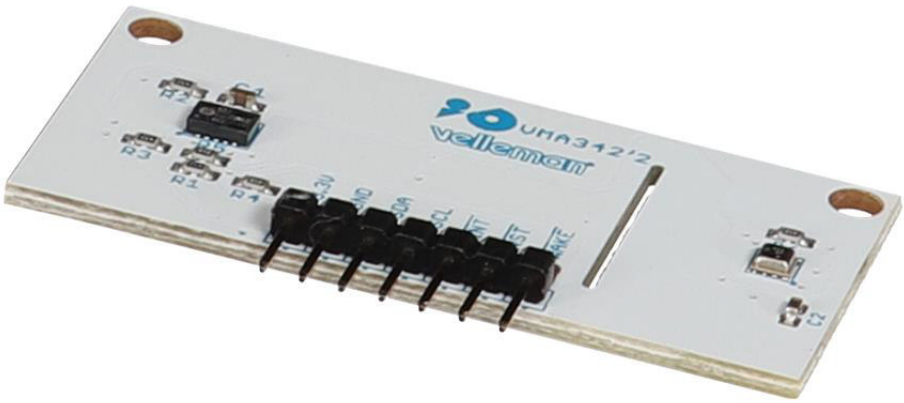
WHADDA

EXCITING ELECTRONICS

USER MANUAL

EN air quality sensor combo board

WPSE342



whadda.com

Introduction

To all residents of the European Union
Important environmental information about this product
This symbol on the device or the package indicates that disposal of the device after its lifecycle could harm the environment. Do not dispose of the unit (or batteries) as unsorted municipal waste; it should be taken to a specialized company for recycling. This device should be returned to your distributor or to a local recycling service. Respect the local environmental rules.
If in doubt, contact your local waste disposal authorities.

Thank you for choosing Whadda! Please read the manual thoroughly before bringing this device into service. If the device was damaged in transit, do not install or use it and contact your dealer.

Safety Instructions

Read and understand this manual and all safety signs before using this appliance.

For indoor use only.

- This device can be used by children aged from 8 years and above, and persons with reduced physical, sensory or mental capabilities or lack of experience and knowledge if they have been given supervision or instruction concerning the use of the device in a safe way and understand the hazards involved. Children shall not play with the device. Cleaning and user maintenance shall not be made by children without supervision.

General Guidelines

- Refer to the Velleman® Service and Quality Warranty on the last pages of this manual.
- All modifications of the device are forbidden for safety reasons. Damage caused by user modifications to the device is not covered by the warranty.
- Only use the device for its intended purpose. Using the device in an unauthorized way will void the warranty.
- Damage caused by disregard of certain guidelines in this manual is not covered by the warranty and the dealer will not accept responsibility for any ensuing defects or problems.
- Nor Velleman Group nv nor its dealers can be held responsible for any damage (extraordinary, incidental or indirect) – of any nature (financial, physical...) arising from the possession, use or failure of this product.
- Keep this manual for future reference.

What is Arduino®

Arduino® is an open-source prototyping platform based on easy-to-use hardware and software. Arduino® boards are able to read inputs – light-on sensor, a finger on a button or a Twitter message – and turn it into an output – activating of a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so, you use the Arduino programming language (based on Wiring) and the Arduino® software IDE (based on Processing). Additional shields/modules/components are required for reading a twitter message or publishing online. Surf to www.arduino.cc for more information.

Product Overview

This air quality combo board senses the atmospheric quality by using the popular CCS811 and BME280 ICs. It provides a variety of environmental data including barometric pressure, humidity, temperature, TVOCs and equivalent CO2 (or eCO2) levels. Communication is possible through the I^2C protocol.
The CCS811 is an exceedingly popular sensor, providing readings for equivalent CO2 (or eCO2) in parts per million (ppm) and total volatile organic compounds (TVOC) in the parts per billion (ppb).
The CCS811 also allows to fine-tune its readings if it has access to the current humidity and temperature.

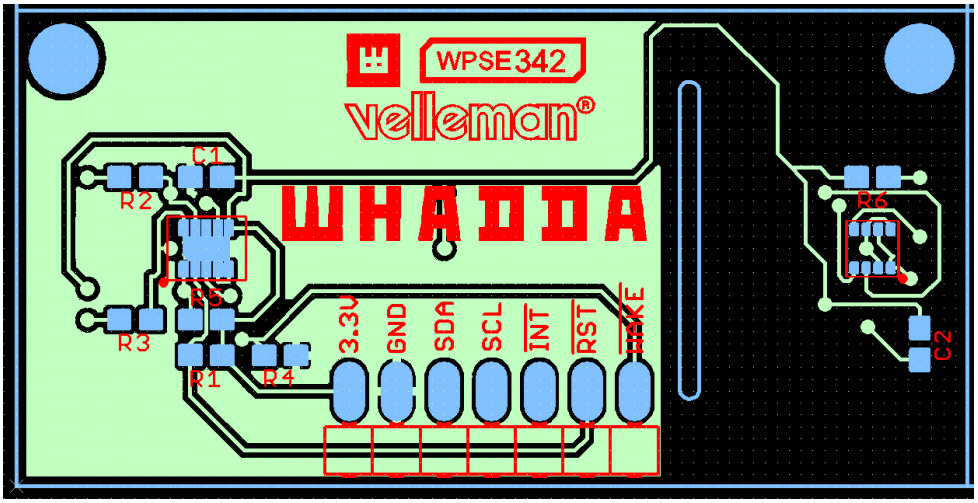
Specifications

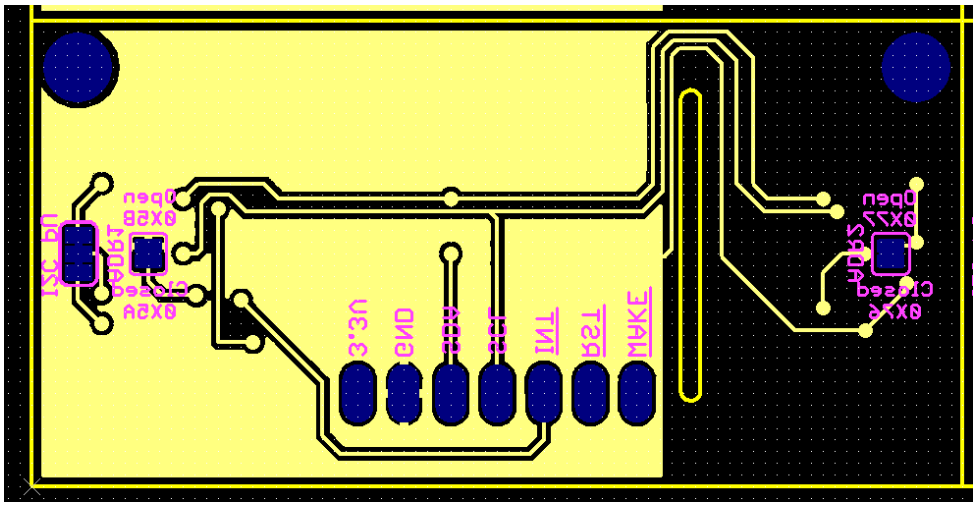
- operation voltage: 3.3 V
- eCO2 sensing: 400-8,192 ppm
- total volatile organic compound (TVOC): 0-1,187 ppm
- temperature range: -40 °C to 85 °C (-40 °F to 185 °F)
- humidity range: 0-100 % RH, ≤ 3 % from 20-80 %
- pressure range: 30,000-110,000 Pa
  - relative accuracy: 12 Pa
  - absolute accuracy: 100 Pa
- altitude range: 0-9.2 km (0-30,000 ft)
  - relative accuracy: 1 m (3.3 ft) @ MSL
  - 2 m (6.6 ft) @ 9.2 km (30,000 ft)
- dimensions:
  - length: 51 mm
  - width: 30.4 mm
  - height: 5.4 mm
- weight: 4.80 g

Features

- CCS811 and BME280 ICs
- eCO2 sensor
- TVOC sensor
- temperature sensor
- humidity sensor
- pressure sensor
- altitude sensor

Circuit Diagram



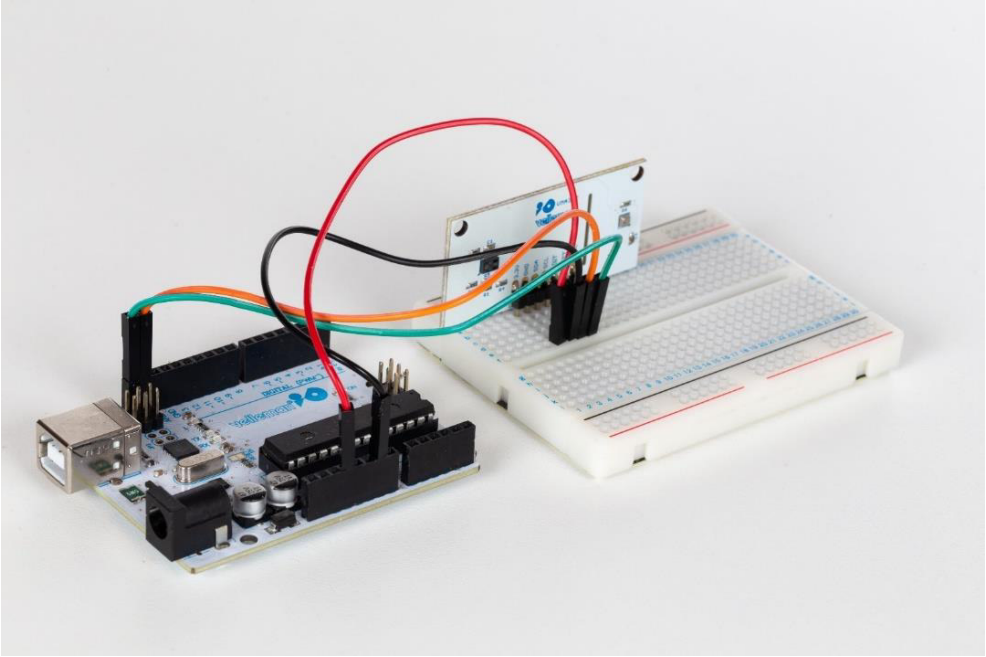


Connection to Arduino® – Pin Assignment

WPSE342	Arduino® Uno	Arduino® Mega
3.3 V	3.3 V	3.3 V
GND	GND	GND
SDA	SDA (A4)	20
SCL	SCL (A5)	21

Using the Air Quality Sensor Module

We are going to use I²C communication with the BME280 and CSS811 sensor module.
For that, connect the air quality sensor to the Arduino® Uno **SDA** and **SCL** pins, as shown in the following schematic diagram below or check the pin assignment table above.

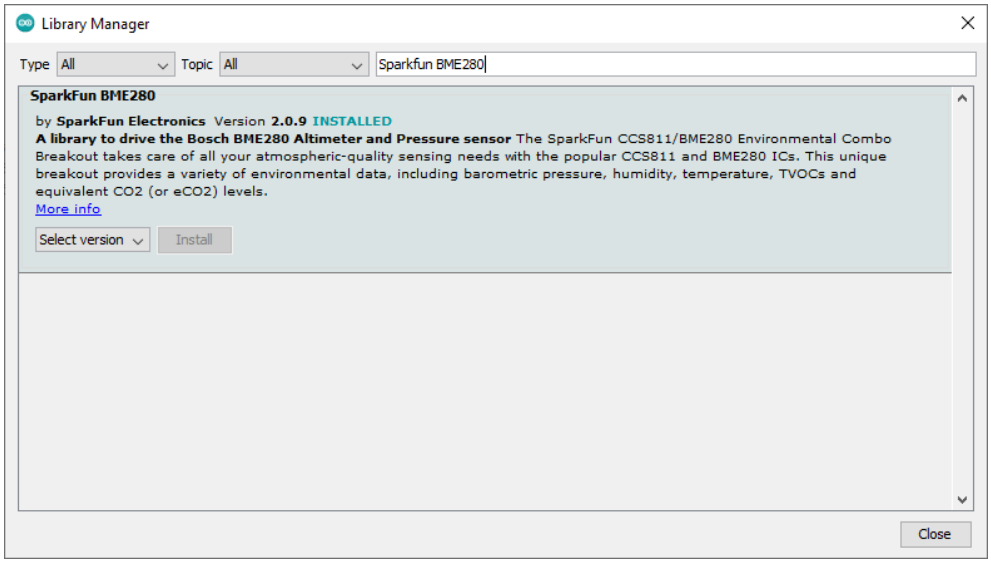


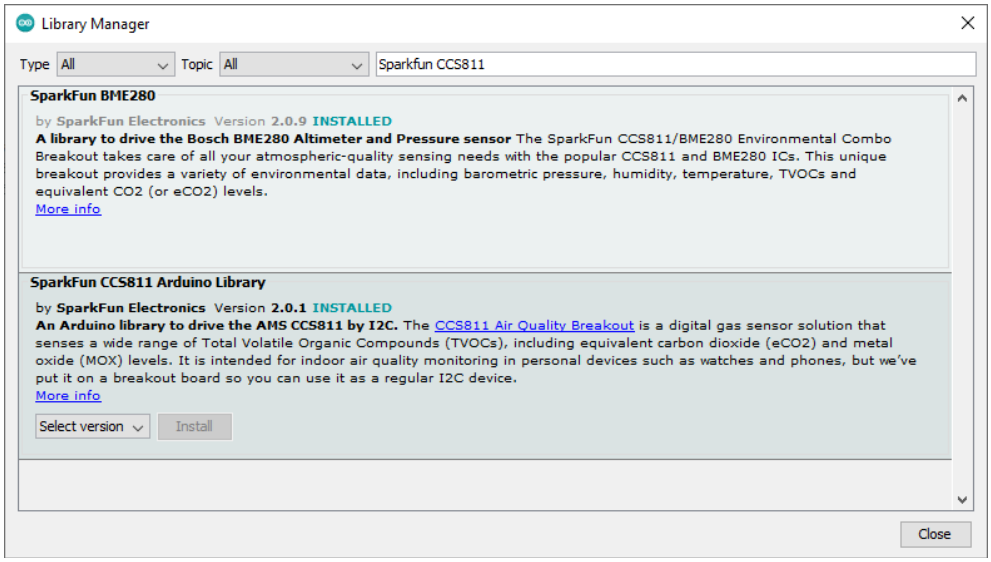
Installing the BME280 and CCS811 Libraries

To get data readings from the air quality sensor board, you need to use the Sparkfun BME280 and CCS811 libraries. Follow the next steps to install the library in your Arduino® IDE.

Open your Arduino® IDE and go to **Sketch** > **Include Library** > **Manage Libraries**. The Library Manager should open.

- Search for **Sparkfun bme280** in the search box and install the library.
- Search for **Sparkfun CCS811** in the search box and install the library.





Coding

To make sure everything is correctly wired, and the air quality sensor is working, download the example code from our product page:

https://whadda.com/product/air-quality-sensor-combo-board-wpse342/

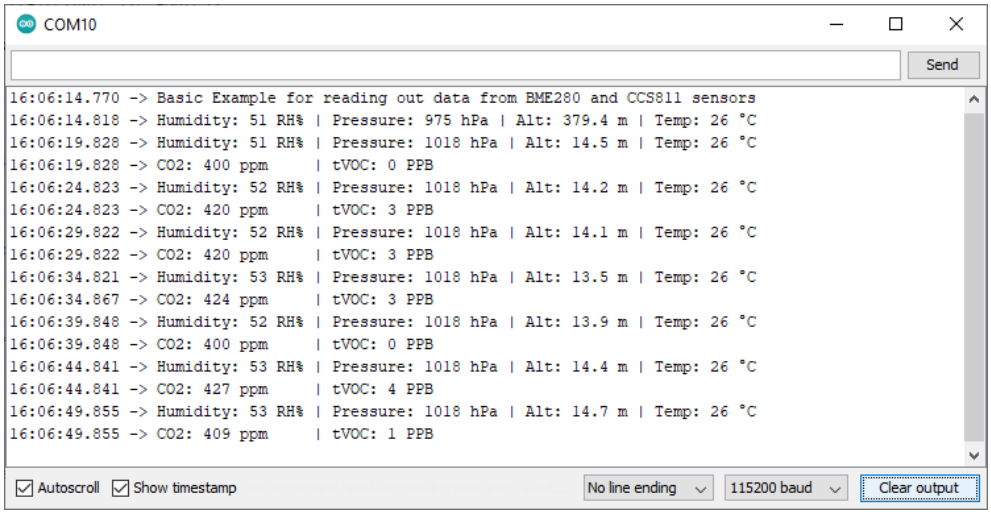
You can download the code from the **Downloads** section or copy paste the code here below.

```
/*
  @@@@@@@@@@@@@@@@@@@@

  @@@@@@@@@@@@@@@@@@@@            @@@@    @@@    @@@@    @@@@    @@@@       @@@@@@
@@        @@@@@@@@@@@@@@    @@@@@@@@@@@@@       @@@@@@@@
  @@@@@@@@@@@@@@@@@@@@            @@@@    @@@    @@@@    @@@@    @@@@       @@@@@@@
@@        @@@@@@@@@@@@@@    @@@@@@@@@@@@@       @@@@@@@@
  @@@@@  @@@  @@@  @@@@@            @@@@    @@@    @@@@    @@@@    @@@@       @@@@  @
@@@         @@@@@   @@@@@      @@@@@   @@@@       @@@@   @@@@
  @@@@@              @@@@@            @@@@   @@@@    @@@@    @@@@@@@@@@@       @@@@  @
@@@         @@@@@   @@@@@      @@@@@   @@@@       @@@@   @@@@
  @@@@@   @    @   @@@@@            @@@@   @@@@    @@@@    @@@@@@@@@@@       @@@@@@@
@@@         @@@@@   @@@@@      @@@@@   @@@@       @@@@@@@@@@
  @@@@@              @@@@@            @@@@  @@@@@   @@@@@    @@@@    @@@@      @@@@@@@@
@@@       @@@@@@   @@@@@    @@@@@@   @@@@       @@@@@@@@@@
  @@@@@@@@@@@@@@@@@@@@            @@@@@@@@@@@@@@@@@@    @@@@    @@@@      @@@@    @
@@@@      @@@@@@@@@@@@@    @@@@@@@@@@@@@      @@@@    @@@@
  @@@@@@@@@@@@@@@@@@@@
*/

/*
  Whadda WPSE342 Air Quality combo board sensor:

  This air quality combo board senses the atmospheric-quality by using the popular CCS811 and BME280 ICs. It provides a variety of environmental data including:
  barometric pressure, humidity, temperature, TVOCs and equivalent CO2 (or eCO2) levels.
  Communication is possible through the I²C protocol.

  The CCS811 is an exceedingly popular sensor, providing readings for equivalent CO2 (or eCO2) in parts per million (PPM)
  and total volatile organic compounds (TVOC) in the parts per billion (PPB).
  The CCS811 also has a feature that allows it to fine-tune its readings if it has access to the current humidity and temperature.


  Pin Configuration Sensor board to arduino using I²C interface:
  ---------------------------------------------------------------

  WPSE342 | Arduino Uno

  3V3       =    3V3 (VCC)
  GND     =  GND (Ground)
  SDA     =  UNO SDA (A4) / Mega SDA (44 - IDE 20)
  SCL       =    UNO SCL (A5) / Mega SCL (43 - IDE 21)

  Required Libraries:
  -------------------
  SparkFunBME280.h
  SparkFunCCS811.h
```

```
37.
38.   Standard Arduino Library:
39.   ------------------------
40.
41.   Wire.h
42.
43.
44.   For more informarion about WPSE342 Air Quality sensor, consult the manual at the
    WPSE342 product page on https://whadda.com/product/air-quality-sensor-combo-board-
    wpse342/
45.
46. */
47.
48. #include <Wire.h>
49. #include "SparkFunBME280.h" //Click here to get the library:
    http://librarymanager/All#SparkFun_BME280
50. #include "SparkFunCCS811.h" //Click here to get the library:
    http://librarymanager/All#SparkFun_CCS811
51.
52. BME280 myBME280;
53.
54. #define CCS811_ADDR 0x5B //Default I2C Address
55. //#define CCS811_ADDR 0x5A //Alternate I2C Address
56.
57. CCS811 myCCS811(CCS811_ADDR);
58.
59.
60. void setup() {
61.
62.  // Open serial communications and wait for port to open:
63.
64.  Serial.begin(115200);
65.
66.  Serial.println("Basic Example for reading out data from BME280 and CCS811 sensors");
67.
68.  Wire.begin();
69.
70.  if (myBME280.beginI2C() == false) //Begin communication over I2C
71.   {
72.     Serial.println("BME280 sensor did not respond. Please check wiring. Freezing...");
73.     while(1); //Freeze
74.   }
75.
76.   myBME280.setReferencePressure(101957); //Adjust the sea level pressure used for
    altitude calculations
77.
78.   if (myCCS811.begin() == false)
79.   {
```

```
80.     Serial.print("CCS811 error did not respond. Please check wiring. Freezing...");
81.     while (1)
82.       ;
83.   }
84. }
85.
86. void loop() {
87.
88.   // Print out the raw data of sensors in Float, to serial monitor
89.
90.   Serial.print("Humidity: ");
91.   Serial.print(myBME280.readFloatHumidity(), 0);
92.   Serial.print(" RH%");
93.
94.   Serial.print(" | Pressure: ");
95.   Serial.print(myBME280.readFloatPressure() /100.0F, 0);
96.   Serial.print(" hPa");
97.
98.   Serial.print(" | Alt: ");
99.   Serial.print(myBME280.readFloatAltitudeMeters(), 1);  //Show Altitude in meters.
100.   //Serial.print(myBME280.readFloatAltitudeFeet(), 1);  //Show Altitude in feets.
101.   Serial.print(" m");
102.
103.   Serial.print(" | Temp: ");
104.   //Serial.print(myBME280.readTempF(), 0);  // Show temp. in °Fahrenheit
105.   Serial.print(myBME280.readTempC(), 0);    // Show temp. in °Celsius
106.   Serial.print(" °C");
107.
108.   Serial.println();
109.
110.
111.   //Check to see if data is ready with .dataAvailable()
112.   if (myCCS811.dataAvailable())
113.   {
114.     //If so, have the sensor read and calculate the results.
115.     //Get them later
116.     myCCS811.readAlgorithmResults();
117.
118.     Serial.print("CO2: ");
119.     //Returns calculated CO2 reading
120.     Serial.print(myCCS811.getCO2());
121.     Serial.print(" ppm");
122.     Serial.print("     | tVOC: ");
123.     //Returns calculated TVOC reading
124.     Serial.print(myCCS811.getTVOC());
125.     Serial.print(" PPB");
126.     Serial.println();
127.   }
128.   delay(5000); //Don't spam the I2C bus
129. }
```

Now upload the code to your Arduino® Uno board. Make sure to select the correct **board** and **COM port**. Open the serial monitor with baud rate **115200**, normally the basic sensor float data will be presented in the serial monitor. If everything is working properly, you will see a similar message on the serial monitor.



Code Explanation

Libraries

The basic example code starts by including the required libraries to interface with the onboard sensors of the air quality sensor combo board. These are the standard Arduino® **Wire** library using the I²C interface, the **Sparkfun BME280** and **Sparkfun CCS811** library.

```
#include <Wire.h>
#include "SparkFunBME280.h"
#include "SparkFunCCS811.h"
```

I²C

We are using the I²C interface protocol by default. We need to create 2 objects: a **BME280** object called **myBME280**, and a **CCS811** object called **myCCS811**.

```
BME280 myBME280;
CCS811 myCCS811(CCS811_ADDR);
```

We initialize the CCS811 sensor with address **0x5B** (standard).

```
#define CCS811_ADDR 0x5B
```

Setup()

In the Setup(), we start a serial communication at baud rate **115200**.

```
Serial.begin(115200);
```

Estimate the altitude in meters, based on the pressure at the sea level. Google **sea level pressure map** for more information.

http://weather.unisys.com/surface/sfc_con.php?image=pr&inv=0&t=cur

https://www.atmos.illinois.edu/weather/tree/viewer.pl?launch/sfcslp

```
myBME280.setReferencePressure(101500); //Adjust the sea level pressure used for altitude calculations
```

Initializing the Sensors

BME280

```
if (myBME280.beginI2C() == false) //Begin communication over I2C
 {
    Serial.println("BME280 sensor did not respond. Please check wiring. Freezing...");
    while(1); //Freeze
  }
```

CCS811

```
if (myCCS811.begin() == false)
  {
    Serial.print("CCS811 error did not respond. Please check wiring. Freezing...");
    while(1)
  ;
}
```

Reading Data from the Sensors

BME280

```
myBME280.readFloatHumidity()
myBME280.readFloatPressure()
myBME280.readFloatAltitudeMeters()
myBME280.readFloatAltitudeFeet()
myBME280.readTempF() // Show temp. in °Fahrenheit
myBME280.readTempC() // Sowh temp. in °Celsius
```

CCS811

```
myCCS811.getCO2()
myCCS811.getTVOC()
```

whadda.com