CAP018-K01

MODULE CAPTEUR DE CONDUCTIVITE ET QUALITE DE L’EAU DFR0300-H K=1

SKU:DFR0300

Introduction

DFRobot Gravity: analog electrical conductivity meter V2 is specially used to measure the electrical conductivity of aqueous solutions, and then to evaluate the water quality. This sensor is often used in water culture, aquaculture, environmental water detection and other fields.

This product, as an upgraded version of electrical conductivity meter V1, which greatly improves the user experience and measurement precision. It supports 3~5v wide voltage input, and is compatible with 5V and 3.3V main control board. The output signal is filtered by hardware and has low jitter. The excitation source adopts an AC signal, which effectively reduces the polarization effect, improves the precision and prolongs the life of the probe. The software library uses a two-point calibration method, and can automatically identify the standard buffer solution, giving the user a simple and convenient sensor.

With this product, a main control board (such as Arduino) and the software library, you can quickly build an electrical conductivity meter annd immeditly begin plug and play without welding or soldering. DFRobot provides a variety of water quality sensor products with uniform sizes and interfaces, which not only meet the needs of various water quality testing but are also suitable for the DIY of multi-parameter water quality tester.

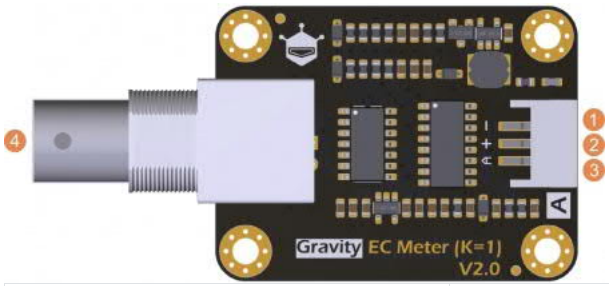
Conductivity is the reciprocal of an objects resistivity, which is related to the ability of the material to carry the current. In a liquid, the solution's conductivity is a measure of its ability to conduct electricity. Conductivity is an important parameter of water quality. It can reflect the extent of electrolytes present in water.

⚠	**The probe is a laboratory-grade probe. Do not immerse in liquid for a long time. Otherwise this will shorten the life of the probe.** **Platinum black layer is attached to the surface of the sheet metal in the probe. It should avoid any object touching it. It can only be washed with distilled water, otherwise, the platinum black layer will be damaged, resulting in the inaccurate measurement.**

Specification

- **Signal Conversion Board (Transmitter) V2**
  - Supply Voltage: 3.0~5.0V
  - Output Voltage: 0~3.4V
  - Probe Connector: BNC
  - Signal Connector: PH2.0-3Pin
  - Measurement Accuracy: ±5% F.S.
  - Board size: 42mm*32mm/1.65in*1.26in
- **Electrical Conductivity Probe**
  - Probe Type: Laboratory Grade
  - Cell Constant: 1.0
  - Support Detection Range: 0~20ms/cm
  - Recommended Detection Range: 1~15ms/cm
  - Temperature Range: 0~40°C
  - Probe Life: >0.5 year (depending on frequency of use)
  - Cable Length: 100cm

Board Overview



Num	Label	Description
1	-	Power GND(0V)
2	+	Power VCC(3.0~5.0V)

Num	Label	Description
3	A	Analog Signal Output(0~3.4V)
4	BNC	Probe Connector

Tutorial

This tutorial will demonstrate how to use this electrical conductivity meter for calibration and measurement. Please read each step carefully.

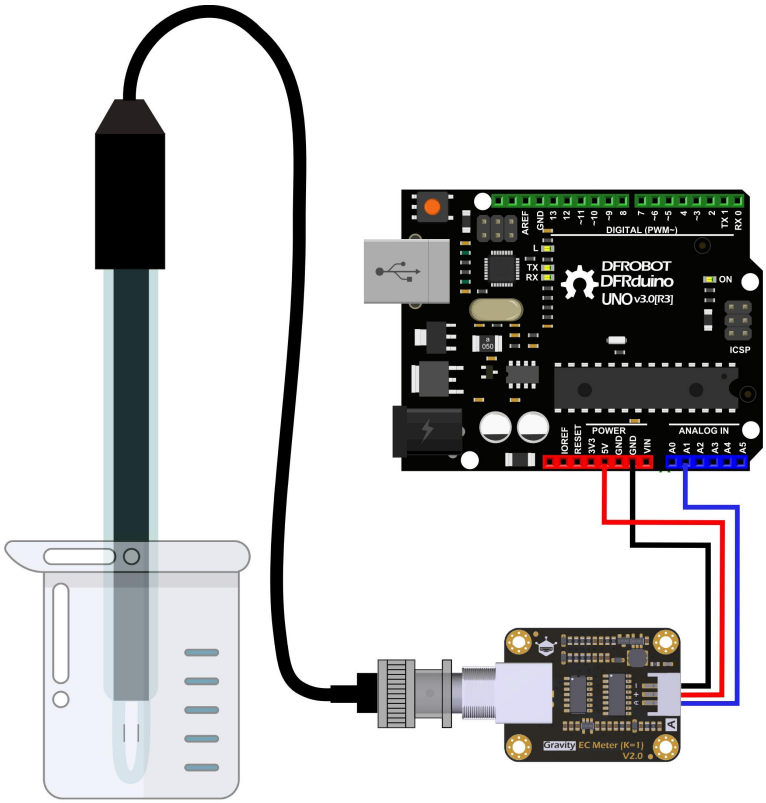
1. In order to ensure the measurement accuracy, it is strongly recommended to add a temperature sensor to measure the temperature and achieve automatic temperature compensation. DS18B20 waterproof temperature sensor can be used.
2. Before measuring another liquid, be sure to wash the probe and absorb residual water-drops with paper to prevent contamination of the liquid. You can flush the probe with distilled water.

Requirements

- **Hardware**
  - DFRduino UNO R3 (or similar) x 1
  - Analog Electrical Conductivity Meter Board(K=1) x1
  - Electrical Conductivity Probe(K=1) x1
  - Standard Buffer Solution 1413us/cm x1
  - Standard Buffer Solution 12.88ms/cm x1
  - Gravity 3pin Sensor Cable (or several DuPont cables) x1
  - Test Solution x1

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

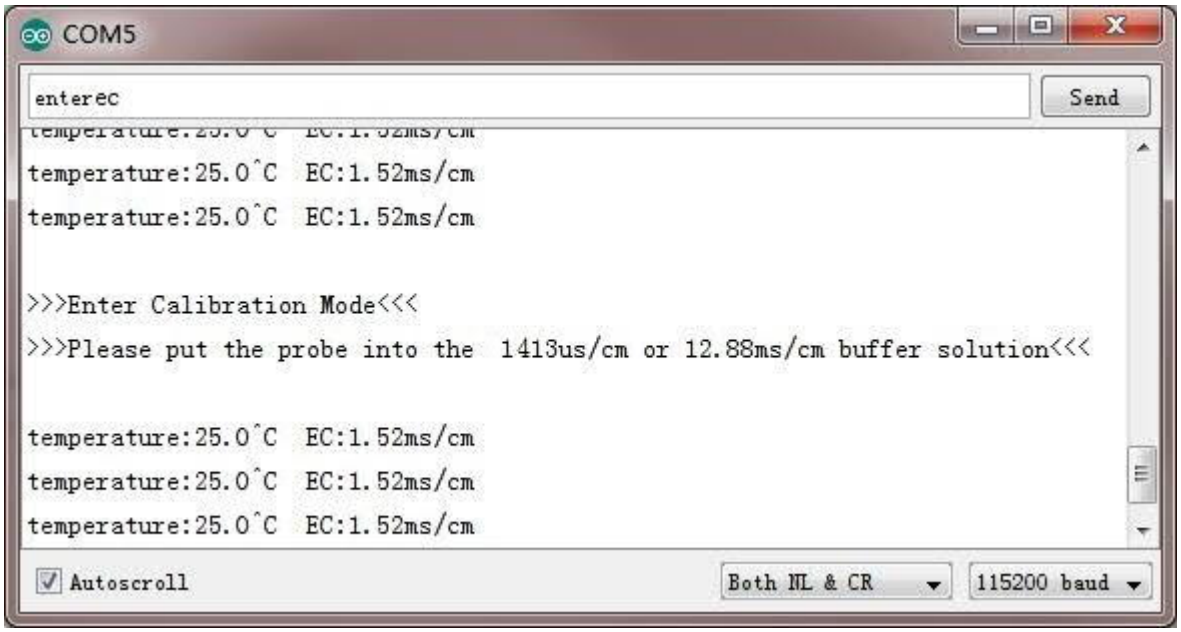
Connection Diagram



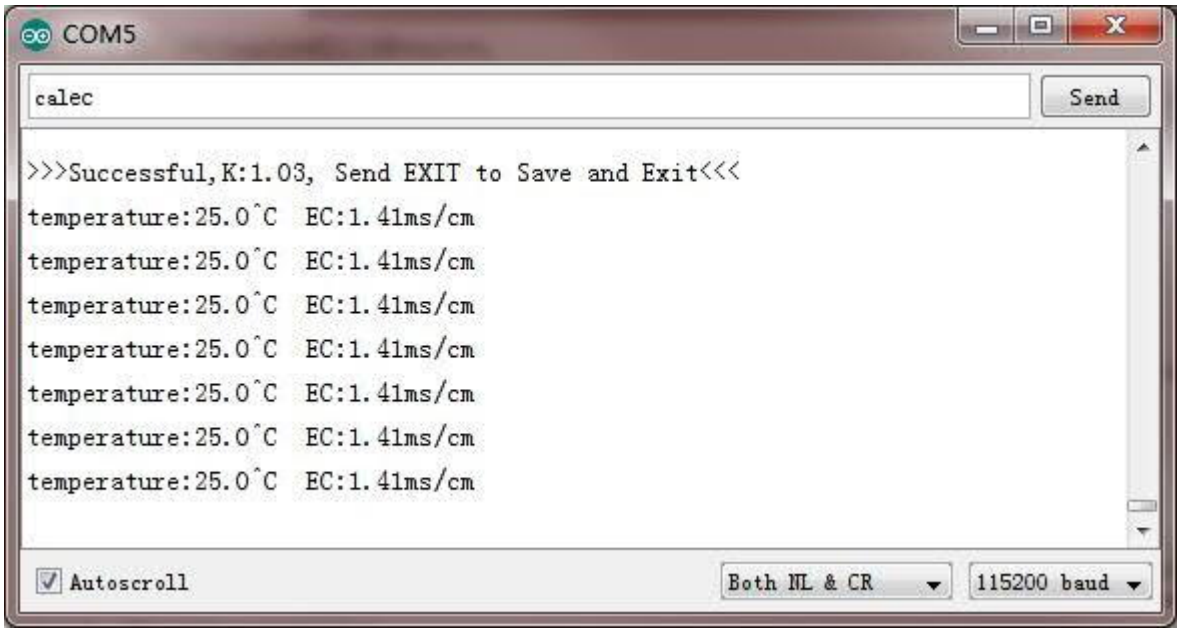
Calibration

To ensure accuracy, the probe needs to be calibrated for its first use and after not being used for an extended period of time. This tutorial uses two-point calibration and therefore requires standard buffer solutions of 1413us/cm and 12.88ms/cm. The following tutorial shows how to operate two-point calibration.

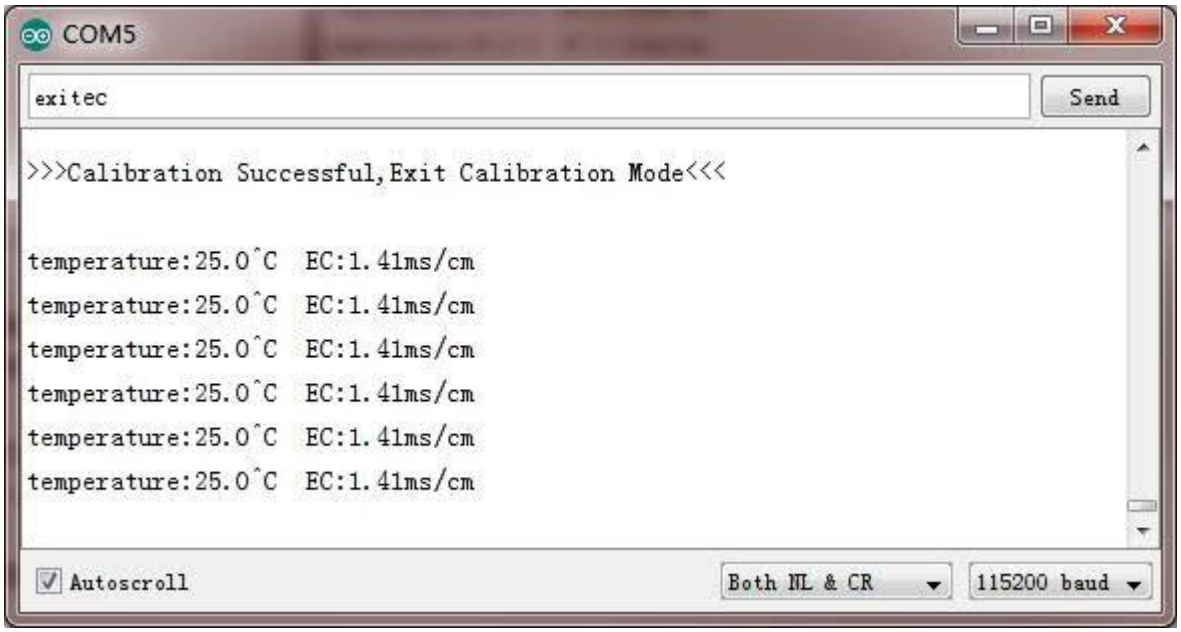
- **1.Upload the sample code to the Arduino board, then open the serial monitor; you can see the temperature and electrical conductivity. If you added a temperature sensor, be sure to write the corresponding function and call it.**
- **2.Wash the probe with distilled water, then absorb the residual water-drops with paper. Insert the probe into the 1413us/cm standard buffer solution, stir gently, until the values are stable.**
- **3.After the values are stable, the first point can be calibrated. Specific steps are as follows:**
    - 1. Input `enterec` command in the serial monitor to enter the calibration mode.



- 2. Input `calec` commands to start the calibration. The program will automatically identify which of the two standard buffer solutions is present: either 1413us/cm and 12.88ms/cm. In this step, the standard buffer solution is 1413us/cm.



- 3. After the calibration, input `exitec` command to save the relevant parameters and exit the calibration mode. **Note: Only after input `exitec` command in the serial monitor can the relevant parameters be saved.**



- 4.After the above steps, the first point calibration is completed. The second point calibration will be performed below.
    - **4. Wash the probe with distilled water, then absorb the residual water-drops with paper. Insert the probe into the 12.88ms/cm standard buffer solution, stir gently, until the values are stable.**
    - **5.After the values are stable, the second point can be calibrated. As same with the first calibration step, the specific steps are as follows:**

- 1. Input `enterec` command in the serial monitor to enter the calibration mode.
- 2. Input `calec` commands to start the calibration. The program will automatically identify which of the two standard buffer solutions is present: either 1413us/cm and 12.88ms/cm. In this step, the standard buffer solution is 12.88ms/cm.
- 3. After the calibration, input `exitec` command to save the relevant parameters and exit the calibration mode. **Note: Only after input `exitec` command in the serial monitor can the relevant parameters be saved.**

- 4.After the above steps, the second point calibration is completed.
    - **6.After completing the above steps, the two-point calibration is completed, and then it can be used for actual measurement. The relevant parameters in the calibration process have been saved to the EEPROM of the main control board.**

Sample Code

Please download DFRobot_EC Library first,then install it. How to install Libraries in Arduino IDE

```
/*
 * file DFRobot_EC.ino
 * @ https://github.com/DFRobot/DFRobot_EC
 *
 * This is the sample code for Gravity: Analog Electrical Conductivity Sensor /
Meter Kit V2 (K=1.0), SKU: DFR0300.
```

```
 * In order to guarantee precision, a temperature sensor such as DS18B20 is
needed, to execute automatic temperature compensation.
 * You can send commands in the serial monitor to execute the calibration.
 * Serial Commands:
 *   enterec -> enter the calibration mode
 *   calec -> calibrate with the standard buffer solution, two buffer
solutions(1413us/cm and 12.88ms/cm) will be automaticlly recognized
 *   exitec -> save the calibrated parameters and exit from calibration mode
 *
 * Copyright   [DFRobot](http://www.dfrobot.com), 2018
 * Copyright   GNU Lesser General Public License
 *
 * version  V1.0
 * date  2018-03-21
 */

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

#define EC_PIN A1
float voltage,ecValue,temperature = 25;
DFRobot_EC ec;

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

void loop()
{
    static unsigned long timepoint = millis();
    if(millis()-timepoint>1000U)  //time interval: 1s
    {
      timepoint = millis();
      voltage = analogRead(EC_PIN)/1024.0*5000;   // read the voltage
      //temperature = readTemperature();          // 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,1);
      Serial.print("^C  EC:");
      Serial.print(ecValue,2);
      Serial.println("ms/cm");
    }
    ec.calibration(voltage,temperature);          // calibration process by Serail
CMD
}

float readTemperature()
{
  //add your code here to get the temperature from your temperature sensor
}
```