RAK1906 WisBlock Environmental Sensor Datasheet

Overview

Description

The RAK1906 WisBlock Environmental Sensor module, part of the RAK WisBlock Sensor series, is a 4-in-1 digital sensor board that comprises gas, humidity pressure, and temperature sensor based on the Bosch® BME680 module. The RAK1906 is ideal for applications such as indoor air quality, home automation, and building IoT solutions.

Features

- **Temperature range**: -40 °C to 85 °C
- **Humidity range**: 0 to 100%
- **Pressure range**: 300 hPa to 1100 hPa
- **Gas sensor response time**: < 1 sec
- **Gas sensor output**: direct output of IAQ (Indoor Air Quality) index
- **Module size**: 10 x 10 mm

Specifications

Overview

Mounting

The RAK1906 module can be mounted on the slots: A, B, C, or D of a WisBlock Base board. Figure 2 shows the mounting mechanism of the RAK1906 on a WisBlock Base board, such as the RAK5005-O.

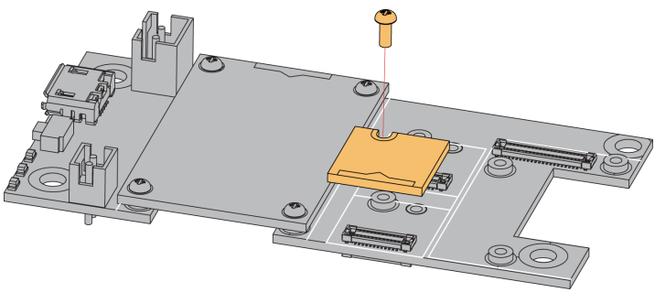


Figure 1: RAK1906 WisBlock Environmental Sensor Mounting

Hardware

Chipset

Vendor	Part number
BOSCH	BME680

Pin Definition

The RAK1906 WisBlock Environmental Sensor module comprises a standard WisIO connector. The WisIO connector allows the RAK1906 module to be mounted on a WisBlock Base board, such as RAK5005-O. The pin order of the connector and the definition of the pinout are shown in Figure 3.

> 📝 **NOTE:**
>
> Only the I2C related pin, VDD, and GND are connected to this module.

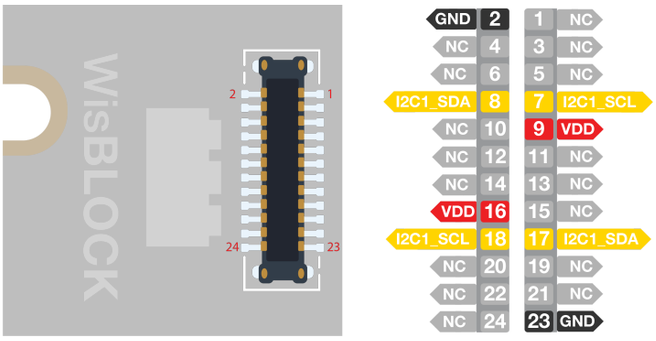


Figure 2: RAK1906 WisBlock Environmental Sensor Pinout

Sensors

Temperature Sensor

Parameter	Symbol	Condition	Min	Typ	Max	Unit
Operating Temperature Range	T_A	Operational	-40	25	85	°C
Supply Current	$I_{DD,T}$	1 Hz Forced Mode, Temperature Measurement Only		1.0		µA
Absolute Accuracy Temperature	$A_{T,25}$	25 °C		±0.5		°C
	$A_{T,full}$	0-65 °C		±1.0		°C
Output Resolution	R_T	API Output Resolution		0.01		°C

Humidity Sensor

Parameter	Symbol	Condition	Min	Typ	Max	Unit
Operating Range			-40	25	85	°C
			0		100	% r.H.
Full Accuracy Range			0		65	°C
			10		90	% r.H.
Supply Current	$I_{DD,H}$	1 Hz Forced Mode, Temperature and Humidity Measurement		2.1	2.8	µA
Absolute Accuracy	A_h	20-80% r.H., 25 °C, including hysteresis		±3		% r.H.

Pressure Sensor

Parameter	Symbol	Condition	Min	Typ	Max	Unit
Operating Temperature Range	T_A	Operational	-40	25	85	°C
		Full Accuracy	0		65	
Operating Pressure range	P	Full Accuracy	300		1100	hPa
Supply Current	$I_{DD,LP}$	1 Hz Forced Mode, Pressure and Temperature, Lowest Power		3.1	4.2	µA
Temperature Coefficient Of Offset	TCO_P	25-40 °C, 900 hPa		±1.3		Pa,K
				±10.9		cm/K
Absolute Accuracy Pressure	$A_{p,full}$	300-1100 hPa 0-65 °C		±0.6		hPa

AQI Sensor

Parameter	Symbol	Condition	Min	Typ	Max	Unit
Accuracy Status	AX_{IAQ}	Android Compatible	0		3	
IAQ Resolution	IAQ_{rs}			1		
IAQ Range	IAQ_{rg}		0		500	
Sensor-to-sensor Deviation	IAQ_{S2S}	All Operation Modes		±15% ±15		
Durability To Siloxanes	IAQ_{S2S}	Sensor-to-sensor Deviation		±15% ±15		
	IAQ_{drift}	Drift at Low & High Concentrations		±1% ±4		

Electrical Characteristics

Recommended Operating Conditions

Symbol	Description	Min.	Nom.	Max.	Unit
V_{DD}	Power supply for the module	1.71	1.8	3.6	V
I_{sleep}	Sleep current	-	0.15	-	uA
I_{DD}	Humidity Measure current	-	340	-	uA
I_{DD}	Pressure Measure current	-	714	-	uA
I_{DD}	Temperature Measure current	-	350	-	uA

Mechanical Characteristics

Board Dimensions

Figure 3 shows the dimensions and the mechanic drawing of the RAK1906 module.

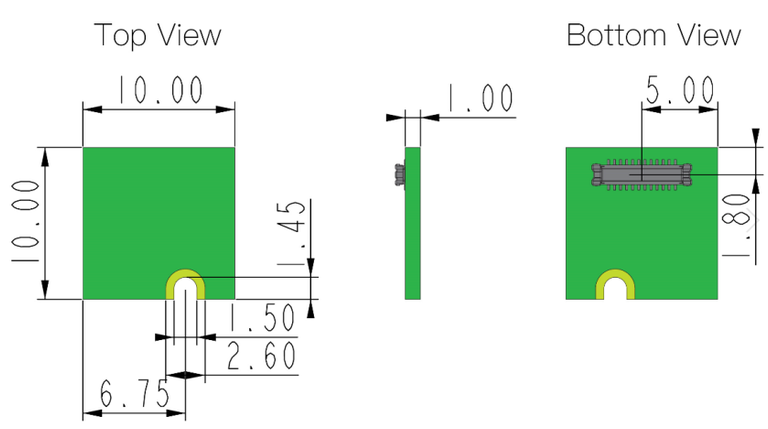
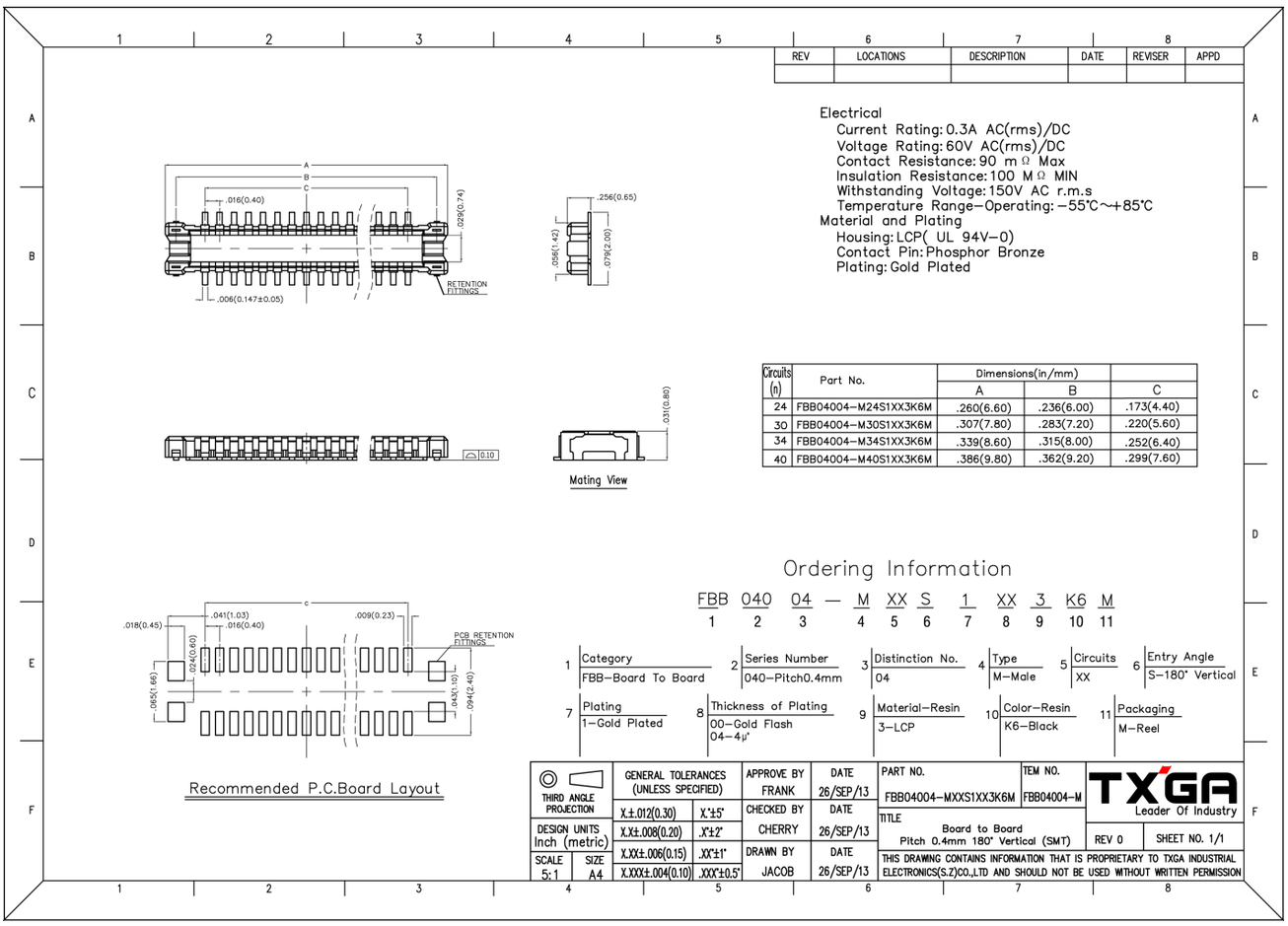


Figure 3: RAK1906 WisBlock Environmental Sensor Mechanic Drawing

WisConnector PCB Layout



Circuits (n)	Part No.	Dimensions(in/mm) A	B	C
24	FBB04004—M24S1XX3K6M	.260(6.60)	.236(6.00)	.173(4.40)
30	FBB04004—M30S1XX3K6M	.307(7.80)	.283(7.20)	.220(5.60)
34	FBB04004—M34S1XX3K6M	.339(8.60)	.315(8.00)	.252(6.40)
40	FBB04004—M40S1XX3K6M	.386(9.80)	.362(9.20)	.299(7.60)

Figure 4: WisConnector PCB footprint and recommendations

Schematic Diagram

Figure 5 shows the schematic of the RAK1906 module.

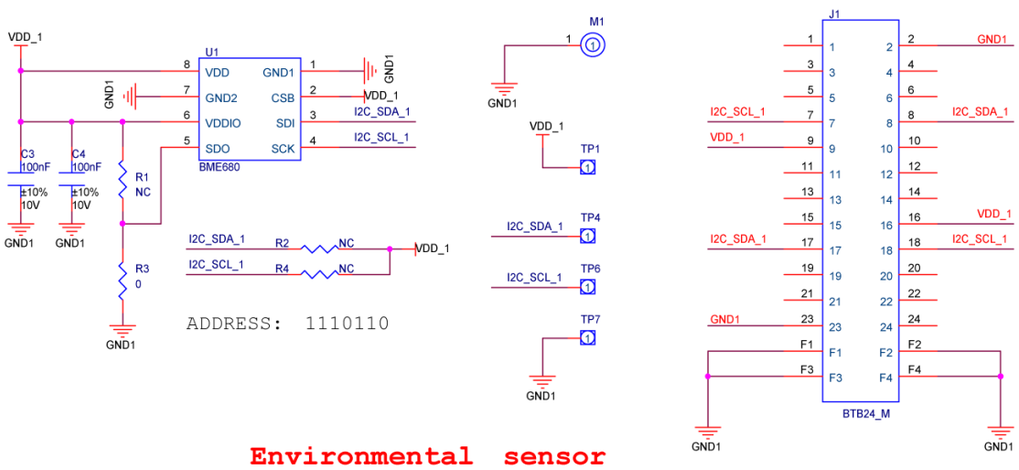


Figure 5: RAK1906 WisBlock Environmental Sensor schematics

Last Updated: 1/10/2022, 2:07:11 AM

RAK1906 Quick Start Guide

Prerequisite

What Do You Need?

Before going through each and every step on using RAK1906 WisBlock module, make sure to prepare the necessary items listed below:

Hardware

- RAK1906
- Your choice of WisBlock Base
- Your choice of WisBlock Core
- USB Cable
- Li-Ion/LiPo battery (optional)
- Solar charger (optional)

Software

- Download and install Arduino IDE .
- Install RAKwireless Arduino BSP and add the RAKwireless Core boards to your **Arduino Boards Manager**. Follow the steps in the Github repo .

Product Configuration

Hardware Setup

WisBlock can integrate this module which makes it easy to build up an environmental temperature and humidity data acquisition system.

For more information about RAK1906, refer to the Datasheet.

The RAK1906 module gives information about:

- Air pressure
- Air quality
- Ambient Air Temperature
- Environment Humidity

RAK1906 module can be connected to any slot of WisBlock Base to communicate with the WisBlock Core. It will work on **SLOT A to D**. Also, always secure the connection of the WisBlock module by using the compatible screws.

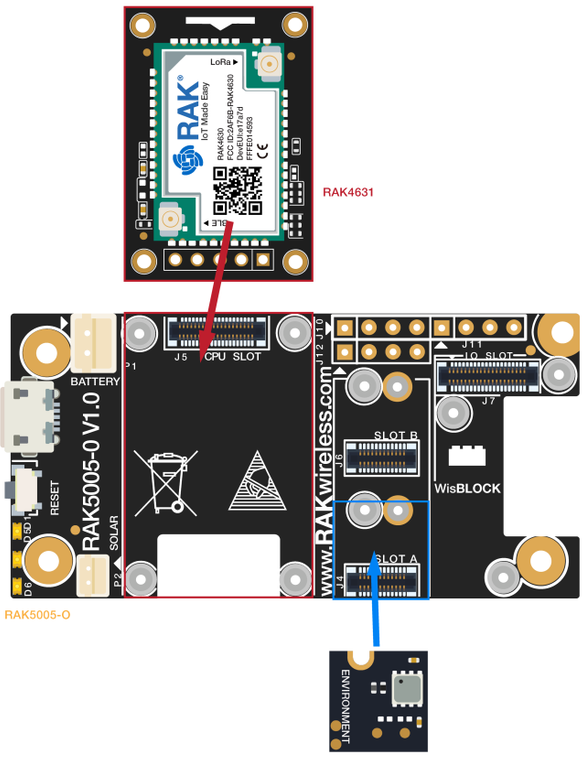


Figure 1: RAK1906 connection to WisBlock Base

Assembling and Disassembling of WisBlock Modules

Assembling

As shown in **Figure 2**, the location for Slot A, B, C, and D are properly marked by silkscreen. Slots C and D are located on bottom of WisBlock Base. Follow carefully the procedure defined in RAK5005-O module assembly/disassembly instructions to attach a WisBlock module. Once attached, carefully fix the module with a M1.2 x 3 mm screw.

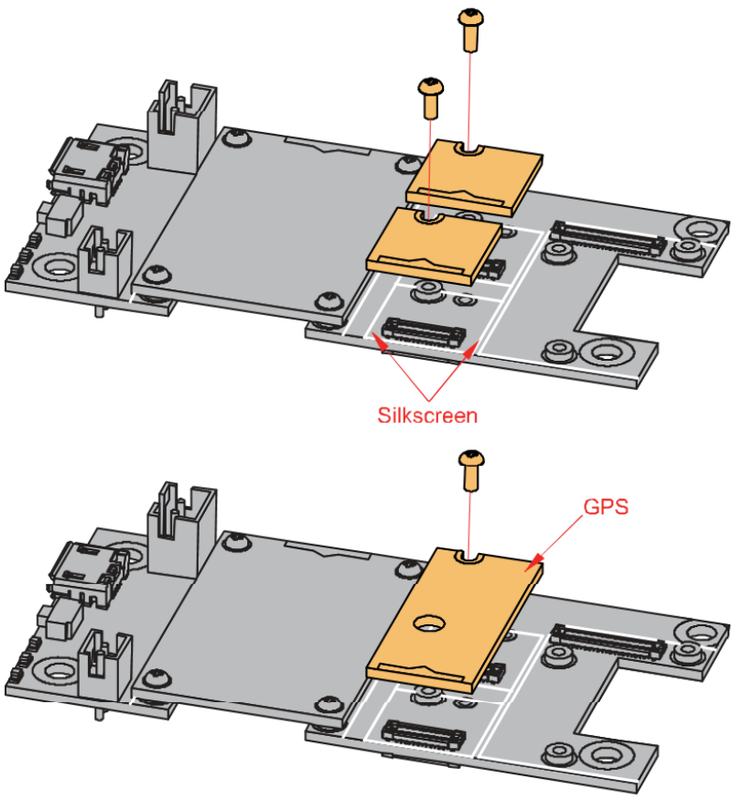


Figure 2: RAK1906 connection to WisBlock Base

Disassembling

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

1. First, remove the screws.

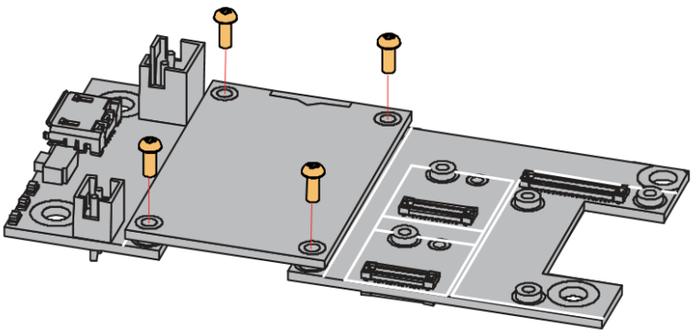


Figure 3: Removing screws from the WisBlock module

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

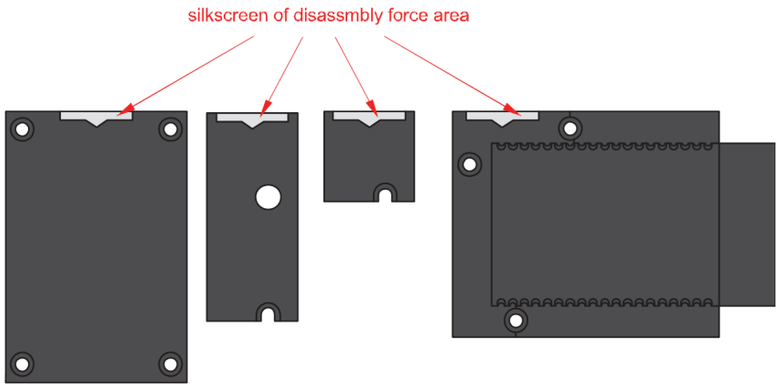


Figure 4: Detaching silkscreen on the WisBlock module

3. Apply force to the module at the position of the connector, as shown in **Figure 5**, to detach the module from the baseboard.

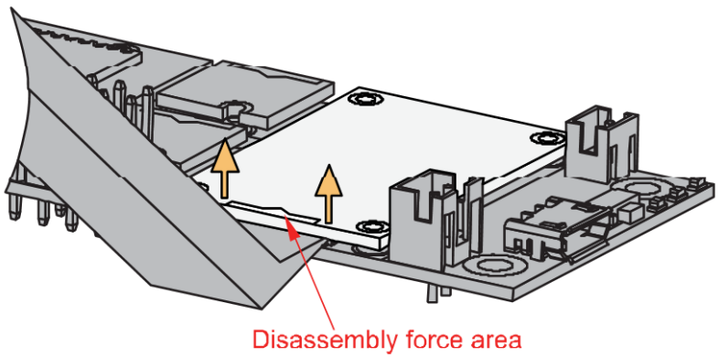


Figure 5: Applying even forces on the proper location of a WisBlock module

> 📝 **NOTE**
>
> If you will connect other modules to the remaining WisBlock Base slots, check on the WisBlock Pin Mapper tool for possible conflicts. RAK1906 uses I2C communication lines, and it can cause possible conflict especially on some IO modules.

After all this setup, you can now connect the battery (optional) and USB cable to start programming the RAK1906.

> ⚠️ **WARNING**
>
> - Battery 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 fire.
> - Make sure the battery wires match the polarity on the RAK WisBlock Base Board. Not all batteries have the same wiring.
> - 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.

Software Configuration and Example

The RAK1906 is a 4-in-1 Environmental Sensor board that contains the Bosch BME680 chip. It is specially designed for wearables where size and low power consumption are critical requirements. The BME680 contains a small Metal-Oxide (MOX) sensor and an I2C interface used to communicate with WisBlock Core.

> ⚠️ **WARNING**
>
> The BME680 requires a burn-in period to collect accurate readings. The burn-in procedure can be achieved by reading all sensors once every 5 seconds for at least 20 minutes the first time that you use it. For each subsequent use, the readings should stabilize 2-3 minutes after power-up.

Initial Test of the RAK1906 WisBlock Module

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

1. First, you need to select the WisBlock Core you have, as shown in **Figure 6** to **Figure 8**.

RAK4631 WisBlock Core

RAK4631 Wisblock Core

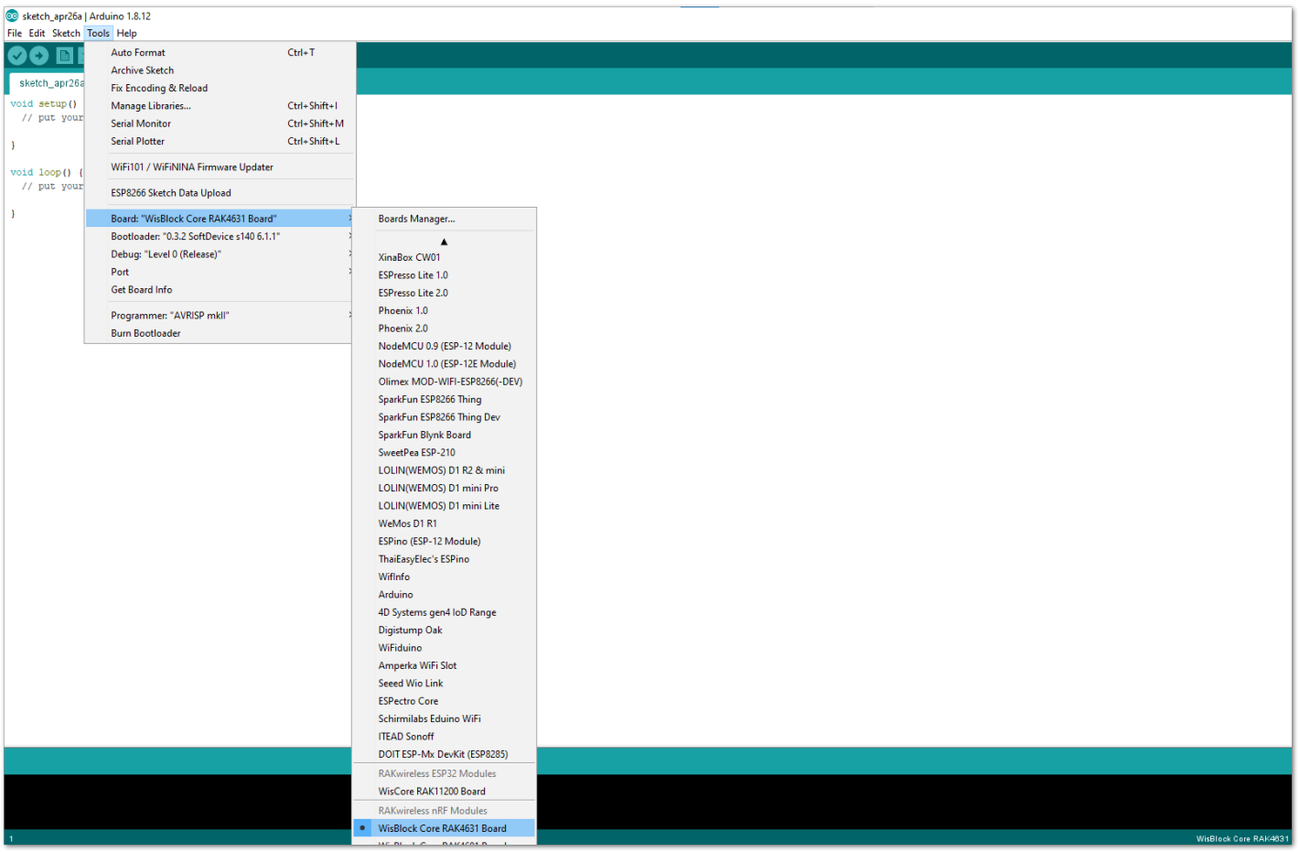


Figure 6: Selecting RAK4631 as WisBlock Core

RAK11200 WisBlock Core

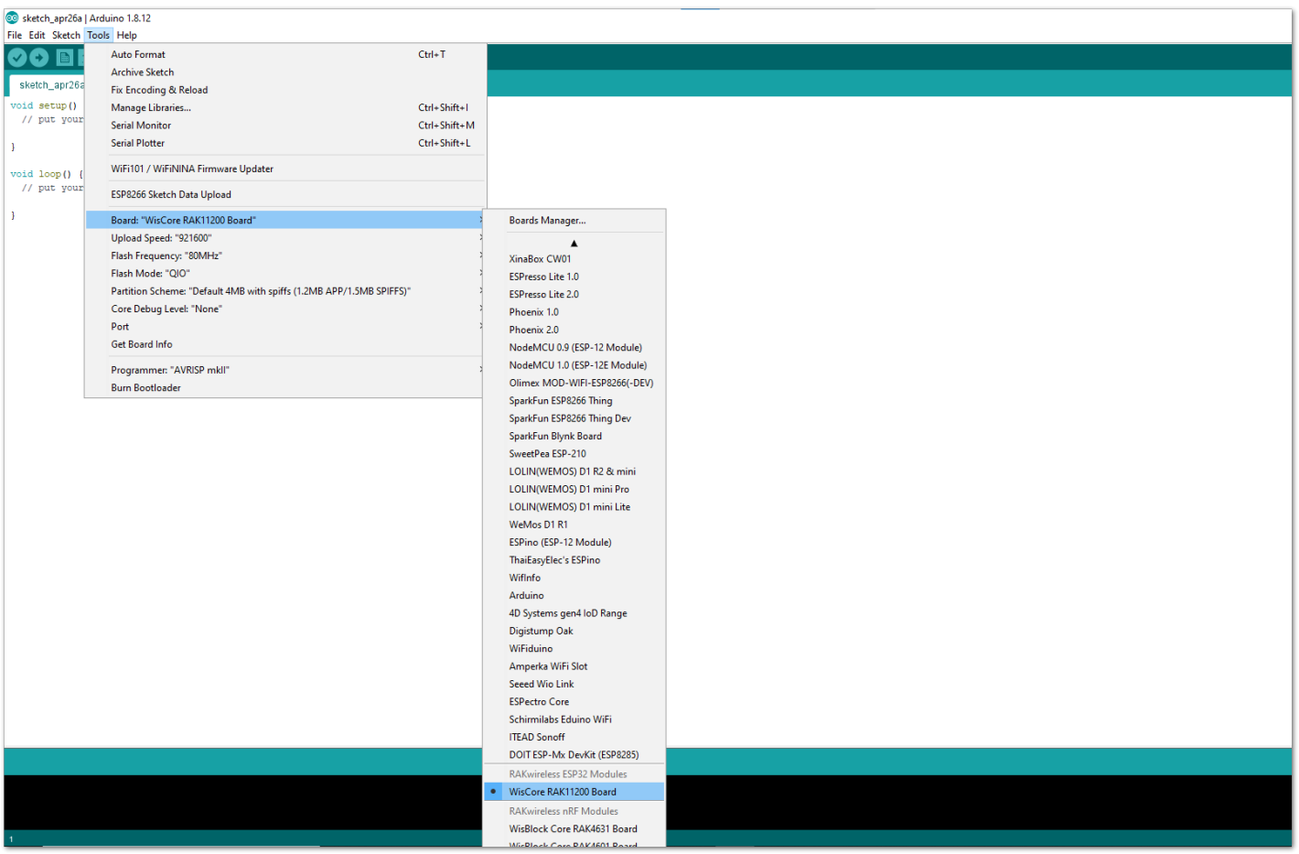


Figure 7: Selecting RAK11200 as WisBlock Core

RAK11310 WisBlock Core

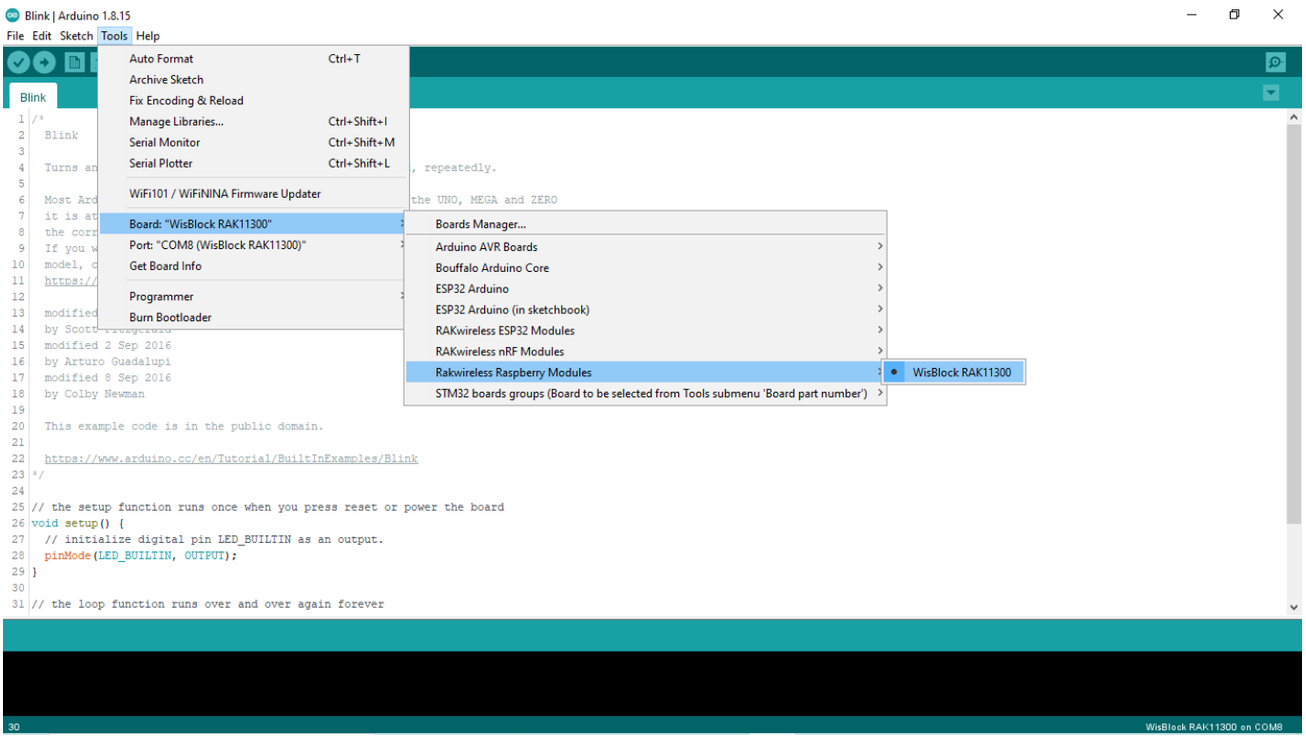


Figure 8: Selecting RAK11310 as WisBlock Core

2. The RAK1906_Environment_BME680 example in Github will work on all WisBlock Core. You can open the example codes depending on your WisBlock Core, as shown in **Figure 9** to **Figure 11**.

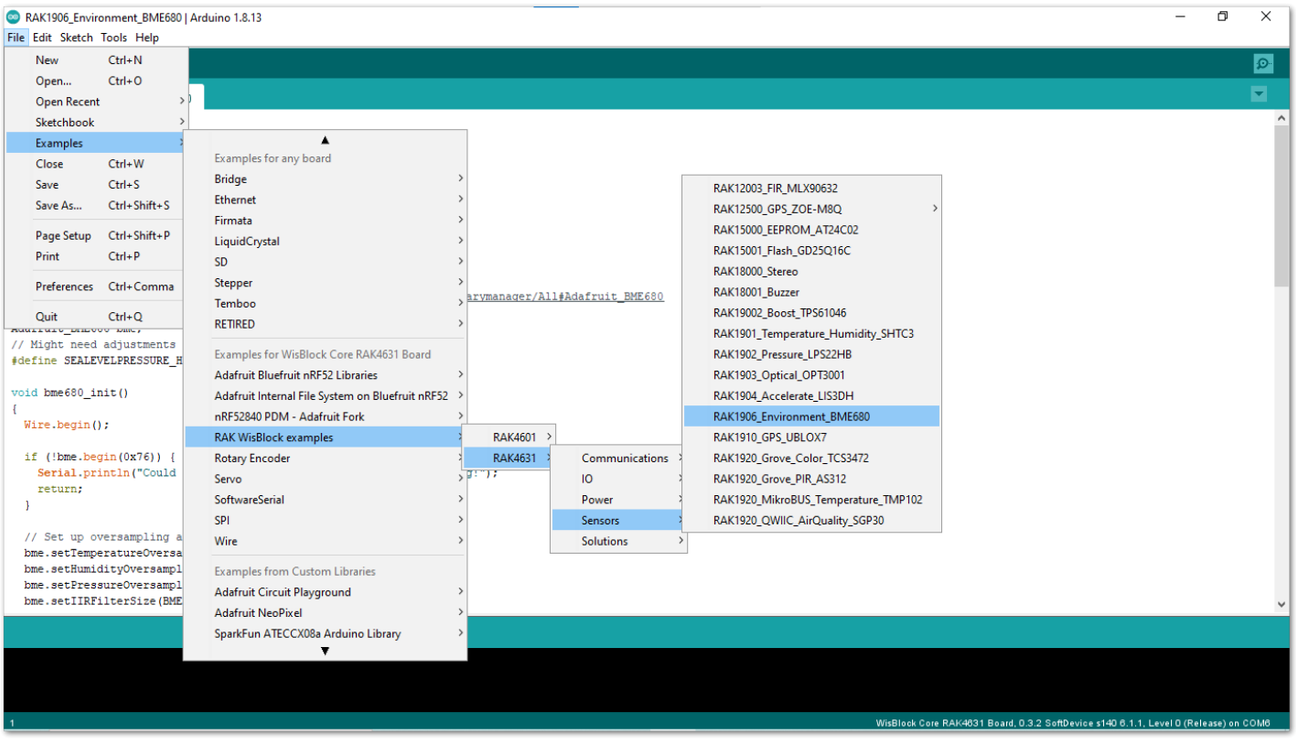


Figure 9: Opening RAK1906 example code for RAK4631 WisBlock Core

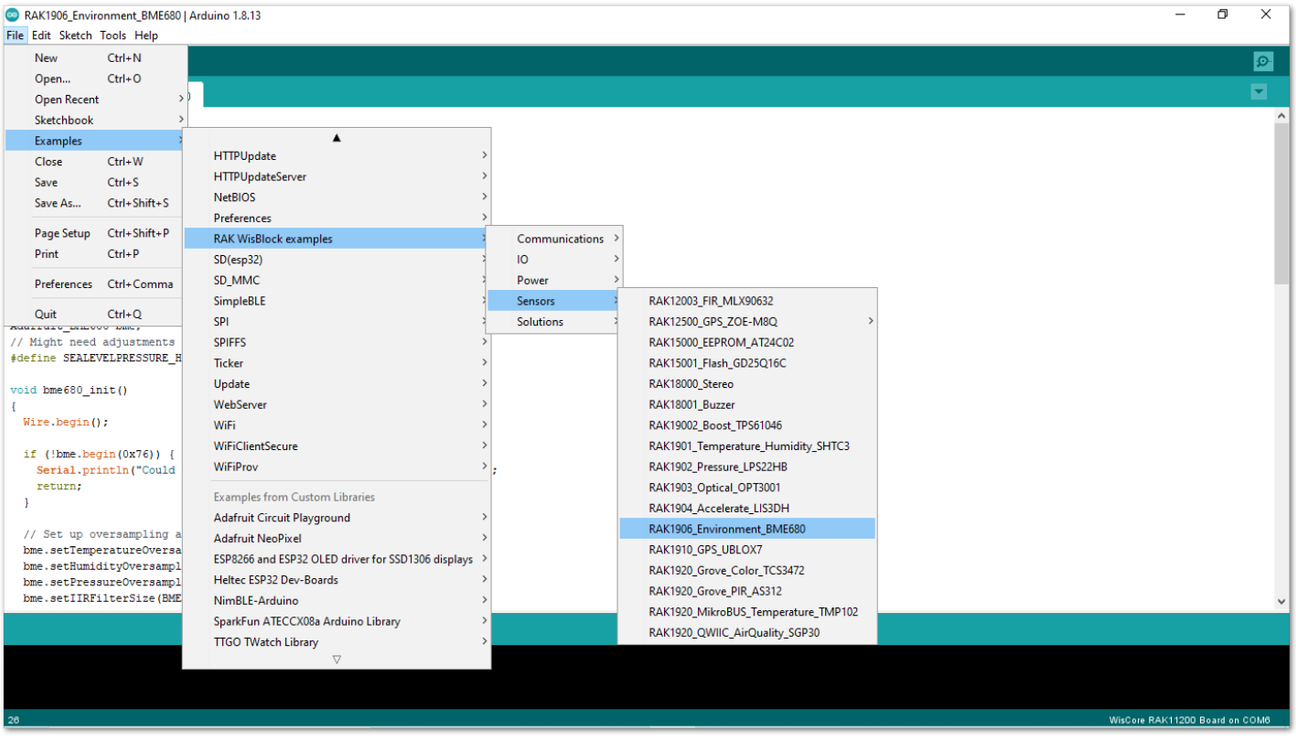


Figure 10: Opening RAK1906 example code for RAK11200 WisBlock Core

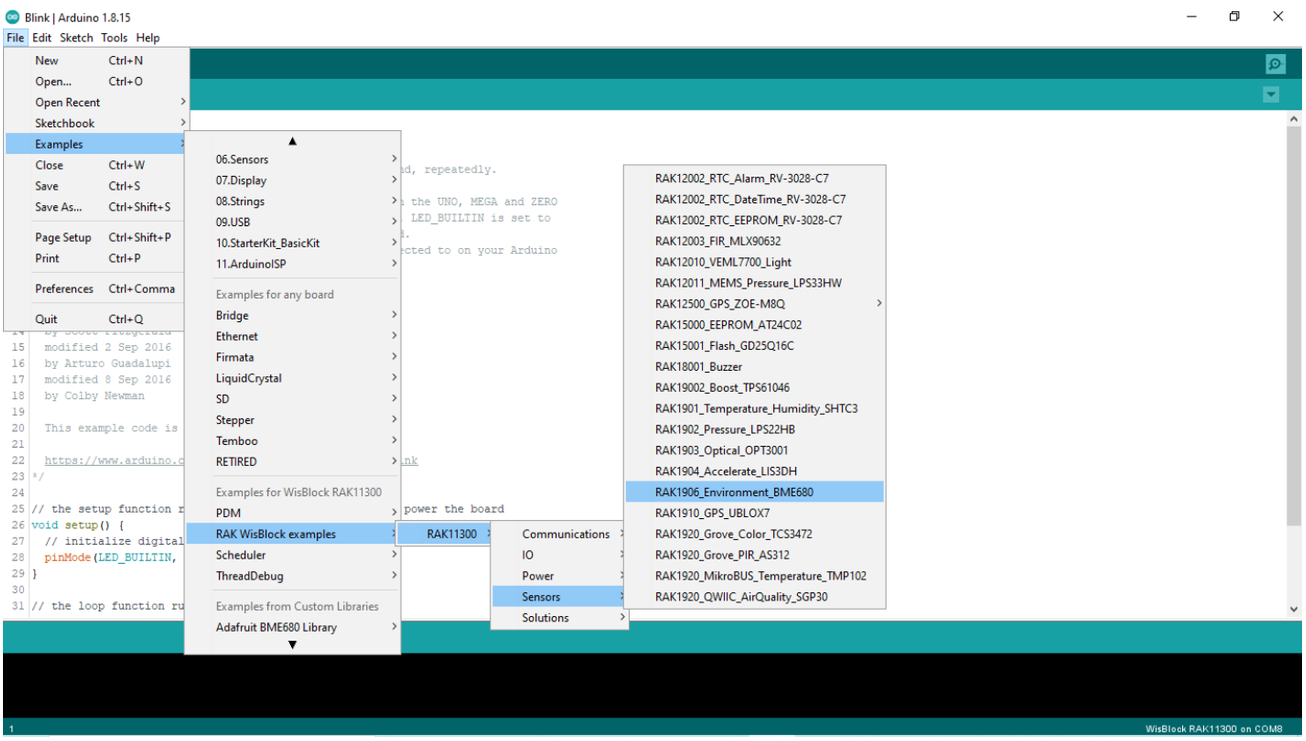


Figure 11: Opening RAK1906 example code for RAK11310 WisBlock Core

3. Once the example code is open, install the Adafruit BME680 library by clicking the yellow highlighted link, as shown in **Figure 12** and **Figure 13**.

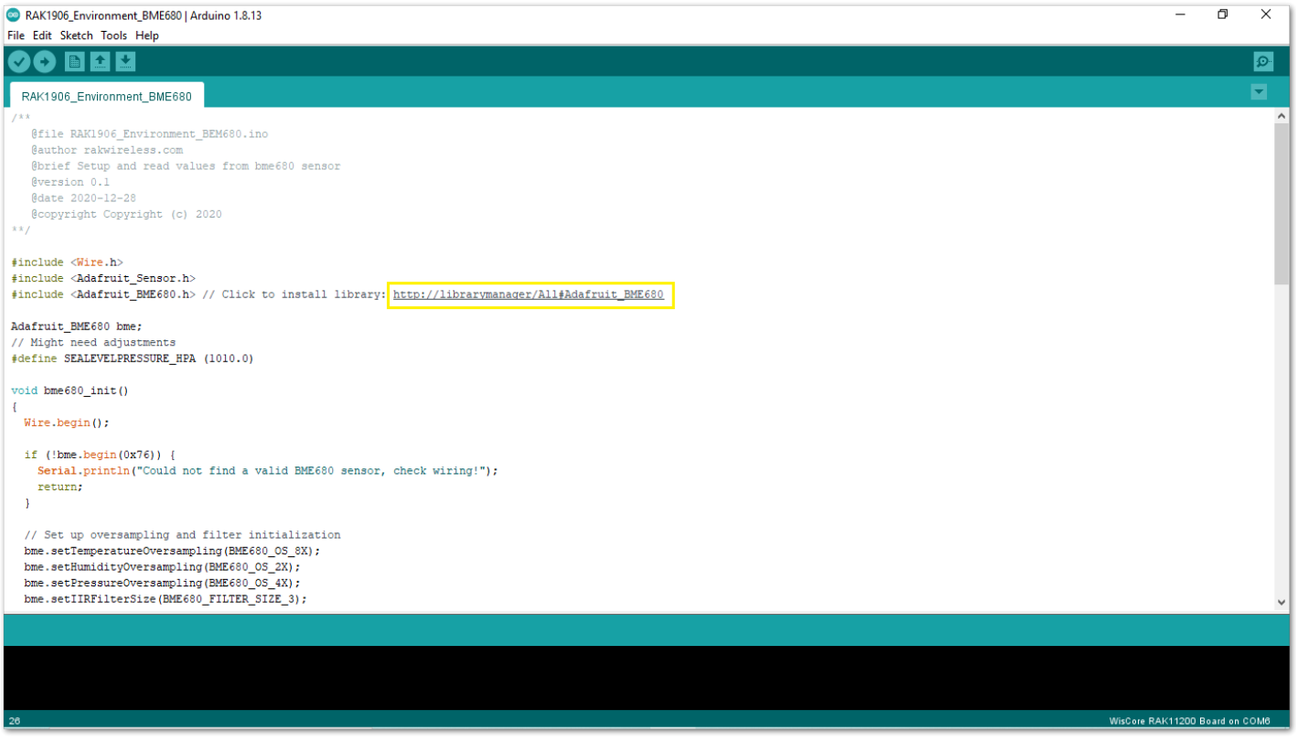


Figure 12: Select Adafruit BME680 library

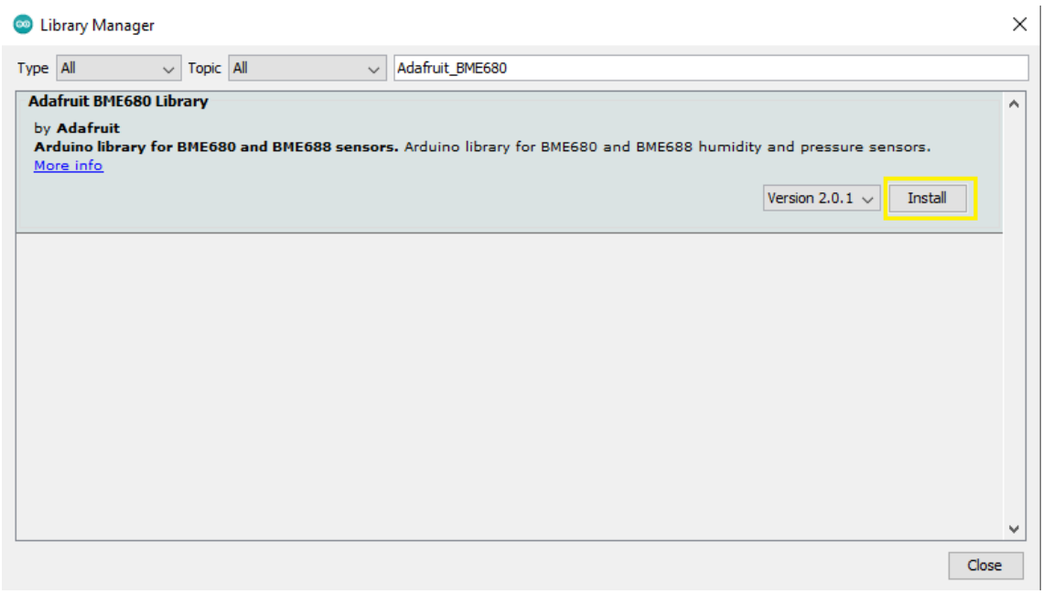


Figure 13: Installing Adafruit BME680 library

During the installation, you will be asked to install the **Adafruit Unified Sensor** library. Install it as well.

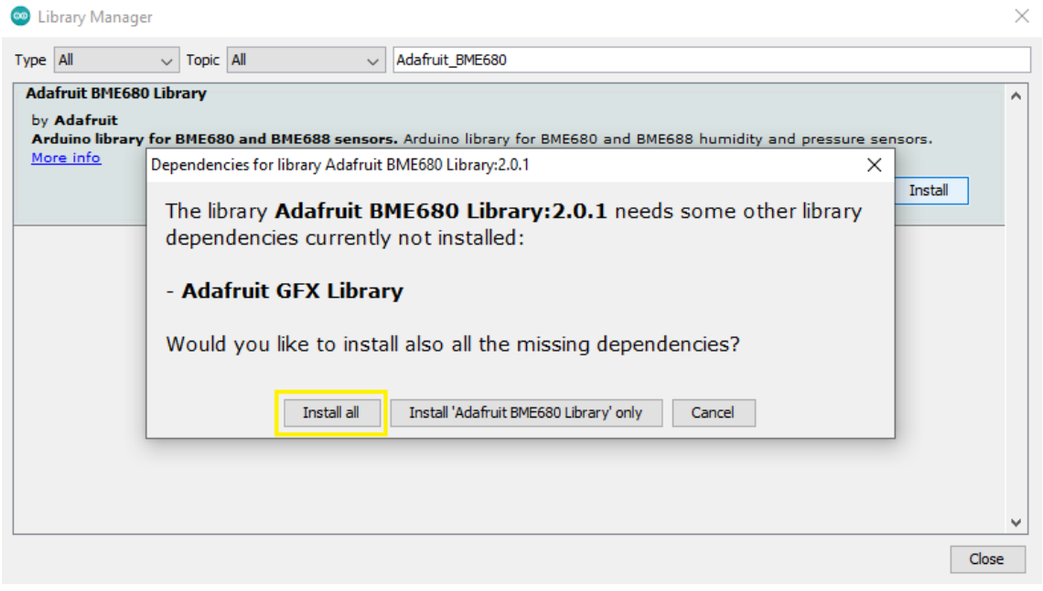


Figure 14: Installing Adafruit Unified Sensor Library

4. After successful installation of the library, you can now select the right serial port and upload the code, as shown in **Figure 15** and **Figure 16**.

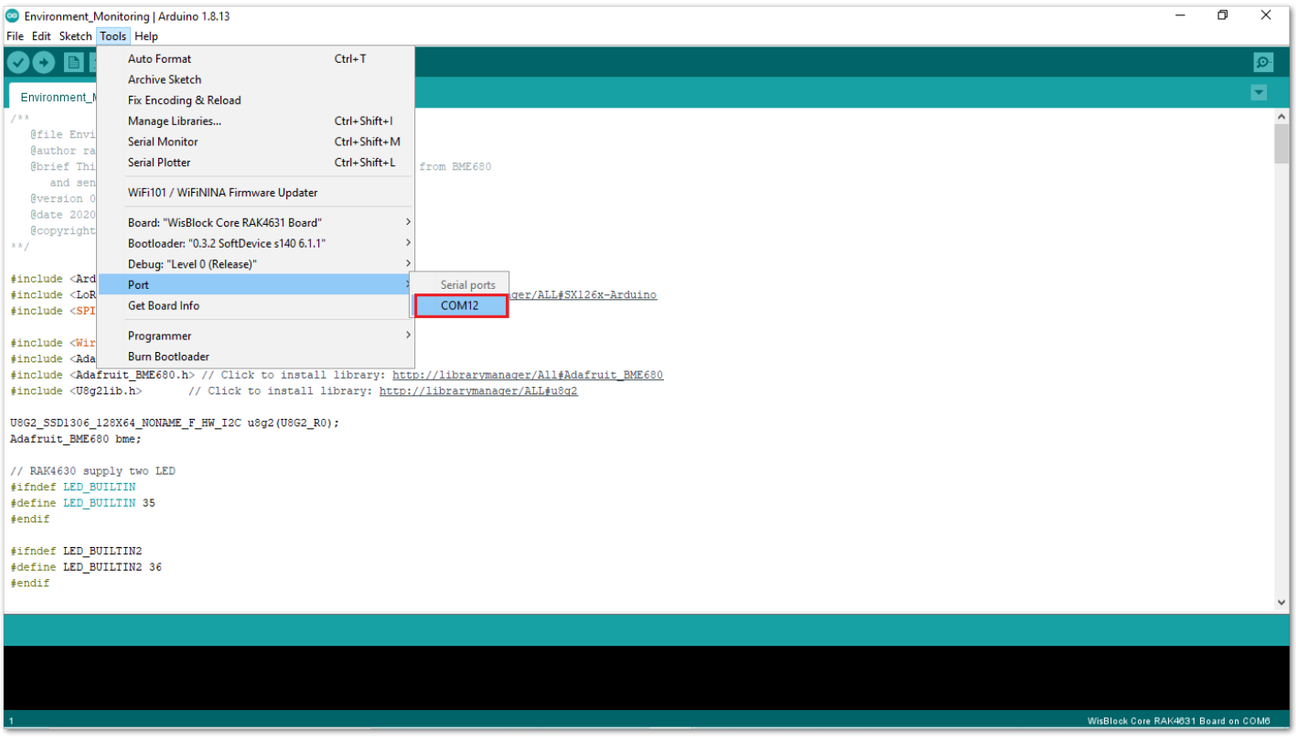


Figure 15: Selecting the correct Serial Port

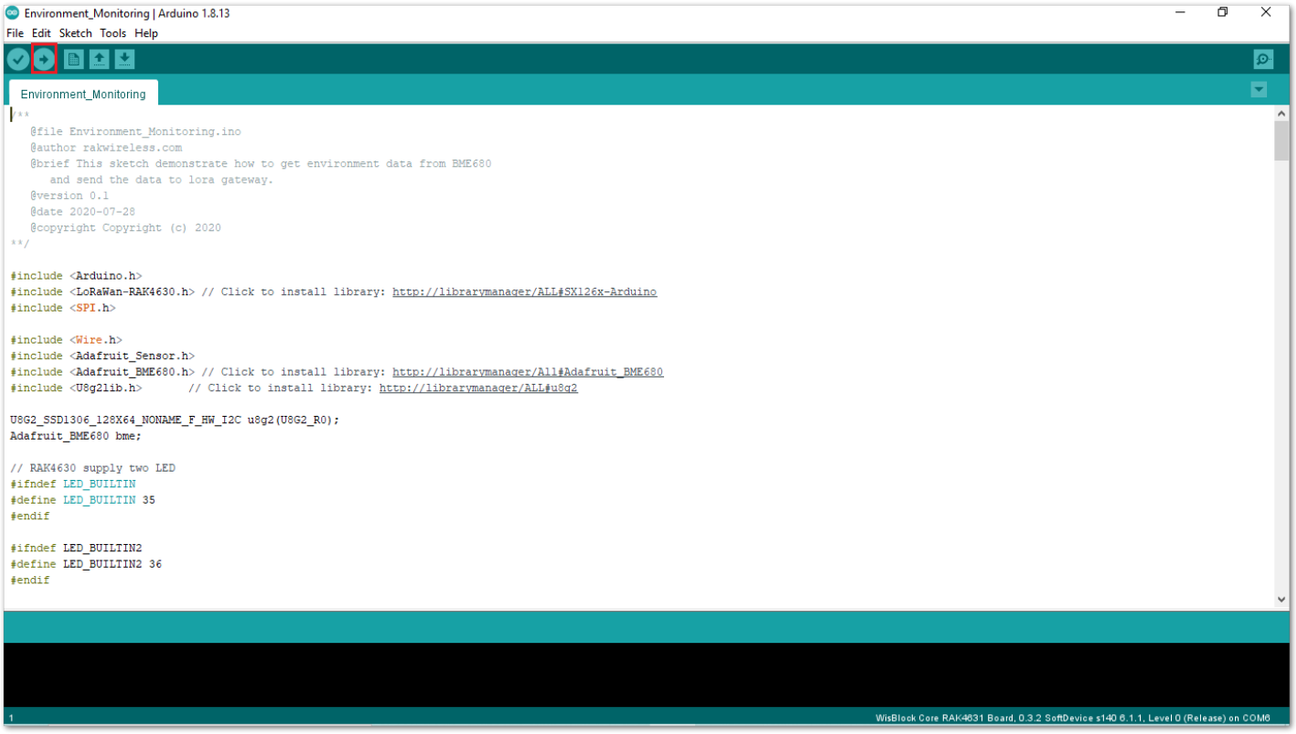


Figure 16: Uploading the RAK1906 example code

The RAK1906_Environment_BME680 application gets BME680 sensor data every 5 seconds and print the results on serial port.

```
Temperature = 30.66 *C
Pressure = 1013.80 hPa
Humidity = 44.38 %
Gas = 3.89 KOhms

Temperature = 30.69 *C
Pressure = 1013.78 hPa
Humidity = 44.59 %
Gas = 3.88 KOhms

Temperature = 30.72 *C
Pressure = 1013.82 hPa
Humidity = 44.37 %
Gas = 3.87 KOhms
```

LoRaWAN Environment Monitoring with RAK1906

This solution shows how to create an environmental data sensor node. The sensor node measures air temperature, air humidity, barometric pressure, and gas levels. After the node joins to server successfully, it then transmits this data periodically to a LoRaWan® server.

These are the quick links that go directly to the specific WisBlock Core module solution that you can use in your RAK1906 project.

- RAK1906 solution for RAK4631
- RAK1906 solution for RAK11310

RAK4631 Solution Requirements

- WisBlock Base RAK5005-O * 1pcs
- WisBlock Core RAK4631 * 1pcs
- WisBlock Sensor RAK1906 * 1pcs
- WisBlock Display RAK1921 * 1pcs
- LoRa Gateway * 1pcs
- Arduino IDE
- RAK4630 BSP
- Adafruit BME680 Library
- SX126x-Arduino Library
- U8g2 library

RAK11310 Solution Requirements

- WisBlock Base RAK5005-O * 1pcs
- WisBlock Core RAK11310 * 1pcs
- WisBlock Sensor RAK1906 * 1pcs
- WisBlock Display RAK1921 * 1pcs
- LoRa Gateway * 1pcs
- Arduino IDE
- Arduino BSP for RAK11300 module
- Adafruit BME680 Library
- SX126x-Arduino Library
- U8g2 library

Configuring LoRaWAN

There are configurations that you need to setup to ensure that the device can join a LoRaWAN Network server.

The steps below will explain the default settings and how to configure it.

1. Setup the LoRa Region.

Default region is EU868.

```
LoRaMacRegion_t g_CurrentRegion = LORAMAC_REGION_EU868;
```

You can change this to a region that is applicable to you like `LORAMAC_REGION_US915`, `LORAMAC_REGION_AU915`, etc.

2. Setup the LoRaWAN activation method.

Default is **OTAA**.

```
bool doOTAA = true;
```

To configure the device to ABP, you need to make this boolean variable `false`.

3. Setup the message type if confirmed or not.

Default is **confirmed message**.

```
lmh_confirm g_CurrentConfirm = LMH_CONFIRMED_MSG;
```

You can change to unconfirmed message by changing the value to `LMH_UNCONFIRMED_MSG`.

4. Setup device class.

Default is **Class A**.

```
DeviceClass_t g_CurrentClass = CLASS_A;
```

You can change this to **CLASS_B** (still under development) or **CLASS_C**.

5. Setup the keys.

- Configuration keys for OTAA Activation:

```
uint8_t nodeDeviceEUI[8] = {0x88, 0x88, 0x88, 0x88, 0x88, 0x88, 0x33, 0x33};
uint8_t nodeAppEUI[8] = {0xB8, 0x27, 0xEB, 0xFF, 0xFE, 0x39, 0x00, 0x00};
uint8_t nodeAppKey[16] = {0x88, 0x88, 0x88, 0x88, 0x88, 0x88, 0x88, 0x88, 0x88, 0x88, 0x88, 0x88,
```

- Configuration keys for ABP Activation:

```
uint32_t nodeDevAddr = 0x260116F8;
uint8_t nodeNwsKey[16] = {0x7E, 0xAC, 0xE2, 0x55, 0xB8, 0xA5, 0xE2, 0x69, 0x91, 0x51, 0x96, 0x06,
uint8_t nodeAppsKey[16] = {0xFB, 0xAC, 0xB6, 0x47, 0xF3, 0x58, 0x45, 0xC7, 0x50, 0x7D, 0xBF, 0x16
```

6. Setup uplink period.

The default is **20000 mS**.

```
#define LORAWAN_APP_INTERVAL 20000
```

Data Format

Byte1	Byte2	Byte3	Byte4	Byte5
1	Temperature1	Temperature2	Humidity1	Humidity2

Byte6	Byte7	Byte8	Byte9	Byte10	Byte11	Byte12	Byte13
Barometric1	Barometric2	Barometric3	Barometric4	Gas1	Gas2	Gas3	Gas4

- Byte1 is a marker for the data type, here always 1.
- Every byte value is MSB first.
- Temperature is sent as two bytes, enlarged 100 times for accuracy. For example, 2510 means 25.01 C, sent as 0x09 0xCE.
- Humidity is sent as two bytes, enlarged 100 times for accuracy. For example, 4173 means 41.73%, sent as 0x10 0x4D.
- Barometric pressure is sent as four bytes, enlarged 100 times for accuracy. For example, 100945 means 1009.45 hPa, sent as 0x00 0x01 0x8A 0x51.
- Gas resistance is sent as four bytes. For example, 63560 Ohm, sent as 0x00 0x00 0xF8 0x48.

Attention

If this example is implemented for the Region US915, DR0 cannot be used because the package size is too large to fit into the allowed payload.

When you successfully uploaded the example sketch, open the Serial Monitor of the Arduino IDE to see the sensor's reading logs. If you see the logs, as shown below, then your RAK1906 is properly communicating to the WisBlock core.

```
10:30:07.777 -> ====================================
10:30:07.777 -> Welcome to RAK4630 LoRaWan!!!
10:30:07.777 -> Type: OTAA
10:30:07.777 -> Region: EU868
10:30:07.777 -> ====================================
10:30:07.777 -> Chip ID=0x61
10:30:12.974 -> OTAA Mode, Network Joined!
10:30:33.999 -> Sending frame now...
10:30:33.999 -> result: Tem:28.70C Hum:67.00% Pres:948.50KPa Gas:27646Ohms
10:30:33.999 -> lmh_send ok count 1
10:30:53.979 -> Sending frame now...
10:30:53.979 -> result: Tem:28.70C Hum:67.00% Pres:948.50KPa Gas:242488Ohms
10:30:54.012 -> lmh_send ok count 2
10:31:14.006 -> Sending frame now...
10:31:14.006 -> result: Tem:28.70C Hum:67.00% Pres:948.50KPa Gas:457174Ohms
10:31:14.006 -> lmh_send ok count 3
10:31:33.997 -> Sending frame now...
10:31:33.997 -> result: Tem:28.69C Hum:67.00% Pres:948.50KPa Gas:74102Ohms
10:31:34.032 -> lmh_send ok count 4
```

Data arrives at LoRaWAN server.

Figure 17: LoRaWAN Environment Monitoring example

Last Updated: 1/10/2022, 2:07:11 AM