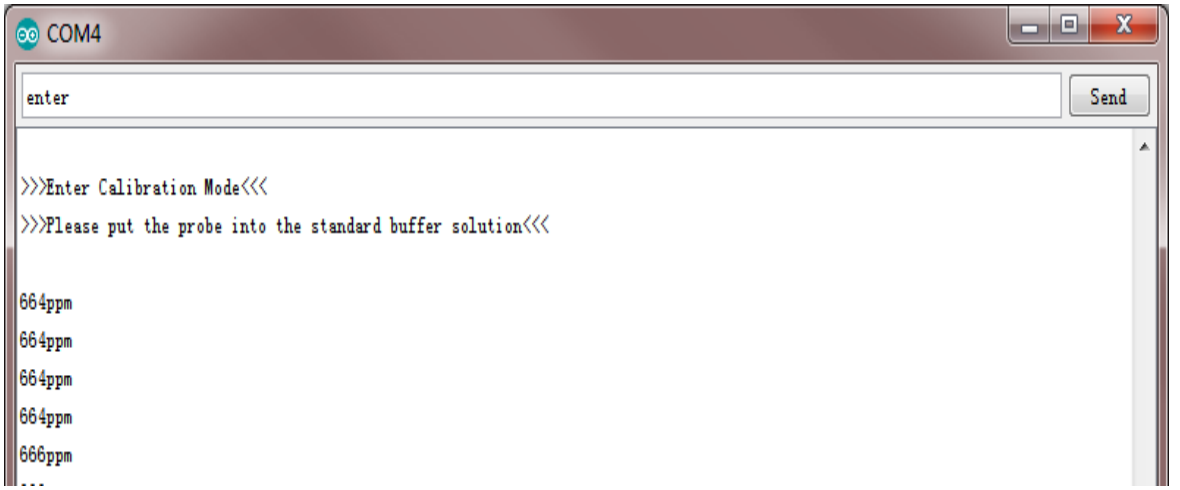
```
/***************************************************
 DFRobot Gravity: Analog TDS Sensor/Meter
 <https://www.dfrobot.com/wiki/index.php/Gravity:_Analog_TDS_Sensor_/_Meter_

 ****************************************************
 This sample code shows how to read the tds value and calibrate it with the
 707ppm(1413us/cm)@25^c standard buffer solution is recommended.

 Created 2018-1-3
 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 with Arduino IDE 1.0.5 r2 and 1.8.2.
 2. Calibration CMD:
     enter -> enter the calibration mode
     cal:tds value -> calibrate with the known tds value(25^c). e.g.cal:707
     exit -> save the parameters and exit the calibration mode
 ****************************************************/

#include <EEPROM.h>
#include "GravityTDS.h"

#define TdsSensorPin A1
GravityTDS gravityTds;
```

```
float temperature =25,tdsValue =0;voidsetup(){Serial.begin(115200);
    gravityTds.setPin(TdsSensorPin);
    gravityTds.setAref(5.0);//reference voltage on ADC, default 5.0V on Ardu
    gravityTds.setAdcRange(1024);//1024 for 10bit ADC;4096 for 12bit ADC
    gravityTds.begin();//initialization}voidloop(){//temperature = readTempe
    gravityTds.setTemperature(temperature);// set the temperature and execut
    gravityTds.update();//sample and calculate
    tdsValue = gravityTds.getTdsValue();// then get the valueSerial.print(td
```

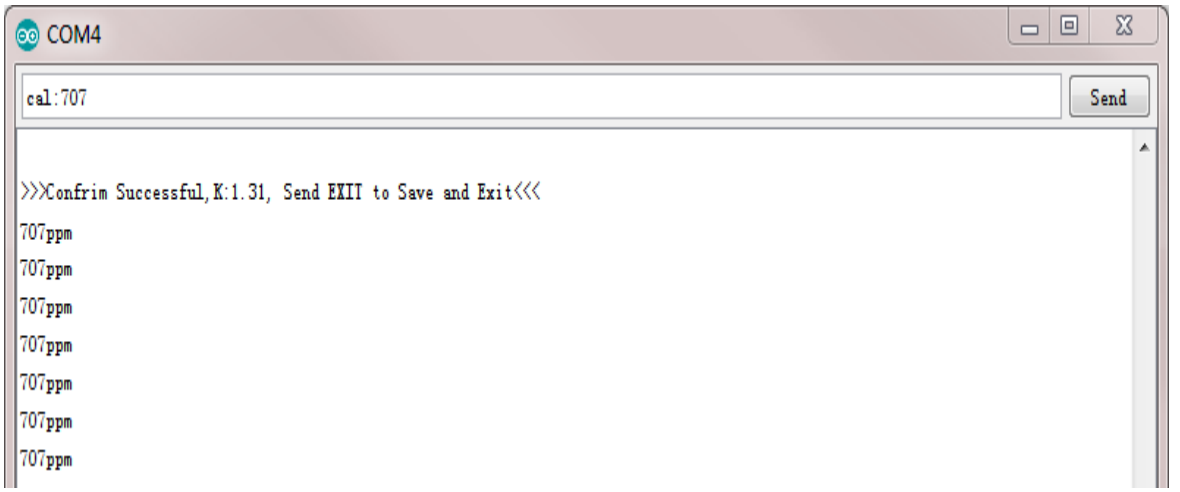
```
,0);Serial.println("ppm");delay(1000);}
```

Calibration Step

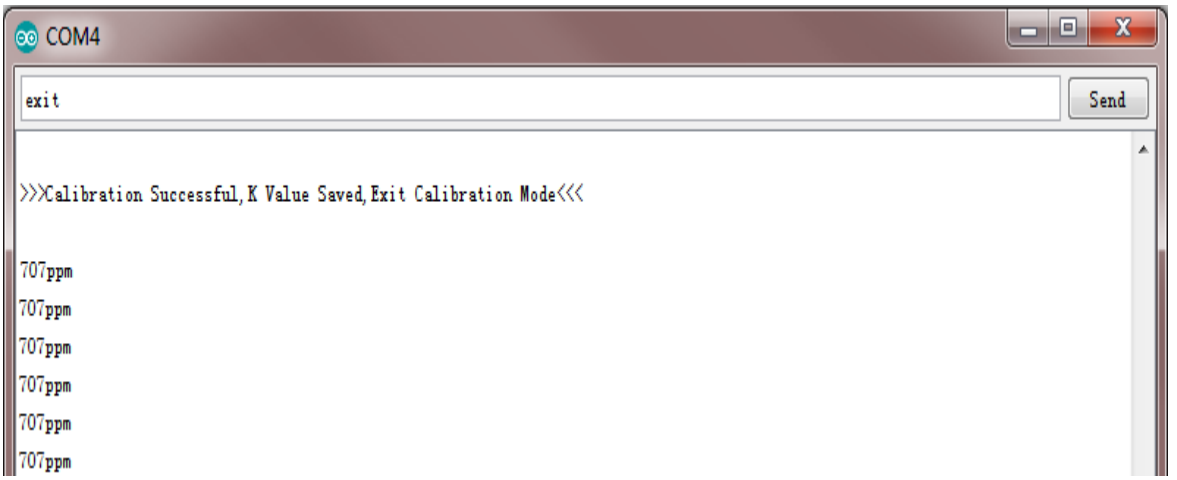
- Uploaded the sample code to your controller board, then open the serial monitor.
- Clean the TDS probe, then dry it with absorbent paper. Insert the probe into the buffer solution of known electrical conductivity or TDS value, then stir gently and wait for stable readings. If you do not have the standard buffer solution, a TDS pen can also measure the TDS value of the liquid solution.
- Input command "enter" to enter the calibration mode.



- Input command "cal:tds value" to calibrate the sensor.In this example, I use the 707ppm buffer solution, so I need to input command "cal:707".



- Input command "exit" to save and exit.



- After the calibration, you can use the TDS sensor in your application now.

FAQ

Q1. Does this sensor have a temperature sensor? How to make the temperature compensation?

A1. 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.

More Documents

- Schematic
- Layout with Dimension
- CD4060BM96 Datasheet
- LMV324A-SR Datasheet
- DFRobot Gravity TDS Sensor Library(Github)

Get **Gravity: Analog TDS Sensor/Meter for Arduino** from DFRobot Store or **DFRobot Distributor**.

Turn to the Top