

# RAK12023 WisBlock Soil Moisture Sensor Connector Module Datasheet

## Overview

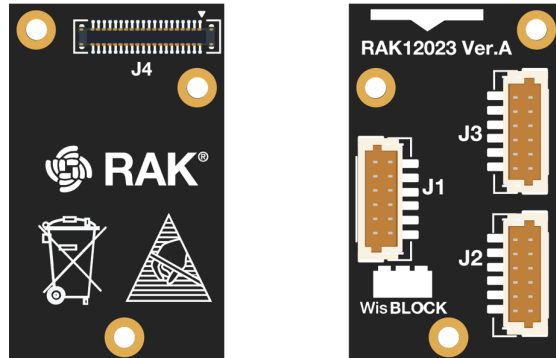


Figure 1: RAK12023 WisBlock Soil Moisture Sensor Connector

## Description

RAK12023 is a soil moisture connector module. It has one (1) standard WisBlock IO connector, which you can connect with the WisBlock Base such as RAK5005-O. It also has three (3) connectors dedicated to the RAK12035. RAK12035 is a separate soil moisture sensor probe that can be connected to the RAK12023 module. RAK12023 is capable of a connection of up to three RAK12035 soil moisture sensor probes simultaneously.

## Features

- A Separate sensor PCB module [RAK12035](#) is required.
- Three (3) available connectors for the soil sensor PCB module RAK12035.
- Module size: 15 x 25 mm

## Specifications

### Overview

### Mounting

The RAK12023 module can be mounted on the IO slot of the WisBlock Base board. **Figure 2** shows the mounting mechanism of the RAK12023 on a WisBlock Base module, such as the RAK5005-O.

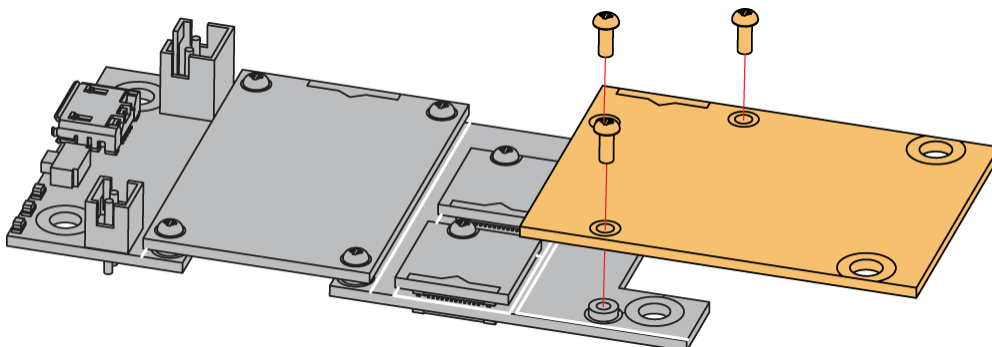


Figure 2: RAK12023 Mounting Mechanism on a WisBlock Base Module

## Hardware

The hardware specification is categorized into three (3) parts. It discusses the pinouts and their corresponding functions and diagrams of the module. Also, it shows the mechanical characteristics of the RAK12023 WisBlock

Soil Moisture Sensor Connector Module.

## Pin Definition

The RAK12023 module has a 40-pin WisConnector that is compatible to the WisBlock Base IO Slot. The pin order of the connector and the pinout definition is shown in **Figure 3**.

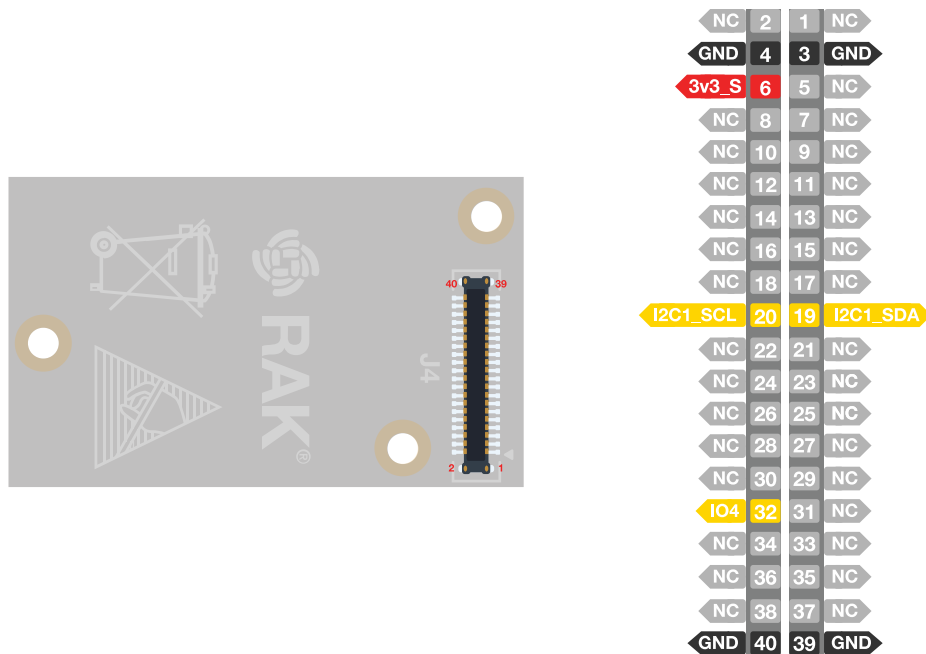


Figure 3: RAK12023 Pinout Schematic

### NOTE

- Only I2C related pin, IO4, 3V3\_S, and GND are connected to the WisConnector.
- 3V3\_S voltage output from the WisBlock Base that powers the RAK12023 module can be controlled by the WisBlock Core via WB\_IO2 (WisBlock IO2 pin). This makes the module ideal for low-power IoT projects since the WisBlock Core can totally disconnect the power of the RAK12023 module.

## Mechanical Characteristics

### Board Dimensions

Figure 4 shows the mechanical dimensions of the RAK12023 Module.

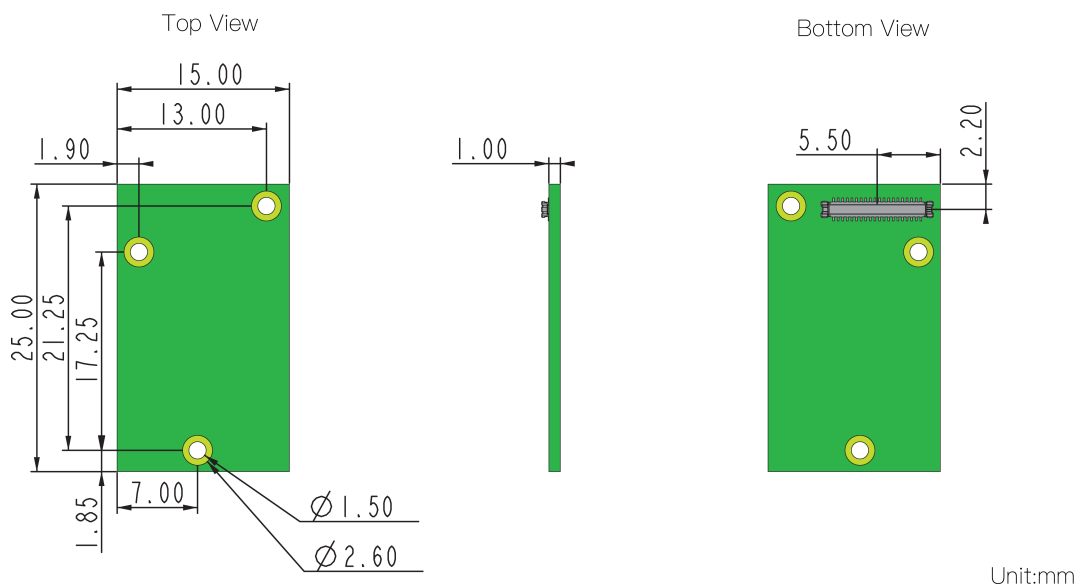


Figure 4: RAK12023 Mechanical Dimensions

## WisConnector PCB Layout

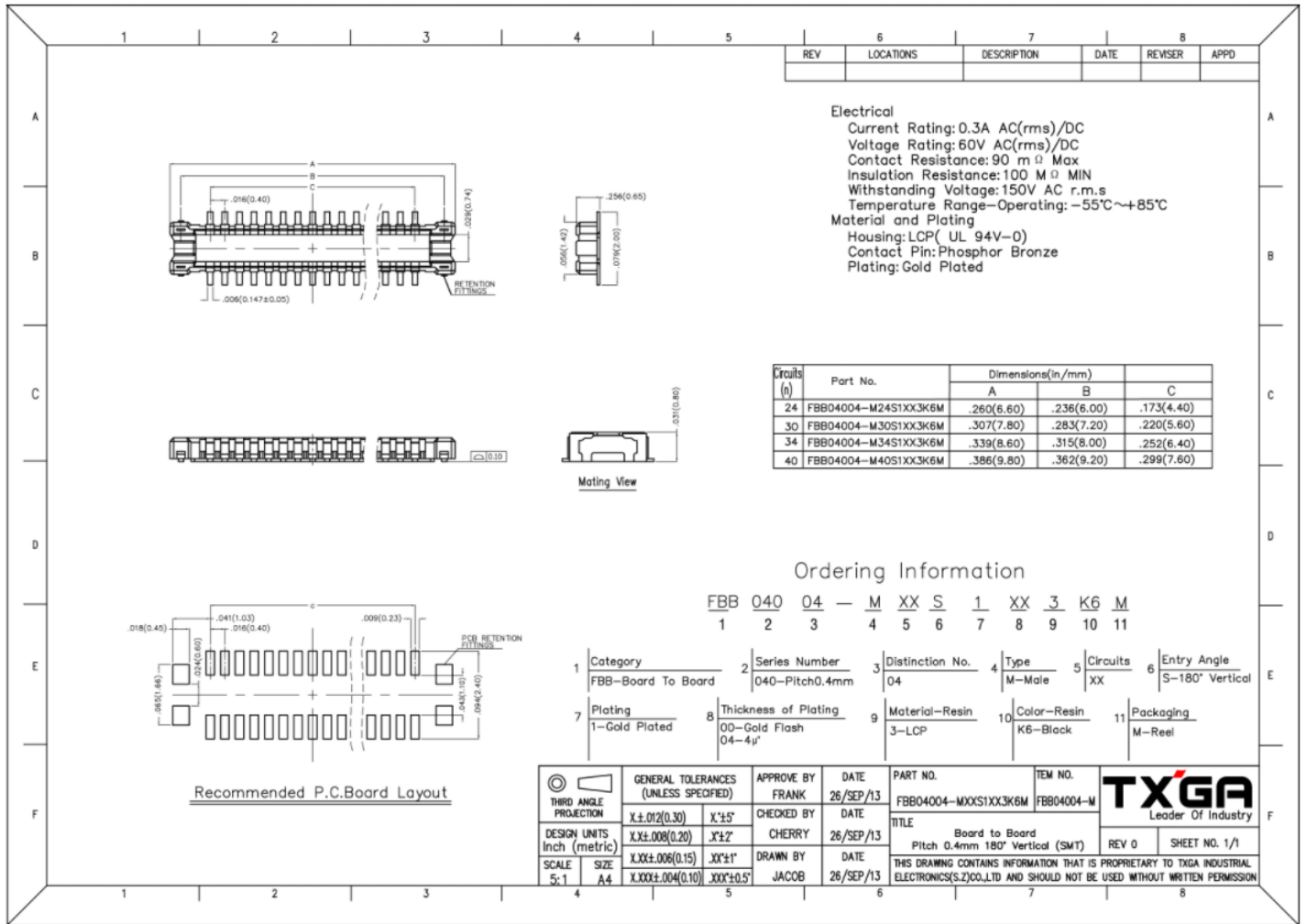


Figure 5: WisConnector PCB Footprint and Recommendations

## Schematic Diagram

Figure 6 shows the schematic of the RAK12023 module.

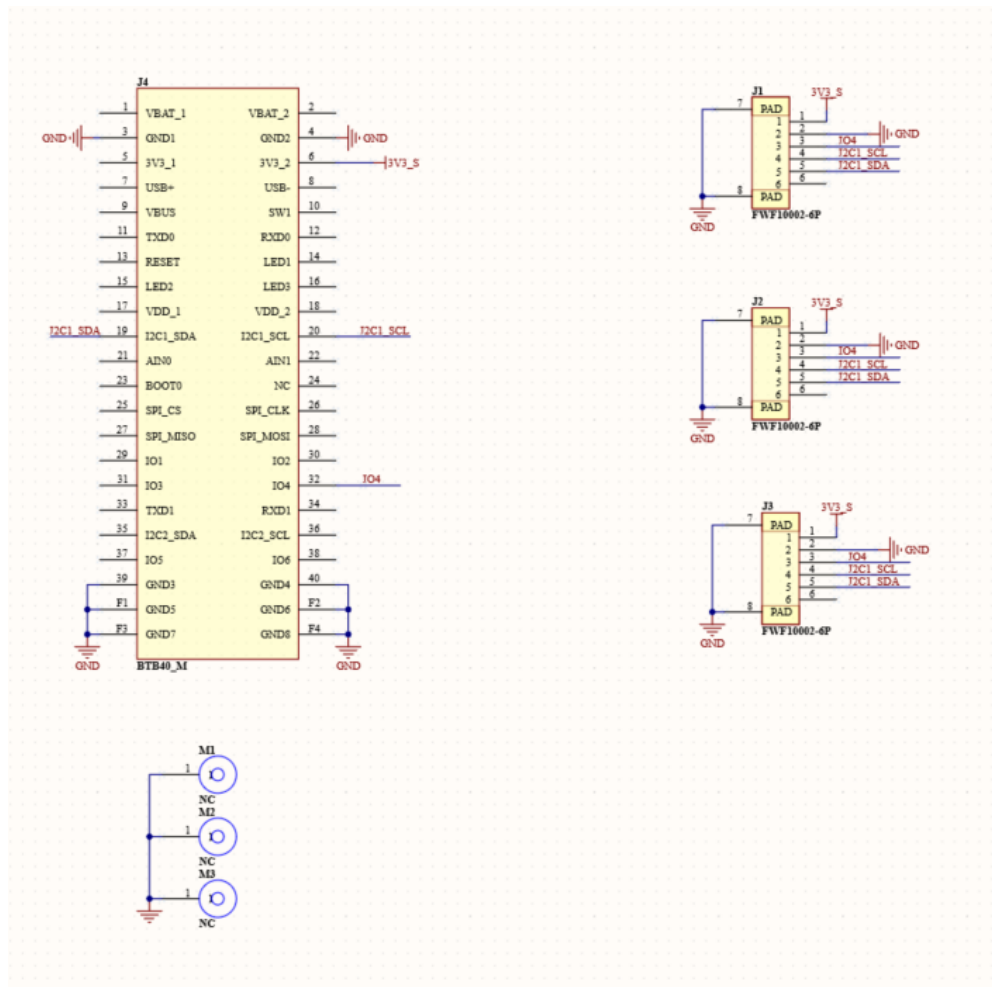


Figure 6: RAK12023 WisBlock Module Schematics

# RAK12023 Quick Start Guide






## Prerequisite

---



### What Do You Need?

Before going through each and every step on using the RAK12023 WisBlock module, make sure to prepare the necessary items listed below:

### Hardware

- [RAK12023 & RAK12035 WisBlock Soil Moisture Sensor Module](#) 
- Your choice of [WisBlock Base](#) 
- Your choice of [WisBlock Core](#) 
- USB Cable
- [Li-Ion/LiPo battery \(optional\)](#) 
- [Solar charger \(optional\)](#) 

### Software

- Download and install the [ArduinoIDE](#)  .
- To add the RAKwireless Core boards on your Arduino board, install the RAKwireless Arduino BSP. Follow the steps in the [Github repo](#)  .

## Product Configuration

---

### Hardware Setup

RAK12023 is a soil moisture connector module. It has one (1) standard WisBlock IO connector, which you can connect with the WisBlock Base such as RAK5005-O. It also has three (3) connectors dedicated to the RAK12035. RAK12035 is a separate soil moisture sensor probe that can be connected to the RAK12023 module. RAK12023 is capable of a connection of up to three RAK12035 soil moisture sensor probes simultaneously.

For more information about RAK12023, refer to the [Datasheet](#).

The RAK12023 module gives information about:

- Soil Moisture
- Soil Temperature

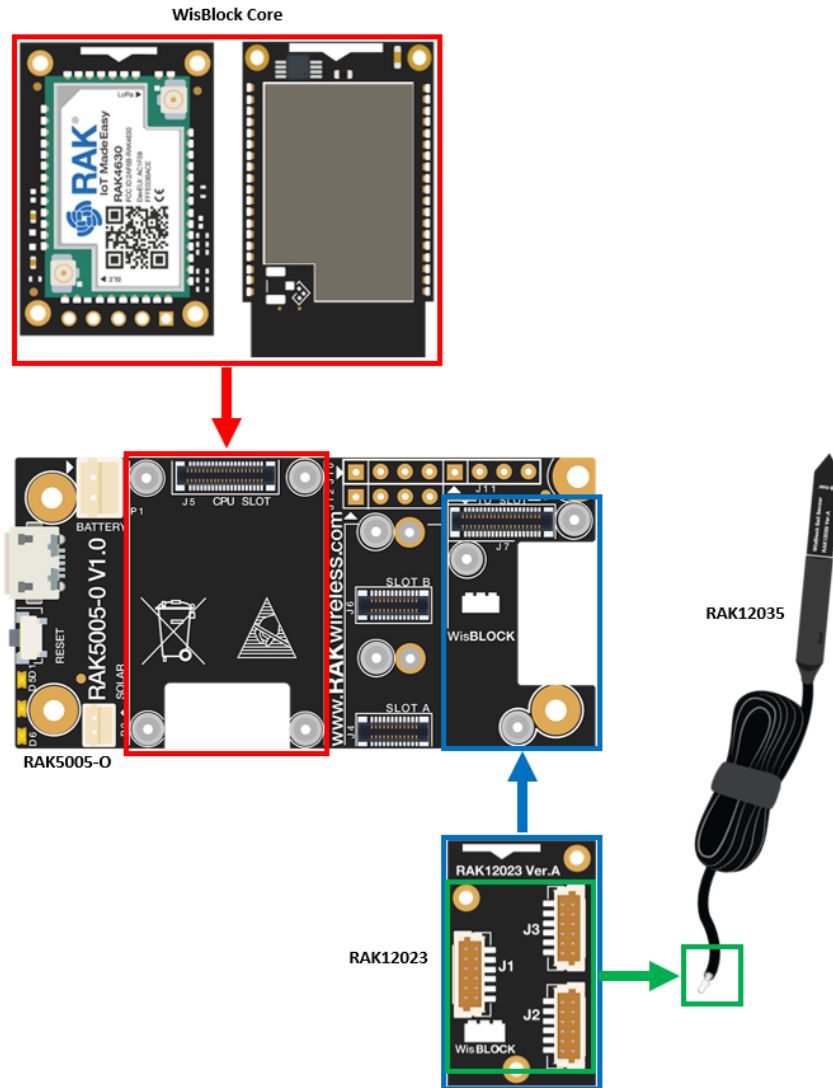


Figure 1: RAK12023 connection to WisBlock Base

## RAK12035 Soil Moisture Sensor Probe

A separate sensor PCB module RAK12035 is required. The RAK12035 is a sensor probe that you need to bury into the soil to measure the moisture of the soil.

For more information about RAK12035, refer to the [RAK12035 Soil Moisture Sensor Overview](#) .

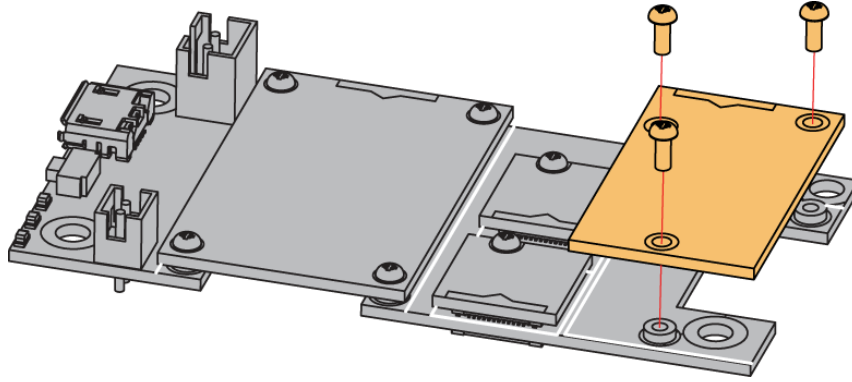


Figure 2: RAK12035 Soil Moisture Sensor

## Assembling and Disassembling of WisBlock Modules

## Assembling

The RAK12023 module can be mounted on the IO slot of the WisBlock Base board, as shown in **Figure 3**. Also, always secure the connection of the WisBlock module by using compatible screws.

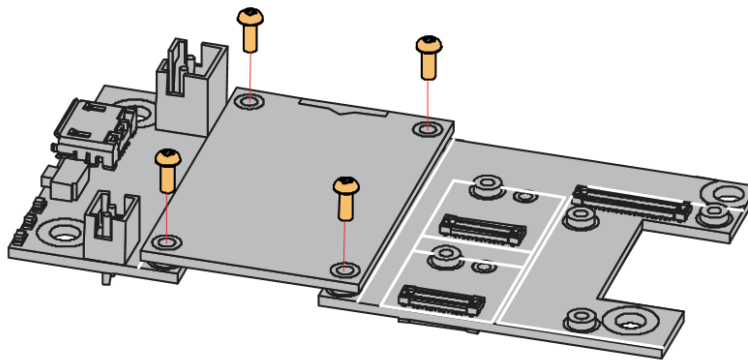


**Figure 3:** RAK12023 mounting connection to WisBlock Base module

## Disassembling

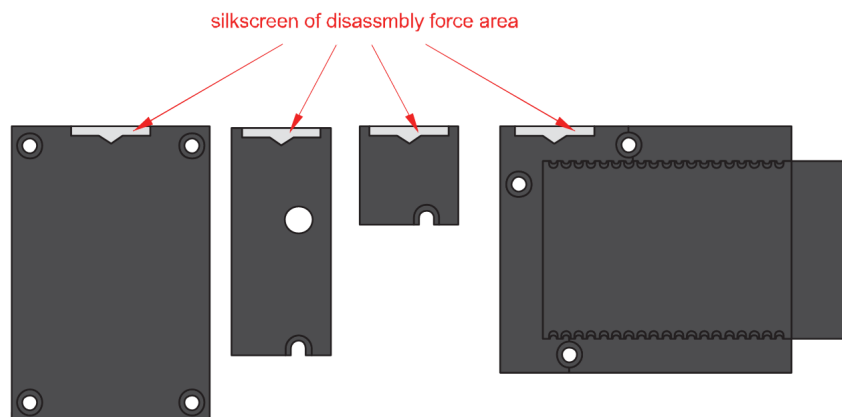
The procedure in disassembling any type of WisBlock modules is the same.

1. First, remove the screws.



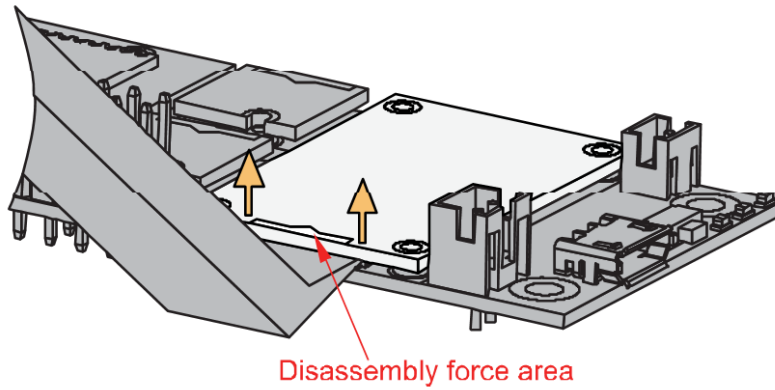
**Figure 4:** Removing screws from the WisBlock module

2. Once the screws are removed, check the silkscreen of the module to find the correct location where force can be applied.



**Figure 5:** Detaching silkscreen on the WisBlock module

3. Apply force to the module at the position of the connector, as shown in **Figure 6**, to detach the module from the baseboard.



**Figure 6:** Applying even forces on the proper location of a WisBlock module

 **NOTE**

If you will connect other modules to the remaining WisBlock Base slots, check on the [WisBlock Pin Mapper](#) tool for possible conflicts.

After all this setup, you can now connect the battery (optional) and USB cable to start programming your WisBlock Core.

 **WARNING**

- Batteries can cause harm if not handled properly.
- Only 3.7-4.2 V Rechargeable LiPo batteries are supported. It is highly recommended not to use other types of batteries with the system unless you know what you are doing.
- If a non-rechargeable battery is used, it has to be unplugged first before connecting the USB cable to the USB port of the board to configure the device. Not doing so might damage the battery or cause a fire.
- Only 5 V solar panels are supported. Do not use 12 V solar panels. It will destroy the charging unit and eventually other electronic parts.
- Make sure the battery wires match the polarity on the WisBlock Base board. Not all batteries have the same wiring.

## Software Configuration and Example

### Initial Test of the RAK12023 WisBlock Module

If you already installed the [RAKwireless Arduino BSP](#), the WisBlock Core and example code should now be available on the Arduino IDE.



**⚠ WARNING**

Before you use the Soil Moisture sensor, you need to calibrate it. The following steps below will show you the example sketch for calibration. The calibration values will be saved on the RAK12035 sensor itself so that you can read them back later from the sensor. In the first step, the sensor is calibrated by reading the sensor values while the sensor is in the air. In the next step submerge the sensor in water (only to the white line on the sensor). After the calibration is finished, the example sketch will save the calibration values directly in the sensor. The value measured in the air corresponds to 0% RH, the value measured in water corresponds to 100% RH.

**REMARK**

The calibration method in air and water is not very precise. To get more precise calibration values, the calibration should be performed in dry soil and water-saturated soil. Some information about such a calibration procedure is shown in [Capacitive Soil Moisture Sensor Calibration with Arduino](#) . Another source for such a calibration in the soil can be found in [Calibrating Soil Moisture Sensors](#) .

1. You need to select first the WisBlock Core you have.

**RAK4631 Board**

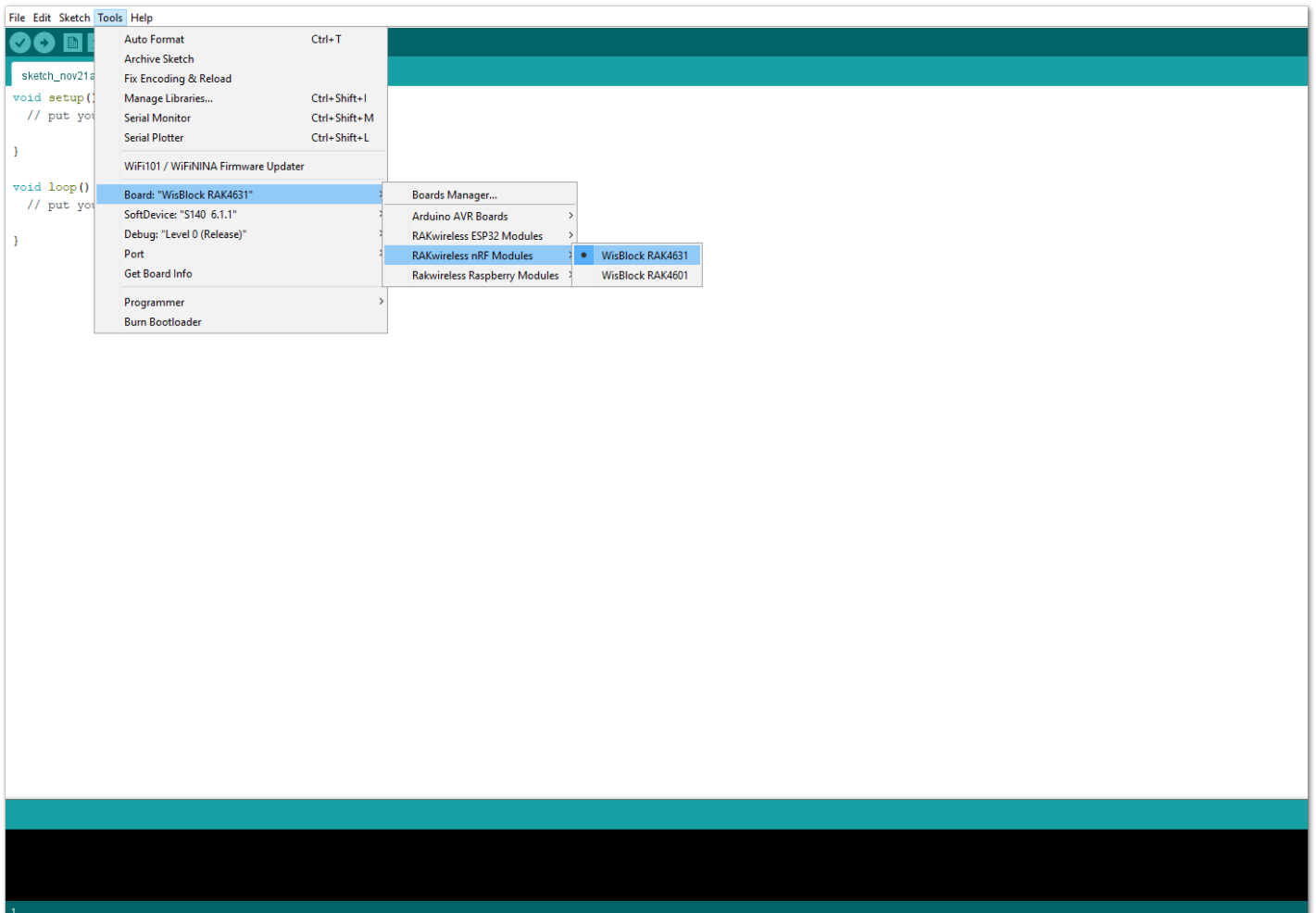
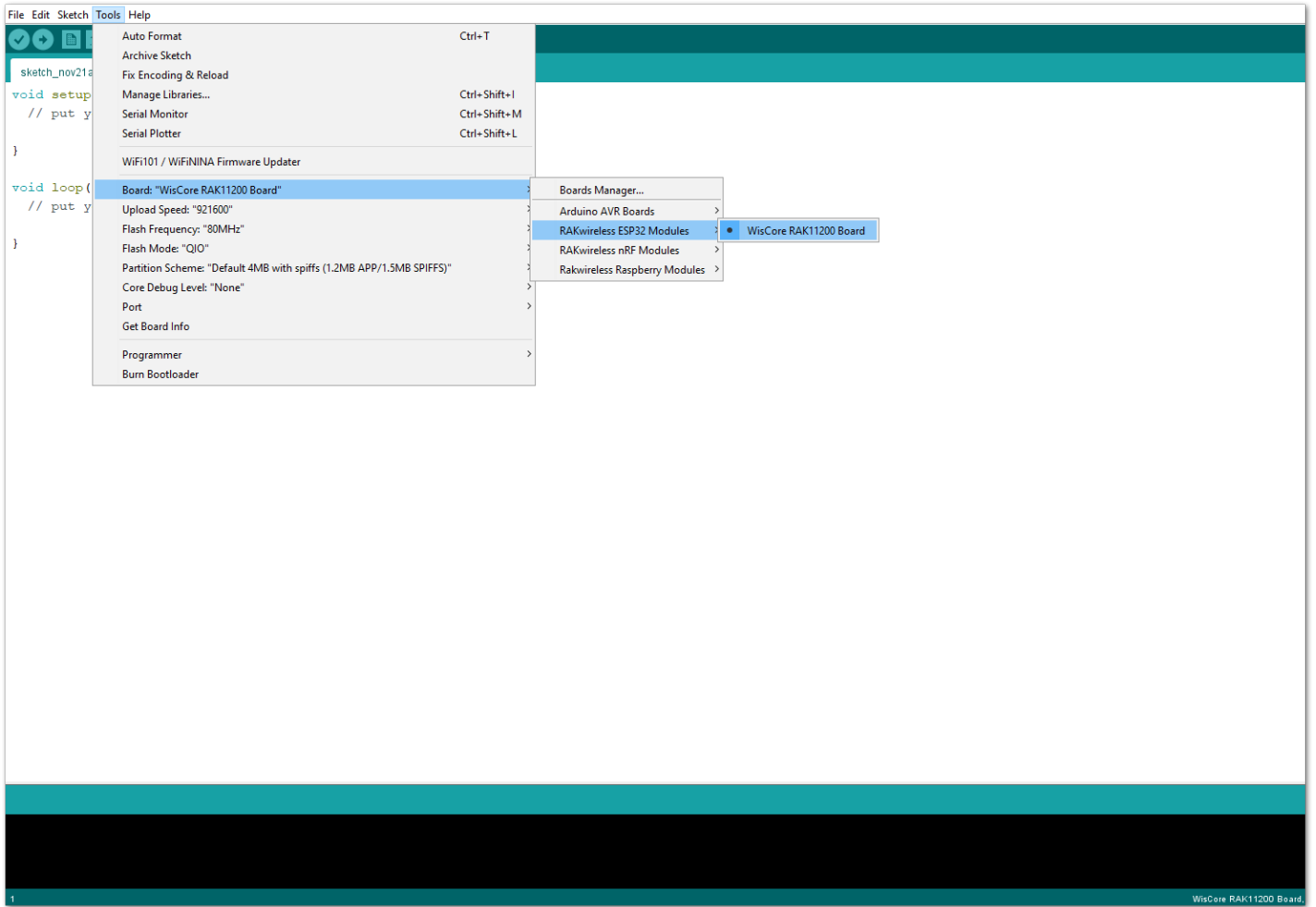


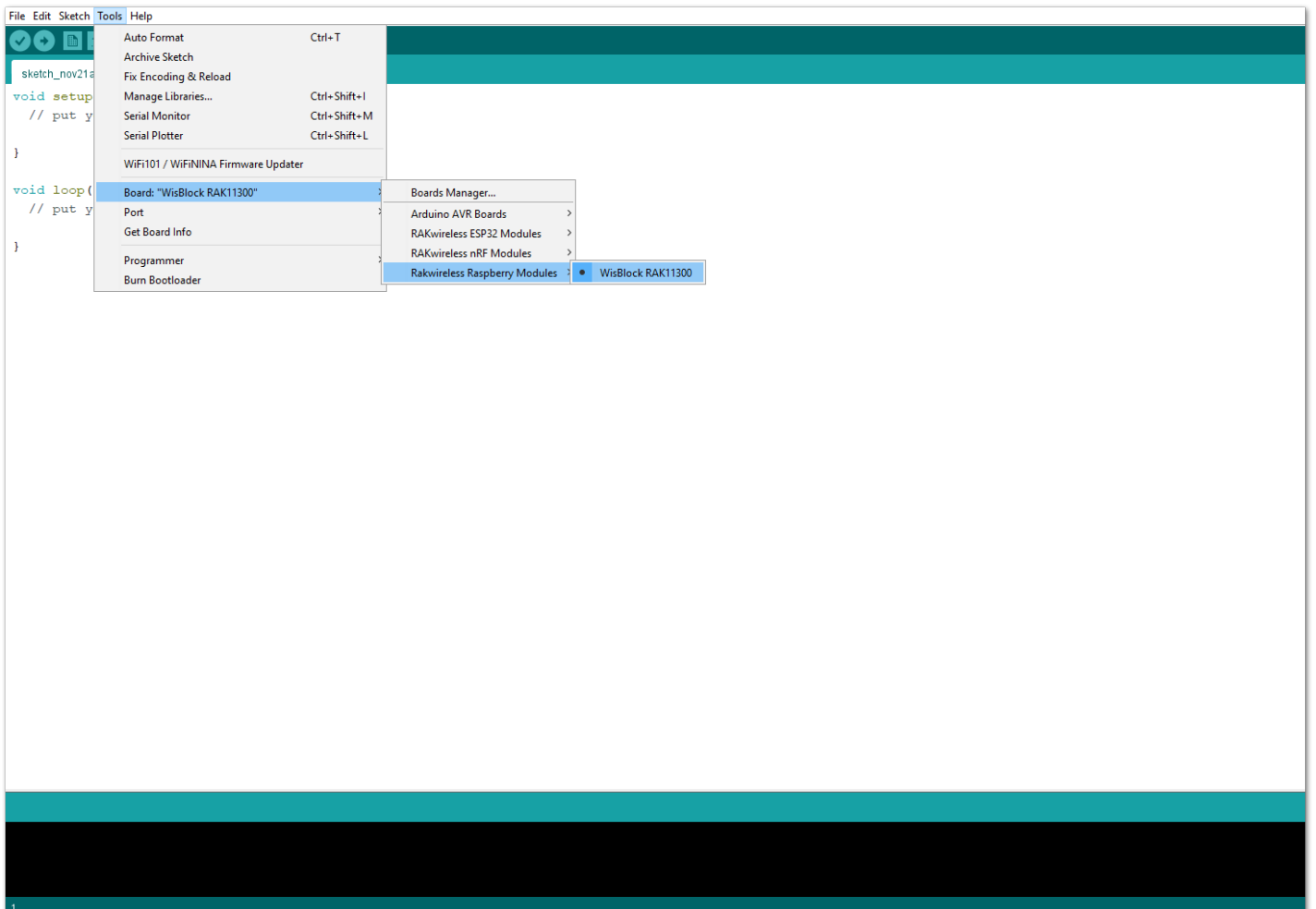
Figure 7: Selecting RAK4631 as WisBlock Core

**RAK11200 Board**



**Figure 8:** Selecting RAK11200 as WisBlock Core

## RAK11310 Board



**Figure 9:** Selecting RAK11310 as WisBlock Core

2. Next, copy the following sample code into your Arduino IDE:

```

/**
 * @file RAK12035_Calibration.ino
 * @author Bernd Giesecke (bernd.giesecke@rakwireless.com)
 * @brief Example for sensor calibration
 * @version 0.1
 * @date 2021-11-20
 *
 * @copyright Copyright (c) 2021
 *
 */
#include <Arduino.h>
#include <Wire.h>
#include "RAK12035_SoilMoisture.h" // Click here to get the library: http://librarymanager/All#RAK12035\_SoilMoisture

/** Sensor */
RAK12035 sensor;

/** Set these two values after the sensor calibration was done */
uint16_t zero_val = 845;
uint16_t hundred_val = 635;

void setup()
{
    // Initialize the built in LED
    pinMode(LED_GREEN, OUTPUT);
    digitalWrite(LED_GREEN, LOW);

    // Initialize the connection status LED
    pinMode(LED_BLUE, OUTPUT);
    digitalWrite(LED_BLUE, HIGH);

    // Initialize Serial for debug output
    Serial.begin(115200);

    time_t serial_timeout = millis();
    // On nRF52840 the USB serial is not available immediately
    while (!Serial)
    {
        if ((millis() - serial_timeout) < 5000)
        {
            delay(100);
            digitalWrite(LED_GREEN, !digitalRead(LED_GREEN));
        }
        else
        {
            break;
        }
    }
    digitalWrite(LED_GREEN, LOW);

    Wire.begin();

    // Initialize sensor
    sensor.begin();

    // Get sensor firmware version
    uint8_t data = 0;
    sensor.get_sensor_version(&data);
    Serial.print("Sensor Firmware version: ");
    Serial.println(data, HEX);
    Serial.println();
}

```



```
// Because each sensor has differences, it is required to calibrate the sensors
// First get the values of a sensor in dry air
Serial.println("Sensor Calibration");
Serial.println("Keep the sensor in a dry environment (air) and press ENTER to start the first");
Serial.println("The calibration steps takes some time as an average is calculated out of 100");

Serial.println("\nPress ENTER when ready.");

while (!Serial.available())
{
    delay(200);
}
Serial.println("Start dry calibration.");
while (Serial.available())
{
    Serial.read();
}

uint16_t new_reading = 0;
uint16_t new_value = 0;
digitalWrite(LED_GREEN, LOW);
digitalWrite(LED_BLUE, HIGH);

sensor.get_sensor_capacitance(&new_value);
for (int readings = 0; readings < 100; readings++)
{
    sensor.get_sensor_capacitance(&new_reading);
    new_value += new_reading;
    new_value = new_value / 2;
    delay(250);
    digitalWrite(LED_GREEN, !digitalRead(LED_GREEN));
    digitalWrite(LED_BLUE, !digitalRead(LED_BLUE));
}
// Save the result
zero_val = new_value;
Serial.print("Calibration in dry air finished, average capacitance is ");
Serial.println(new_value);
Serial.println("=====");

Serial.println("Now submerge the sensor into water up to the white line on the sensor and press ENTER");
Serial.println("The calibration steps takes some time as an average is calculated out of 100");

Serial.println("\nPress ENTER when ready.");

while (!Serial.available())
{
    delay(200);
}
Serial.println("Start wet calibration.");
while (Serial.available())
{
    Serial.read();
}

new_value = 0;
digitalWrite(LED_GREEN, LOW);
digitalWrite(LED_BLUE, HIGH);

sensor.get_sensor_capacitance(&new_value);
for (int readings = 0; readings < 100; readings++)
{
    sensor.get_sensor_capacitance(&new_reading);
    new_value += new_reading;
}
```



```
        new_value = new_value / 2;
        delay(250);
        digitalWrite(LED_GREEN, !digitalRead(LED_GREEN));
        digitalWrite(LED_BLUE, !digitalRead(LED_BLUE));
    }
    // Save the result
    hundred_val = new_value;
    Serial.print("Calibration in water finished, average capacitance is ");
    Serial.println(new_value);
    Serial.println("=====");
    digitalWrite(LED_GREEN, LOW);
    digitalWrite(LED_BLUE, LOW);

    /** Here we are saving the calibration values in the sensor itself */
    sensor.set_dry_cal(zero_val);
    sensor.set_wet_cal(hundred_val);
}

void loop()
{
    // Read capacitance
    uint16_t capacitance = 0;
    sensor.get_sensor_capacitance(&capacitance);
    Serial.print("Soil Moisture Capacitance: ");
    Serial.println(capacitance);

    // Read moisture in %
    // after calibration, we get the Capacitance in air and in water, like zero_val and B. zero_val
    // So the humidity is humidity = (Capacitance-A) / ((B-A)/100.0)
    uint8_t moisture = 0;
    sensor.get_sensor_moisture(&moisture);
    Serial.print("Soil Moisture humidity(%): ");
    Serial.println(moisture);

    // Read temperature
    uint16_t temperature = 0;
    sensor.get_sensor_temperature(&temperature);
    Serial.print("Soil Moisture Temperature: ");
    Serial.print(temperature / 10);
    Serial.println(" *C");
    Serial.println("=====");
    delay(1000);
}
```

#### NOTE

The above sample code is the [RAK12035 Calibration](#) code. If you experience any error in compiling the example sketch, check the updated code for your WisBlock Core Module that can be found on the [RAK12035 WisBlock Example Code Repository](#) and this sample code in Github will work on all WisBlock Core.

Other sample code in the WisBlock Github Repository:

- [RAK12035 Change I2C Address](#)
- [RAK12035 Low Power](#)
- [RAK12035 Soil Moisture](#)

3. Once the example code is open, install the [RAK12035\\_SoilMoisture](#) library by clicking the yellow highlighted link, as shown in **Figure 10** and **Figure 11**.

```
File Edit Sketch Tools Help
sketch_feb08a $
/**
 * @file RAK12035_Calibration.ino
 * @author Bernd Giesecke (bernd.giesecke@rakwireless.com)
 * @brief Example for sensor calibration
 * @version 0.1
 * @date 2021-11-20
 *
 * @copyright Copyright (c) 2021
 */
#include <Arduino.h>
#include <Wire.h>
#include "RAK12035_SoilMoisture.h" // Click here to get the library: http://librarymanager/All#RAK12035\_SoilMoisture

/** Sensor */
RAK12035 sensor;

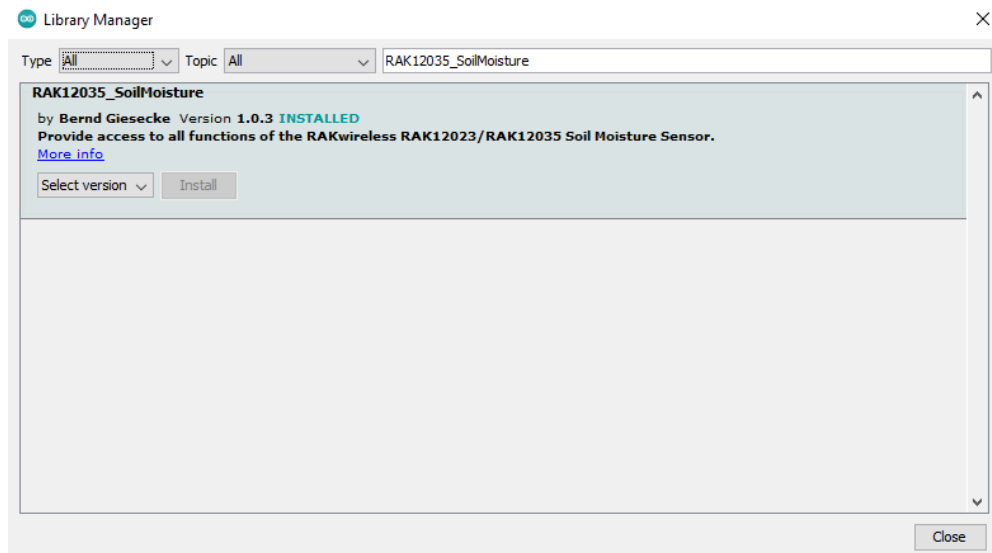
/** Set these two values after the sensor calibration was done */
uint16_t zero_val = 845;
uint16_t hundred_val = 635;

void setup()
{
  // Initialize the built in LED
  pinMode(LED_GREEN, OUTPUT);
  digitalWrite(LED_GREEN, LOW);

  // Initialize the connection status LED
  pinMode(LED_BLUE, OUTPUT);
  digitalWrite(LED_BLUE, HIGH);

  // Initialize Serial for debug output
  Serial.begin(115200);
}
```

**Figure 10:** Accessing the library used for RAK12023 Module



**Figure 11:** Installing the compatible library for RAK12023 Module

4. After the successful installation of the library, you can now select the right serial port and upload the code, as shown in **Figure 12** and **Figure 13**.

**NOTE**

If you're using the RAK11200 as your WisBlock Core, the RAK11200 requires the **Boot0** pin to be configured properly first before uploading. If not done properly, uploading the source code to RAK11200 will fail. Check the full details on the [RAK11200 Quick Start Guide](#) .

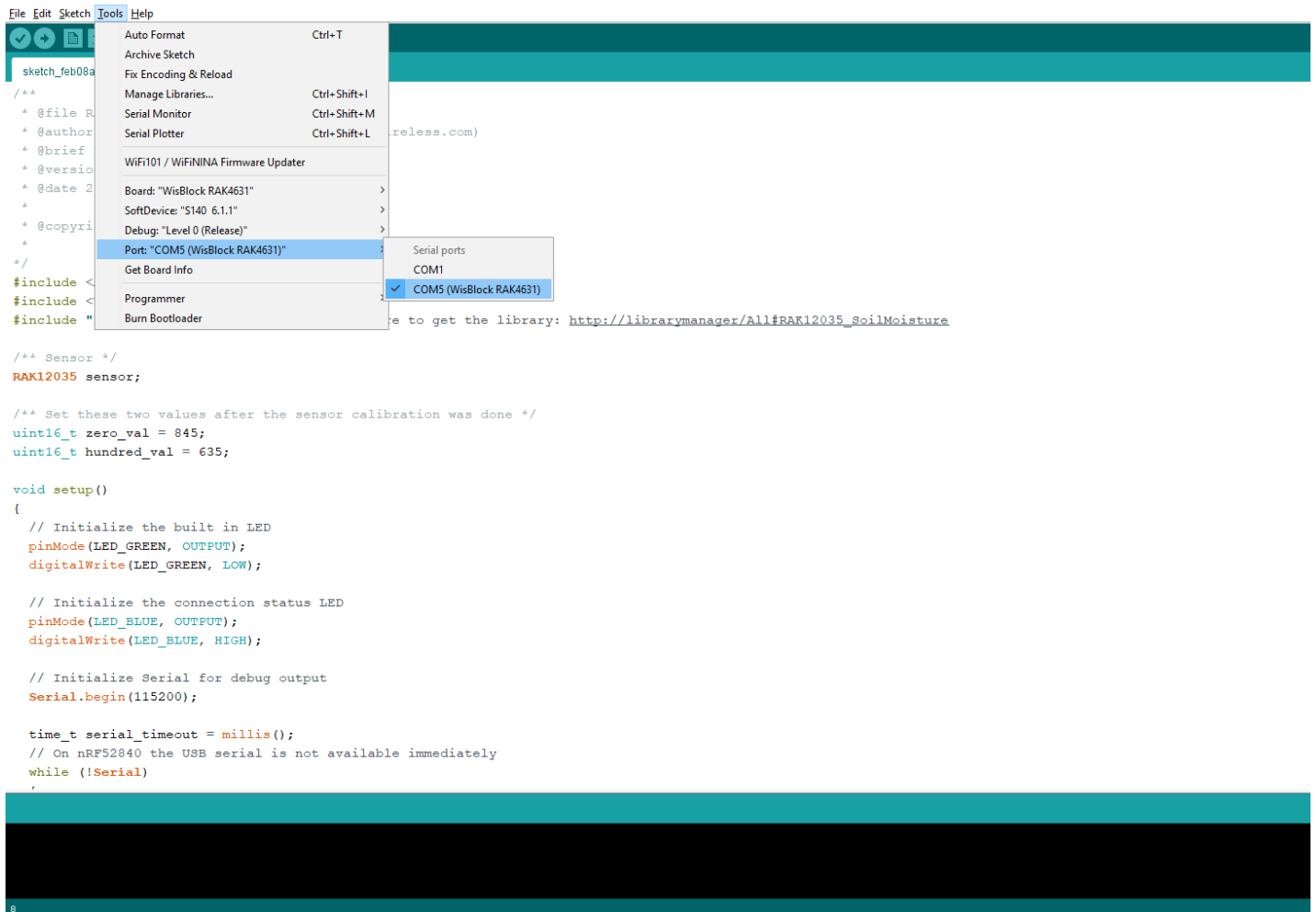


Figure 12: Selecting the correct Serial Port

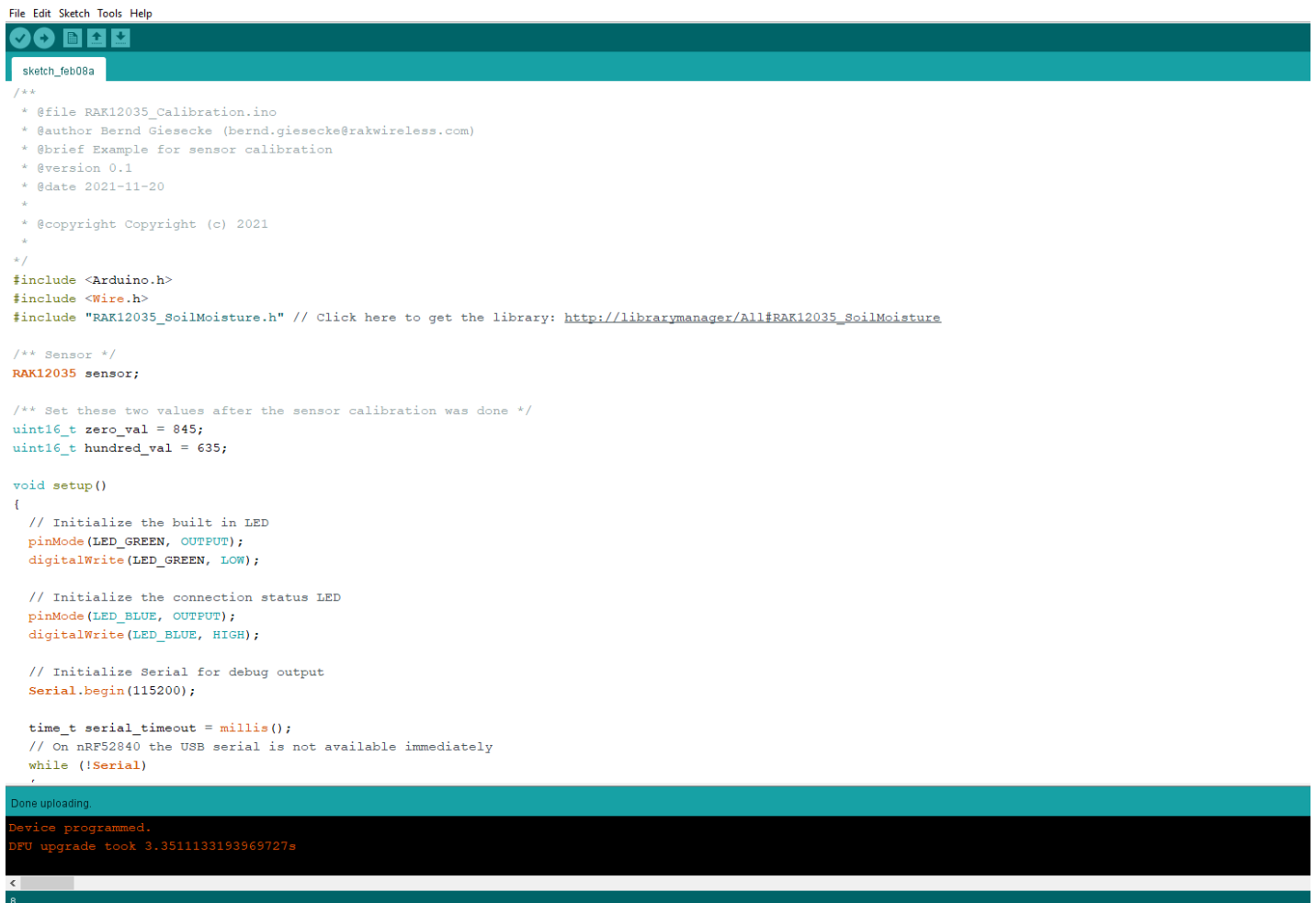


Figure 13: Uploading the RAK12023 example code

4. When you successfully uploaded the example sketch, open the Serial Monitor of the Arduino IDE and select the settings of the Serial Monitor, as shown below in **Figure 14**. Set it into "Newline" and set the baud rate correctly.

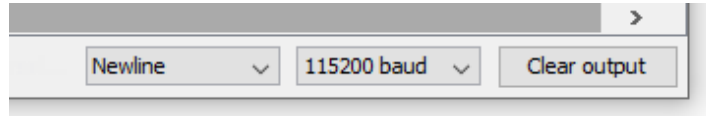


Figure 14: Setting up the Serial Monitor

5. Finally, after setting up the Serial Monitor, you'll now be able to see the readings as shown below in **Figure 15**. Follow the steps carefully for you to be able to calibrate your sensor correctly.

 **NOTE**

Upon calibration and before submerging the sensor into the water, following the [Recommended Depth](#) of the RAK12035 Sensor probe is **required**.

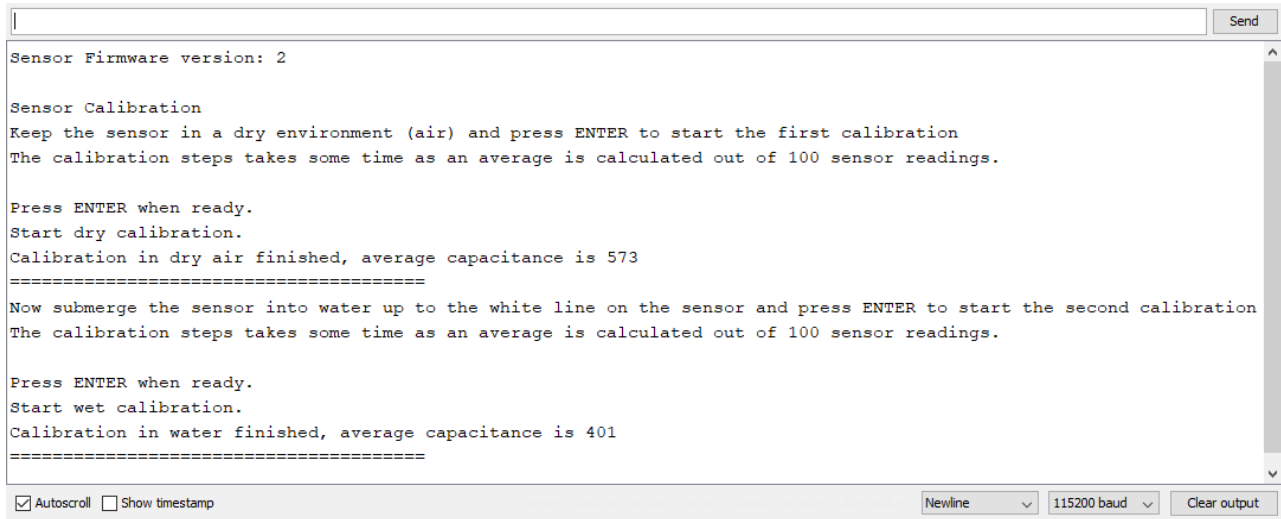


Figure 15: Calibrating the RAK12035 Sensor