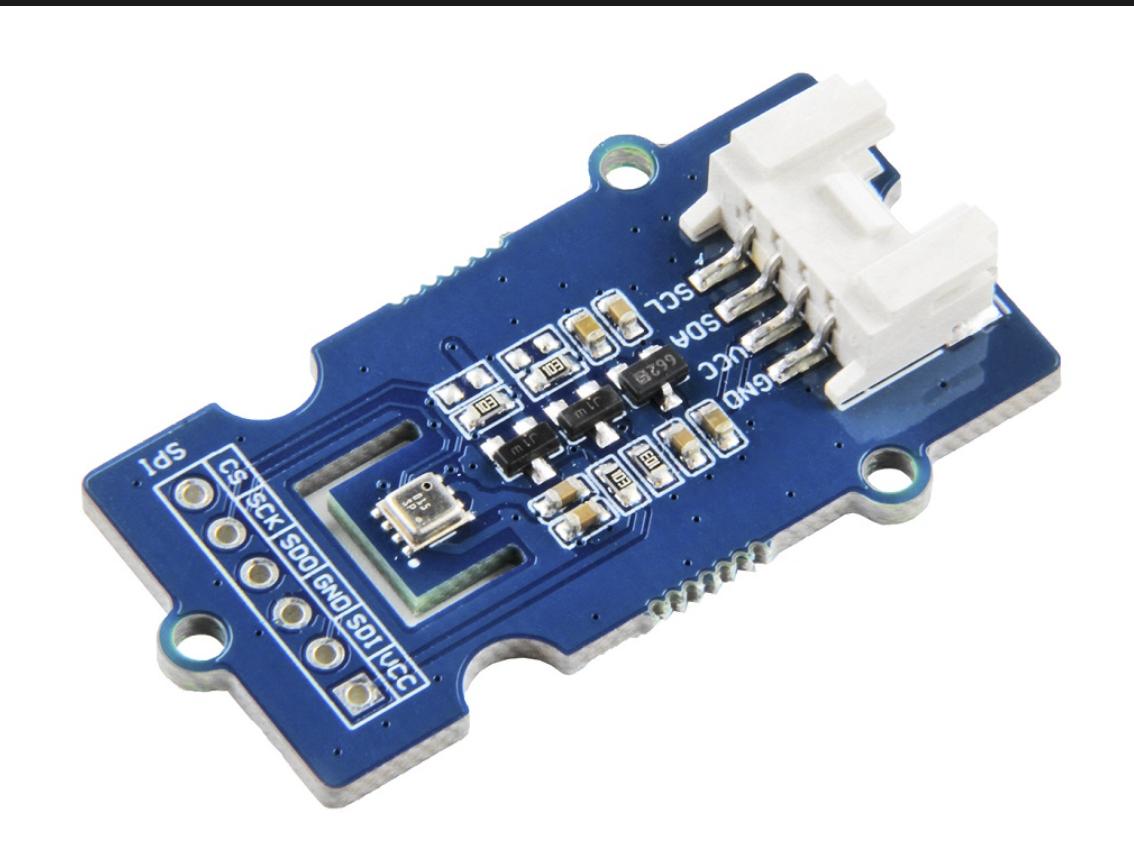


# Sensor680 DA Install

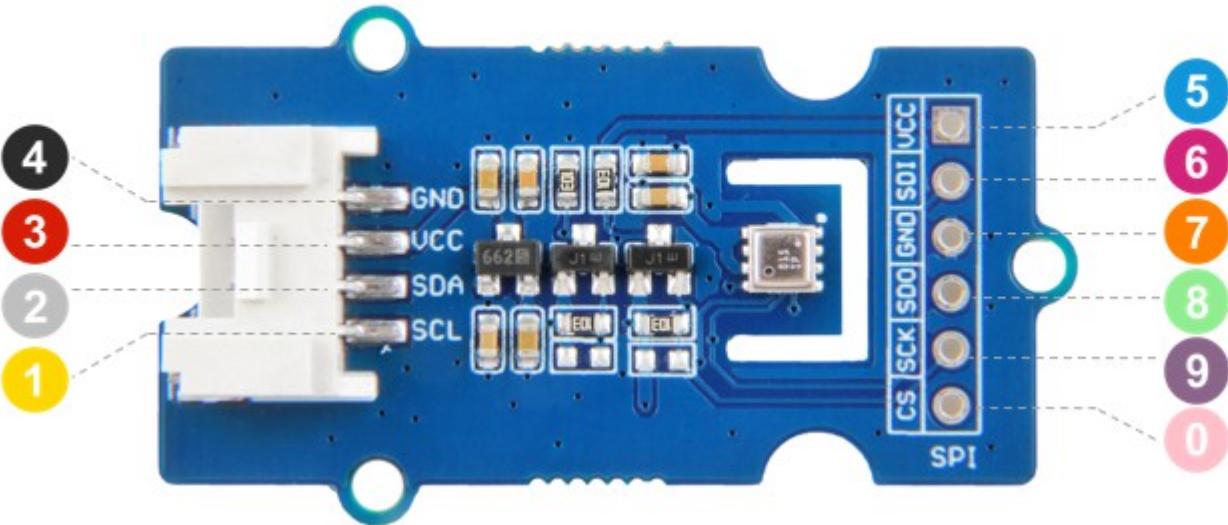
All planned updates to the wiki platform has been publicly available. Your contributions will be essential to us!

[Home](#) > [Grove](#) > [Grove Sensor](#) > [Multiple in one](#) > [Grove - Temperature Humidity Pressure Gas Sensor\(BME680\)](#)

# Grove - Temperature Humidity Pressure Gas Sensor(BME680)



The Grove-Temperature&Humidity&Pressure&Gas Sensor(BME680) is a multiple function sensor which can measure

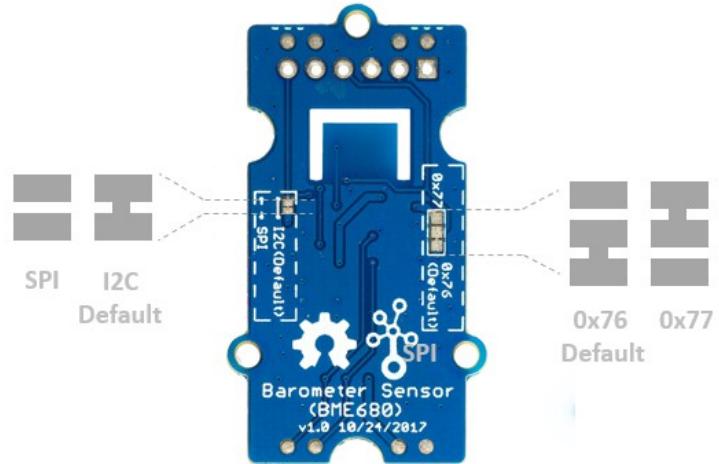


## I2C

- ④ GND: connect this module to the system GND
- ③ VCC: you can use 5V or 3.3V for this module
- ② SDA: serial data
- ① SCL: serial clock

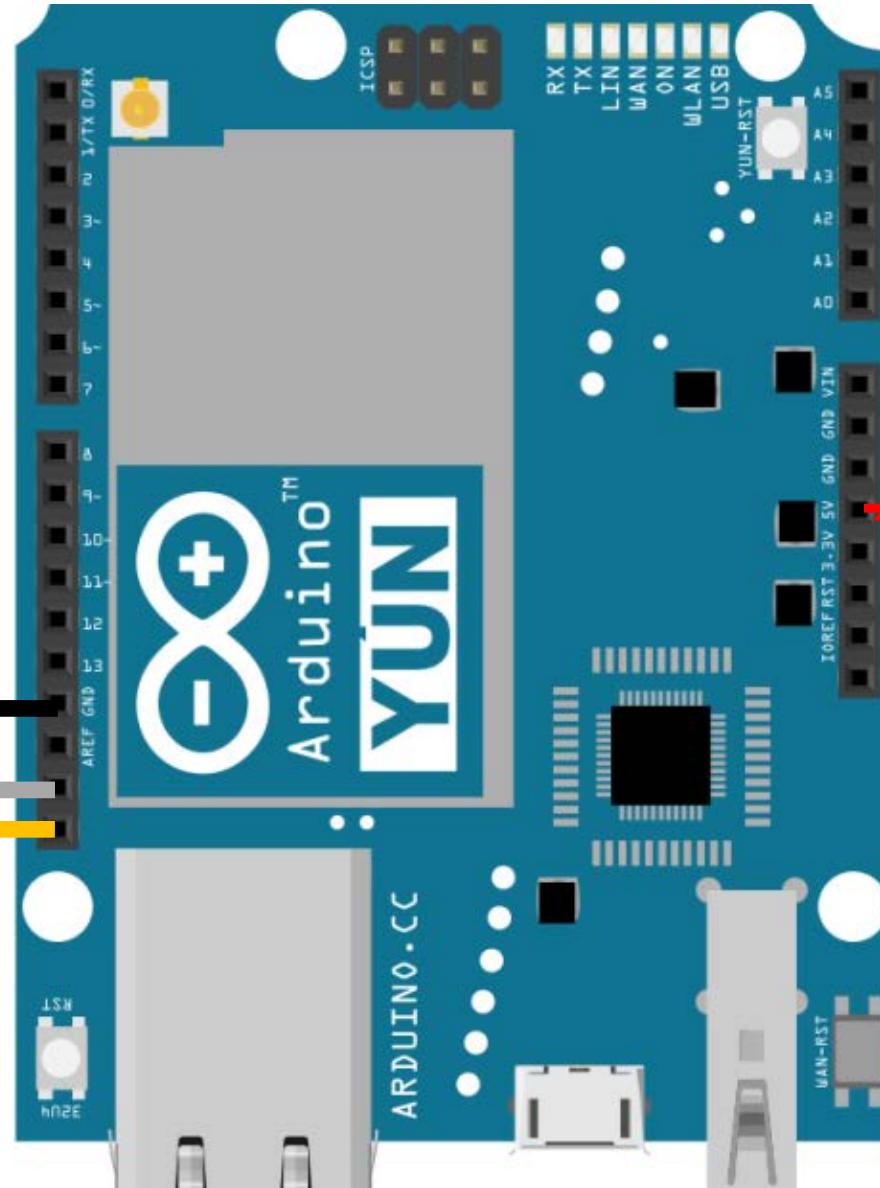
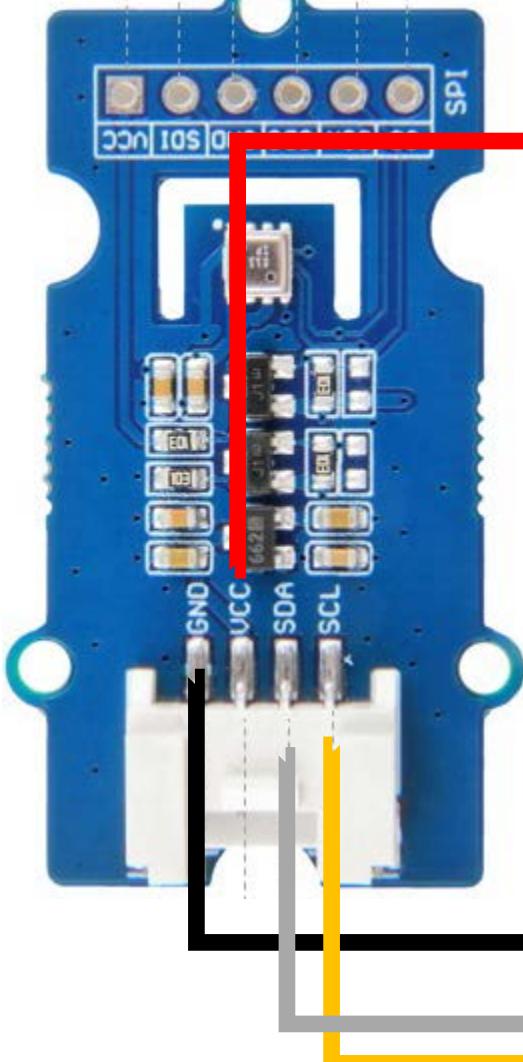
## SPI

- ⑤ VCC: you can use 5V or 3.3V for this module
- ⑥ SDI: serial data input; data input/output in 3-wire mode
- ⑦ GND: connect the system GND with this module
- ⑧ SDO: serial data output; hi-Z in 3-wire mode
- ⑨ SCK: serial clock
- ⑩ CS: chip select, active low



SPI    I2C  
Default

0x76    0x77  
Default



<https://github.com/IoTtalk/Sensor680>

The image shows two open GitHub browser tabs side-by-side.

**Left Tab:** IoTalk/Sensor680: SensorBox

- Code: master
- Issues: 1
- Pull requests: 0
- File list:
  - Jyneda Update README.md
  - IDA\_BME680
  - DAI.py
  - DAN.py
  - README.md
  - autoStart.py
  - config.py

**Right Tab:** Sensor680/IDA\_BME680 at master

- Code: master
- Issues: 0
- Pull requests: 0
- Actions: 0
- Projects: 0
- Wiki: 0
- Security: 0
- Insights: 0
- Settings: 0

**Content Area:**

**Upload Status:** Jyneda Add files via upload ✓

Name	Last commit message
..	
IDA_BME680.ino	Add files via upload

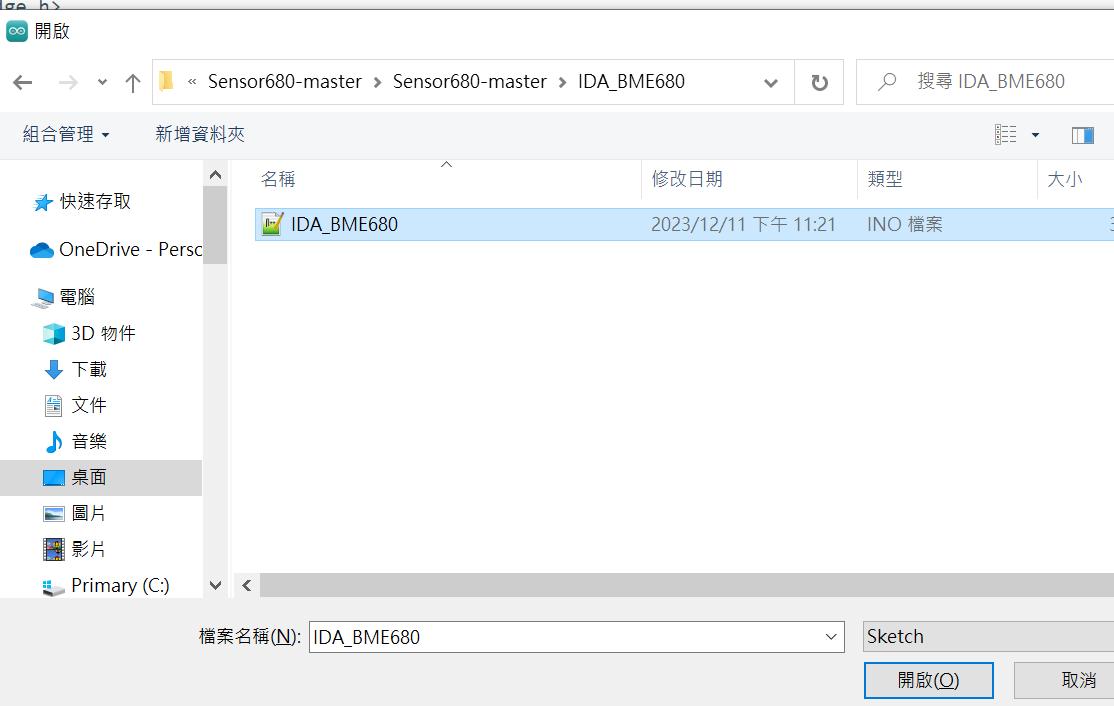


IDA\_BME680.ino

```

1  #include "seeded_bme680.h"
2  #include <SoftwareSerial.h>
3  #include <Bridge.h>
4
5  #define BME_S
6  #define BME_M
7  #define BME_M
8  #define BME_C
9  #define IIC_A
10 Seeed_BME680
11
12 void setup()
13 {
14     Serial.begin(115200);
15     while (!Serial)
16     {
17         delay(1000);
18     }
19     pinMode(17, INPUT);
20     Bridge.begin();
21 }
22
23 void loop() {
24     char D13[10];
25     char value[10];
26     unsigned long reg_done;
27
28     Bridge.getReg(D13, reg_done, D13, 2);

```

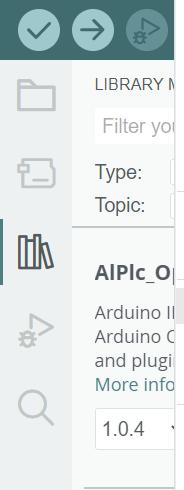


Output Serial Monitor ×

Message (Enter to send message to 'Arduino Yún' on 'COM4')

Temp: 28.08 C  
 Humidity: 71.21 %  
 Pressure: 100940 Pa  
 Gas: 64986 ohms

=====  
 Temp: 28.08 C  
 Humidity: 71.22 %  
 Pressure: 100942 Pa  
 Gas: 64738 ohms



1.0.4

Programmer  
Burn Bootloader**INSTALL****Arduino Cloud Provider****Examples** by ArduinoExamples of how to connect various Arduino boards to cloud providers  
More info

1.2.1

**INSTALL****Arduino Low Power** by ArduinoPower save primitives features for SAMD and nRF52 32bit boards With this library you can manage the low power states of...  
More info

1.2.2

**INSTALL**

Auto Format Ctrl + T

Archive Sketch

Ctrl + Shift + I

Manage Libraries...

Ctrl + Shift + M

Serial Monitor

Serial Plotter

Firmware Updater

Upload SSL Root Certificates

Board: "Arduino Yún"

Port

Get Board Info

Programmer

Burn Bootloader

Boards Manager... Ctrl + Shift + B

● Arduino AVR Boards

Arduino Yún

Arduino Uno

Arduino Uno Mini

Arduino Duemilanove or Diecimila

Arduino Nano

Arduino Mega or Mega 2560

Arduino Mega ADK

Arduino Leonardo

Arduino Leonardo ETH

Arduino Micro

Arduino Esplora

Arduino Mini

Arduino Ethernet

Arduino Fio

Arduino BT

LilyPad Arduino USB

LilyPad Arduino

Arduino Pro or Pro Mini

Arduino NG or older

Arduino Robot Control

Arduino Robot Motor

Arduino Gemma

Adafruit Circuit Playground

Arduino Yún Mini

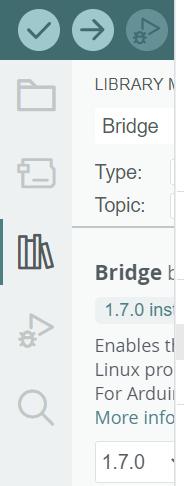
Arduino Industrial 101

Linino One

Arduino Uno WiFi

Output

已安装的库



Auto Format Ctrl + T

Archive Sketch

Manage Libraries... Ctrl + Shift + I

Serial Monitor Ctrl + Shift + M

Serial Plotter

Firmware Updater

Upload SSL Root Certificates

**Bridge**

1.7.0 installed

Enables the Bridge library

Linux provides a bridge driver for Arduino

For Arduino Uno, Duemilanove, and Mega

More info

1.7.0

Board: "Arduino Yún"

Port

Get Board Info

Programmer

Burn Bootloader

**AD7193** by Anne Mahaffey  
<anne@engineerbynights.com>  
Library for AD7193 Sigma-Delta ADC with  
PGA Library for the AD7193 ADC - useful  
for interfacing with various sensors, suc...  
[More info](#)

1.0.0 [▼](#) [INSTALL](#)

**ADS1232** by Max Sanchez.  
<msanchez@hardmax.com.pe>  
Simple Library to interface the Texas  
Instruments ADS1232 ADC. A Simple Non-  
blocking Library for reading from Texas...  
[More info](#)

1.0.2 [▼](#) [INSTALL](#)

**BM12O2321-A** by  
BESTMODULES  
Arduino library for 9-bit UART access to  
the BM12O2321-A/BMD12K232 that H-  
bridge Drive Module we can use this...  
[More info](#)

```
#include "seeed_bme680.h"
#include <SoftwareSerial.h>
#include <Bridge.h>

#define BME_SCK 13
#define BME_MISO 12
#define BME_MOSI 11

Board: "Arduino Yún"
Port
  Serial ports
    COM4 (Arduino Yún) ; /* IIC PROTOCOL D2(SDA), D3(SCL)*/

Programmer
Burn Bootloader

14
15 while (!bme680.init()) {
16   |   Serial.println("bme680 init failed ! can't find device!");
17   |   delay(10000);
18 }
19 pinMode(13,OUTPUT); //IoTtalk successful registration notification
20 Bridge.begin(); // D0 and D1 are used by the Bridge library.
21 }

22
23 void loop() {
24   |   char D13[2];
25   |   char valueStr[23];
```

## Output

项目使用 20930 字节 (72%) 的程序存储空间。最大值为 28672 字节。  
全局变量使用 1161 个字节 (45%) 的动态内存，剩下 1399 个字节用于局部变量。最大值为 2560 字节。

[https://github.com/Seeed-Studio/Seeed\\_Arduino\\_BME68x](https://github.com/Seeed-Studio/Seeed_Arduino_BME68x)

S Grove - Temperature Humidit x S Seeed-Studio/Seeed\_Arduino x S How to install an Arduino libr x +

https://github.com/Seeed-Studio/Seeed\_Arduino\_BME68x

Seeed-Studio / Seeed\_Arduino\_BME68x

Type  to search

Code Issues 1 Pull requests Actions Projects Security Insights

**⚠ GitHub users are now required to enable two-factor authentication as an additional security measure. Your activity on GitHub includes you in this requirement. You will need to enable two-factor authentication on your account before January 19, 2024, or be restricted from account actions.** [Enable 2FA](#)

 **Seeed\_Arduino\_BME68x** Public

Watch 11 Fork 2 Star 10

master 1 branch 0 tags

Go to file Add file Code

Local Codespaces

Clone

HTTPS SSH GitHub CLI

[https://github.com/Seeed-Studio/Seeed\\_Arduino](https://github.com/Seeed-Studio/Seeed_Arduino)

Use Git or checkout with SVN using the web URL.

Open with GitHub Desktop

Download ZIP

**About**

Grove - Temperature, Humidity, Pressure and Gas Sensor (BME68x)

Readme

MIT license

Activity

10 stars

11 watching

2 forks

Report repository

**Releases**

No releases published

**Packages**

No packages published

File	Description	Last Commit
examples/seeed_bme680_test	Pretty printed the Arduino code	4 years ago
.gitlab-ci.yml	Seeed:Arduino: fix travis.yml warning	5 years ago
.travis.yml	Seeed:Arduino: fix travis.yml warning	5 years ago
License.txt	update	5 years ago
README.md	Seeed:Arduino: Add travis build configuration	5 years ago
bme680.cpp	Pretty printed the Arduino code	5 years ago
bme680.h	Pretty printed the Arduino code	5 years ago
bme680_defs.h	Pretty printed the Arduino code with astyle	4 years ago
keywords.txt	update	5 years ago
library.properties	update	5 years ago
seeed_bme680.cpp	Fix Compiler Warnings	2 years ago



- Verify/Compile Ctrl + R
- Upload Ctrl + U
- Configure and Upload
- Upload Using Programmer Ctrl + Shift + U
- Export Compiled Binary Alt + Ctrl + S
- Optimize for Debugging
- Show Sketch Folder Alt + Ctrl + K
- Include Library >
- Add File...

and plugins for supporting the Arduino...

[More info](#)

1.0.4 < [INSTALL](#)

#### AIPic\_PMC by Arduino

Arduino IDE PLC runtime library for Arduino Portenta Machine Control This is the runtime library and plugins for...

[More info](#)

1.0.4 < [INSTALL](#)

#### Arduino Cloud Provider

Examples by Arduino

Examples of how to connect various Arduino boards to cloud providers

[More info](#)

1.2.1 < [INSTALL](#)

#### Arduino Low Power by Arduino

Power save primitives features for SAMD and nRF52 32bit boards With this library you can manage the low power states of...

[More info](#)

1.2.2 < [INSTALL](#)

BME680.ino

```

1 #include "seeed_bme680.h"
2 #include <SoftwareSerial.h>
3 #include <Bridge.h>

4 #define BME_SCK 13
5 #define BME_MISO 12
6 */

7 /* IIC PROTOCOL D2(SDA), D3(SCL)*/

8 // or debug message

9 void init() {
10   Serial.println("BME680 init started");
11   if (!BME.begin(BME_SCK, BME_MISO)) {
12     Serial.println("BME init failed ! can't find device!");
13   }
14 }

15 // talk successful registration notification
16 // and D1 are used by the Bridge library.

```

Output

- Arduino libraries
- Arduino\_Builtin
- EEPROM
- Ethernet
- Firmata
- HID
- Keyboard
- LiquidCrystal
- Mouse
- SD
- Servo
- SoftwareSerial
- SPI
- Stepper
- TFT
- Wire

Contributed libraries

- Seeed BME680

```

1 // Seeed BME680 library for Arduino
2 // Version 1.0.0
3 // Author: Seeed Studio
4 // Date: 2018-08-01
5 // URL: https://github.com/Seeed-Studio/Seeed-BME680-Arduino
6 // Description: This library provides a simple interface for the Seeed BME680 sensor.
7 // The BME680 is a multi-sensor module that integrates a BME280 temperature and humidity sensor,
8 // a CO2 sensor, a VOC sensor, and a gas sensor. It also includes an I2C interface for
9 // communication with the host microcontroller.
10 // To use this library, you need to include the Seeed-BME680.h header file in your sketch
11 // and call the BME.begin() function to initialize the sensor.
12 // Note: The BME680 uses the I2C protocol, so you need to make sure that your
13 // Arduino board has an I2C interface and that it is properly connected to the sensor.
14 // You can also use the SoftwareSerial library to implement an alternative serial
15 // interface for the BME680.
16 // The BME680 has several pins that you can use for different purposes:
17 // - SDA: I2C data pin
18 // - SCL: I2C clock pin
19 // - VCC: Power supply pin
20 // - GND: Ground pin
21 // - ADO: Analog output pin
22 // - AIN: Analog input pin
23 // - BME_SCK: BME680 SCK pin
24 // - BME_MISO: BME680 MISO pin
25 // - BME_MOSI: BME680 MOSI pin
26 // - BME_CS: BME680 CS pin
27 // - BME_VDD: BME680 VDD pin
28 // - BME_VDDA: BME680 VDDA pin
29 // - BME_VDDG: BME680 VDDG pin
30 // - BME_VDDP: BME680 VDDP pin
31 // - BME_VDDV: BME680 VDDV pin
32 // - BME_VDDW: BME680 VDDW pin
33 // - BME_VDDX: BME680 VDDX pin
34 // - BME_VDDY: BME680 VDDY pin
35 // - BME_VDDZ: BME680 VDDZ pin
36 // - BME_VDDA: BME680 VDDA pin
37 // - BME_VDDG: BME680 VDDG pin
38 // - BME_VDDP: BME680 VDDP pin
39 // - BME_VDDV: BME680 VDDV pin
40 // - BME_VDDW: BME680 VDDW pin
41 // - BME_VDDX: BME680 VDDX pin
42 // - BME_VDDY: BME680 VDDY pin
43 // - BME_VDDZ: BME680 VDDZ pin
44 // - BME_VDDA: BME680 VDDA pin
45 // - BME_VDDG: BME680 VDDG pin
46 // - BME_VDDP: BME680 VDDP pin
47 // - BME_VDDV: BME680 VDDV pin
48 // - BME_VDDW: BME680 VDDW pin
49 // - BME_VDDX: BME680 VDDX pin
50 // - BME_VDDY: BME680 VDDY pin
51 // - BME_VDDZ: BME680 VDDZ pin
52 // - BME_VDDA: BME680 VDDA pin
53 // - BME_VDDG: BME680 VDDG pin
54 // - BME_VDDP: BME680 VDDP pin
55 // - BME_VDDV: BME680 VDDV pin
56 // - BME_VDDW: BME680 VDDW pin
57 // - BME_VDDX: BME680 VDDX pin
58 // - BME_VDDY: BME680 VDDY pin
59 // - BME_VDDZ: BME680 VDDZ pin
60 // - BME_VDDA: BME680 VDDA pin
61 // - BME_VDDG: BME680 VDDG pin
62 // - BME_VDDP: BME680 VDDP pin
63 // - BME_VDDV: BME680 VDDV pin
64 // - BME_VDDW: BME680 VDDW pin
65 // - BME_VDDX: BME680 VDDX pin
66 // - BME_VDDY: BME680 VDDY pin
67 // - BME_VDDZ: BME680 VDDZ pin
68 // - BME_VDDA: BME680 VDDA pin
69 // - BME_VDDG: BME680 VDDG pin
70 // - BME_VDDP: BME680 VDDP pin
71 // - BME_VDDV: BME680 VDDV pin
72 // - BME_VDDW: BME680 VDDW pin
73 // - BME_VDDX: BME680 VDDX pin
74 // - BME_VDDY: BME680 VDDY pin
75 // - BME_VDDZ: BME680 VDDZ pin
76 // - BME_VDDA: BME680 VDDA pin
77 // - BME_VDDG: BME680 VDDG pin
78 // - BME_VDDP: BME680 VDDP pin
79 // - BME_VDDV: BME680 VDDV pin
80 // - BME_VDDW: BME680 VDDW pin
81 // - BME_VDDX: BME680 VDDX pin
82 // - BME_VDDY: BME680 VDDY pin
83 // - BME_VDDZ: BME680 VDDZ pin
84 // - BME_VDDA: BME680 VDDA pin
85 // - BME_VDDG: BME680 VDDG pin
86 // - BME_VDDP: BME680 VDDP pin
87 // - BME_VDDV: BME680 VDDV pin
88 // - BME_VDDW: BME680 VDDW pin
89 // - BME_VDDX: BME680 VDDX pin
90 // - BME_VDDY: BME680 VDDY pin
91 // - BME_VDDZ: BME680 VDDZ pin
92 // - BME_VDDA: BME680 VDDA pin
93 // - BME_VDDG: BME680 VDDG pin
94 // - BME_VDDP: BME680 VDDP pin
95 // - BME_VDDV: BME680 VDDV pin
96 // - BME_VDDW: BME680 VDDW pin
97 // - BME_VDDX: BME680 VDDX pin
98 // - BME_VDDY: BME680 VDDY pin
99 // - BME_VDDZ: BME680 VDDZ pin
100 // - BME_VDDA: BME680 VDDA pin
101 // - BME_VDDG: BME680 VDDG pin
102 // - BME_VDDP: BME680 VDDP pin
103 // - BME_VDDV: BME680 VDDV pin
104 // - BME_VDDW: BME680 VDDW pin
105 // - BME_VDDX: BME680 VDDX pin
106 // - BME_VDDY: BME680 VDDY pin
107 // - BME_VDDZ: BME680 VDDZ pin
108 // - BME_VDDA: BME680 VDDA pin
109 // - BME_VDDG: BME680 VDDG pin
110 // - BME_VDDP: BME680 VDDP pin
111 // - BME_VDDV: BME680 VDDV pin
112 // - BME_VDDW: BME680 VDDW pin
113 // - BME_VDDX: BME680 VDDX pin
114 // - BME_VDDY: BME680 VDDY pin
115 // - BME_VDDZ: BME680 VDDZ pin
116 // - BME_VDDA: BME680 VDDA pin
117 // - BME_VDDG: BME680 VDDG pin
118 // - BME_VDDP: BME680 VDDP pin
119 // - BME_VDDV: BME680 VDDV pin
120 // - BME_VDDW: BME680 VDDW pin
121 // - BME_VDDX: BME680 VDDX pin
122 // - BME_VDDY: BME680 VDDY pin
123 // - BME_VDDZ: BME680 VDDZ pin
124 // - BME_VDDA: BME680 VDDA pin
125 // - BME_VDDG: BME680 VDDG pin
126 // - BME_VDDP: BME680 VDDP pin
127 // - BME_VDDV: BME680 VDDV pin
128 // - BME_VDDW: BME680 VDDW pin
129 // - BME_VDDX: BME680 VDDX pin
130 // - BME_VDDY: BME680 VDDY pin
131 // - BME_VDDZ: BME680 VDDZ pin
132 // - BME_VDDA: BME680 VDDA pin
133 // - BME_VDDG: BME680 VDDG pin
134 // - BME_VDDP: BME680 VDDP pin
135 // - BME_VDDV: BME680 VDDV pin
136 // - BME_VDDW: BME680 VDDW pin
137 // - BME_VDDX: BME680 VDDX pin
138 // - BME_VDDY: BME680 VDDY pin
139 // - BME_VDDZ: BME680 VDDZ pin
140 // - BME_VDDA: BME680 VDDA pin
141 // - BME_VDDG: BME680 VDDG pin
142 // - BME_VDDP: BME680 VDDP pin
143 // - BME_VDDV: BME680 VDDV pin
144 // - BME_VDDW: BME680 VDDW pin
145 // - BME_VDDX: BME680 VDDX pin
146 // - BME_VDDY: BME680 VDDY pin
147 // - BME_VDDZ: BME680 VDDZ pin
148 // - BME_VDDA: BME680 VDDA pin
149 // - BME_VDDG: BME680 VDDG pin
150 // - BME_VDDP: BME680 VDDP pin
151 // - BME_VDDV: BME680 VDDV pin
152 // - BME_VDDW: BME680 VDDW pin
153 // - BME_VDDX: BME680 VDDX pin
154 // - BME_VDDY: BME680 VDDY pin
155 // - BME_VDDZ: BME680 VDDZ pin
156 // - BME_VDDA: BME680 VDDA pin
157 // - BME_VDDG: BME680 VDDG pin
158 // - BME_VDDP: BME680 VDDP pin
159 // - BME_VDDV: BME680 VDDV pin
160 // - BME_VDDW: BME680 VDDW pin
161 // - BME_VDDX: BME680 VDDX pin
162 // - BME_VDDY: BME680 VDDY pin
163 // - BME_VDDZ: BME680 VDDZ pin
164 // - BME_VDDA: BME680 VDDA pin
165 // - BME_VDDG: BME680 VDDG pin
166 // - BME_VDDP: BME680 VDDP pin
167 // - BME_VDDV: BME680 VDDV pin
168 // - BME_VDDW: BME680 VDDW pin
169 // - BME_VDDX: BME680 VDDX pin
170 // - BME_VDDY: BME680 VDDY pin
171 // - BME_VDDZ: BME680 VDDZ pin
172 // - BME_VDDA: BME680 VDDA pin
173 // - BME_VDDG: BME680 VDDG pin
174 // - BME_VDDP: BME680 VDDP pin
175 // - BME_VDDV: BME680 VDDV pin
176 // - BME_VDDW: BME680 VDDW pin
177 // - BME_VDDX: BME680 VDDX pin
178 // - BME_VDDY: BME680 VDDY pin
179 // - BME_VDDZ: BME680 VDDZ pin
180 // - BME_VDDA: BME680 VDDA pin
181 // - BME_VDDG: BME680 VDDG pin
182 // - BME_VDDP: BME680 VDDP pin
183 // - BME_VDDV: BME680 VDDV pin
184 // - BME_VDDW: BME680 VDDW pin
185 // - BME_VDDX: BME680 VDDX pin
186 // - BME_VDDY: BME680 VDDY pin
187 // - BME_VDDZ: BME680 VDDZ pin
188 // - BME_VDDA: BME680 VDDA pin
189 // - BME_VDDG: BME680 VDDG pin
190 // - BME_VDDP: BME680 VDDP pin
191 // - BME_VDDV: BME680 VDDV pin
192 // - BME_VDDW: BME680 VDDW pin
193 // - BME_VDDX: BME680 VDDX pin
194 // - BME_VDDY: BME680 VDDY pin
195 // - BME_VDDZ: BME680 VDDZ pin
196 // - BME_VDDA: BME680 VDDA pin
197 // - BME_VDDG: BME680 VDDG pin
198 // - BME_VDDP: BME680 VDDP pin
199 // - BME_VDDV: BME680 VDDV pin
200 // - BME_VDDW: BME680 VDDW pin
201 // - BME_VDDX: BME680 VDDX pin
202 // - BME_VDDY: BME680 VDDY pin
203 // - BME_VDDZ: BME680 VDDZ pin
204 // - BME_VDDA: BME680 VDDA pin
205 // - BME_VDDG: BME680 VDDG pin
206 // - BME_VDDP: BME680 VDDP pin
207 // - BME_VDDV: BME680 VDDV pin
208 // - BME_VDDW: BME680 VDDW pin
209 // - BME_VDDX: BME680 VDDX pin
210 // - BME_VDDY: BME680 VDDY pin
211 // - BME_VDDZ: BME680 VDDZ pin
212 // - BME_VDDA: BME680 VDDA pin
213 // - BME_VDDG: BME680 VDDG pin
214 // - BME_VDDP: BME680 VDDP pin
215 // - BME_VDDV: BME680 VDDV pin
216 // - BME_VDDW: BME680 VDDW pin
217 // - BME_VDDX: BME680 VDDX pin
218 // - BME_VDDY: BME680 VDDY pin
219 // - BME_VDDZ: BME680 VDDZ pin
220 // - BME_VDDA: BME680 VDDA pin
221 // - BME_VDDG: BME680 VDDG pin
222 // - BME_VDDP: BME680 VDDP pin
223 // - BME_VDDV: BME680 VDDV pin
224 // - BME_VDDW: BME680 VDDW pin
225 // - BME_VDDX: BME680 VDDX pin
226 // - BME_VDDY: BME680 VDDY pin
227 // - BME_VDDZ: BME680 VDDZ pin
228 // - BME_VDDA: BME680 VDDA pin
229 // - BME_VDDG: BME680 VDDG pin
230 // - BME_VDDP: BME680 VDDP pin
231 // - BME_VDDV: BME680 VDDV pin
232 // - BME_VDDW: BME680 VDDW pin
233 // - BME_VDDX: BME680 VDDX pin
234 // - BME_VDDY: BME680 VDDY pin
235 // - BME_VDDZ: BME680 VDDZ pin
236 // - BME_VDDA: BME680 VDDA pin
237 // - BME_VDDG: BME680 VDDG pin
238 // - BME_VDDP: BME680 VDDP pin
239 // - BME_VDDV: BME680 VDDV pin
240 // - BME_VDDW: BME680 VDDW pin
241 // - BME_VDDX: BME680 VDDX pin
242 // - BME_VDDY: BME680 VDDY pin
243 // - BME_VDDZ: BME680 VDDZ pin
244 // - BME_VDDA: BME680 VDDA pin
245 // - BME_VDDG: BME680 VDDG pin
246 // - BME_VDDP: BME680 VDDP pin
247 // - BME_VDDV: BME680 VDDV pin
248 // - BME_VDDW: BME680 VDDW pin
249 // - BME_VDDX: BME680 VDDX pin
250 // - BME_VDDY: BME680 VDDY pin
251 // - BME_VDDZ: BME680 VDDZ pin
252 // - BME_VDDA: BME680 VDDA pin
253 // - BME_VDDG: BME680 VDDG pin
254 // - BME_VDDP: BME680 VDDP pin
255 // - BME_VDDV: BME680 VDDV pin
256 // - BME_VDDW: BME680 VDDW pin
257 // - BME_VDDX: BME680 VDDX pin
258 // - BME_VDDY: BME680 VDDY pin
259 // - BME_VDDZ: BME680 VDDZ pin
260 // - BME_VDDA: BME680 VDDA pin
261 // - BME_VDDG: BME680 VDDG pin
262 // - BME_VDDP: BME680 VDDP pin
263 // - BME_VDDV: BME680 VDDV pin
264 // - BME_VDDW: BME680 VDDW pin
265 // - BME_VDDX: BME680 VDDX pin
266 // - BME_VDDY: BME680 VDDY pin
267 // - BME_VDDZ: BME680 VDDZ pin
268 // - BME_VDDA: BME680 VDDA pin
269 // - BME_VDDG: BME680 VDDG pin
270 // - BME_VDDP: BME680 VDDP pin
271 // - BME_VDDV: BME680 VDDV pin
272 // - BME_VDDW: BME680 VDDW pin
273 // - BME_VDDX: BME680 VDDX pin
274 // - BME_VDDY: BME680 VDDY pin
275 // - BME_VDDZ: BME680 VDDZ pin
276 // - BME_VDDA: BME680 VDDA pin
277 // - BME_VDDG: BME680 VDDG pin
278 // - BME_VDDP: BME680 VDDP pin
279 // - BME_VDDV: BME680 VDDV pin
280 // - BME_VDDW: BME680 VDDW pin
281 // - BME_VDDX: BME680 VDDX pin
282 // - BME_VDDY: BME680 VDDY pin
283 // - BME_VDDZ: BME680 VDDZ pin
284 // - BME_VDDA: BME680 VDDA pin
285 // - BME_VDDG: BME680 VDDG pin
286 // - BME_VDDP: BME680 VDDP pin
287 // - BME_VDDV: BME680 VDDV pin
288 // - BME_VDDW: BME680 VDDW pin
289 // - BME_VDDX: BME680 VDDX pin
290 // - BME_VDDY: BME680 VDDY pin
291 // - BME_VDDZ: BME680 VDDZ pin
292 // - BME_VDDA: BME680 VDDA pin
293 // - BME_VDDG: BME680 VDDG pin
294 // - BME_VDDP: BME680 VDDP pin
295 // - BME_VDDV: BME680 VDDV pin
296 // - BME_VDDW: BME680 VDDW pin
297 // - BME_VDDX: BME680 VDDX pin
298 // - BME_VDDY: BME680 VDDY pin
299 // - BME_VDDZ: BME680 VDDZ pin
300 // - BME_VDDA: BME680 VDDA pin
301 // - BME_VDDG: BME680 VDDG pin
302 // - BME_VDDP: BME680 VDDP pin
303 // - BME_VDDV: BME680 VDDV pin
304 // - BME_VDDW: BME680 VDDW pin
305 // - BME_VDDX: BME680 VDDX pin
306 // - BME_VDDY: BME680 VDDY pin
307 // - BME_VDDZ: BME680 VDDZ pin
308 // - BME_VDDA: BME680 VDDA pin
309 // - BME_VDDG: BME680 VDDG pin
310 // - BME_VDDP: BME680 VDDP pin
311 // - BME_VDDV: BME680 VDDV pin
312 // - BME_VDDW: BME680 VDDW pin
313 // - BME_VDDX: BME680 VDDX pin
314 // - BME_VDDY: BME680 VDDY pin
315 // - BME_VDDZ: BME680 VDDZ pin
316 // - BME_VDDA: BME680 VDDA pin
317 // - BME_VDDG: BME680 VDDG pin
318 // - BME_VDDP: BME680 VDDP pin
319 // - BME_VDDV: BME680 VDDV pin
320 // - BME_VDDW: BME680 VDDW pin
321 // - BME_VDDX: BME680 VDDX pin
322 // - BME_VDDY: BME680 VDDY pin
323 // - BME_VDDZ: BME680 VDDZ pin
324 // - BME_VDDA: BME680 VDDA pin
325 // - BME_VDDG: BME680 VDDG pin
326 // - BME_VDDP: BME680 VDDP pin
327 // - BME_VDDV: BME680 VDDV pin
328 // - BME_VDDW: BME680 VDDW pin
329 // - BME_VDDX: BME680 VDDX pin
330 // - BME_VDDY: BME680 VDDY pin
331 // - BME_VDDZ: BME680 VDDZ pin
332 // - BME_VDDA: BME680 VDDA pin
333 // - BME_VDDG: BME680 VDDG pin
334 // - BME_VDDP: BME680 VDDP pin
335 // - BME_VDDV: BME680 VDDV pin
336 // - BME_VDDW: BME680 VDDW pin
337 // - BME_VDDX: BME680 VDDX pin
338 // - BME_VDDY: BME680 VDDY pin
339 // - BME_VDDZ: BME680 VDDZ pin
340 // - BME_VDDA: BME680 VDDA pin
341 // - BME_VDDG: BME680 VDDG pin
342 // - BME_VDDP: BME680 VDDP pin
343 // - BME_VDDV: BME680 VDDV pin
344 // - BME_VDDW: BME680 VDDW pin
345 // - BME_VDDX: BME680 VDDX pin
346 // - BME_VDDY: BME680 VDDY pin
347 // - BME_VDDZ: BME680 VDDZ pin
348 // - BME_VDDA: BME680 VDDA pin
349 // - BME_VDDG: BME680 VDDG pin
350 // - BME_VDDP: BME680 VDDP pin
351 // - BME_VDDV: BME680 VDDV pin
352 // - BME_VDDW: BME680 VDDW pin
353 // - BME_VDDX: BME680 VDDX pin
354 // - BME_VDDY: BME680 VDDY pin
355 // - BME_VDDZ: BME680 VDDZ pin
356 // - BME_VDDA: BME680 VDDA pin
357 // - BME_VDDG: BME680 VDDG pin
358 // - BME_VDDP: BME680 VDDP pin
359 // - BME_VDDV: BME680 VDDV pin
360 // - BME_VDDW: BME680 VDDW pin
361 // - BME_VDDX: BME680 VDDX pin
362 // - BME_VDDY: BME680 VDDY pin
363 // - BME_VDDZ: BME680 VDDZ pin
364 // - BME_VDDA: BME680 VDDA pin
365 // - BME_VDDG: BME680 VDDG pin
366 // - BME_VDDP: BME680 VDDP pin
367 // - BME_VDDV: BME680 VDDV pin
368 // - BME_VDDW: BME680 VDDW pin
369 // - BME_VDDX: BME680 VDDX pin
370 // - BME_VDDY: BME680 VDDY pin
371 // - BME_VDDZ: BME680 VDDZ pin
372 // - BME_VDDA: BME680 VDDA pin
373 // - BME_VDDG: BME680 VDDG pin
374 // - BME_VDDP: BME680 VDDP pin
375 // - BME_VDDV: BME680 VDDV pin
376 // - BME_VDDW: BME680 VDDW pin
377 // - BME_VDDX: BME680 VDDX pin
378 // - BME_VDDY: BME680 VDDY pin
379 // - BME_VDDZ: BME680 VDDZ pin
380 // - BME_VDDA: BME680 VDDA pin
381 // - BME_VDDG: BME680 VDDG pin
382 // - BME_VDDP: BME680 VDDP pin
383 // - BME_VDDV: BME680 VDDV pin
384 // - BME_VDDW: BME680 VDDW pin
385 // - BME_VDDX: BME680 VDDX pin
386 // - BME_VDDY: BME680 VDDY pin
387 // - BME_VDDZ: BME680 VDDZ pin
388 // - BME_VDDA: BME680 VDDA pin
389 // - BME_VDDG: BME680 VDDG pin
390 // - BME_VDDP: BME680 VDDP pin
391 // - BME_VDDV: BME680 VDDV pin
392 // - BME_VDDW: BME680 VDDW pin
393 // - BME_VDDX: BME680 VDDX pin
394 // - BME_VDDY: BME680 VDDY pin
395 // - BME_VDDZ: BME680 VDDZ pin
396 // - BME_VDDA: BME680 VDDA pin
397 // - BME_VDDG: BME680 VDDG pin
398 // - BME_VDDP: BME680 VDDP pin
399 // - BME_VDDV: BME680 VDDV pin
400 // - BME_VDDW: BME680 VDDW pin
401 // - BME_VDDX: BME680 VDDX pin
402 // - BME_VDDY: BME680 VDDY pin
403 // - BME_VDDZ: BME680 VDDZ pin
404 // - BME_VDDA: BME680 VDDA pin
405 // - BME_VDDG: BME680 VDDG pin
406 // - BME_VDDP: BME680 VDDP pin
407 // - BME_VDDV: BME680 VDDV pin
408 // - BME_VDDW: BME680 VDDW pin
409 // - BME_VDDX: BME680 VDDX pin
410 // - BME_VDDY: BME680 VDDY pin
411 // - BME_VDDZ: BME680 VDDZ pin
412 // - BME_VDDA: BME680 VDDA pin
413 // - BME_VDDG: BME680 VDDG pin
414 // - BME_VDDP: BME680 VDDP pin
415 // - BME_VDDV: BME680 VDDV pin
416 // - BME_VDDW: BME680 VDDW pin
417 // - BME_VDDX: BME680 VDDX pin
418 // - BME_VDDY: BME680 VDDY pin
419 // - BME_VDDZ: BME680 VDDZ pin
420 // - BME_VDDA: BME680 VDDA pin
421 // - BME_VDDG: BME680 VDDG pin
422 // - BME_VDDP: BME680 VDDP pin
423 // - BME_VDDV: BME680 VDDV pin
424 // - BME_VDDW: BME680 VDDW pin
425 // - BME_VDDX: BME680 VDDX pin
426 // - BME_VDDY: BME680 VDDY pin
427 // - BME_VDDZ: BME680 VDDZ pin
428 // - BME_VDDA: BME680 VDDA pin
429 // - BME_VDDG: BME680 VDDG pin
430 // - BME_VDDP: BME680 VDDP pin
431 // - BME_VDDV: BME680 VDDV pin
432 // - BME_VDDW: BME680 VDDW pin
433 // - BME_VDDX: BME680 VDDX pin
434 // - BME_VDDY: BME680 VDDY pin
435 // - BME_VDDZ: BME680 VDDZ pin
436 // - BME_VDDA: BME680 VDDA pin
437 // - BME_VDDG: BME680 VDDG pin
438 // - BME_VDDP: BME680 VDDP pin
439 // - BME_VDDV: BME680 VDDV pin
440 // - BME_VDDW: BME680 VDDW pin
441 // - BME_VDDX: BME680 VDDX pin
442 // - BME_VDDY: BME680 VDDY pin
443 // - BME_VDDZ: BME680 VDDZ pin
444 // - BME_VDDA: BME680 VDDA pin
445 // - BME_VDDG: BME680 VDDG pin
446 // - BME_VDDP: BME680 VDDP pin
447 // - BME_VDDV: BME680 VDDV pin
448 // - BME_VDDW: BME680 VDDW pin
449 // - BME_VDDX: BME680 VDDX pin
450 // - BME_VDDY: BME680 VDDY pin
451 // - BME_VDDZ: BME680 VDDZ pin
452 // - BME_VDDA: BME680 VDDA pin
453 // - BME_VDDG: BME680 VDDG pin
454 // - BME_VDDP: BME680 VDDP pin
455 // - BME_VDDV: BME680 VDDV pin
456 // - BME_VDDW: BME680 VDDW pin
457 // - BME_VDDX: BME680 VDDX pin
458 // - BME_VDDY: BME680 VDDY pin
459 // - B
```



Verify/Compile Ctrl + R  
 Upload Ctrl + U  
 Configure and Upload  
 Upload Using Programmer Ctrl + Shift + U  
 Export Compiled Binary Alt + Ctrl + S  
 Optimize for Debugging  
 Show Sketch Folder Alt + Ctrl + K  
 Include Library... Add File...  
 #define IIC\_ADDR uint8\_t(0x70)

```
10 Seeed_BME680 bme680(IIC_ADDR);
11
12 void setup() {
13     Serial.begin(115200); //for
14
15     while (!bme680.init()) {
16         Serial.println("bme680 i
17         delay(10000);
18     }
19     pinMode(13,OUTPUT); //IoTta
20     Bridge.begin(); // D0 and
21 }
22
23 void loop() {
24     char D13[2];
25     char valueStr[23];
```

## Output

```
C:\Users\Jyneda\Desktop\Sensor680-mast
#included <Bridge.h>
| | | |
compilation terminated.

exit status 1
```

Compilation error: Bridge.h: No such file or directory

- Manage Libraries... Ctrl + Shift + I
- Add .ZIP Library...
- Arduino libraries
- Arduino\_Builtin
- EEPROM
- Ethernet
- Firmata
- HID
- Keyboard
- LiquidCrystal
- Mouse
- SD
- Servo
- SoftwareSerial
- SPI
- Stepper
- TFT
- Wire
- Contributed libraries
- Seeed BME680

A\_BME680.ino:3:10: fatal error: Bridge.h: No such file or directory

✖ Compilation error: Bridge.h: No such file or directory

COPY ERROR MESSAGES

Arduino Yún

√

○

...



## LIBRARY MANAGER

**Bridge**

Type: All

Topic: All

**Bridge** by Arduino ...

Enables the communication between the Linux processor and the microcontroller. For Arduino/Genuino Yún, Yún Shield an...

[More info](#)

1.7.0

**INSTALL****AD7193** by Anne Mahaffey  
[anne@engineerbynights.com](mailto:<anne@engineerbynights.com>)

Library for AD7193 Sigma-Delta ADC with PGA Library for the AD7193 ADC - useful for interfacing with various sensors, suc...

[More info](#)

1.0.0

**INSTALL****ADS1232** by Max Sanchez.  
[msanchez@hardmax.com.pe](mailto:<msanchez@hardmax.com.pe>)

Simple Library to interface the Texas Instruments ADS1232 ADC. A Simple Non-blocking Library for reading from Texas...

[More info](#)

1.0.2

**INSTALL****BM12O2321-A** by  
BESTMODULES

Arduino library for 9-bit UART access to the BM12O2321-A/BMD12K232 that H-bridge Drive Module we can use this...

[More info](#)

1.0.1

**INSTALL**

IDA\_BME680.ino

```

1 #include "seeed_bme680.h"
2 #include <SoftwareSerial.h>
3 #include <Bridge.h>
4
5 #define BME_SCK 13
6 #define BME_MISO 12
7 #define BME_MOSI 11
8 #define BME_CS 10
9 #define IIC_ADDR uint8_t(0x76)
10 Seeed_BME680 bme680(IIC_ADDR); /* IIC PROTOCOL D2(SDA), D3(SCL)*/
11
12 void setup() {
13     Serial.begin(115200); //for debug message
14
15     while (!bme680.init()) {
16         Serial.println("bme680 init failed ! can't find device!");
17         delay(10000);
18     }
19     pinMode(13,OUTPUT); //IoTtalk successful registration notification
20     Bridge.begin();    // D0 and D1 are used by the Bridge library.
21 }
22
23 void loop() {
24     char D13[2];
25     char valueStr[23];

```

## Output

```
C:\Users\Jyneda\Desktop\Sensor680-master\Sensor680-master\IDA_BME680\IDA_BME680.ino:3:10: fatal error: Bridge.h: No such file or directory
| #include <Bridge.h>
| | | | ^~~~~~
compilation terminated.

exit status 1

Compilation error: Bridge.h: No such file or directory
```

✖ Compilation error: Bridge.h: No such file or directory

**COPY ERROR MESSAGES**



Arduino Yún

Verify

**LIBRARY MANAGER**

**Bridge**

Type: All

Topic: All

**Bridge by Arduino**  
1.7.0 installed

Enables the communication between the Linux processor and the microcontroller. For Arduino/Genuino Yún, Yún Shield an...

More info

1.7.0 **REMOVE**

---

**AD7193** by Anne Mahaffey  
<anne@engineerbynights.com>

Library for AD7193 Sigma-Delta ADC with PGA Library for the AD7193 ADC - useful for interfacing with various sensors, suc...

More info

1.0.0 **INSTALL**

---

**ADS1232** by Max Sanchez.  
<msanchez@hardmax.com.pe>

Simple Library to interface the Texas Instruments ADS1232 ADC. A Simple Non-blocking Library for reading from Texas...

More info

1.0.2 **INSTALL**

---

**BM12O2321-A** by  
BESTMODULES

Arduino library for 9-bit UART access to the BM12O2321-A/BMD12K232 that H-bridge Drive Module we can use this...

More info

```
IDA_BME680.ino
1 #include "seeed_bme680.h"
2 #include <SoftwareSerial.h>
3 #include <Bridge.h>
4
5 #define BME_SCK 13
6 #define BME_MISO 12
7 #define BME_MOSI 11
8 #define BME_CS 10
9 #define IIC_ADDR uint8_t(0x76)
10 Seeed_BME680 bme680(IIC_ADDR); /* IIC PROTOCOL D2(SDA), D3(SCL) */
11
12 void setup() {
13     Serial.begin(115200); //for debug message
14
15     while (!bme680.init()) {
16         Serial.println("bme680 init failed ! can't find device!");
17         delay(10000);
18     }
19     pinMode(13,OUTPUT); //IoTtalk successful registration notification
20     Bridge.begin();    // D0 and D1 are used by the Bridge library.
21 }
22
23 void loop() {
24     char D13[2];
25     char valueStr[23];
```

Output

项目使用 20930 字节 (72%) 的程序存储空间。最大值为 28672 字节。  
全局变量使用 1161 个字节 (45%) 的动态内存，剩下 1399 个字节用于局部变量。最大值为 2560 字节。

Done compiling.





...

IDA\_BME680.ino

```

30
31     if (bme680.read_sensor_data()) {
32         Serial.println("Failed");
33         return;
34     }
35
36     //BME 大氣壓 D2, D3
37     float Temperature = bme680.
38     uint32_t AtmosphericPressure;
39     uint32_t Humidity = bme680.sensor_result_value.humidity;
40     uint32_t Gas = bme680.sensor_result_value.gas;
41     dtostrf(Temperature, 6, 2, valueStr);
42     Bridge.put("Temperature", valueStr);
43     dtostrf(AtmosphericPressure, 8, 0, valueStr);
44     Bridge.put("AtPressure", valueStr);
45     dtostrf(Humidity, 8, 0, valueStr);
46     Bridge.put("Humidity", valueStr);
47     dtostrf(Gas, 8, 0, valueStr);
48     Bridge.put("Gas", valueStr);
49     Serial.print("Temp: ");      Serial.print(Temperature);    Serial.println(" C");// unit: C
50     Serial.print("Humidity: ");  Serial.print(Humidity);       Serial.println(" %"); // unit: %
51     Serial.print("Pressure: ");  Serial.print(AtmosphericPressure);  Serial.println(" Pa"); // unit: Pa
52     Serial.print("Gas: ");      Serial.print(Gas);           Serial.println(" ohms"); // unit: Pa
53     Serial.println("=====***====");
54
55     delay(1000);
56 }
57

```

Output Serial Monitor



项目使用 21048 字节 (73%) 的程序存储空间。最大值为 28672 字节。  
个全局变量使用 1177 个字节 (45%) 的动态内存，剩下 1383 个字节用于局部变量。最大值为 2560 字节。

Connecting to programmer: .

Found programmer: Id = "CATERIN"; type = S

|| Software Version = 1.0; No Hardware Version given.

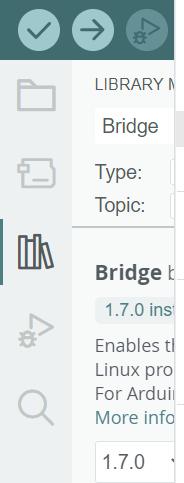
Programmer supports auto addr increment.

Programmer supports buffered memory access with bufsize=128 bytes.

Programmer supports the following devices:

|| Device code: 0x44

(i) Done uploading.

- Auto Format Ctrl + T
- Archive Sketch
- Manage Libraries... Ctrl + Shift + I
- Serial Monitor Ctrl + Shift + M
- Serial Plotter
- Firmware Updater
- Upload SSL Root Certificates
- Board: "Arduino Yún"
- Port: "COM4"
- Get Board Info
- Programmer
- Burn Bootloader

14

```

15   while (!bme680.init()) {
16     Serial.println("bme680 init failed ! can't find device!");
17     delay(10000);
18   }
19   pinMode(13,OUTPUT); //IoTtalk successful registration notification
20   Bridge.begin();    // D0 and D1 are used by the Bridge library.
21 }

22
23 void loop() {
24   char D13[2];
25   char valueStr[23];

```

Output

Message (Enter to send message to 'Arduino Yún' on 'COM4')

New Line 115200 baud

**AD7193** by Anne Mahaffey  
<anne@engineerbynights.com>  
Library for AD7193 Sigma-Delta ADC with  
PGA Library for the AD7193 ADC - useful  
for interfacing with various sensors, suc...  
[More info](#)

1.0.0

**INSTALL**

**ADS1232** by Max Sanchez.  
<m.sanchez@hardmax.com.pe>  
Simple Library to interface the Texas  
Instruments ADS1232 ADC. A Simple Non-  
blocking Library for reading from Texas...  
[More info](#)

1.0.2

**INSTALL**

**BM12O2321-A** by  
BESTMODULES  
Arduino library for 9-bit UART access to  
the BM12O2321-A/BMD12K23 that H-  
bridge Drive Module we can use this...  
[More info](#)

```

=====
Temp: 27.61 C
Humidity: 72 %
Pressure: 100991 Pa
=====
Temp: 27.61 C
Humidity: 72 %
Pressure: 100987 Pa
=====
Temp: 27.62 C
Humidity: 72 %
Pressure: 100993 Pa
=====
```

`:= README.md`

## Sensor680

安裝程序

在Arduino ATmega32u4端，要燒入以下INO檔

Sensor680/IDA BME680/IDA BME680.ino

[https://github.com/IoTtalk/Sensor680/blob/master/IDA\\_BME680/IDA\\_BME680.ino](https://github.com/IoTtalk/Sensor680/blob/master/IDA_BME680/IDA_BME680.ino)

之後SSH登入ArduinoYun Linux介面後，輸入以下指令進行安裝程序。(要先讓Yun連上Internet才可執行下列指令)

```
 wget http://yun.iottalk.tw/yunBME680DA.sh  
 chmod 700 yunBME680DA.sh  
 ./yunBME680DA.sh
```

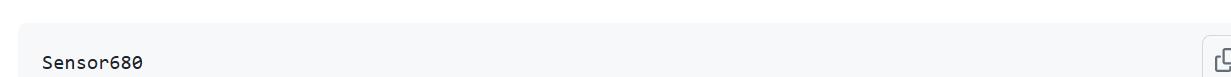
執行完上述指令後，會自動進入編輯 config.py 的狀態

<https://github.com/LoTTalk/Sensor680/blob/master/config.py>

在config.py中，要填寫連線主機資訊(Lines 1 ~ 8)，填寫完成存檔後，斷電重啟板子後，看到白燈閃爍、紅燈亮起，即為成功運作。

Device Model

### 主要用的Le-Tallec-David Model



## Device Features

```
root@Arduino:~# wget http://yun.iottalk.tw/yunBME680DA.sh
--2023-12-11 16:33:17--  http://yun.iottalk.tw/yunBME680DA.sh
Resolving yun.iottalk.tw... 140.113.60.21
Connecting to yun.iottalk.tw|140.113.60.21|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 995 [application/octet-stream]
Saving to: `yunBME680DA.sh'

100%[=====] 995          --.-K/s   in 0s
```

2023-12-11 16:33:17 (11.0 MB/s) - `yunBME680DA.sh' saved [995/995]

```
root@Arduino:~# chmod 700 yunBME680DA.sh
root@Arduino:~# ./yunBME680DA.sh
--2023-12-11 16:33:26--  http://yun.iottalk.tw/packages/distribute_0.6.21-1_ar71xx.ipk
Resolving yun.iottalk.tw... 140.113.60.21
Connecting to yun.iottalk.tw|140.113.60.21|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 154232 (151K) [application/octet-stream]
Saving to: `distribute_0.6.21-1_ar71xx.ipk'
```

100%[=====] 154,232 55.1K/s in 2.7s

2023-12-11 16:33:30 (55.1 KB/s) - `distribute\_0.6.21-1\_ar71xx.ipk' saved [154232/154232]

Installing distribute (0.6.21-1) to root...

```
192.168.50.227 - PuTTY
ServerURL='http://DomainName:9999'
dm_name = 'Sensor680'
d_name = None

MQTT_broker = None
MQTT_port = 1883
MQTT_User = '?'
MQTT_PW = '?'

Comm_interval = 1 # unit:second

odf_list = []

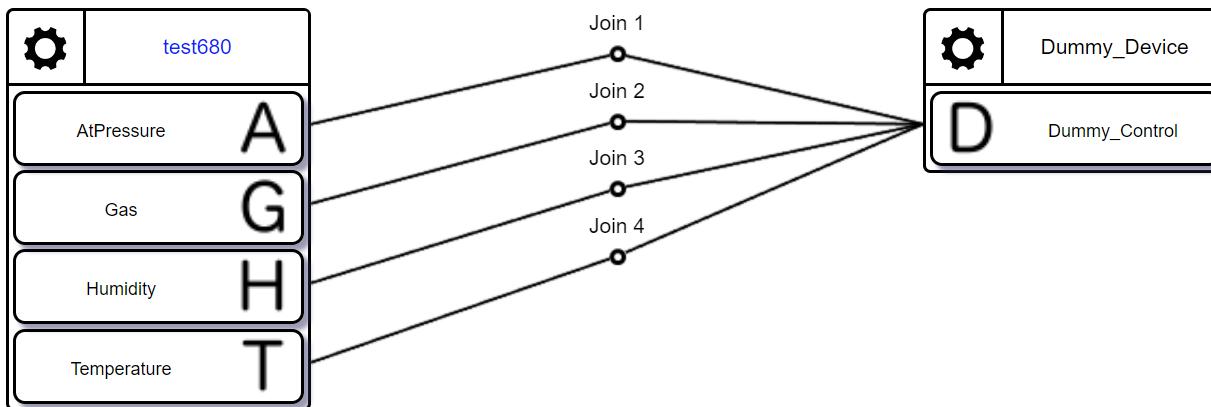
idf_list = [
    ('Temperature', float),
    ('AtPressure', float),
    ('Humidity', float),
    ('Gas', float),
]

~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
- config.py 1/21 4%
```

初步測試，可以輸入 `python DAI.py` 進行執行測試。

如果遭遇**Error 104**，表示Bridge沒運作，可將板子斷電重開即可，不需要再連上板子輸入上述指令。  
看到白燈閃爍、紅燈亮起，就表示運作成功。

```
192.168.50.227 - PuTTY
root@Arduino:~# python DAI.py
Detected features:
    Temperature
    AtPressure
    Humidity
    Gas
IoTtalk Server = http://class.iottalk.tw:9999
This device has successfully registered.
Device name = test680
MQTT broker: class.iottalk.tw
ODF list is empty.
2023-12-11 16:59:20.841214: Push(AtPressure, 100956.0)
2023-12-11 16:59:21.310637: Push(Humidity, 71.02)
2023-12-11 16:59:21.768188: Push(Gas, 64738.0)
2023-12-11 16:59:23.227815: Push(Temperature, 28.05)
2023-12-11 16:59:23.689719: Push(AtPressure, 100952.0)
2023-12-11 16:59:24.152957: Push(Humidity, 71.02)
2023-12-11 16:59:24.608393: Push(Gas, 64590.0)
```

! Model Flush OFF DeleteSimulation OFF Import Export

## IDF Monitor

Sub-stage:	Input	Value
01:00:15		28.04
01:00:18		28.04
01:00:21		28.04
01:00:24		28.02
01:01:23		27.99
01:02:16		28.05

## ODF Monitor

Sub-stage:	Function	Value
01:00:15		28.04
01:00:18		28.04
01:00:21		28.04
01:00:24		28.02
01:01:23		27.99
01:02:16		28.05