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](image)

Hardware

Chipset

<table>
<thead>
<tr>
<th>Vendor</th>
<th>Part number</th>
</tr>
</thead>
<tbody>
<tr>
<td>BOSCH</td>
<td>BME680</td>
</tr>
</tbody>
</table>

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.

![RAK1906 WisBlock Environmental Sensor Pinout](image)

**Sensors**

**Temperature Sensor**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Symbol</th>
<th>Condition</th>
<th>Min</th>
<th>Typ</th>
<th>Max</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating Temperature Range</td>
<td>$T_A$</td>
<td>Operational</td>
<td>-40</td>
<td>25</td>
<td>85</td>
<td>°C</td>
</tr>
<tr>
<td>Supply Current</td>
<td>$I_{DD,T}$</td>
<td>1 Hz Forced Mode, Temperature Measurement Only</td>
<td>1.0</td>
<td>µA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Absolute Accuracy Temperature</td>
<td>$A_{T,25}$</td>
<td>25 °C</td>
<td>±0.5</td>
<td>°C</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$A_{T,full}$</td>
<td>0-65 °C</td>
<td>±1.0</td>
<td>°C</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Humidity Sensor**
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Symbol</th>
<th>Condition</th>
<th>Min</th>
<th>Typ</th>
<th>Max</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating Range</td>
<td></td>
<td></td>
<td>-40</td>
<td>25</td>
<td>85</td>
<td>°C</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td>100</td>
<td></td>
<td>% r.H.</td>
</tr>
<tr>
<td>Full Accuracy</td>
<td></td>
<td></td>
<td>0</td>
<td>65</td>
<td></td>
<td>°C</td>
</tr>
<tr>
<td>Range</td>
<td></td>
<td></td>
<td>0</td>
<td>65</td>
<td></td>
<td>% r.H.</td>
</tr>
<tr>
<td>Supply Current</td>
<td>I_DD,H</td>
<td>1 Hz Forced Mode,</td>
<td>2.1</td>
<td>2.8</td>
<td></td>
<td>µA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Temperature and Humidity</td>
<td>Measurement</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Absolute Accuracy</td>
<td>A_h</td>
<td>20-80% r.H., 25 °C,</td>
<td>±3</td>
<td></td>
<td></td>
<td>% r.H.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>including hysteresis</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Pressure Sensor**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Symbol</th>
<th>Condition</th>
<th>Min</th>
<th>Typ</th>
<th>Max</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating Temperature Range</td>
<td>T_A</td>
<td>Operational</td>
<td>-40</td>
<td>25</td>
<td>85</td>
<td>°C</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Full Accuracy</td>
<td>0</td>
<td>65</td>
<td></td>
<td>°C</td>
</tr>
<tr>
<td>Operating Pressure range</td>
<td>P</td>
<td>Full Accuracy</td>
<td>300</td>
<td></td>
<td>1100</td>
<td>hPa</td>
</tr>
<tr>
<td>Supply Current</td>
<td>I_DD,LP</td>
<td>1 Hz Forced Mode,</td>
<td>3.1</td>
<td>4.2</td>
<td></td>
<td>µA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Pressure and Temperature,</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Lowest Power</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Temperature Coefficient Of Offset</td>
<td>T_CO_P</td>
<td>25-40 °C, 900 hPa</td>
<td>±1.3</td>
<td></td>
<td></td>
<td>Pa,K</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>±10.9</td>
<td></td>
<td>cm/K</td>
</tr>
<tr>
<td>Absolute Accuracy Pressure</td>
<td>A_p,full</td>
<td>300-1100 hPa</td>
<td>±0.6</td>
<td></td>
<td></td>
<td>hPa</td>
</tr>
</tbody>
</table>

**AQI Sensor**
Electrical Characteristics

Recommended Operating Conditions

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Min</th>
<th>Nom.</th>
<th>Max</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>V_{DD}</td>
<td>Power supply for the module</td>
<td>1.71</td>
<td>1.8</td>
<td>3.6</td>
<td>V</td>
</tr>
<tr>
<td>I_{sleep}</td>
<td>Sleep current</td>
<td>-</td>
<td>0.15</td>
<td>-</td>
<td>uA</td>
</tr>
<tr>
<td>I_{DD}</td>
<td>Humidity Measure current</td>
<td>-</td>
<td>340</td>
<td>-</td>
<td>uA</td>
</tr>
<tr>
<td>I_{DD}</td>
<td>Pressure Measure current</td>
<td>-</td>
<td>714</td>
<td>-</td>
<td>uA</td>
</tr>
<tr>
<td>I_{DD}</td>
<td>Temperature Measure current</td>
<td>-</td>
<td>350</td>
<td>-</td>
<td>uA</td>
</tr>
</tbody>
</table>

Mechanical Characteristics

Board Dimensions

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

Figure 5 shows the schematic of the RAK1906 module.

Figure 5: RAK1906 WisBlock Environmental Sensor schematics
**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.
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.
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](image)

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](image)

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](image)
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

RAK11310 WisBlock Core

Figure 7: Selecting RAK11200 as 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
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.
During the installation, you will be asked to install the **Adafruit Unified Sensor** library. Install it as well.
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.

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

```c
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**.

```c
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**.

```c
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**.

```c
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:

```c
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, 0x88, 0x88, 0x88, 0x88};
```

- Configuration keys for ABP Activation:

The default is 20000 mS.

---

```c
#define LORAWAN_APP_INTERVAL 20000
```

---

### Data Format

<table>
<thead>
<tr>
<th>Byte1</th>
<th>Byte2</th>
<th>Byte3</th>
<th>Byte4</th>
<th>Byte5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Temperature1</td>
<td>Temperature2</td>
<td>Humidity1</td>
<td>Humidity2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Byte6</th>
<th>Byte7</th>
<th>Byte8</th>
<th>Byte9</th>
<th>Byte10</th>
<th>Byte11</th>
<th>Byte12</th>
<th>Byte13</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Barometric1</td>
<td>Barometric2</td>
<td>Barometric3</td>
<td>Barometric4</td>
<td>Gas1</td>
<td>Gas2</td>
<td>Gas3</td>
</tr>
</tbody>
</table>

- 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.
Data arrives at LoRaWAN server.

Figure 17: LoRaWAN Environment Monitoring example