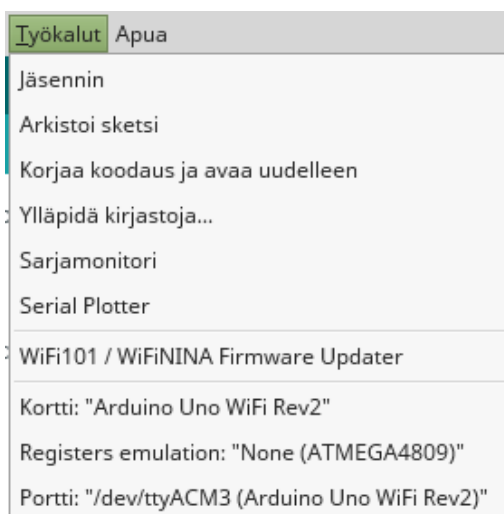


# Sähkönjohtavuussensorin kalibrointi

1. Liitä Arduino Uno Wifi Rev2 USB-johdolla tietokoneeseen.
2. Avaa Arduino IDE.
3. Tarkista, että kohdissa "Kortti", "Portti" sekä "Registers emulation" lukee alla olevat tekstit (Huom! Portin nimi riippuu omasta koneestasi / käyttöjärjestelmästä):



4. Lataa "DFRobot\_EC Library" kirjasto täältä:  
[https://github.com/DFRobot/DFRobot\\_EC/archive/master.zip](https://github.com/DFRobot/DFRobot_EC/archive/master.zip)
5. Asenna kirjastot täällä: Sketsi/Sisällytä kirjasto/Tuo .ZIP kirjasto.
6. Kirjoita seuraava koodi Arduino IDE:seen.

```
/*  
 * 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:
```

```

* enter -> enter the calibration mode
* cal -> calibrate with the standard buffer solution, two buffer solutions(1413us/cm and 12.88ms/cm) will be automatically recognized
* exit -> 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 A4
```

```
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");
  }
}
```

```

}

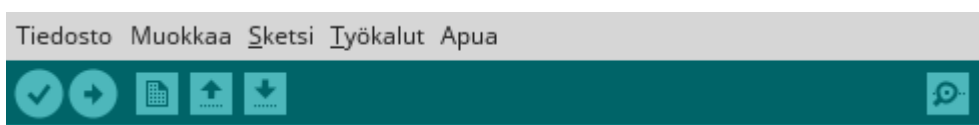
ec.calibration(voltage,temperature); // calibration process by Serail CMD

}

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

```

7. Tarkista koodi painamalla  sekä siirrä koodi Arduino Uno WiFi Rev2 –laitteeseen painamalla :



8. Paina sarjamonitori-näppäintä  ja vaihda NL arvoksi **115200** baudia. Tässä vaiheessa mittauksen pitäisi käynnistyä.

### ***-ITSE KALIBROINTI ALKAA TÄSTÄ-***

9. **OSA 1:** Pese sensorin pää tislattulla vedellä ja kuivaa se huolellisesti.
10. Kasta sensorin pää pakkauksen mukana tulleeseen **1413** us/cm nesteeseen ja odota, että arvot tasaantuvat.
11. Kirjoita sarjamonitoriin **ENTEREC** ja paina lähetä.
12. Kirjoita sarjamonitoriin **CALEC** ja paina lähetä.
13. Kirjoita sarjamonitoriin **EXITEC** ja paina lähetä.

14. **OSA 2:** Pese sensorin pää tislattulla vedellä ja kuivaa se huolellisesti.
15. Kasta sensorin pää pakkauksen mukana tulleetseen 12.88 ms/cm nesteeseen ja odota, että arvot tasaantuvat.
16. Kirjoita sarjamonitoriin **ENTEREC** ja paina lähetä.
17. Kirjoita sarjamonitoriin **CALEC** ja paina lähetä.
18. Kirjoita sarjamonitoriin **EXITEC** ja paina lähetä.

Jos kalibrointi epäonnistuu, niin silloin pitää putsata **EEPROM**. Putsaus onnistuu ajamalla alla oleva koodi laitteeseen. Tämän jälkeen voi ajaa kalibrointi koodin uudestaan.

```
#include <EEPROM.h>
#define KVALUEADDR 0x0A
void setup(){
    for(byte i = 0; i < 8; i++){
        EEPROM.write(KVALUEADDR i, 0xFF);
    }
}
void loop(){
}
```