

# Smart Gases 3.0

## Technical Guide



Document version: v7.0 - 11/2016  
© Libelium Comunicaciones Distribuidas S.L.

## INDEX

<b>1. General .....</b>	<b>5</b>
1.1. General and safety information .....	5
1.2. Conditions of use .....	5
<b>2. New version: Smart Gases v3.0.....</b>	<b>6</b>
<b>3. WaspMote Plug &amp; Sense!.....</b>	<b>7</b>
3.1. Features .....	7
<b>4. General view .....</b>	<b>8</b>
4.1. Specifications .....	8
4.2. Parts included.....	11
4.3. Identification.....	12
4.4. Sensor probes.....	14
4.5. Solar powered .....	14
4.6. Programming the Nodes.....	16
4.7. Radio interfaces .....	18
4.8. Models.....	19
4.8.1. Smart Environment .....	20
<b>5. Gases PRO Sensor Board (Calibrated) .....</b>	<b>22</b>
<b>6. Hardware .....</b>	<b>26</b>
6.1. General Description .....	26
6.2. Specifications .....	26
6.3. Electrical Characteristics.....	27
<b>7. Sensors .....</b>	<b>28</b>
7.1. General considerations in the use of the sensors.....	28
7.2. Starting with the gas sensors .....	29
7.2.1. Sensor calibration .....	29
7.2.2. Gain and load resistor configuration.....	29
7.2.3. Converting the read data .....	30
7.2.4. Calibrated sensors performance.....	30
<b>8. Sensors .....</b>	<b>31</b>
8.1. Temperature, Humidity and Pressure Sensor.....	31
8.1.1. Specifications.....	31
8.1.2. Measurement process .....	32

8.2. Carbon Monoxide (CO) Sensor – TGS2442.....	33
8.2.1. Specifications.....	33
8.2.2. Measurement Process.....	33
8.3. Carbon Dioxide (CO <sub>2</sub> ) Sensor – TGS4161 .....	34
8.3.1. Specifications.....	34
8.3.2. Measurement Process.....	35
8.4. Molecular Oxygen (O <sub>2</sub> ) Sensor – SK-25 .....	36
8.4.1. Specifications.....	36
8.4.2. Measurement Process.....	36
8.5. Nitrogen Dioxide (NO <sub>2</sub> ) Sensor - MiCS-2714.....	37
8.5.1. Specifications.....	37
8.5.2. Measurement process .....	37
8.6. Ammonia (NH <sub>3</sub> ) sensor – TGS2444 .....	38
8.6.1. Specifications.....	38
8.6.2. Measurement Process.....	38
8.7. Methane (CH <sub>4</sub> ) sensor – TGS2611.....	40
8.7.1. Specifications.....	40
8.7.2. Measurement Process.....	40
8.8. Liquefied Petroleum Gas Sensor - TGS2610 .....	41
8.8.1. Specifications.....	41
8.8.2. Measurement Process.....	41
8.9. Air Pollutants I sensor – TGS2600.....	43
8.9.1. Specifications.....	43
8.9.2. Measurement Process.....	43
8.10. Air Pollutants II sensor – TGS2602 .....	45
8.10.1. Specifications .....	45
8.10.2. Measurement Process .....	45
8.11. Solvent Vapors Sensor - TGS2620 .....	46
8.11.1. Specifications .....	46
8.11.2. Measurement Process .....	46
8.12. Ozone (O <sub>3</sub> ) Sensor - MiCS-2614.....	48
8.12.1. Specifications .....	48
8.12.2. Measurement process.....	48
8.13. VOC's Sensor - MiCS-5524.....	49
8.13.1. Specifications .....	49
8.13.2. Measurement process.....	49
8.14. Ultrasound sensor probe (MaxSonar® from MaxBotix™).....	51
8.14.1. Specifications .....	51
8.14.2. Measurement process.....	53
8.14.3. Socket .....	53
8.15. Luminosity Sensor.....	54
8.15.1. Specifications .....	54
8.15.2. Measurement process.....	54

8.16. Design and connections .....	55
8.16.1. Socket 1 .....	55
8.16.2. Socket 2 .....	55
8.16.3. Socket 3 .....	56
8.16.4. Socket 4 .....	56
8.16.5. Socket 5 .....	57
8.16.6. Socket 6 and socket 7 .....	57
8.16.7. Socket 8 .....	58
<b>9. Board configuration and programming .....</b>	<b>59</b>
9.1. Hardware configuration .....	59
9.2. Library .....	59
<b>10. Consumption .....</b>	<b>62</b>
10.1. Power control .....	62
10.2. Consumption table .....	62
10.3. Low Consumption Mode .....	62
<b>11. API Changelog .....</b>	<b>63</b>
<b>12. Maintenance .....</b>	<b>64</b>
<b>13. Disposal and Recycle .....</b>	<b>65</b>
<b>14. Certifications .....</b>	<b>66</b>

# 1. General

## 1.1. General and safety information

In this section, the term "Waspmote" encompasses both the Waspmote device itself and its modules and sensor boards.

- Read through the document "General Conditions of Libelium Sale and Use".
- Do not allow contact of metallic objects with the electronic part to avoid injuries and burns.
- NEVER submerge the device in any liquid.
- Keep the device in a dry place and away from any liquid which may spill.
- Waspmote consists of highly sensitive electronics which is accessible to the exterior, handle with great care and avoid bangs or hard brushing against surfaces.
- Check the product specifications section for the maximum allowed power voltage and amperage range and consequently always use a current transformer and a battery which works within that range. Libelium is only responsible for the correct operation of the device with the batteries, power supplies and chargers which it supplies.
- Keep the device within the specified range of temperatures in the specifications section.
- Do not connect or power the device with damaged cables or batteries.
- Place the device in a place only accessible to maintenance personnel (a restricted area).
- Keep children away from the device in all circumstances.
- If there is an electrical failure, disconnect the main switch immediately and disconnect that battery or any other power supply that is being used.
- If using a car lighter as a power supply, be sure to respect the voltage and current data specified in the "Power Supplies" section.
- If using a battery in combination or not with a solar panel as a power supply, be sure to use the voltage and current data specified in the "Power supplies" section.
- If a software or hardware failure occurs, consult the Libelium Web [Development section](#).
- Check that the frequency and power of the communication radio modules together with the integrated antennas are allowed in the area where you want to use the device.
- Waspmote is a device to be integrated in a casing so that it is protected from environmental conditions such as light, dust, humidity or sudden changes in temperature. The board supplied "as is" is not recommended for a final installation as the electronic components are open to the air and may be damaged.

## 1.2. Conditions of use

- Read the "General and Safety Information" section carefully and keep the manual for future consultation.
- Use Waspmote in accordance with the electrical specifications and the environment described in the "Electrical Data" section of this manual.
- Waspmote and its components and modules are supplied as electronic boards to be integrated within a final product. This product must contain an enclosure to protect it from dust, humidity and other environmental interactions. In the event of outside use, this enclosure must be rated at least IP-65.
- Do not place Waspmote in contact with metallic surfaces; they could cause short-circuits which will permanently damage it.

Further information you may need can be found at <http://www.libelium.com/development/waspmote>

The "General Conditions of Libelium Sale and Use" document can be found at:  
[http://www.libelium.com/development/waspmote/technical\\_service](http://www.libelium.com/development/waspmote/technical_service)

## 2. New version: Smart Gases v3.0

This guide explains the new Smart Gases Sensor Board v3.0. This board was specifically designed for our new product lines Wasp mote v15 and Plug & Sense! v15, released on October 2016.

This board is not compatible with Wasp mote v12 or Plug & Sense! v12, so it is NOT recommended to mix product generations. If you are using previous versions of our products, please use the corresponding guides, available on our [Development website](#).

You can get more information about the generation change on the document "[New generation of Libelium product lines](#)".

Differences of Smart Gases v3.0 with previous versions:

- Included the new BME280 temperature, humidity and air pressure sensor. This tiny, digital and cost-efficient sensor measures 3 parameters with great accuracy and range.
- The BME280 replaces the previous analog Temperature sensor (MCP9700A), the analog Humidity sensor (808H5V5) and the Atmospheric Pressure sensor (MPX4115A).
- Many internal changes have been made on the board circuitry to reduce the BoM. Now some sensors are multiplexed, avoiding duplicate parts.
- Clearer silkscreen for easier connection.
- I2C socket allows the connections of digital sensors.
- Added an I2C isolator chip to avoid affecting to the Wasp mote I2C bus.
- The library has been improved to make the board handling easier.
- New connectors to improve the Plug & Sense! wiring, making it more robust.

## 3. Waspmote Plug & Sense!

The Waspmote Plug & Sense! line allows you to easily deploy Internet of Things networks in an easy and scalable way, ensuring minimum maintenance costs. The platform consists of a robust waterproof enclosure with specific external sockets to connect the sensors, the solar panel, the antenna and even the USB cable in order to reprogram the node. It has been specially designed to be scalable, easy to deploy and maintain.

**Note:** For a complete reference guide download the "Waspmote Plug & Sense! Technical Guide" in the [Development section](#) of the [Libelium website](#).

### 3.1. Features

- Robust waterproof IP65 enclosure
- Add or change a sensor probe in seconds
- Solar powered with internal and external panel options
- Radios available: 802.15.4, 868 MHz, 900 MHz, WiFi, 4G, Sigfox and LoRaWAN
- Over the air programming (OTAP) of multiple nodes at once (via WiFi or 4G radios)
- Special holders and brackets ready for installation in street lights and building fronts
- Graphical and intuitive programming interface Code Generator (coming in 2017)
- Built-in, 3-axes accelerometer
- External, contactless reset with magnet
- External SIM connector for the 4G models
- Fully certified: CE (Europe), FCC (USA), IC (Canada), ANATEL (Brazil), RCM (Australia), PTCRB (USA, cellular connectivity), AT&T (USA, cellular connectivity)



Figure: Waspmote Plug & Sense!

## 4. General view

This section shows main parts of WaspMote Plug & Sense! and a brief description of each one. In later sections all parts will be described deeply.

### 4.1. Specifications

- **Material:** polycarbonate
- **Sealing:** polyurethane
- **Cover screws:** stainless steel
- **Ingress protection:** IP65
- **Impact resistance:** IK08
- **Rated insulation voltage AC:** 690 V
- **Rated insulation voltage DC:** 1000 V
- **Heavy metals-free:** Yes
- **Weatherproof:** true - nach UL 746 C
- **Ambient temperature (min.):** -10 °C
- **Ambient temperature (max.):** 50 °C
- **Approximated weight:** 800 g

In the pictures included below it is shown a general view of WaspMote Plug & Sense! main parts. Some elements are dedicated to node control, others are designated to sensor connection and other parts are just identification elements. All of them will be described along this guide.

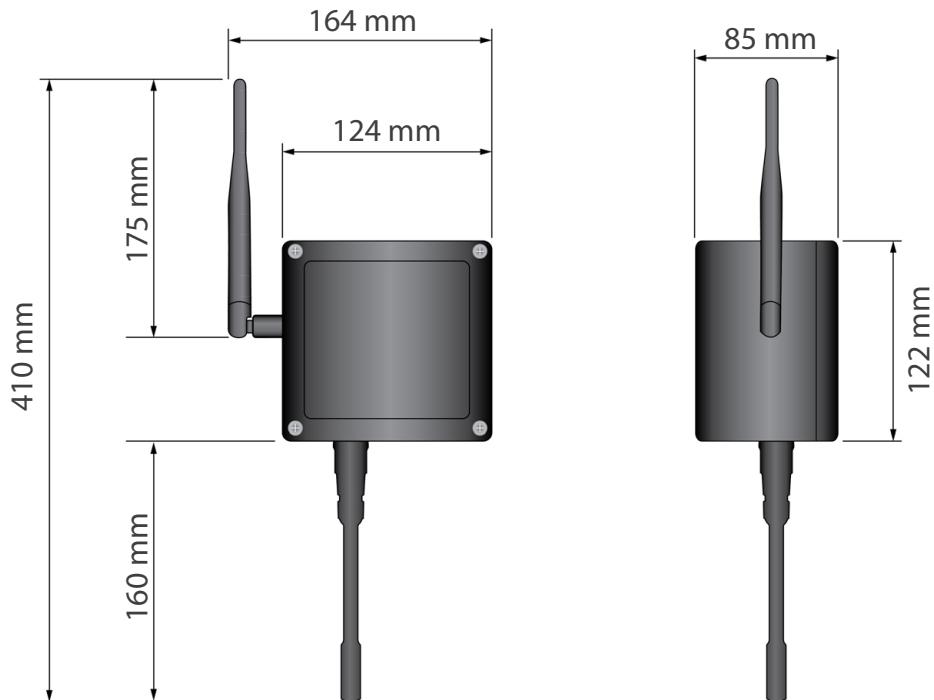


Figure : Main view of WaspMote Plug & Sense!

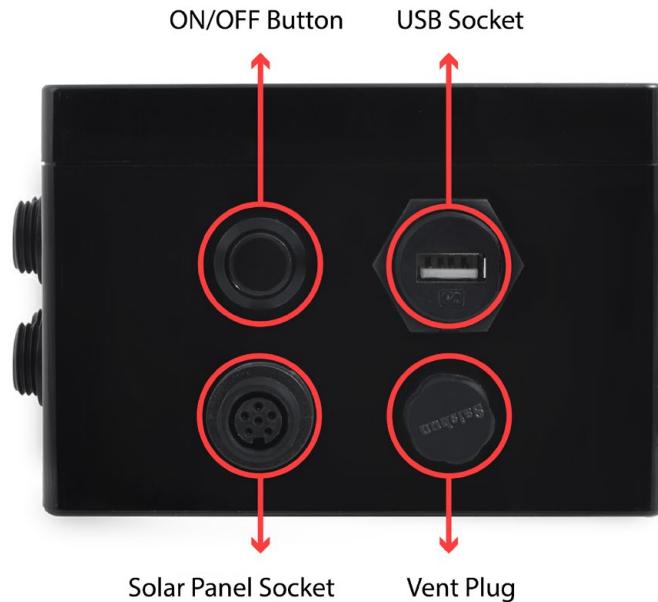


Figure : Control side of the enclosure

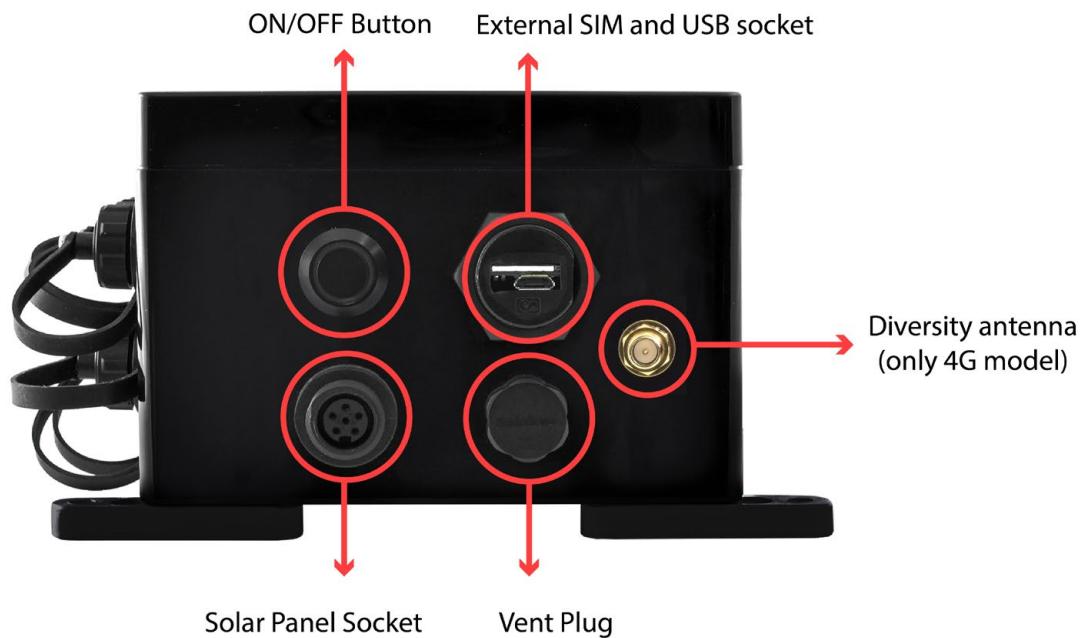


Figure : Control side of the enclosure for 4G model



Figure : Sensor side of the enclosure



Figure : Antenna side of the enclosure



Figure : Front view of the enclosure



Figure : Back view of the enclosure



Figure : Warranty stickers of the enclosure

**Important note:** Do not handle black stickers seals of the enclosure (Warranty stickers). Their integrity is the proof that Wasp mote Plug & Sense! has not been opened. If they have been handled, damaged or broken, the warranty is automatically void.

## 4.2. Parts included

Next picture shows Wasp mote Plug & Sense! and all of its elements. Some of them are optional accessories that may not be included.



Figure : Wasp mote Plug & Sense! accessories: 1 enclosure, 2 sensor probes, 3 external solar panel, 4 USB cable, 5 antenna, 6 cable ties, 7 mounting feet (screwed to the enclosure), 8 extension cord, 9 solar panel cable, 10 wall plugs & screws

## 4.3. Identification

Each Waspmote model is identified by stickers. Next figure shows front sticker.

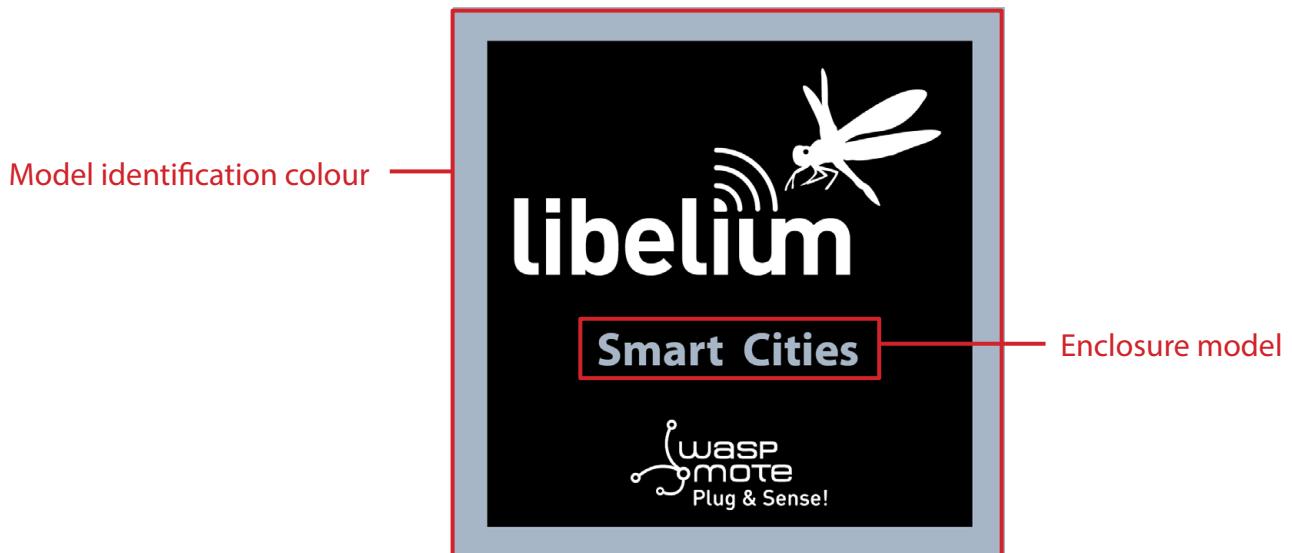


Figure : Front sticker of the enclosure

There are many configurations of Waspmote Plug & Sense! line, all of them identified by one unique sticker. Next image shows all possibilities.



Figure : Different front stickers

Moreover, WaspMote Plug & Sense! includes a back sticker where it is shown identification numbers, radio MAC addresses, etc. It is highly recommended to annotate this information and save it for future maintenance. Next figure shows it in detail.



Figure : Back sticker

Sensor probes are identified too by a sticker showing the measured parameter and the sensor manufacturer reference.



Figure : Sensor probe identification sticker

## 4.4. Sensor probes

Sensor probes can be easily attached by just screwing them into the bottom sockets. This allows you to add new sensing capabilities to existing networks just in minutes. In the same way, sensor probes may be easily replaced in order to ensure the lowest maintenance cost of the sensor network.



Figure : Connecting a sensor probe to WaspMote Plug & Sense!

Go to the [Plug & Sense! Sensor Guide](#) to know more about our sensor probes.

## 4.5. Solar powered

The battery can be recharged using the waterproof USB cable but also the internal or external solar panel options.

The external solar panel is mounted on a 45° holder which ensures the maximum performance of each outdoor installation.



Figure : WaspMote Plug & Sense! powered by an external solar panel

For the internal option, the solar panel is embedded on the front of the enclosure, perfect for use where space is a major challenge.



Figure : Internal solar panel



Figure : Wasp mote Plug & Sense! powered by an internal solar panel

## 4.6. Programming the Nodes

WaspMote Plug & Sense! can be reprogrammed in two ways:

The basic programming is done from the USB port. Just connect the USB to the specific external socket and then to the computer to upload the new firmware.



Figure : Programming a node

Besides, Libelium is developing a graphical and intuitive programming interface, the Code Generator (coming in 2017).

**› WaspMote - Plug & Sense! - Code Generator**

**\* Select Model**  
 Board model:

**\* Sleeping Time**  
 Time (seconds):

**\* Select sensor by socket**  

A:	Select	B:	Select	C:	Select
D:	Select	E:	Select	F:	Select

---

**Additional information**

Add Accelerometer 3 Axis data:  Add GPS coordinates:  WaspMote identifier (nodeID):  (Max 10 characters)

**Select Communication Module**

Communication Module: 
\* Mandatory fields

Figure: Code Generator web application

Over the Air Programming (OTAP) is also possible once the node has been installed (via WiFi or 4G radios). With this technique you can reprogram, wireless, one or more WaspMote sensor nodes at the same time by using a laptop and Meshlium.

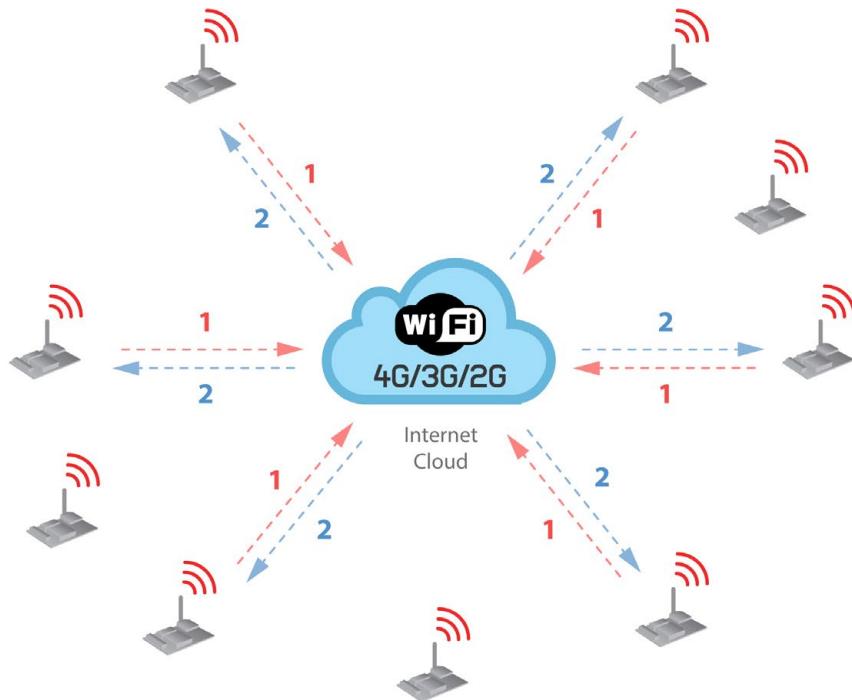


Figure : Typical OTAP process

## 4.7. Radio interfaces

Radio	Protocol	Frequency bands	Transmission power	Sensitivity	Range*	Certification
XBee-PRO 802.15.4 EU	802.15.4	2.4 GHz	10 dBm	-100 dBm	750 m	CE
XBee-PRO 802.15.4	802.15.4	2.4 GHz	18 dBm	-100 dBm	1600 m	FCC, IC, ANATEL, RCM
XBee 868LP	RF	868 MHz	14 dBm	-106 dBm	8.4 km	CE
XBee 900HP US	RF	900 MHz	24 dBm	-110 dBm	15.5 km	FCC, IC
XBee 900HP BR	RF	900 MHz	24 dBm	-110 dBm	15.5 km	ANATEL
XBee 900HP AU	RF	900 MHz	24 dBm	-110 dBm	15.5 km	RCM
WiFi	(HTTP(S), FTP, TCP, UDP)	2.4 GHz	17 dBm	-94 dBm	500 m	CE, FCC, IC, ANATEL, RCM
4G EU/BR	4G/3G/2G (HTTP, FTP, TCP, UDP) GPS	800, 850, 900, 1800, 2100, 2600 MHz	4G: class 3 (0.2 W, 23 dBm)	4G: -102 dBm	- km - Typical base station range	CE, ANATEL
4G US	4G/3G/2G (HTTP, FTP, TCP, UDP) GPS	700, 850, 1700, 1900 MHz	4G: class 3 (0.2 W, 23 dBm)	4G: -103 dBm	- km - Typical base station range	FCC, IC, PTCRB, AT&T
4G AU	4G (HTTP, FTP, TCP, UDP)	700, 1800, 2600 MHz	4G: class 3 (0.2 W, 23 dBm)	4G: -102 dBm	- km - Typical base station range	RCM
Sigfox EU	Sigfox	868 MHz	16 dBm	-126 dBm	- km - Typical base station range	CE
Sigfox US	Sigfox	900 MHz	24 dBm	-127 dBm	- km - Typical base station range	FCC, IC
LoRaWAN EU	LoRaWAN	868 MHz	14 dBm	-136 dBm	> 15 km	CE
LoRaWAN US	LoRaWAN	900 MHz	18.5 dBm	-136 dBm	> 15 km	FCC, IC

\* Line of sight and Fresnel zone clearance with 5dBi dipole antenna.

## 4.8. Models

There are some defined configurations of WaspMote Plug & Sense! depending on which sensors are going to be used. WaspMote Plug & Sense! configurations allow to connect up to six sensor probes at the same time.

Each model takes a different conditioning circuit to enable the sensor integration. For this reason each model allows to connect just its specific sensors.

This section describes each model configuration in detail, showing the sensors which can be used in each case and how to connect them to WaspMote. In many cases, the sensor sockets accept the connection of more than one sensor probe. See the compatibility table for each model configuration to choose the best probe combination for the application.

It is very important to remark that each socket is designed only for one specific sensor, so **they are not interchangeable**. Always be sure you connected probes in the right socket, otherwise they can be damaged.



Figure : Identification of sensor sockets

#### 4.8.1. Smart Environment

Smart Environment model is designed to monitor environmental parameters such as temperature, humidity, atmospheric pressure and some types of gases. The main applications for this WaspMote Plug & Sense! configuration are city pollution measurement, emissions from farms and hatcheries, control of chemical and industrial processes, forest fires, etc. Go to the application section in the [Libelium website](#) for a complete list of services.



Figure: Smart Environment WaspMote Plug & Sense! model

Sensor sockets are configured as shown in the figure below.

Sensor Socket	Sensor probes allowed for each sensor socket	
	Parameter	Reference
A	Carbon monoxide - CO	9229-P
	Ammonia – NH <sub>3</sub>	9233-P
B	Ozone – O <sub>3</sub>	9258-PB
	Hydrocarbons – VOC	9201-PB
C (digital bus)	Temperature, humidity and pressure	9370-P
	Luminosity (Luxes accuracy)	9325-P
	Ultrasound (distance measurement)	9246-P
D	Nitrogen Dioxide – NO <sub>2</sub>	9238-PB
E	Carbon Dioxide – CO <sub>2</sub>	9230-P
	Oxygen – O <sub>2</sub>	9231-P
F	Methane – CH <sub>4</sub>	9232-P
	Liquefied Petroleum Gases: H <sub>2</sub> , CH <sub>4</sub> , ethanol, isobutene	9234-P
	Air pollutants 1: C <sub>4</sub> H <sub>10</sub> , CH <sub>3</sub> CH <sub>2</sub> OH, CO, CH <sub>4</sub>	9235-P
	Air pollutants 2: C <sub>6</sub> H <sub>5</sub> CH <sub>3</sub> , H <sub>2</sub> S, CH <sub>3</sub> CH <sub>2</sub> OH, NH <sub>3</sub> , H <sub>2</sub>	9236-P
	Alcohol derivatives: CH <sub>3</sub> CH <sub>2</sub> OH, H <sub>2</sub> , C <sub>4</sub> H <sub>10</sub> , CO, CH <sub>4</sub>	9237-P

Figure: Sensor sockets configuration for Smart Environment model

**Note:** For more technical information about each sensor probe go to the [Development section](#) on the Libelium website.

## 5. Gases PRO Sensor Board (Calibrated)

This platform enables the user to implement pollution, air quality, industrial, environmental or farming projects with high requirements in terms of high accuracy, reliability and measurement range as the sensors come calibrated from factory. The Gases PRO line is available for both WaspMote and Plug & Sense!

So we offer to our clients two different sensor boards in order to measure gas levels: Gases Sensor Board (Gases) and Gases PRO Sensor Board (*Gases PRO*).

Comparative between Gases and Gases PRO sensor boards:

- Gases PRO integrates 17 different gas sensors. Gases integrates 12. See the complete list of parameters in the tables below.
- Gases PRO is useful to measure the exact value in ppm's of a gas concentration. The Gases line provides just the level of concentration low / medium / high. Complete accuracy ranges are shown in the table below.
- Gases PRO is meant to be used in applications that require accurate readings and exact concentrations thus its sensors are more expensive than the Gases regular version that just provides average levels.
- A Particle Matter Sensor -commonly known as "Dust Sensor"- has been integrated in the Gases PRO board. This sensor allows to measure PM1, PM2.5, and PM10 particles. This sensor comes also calibrated from factory.

### Should I choose Gases or Gases PRO?

Libelium created 2 different systems in terms of accuracy and pricing. The customer should consider Gases PRO if he needs maximum performance sensors for metering in accurate ppm or percentage. If the project just needs to detect gas presence or gas levels, the standard Gases Sensor Board can be enough.

The Gases PRO Sensor Board can read up to 17 gas sensors; it also has a high-end sensor for 3 parameters, temperature, humidity and pressure.

Gases PRO v3.0			
Parameter	Range	Calibration*	Max Consumption
Temperature	-40 to +85 °C	Calibrated ±1 °C (±0.5 °C at 25 °C)	2 µA @ 3V3
Humidity	0 to 100% HR	Calibrated ±3% RH (at 25 °C, range 20 ~ 80% RH)	2.8 µA @ 3V3
Pressure	30 to 110 kPa	Calibrated ±0.1 kPa (range 0 ~ 65 °C)	4.2 µA @ 3V3
Carbon Monoxide for high concentrations CO	0 to 500 ppm	Calibrated ±1 ppm	351 µA @ 3V3
Carbon Monoxide for low concentrations CO	0 to 25 ppm	Calibrated ±0.1 ppm	312 µA @ 3V3
Carbon Dioxide CO <sub>2</sub>	0 to 5000 ppm	Calibrated ±50 ppm (range 0~2500 ppm) ±200 ppm (range 2500~5000 ppm)	85 mA @ 3V3
Molecular Oxygen O <sub>2</sub>	0 to 30%	Calibrated ±0.1%	402 µA @ 3V3
Ozone O <sub>3</sub>	0 to 18 ppm	Calibrated ±0.2 ppm	< 1 mA @ 3V3
Nitric Oxide for high concentrations NO	0 to 250 ppm	Calibrated ±0.5 ppm	441 µA @ 3V3

Gases PRO v3.0			
Parameter	Range	Calibration*	Max Consumption
Nitric Dioxide NO <sub>2</sub>	0 to 20 ppm	Calibrated ±0.2 ppm	335 µA @ 3V3
Sulfur Dioxide SO <sub>2</sub>	0 to 20 ppm	Calibrated ±0.2 ppm	333 µA @ 3V3
Ammonia NH <sub>3</sub>	0 to 100 ppm	Calibrated ±0.5 ppm	338 µA @ 3V3
Methane and other combustible gases CH <sub>4</sub>	0 to 100% / LEL	Calibrated ±0.15% LEL	68 mA @ 3V3
Molecular Hydrogen H <sub>2</sub>	0 to 1000	Calibrated ±10 ppm	520 µA @ 3V3
Hydrogen Sulfide H <sub>2</sub> S	0 to 100 ppm	Calibrated ±0.1 ppm	352 µA @ 3V3
Hydrogen Chloride HCl	0 to 50 ppm	Calibrated ±1 ppm	341 µA @ 3V3
Hydrogen Cyanide HCN	0 to 50 ppm	Calibrated ±0.2 ppm	327 µA @ 3V3
Phosphine PH <sub>3</sub>	0 to 20 ppm	Calibrated ±0.1 ppm	361 µA @ 3V3
Ethylene Oxide ETO	0 to 100 ppm	Calibrated ±1 ppm	360 µA @ 3V3
Chlorine Cl <sub>2</sub>	0 to 50 ppm	Calibrated ±0.1 ppm	353 µA @ 3V3
Isobutane C <sub>4</sub> H <sub>10</sub>	-	-	-
Ethanol CH <sub>3</sub> CH <sub>2</sub> OH	-	-	-
Toluene C <sub>6</sub> H <sub>5</sub> CH <sub>3</sub>	-	-	-
Volatile Organic Compounds	-	-	-
Hydrocarbons	-	-	-
Particle Matter – Dust	0.5 to 16 µm (16 steps) (includes PM1, PM2.5 and PM10)	Calibrated	200+ mA @ 5 V

\* Accuracy values have been calculated at average conditions: 20°C / 101300 Pa. Accuracy levels may differ with different temperature and pressure levels, aging and in presence of third type of gases which cause cross sensitivity. See the "Calibration" chapter for more detail.

Gases v3.0			
Parameter	Range	Calibration	Consumption
Temperature	-40 to +85 °C	Calibrated ±1 °C (±0.5 °C at 25 °C)	2 µA @ 3V3
Humidity	0 to 100% HR	Calibrated ±3% RH (at 25 °C, range 20 ~ 80% RH)	2.8 µA @ 3V3
Pressure	30 to 110 kPa	Calibrated ±0.1 kPa (range 0 ~ 65 °C)	4.2 µA @ 3V3
Carbon Monoxide CO	30 to 1000 ppm	Not calibrated	3 mA @ 5V
Carbon Dioxide CO <sub>2</sub>	350 to 10000 ppm	Not calibrated	50 mA @ 5V
Molecular Oxygen O <sub>2</sub>	0 to 30%	Not calibrated	0 µA
Ozone O <sub>3</sub>	0.010 to 1 ppm	Not calibrated	34 mA
Nitric Oxide NO	-	-	-
Nitric Dioxide NO <sub>2</sub>	0.05 to 5 ppm	Not calibrated	26 mA
Sulfur Dioxide SO <sub>2</sub>	-	-	-
Ammonia NH <sub>3</sub>	10 to 100 ppm	Not calibrated	12 mA @ 5V
Methane and other combustible gases CH <sub>4</sub>	500 to 10000 ppm	Not calibrated	61 mA @ 5V
Molecular Hydrogen H <sub>2</sub>	500 to 10000 ppm	Not calibrated	61 mA @ 5V 46 mA @ 5V
Hydrogen Sulfide H <sub>2</sub> S	0.1 to 3 ppm	Not calibrated	61 mA @ 5V
Hydrogen Chloride HCl	-	-	-
Hydrogen Cyanide HCN	-	-	-
Phosphine PH <sub>3</sub>	-	-	-
Ethylene Oxide ETO	-	-	-
Chlorine Cl <sub>2</sub>	-	-	-
Isobutane C <sub>4</sub> H <sub>10</sub>	50 to 5000 ppm	Not calibrated	61 mA @ 5V
Ethanol CH <sub>3</sub> CH <sub>2</sub> OH	1 to 30 ppm 50 to 5000 ppm	Not calibrated	61 mA @ 5V
Toluene C <sub>6</sub> H <sub>5</sub> CH <sub>3</sub>	1 to 30 ppm	Not calibrated	61 mA @ 5V
Volatile Organic Compounds	30 to 400 ppm	Not calibrated	32 mA
Hydrocarbons	-	Not calibrated	61 mA @ 5V
Particle Matter – Dust	-	-	-

## How are the Gases PRO sensors calibrated?

All the Gases PRO sensors are linear, so they behave according to a simple line like  $y = f(x) = m \cdot x + c$ . On the other hand, the Gases Sensor Board sensors show a logarithmic response, which is harder to handle.

All the gas sensors for Gases PRO are calibrated by the sensor manufacturers, in their laboratories. A two-point calibration process is performed to get maximum accuracy, with controlled concentrations of gas in vacuum chambers. Due to the linear nature of the sensors, those 2 points are enough to correct the 2 possible drifts in  $m$  (slope) and  $c$  (constant offset). Our sensor manufacturers provide a calibration certificate for each individual sensor, with two calibration parameters. Libelium stores these parameters in an EEPROM, non-volatile memory chip inside each gas sensor. When the user performs a sensor reading, the software system designed by Libelium reads the sensor signal and the 2 parameters. All these 3 values are processed by the system. After this, a **temperature compensation calculation** is applied too. From this data we obtain the concentration value in ppm's (which is the final value we want to measure).

The Temperature, Humidity and Pressure sensor for Gases PRO is calibrated in factory too. There are 3 calibration points for temperature, 6 points for humidity and 9 points for pressure. These 3 parameters are automatically inter-compensated and this way the user gets extreme accuracy.

## 6. Hardware

### 6.1. General Description

The Waspmote Gases Sensor Board has been designed to monitor environmental parameters such as temperature, humidity, atmospheric pressure and 14 different types of gases. It allows the inclusion of 7 gases sensors at the same time, the regulation of their power through a system of solid state switches and the amplification of the output signal of each one of them through a non-inverting amplification stage controlled by a digital potentiometer configurable through the Inter-Integrated Circuit Bus, I2C.

The gases which can be monitored are:

- Carbon Monoxide – CO
- Carbon Dioxide – CO<sub>2</sub>
- Molecular Oxygen – O<sub>2</sub>
- Methane – CH<sub>4</sub>
- Molecular Hydrogen – H<sub>2</sub>
- Ammonia – NH<sub>3</sub>
- Isobutane – C<sub>4</sub>H<sub>10</sub>
- Ethanol – CH<sub>3</sub>CH<sub>2</sub>OH
- Toluene – C<sub>6</sub>H<sub>5</sub>CH<sub>3</sub>
- Hydrogen Sulphide – H<sub>2</sub>S
- Nitrogen Dioxide – NO<sub>2</sub>
- Ozone – O<sub>3</sub>
- Volatile Organic Compounds (VOC's)
- Hydrocarbons

### 6.2. Specifications

**Weight:** 20 gr

**Dimensions:** 73.5 x 51 x 1.3 mm

**Temperature Range:** [-20 °C, 65 °C]

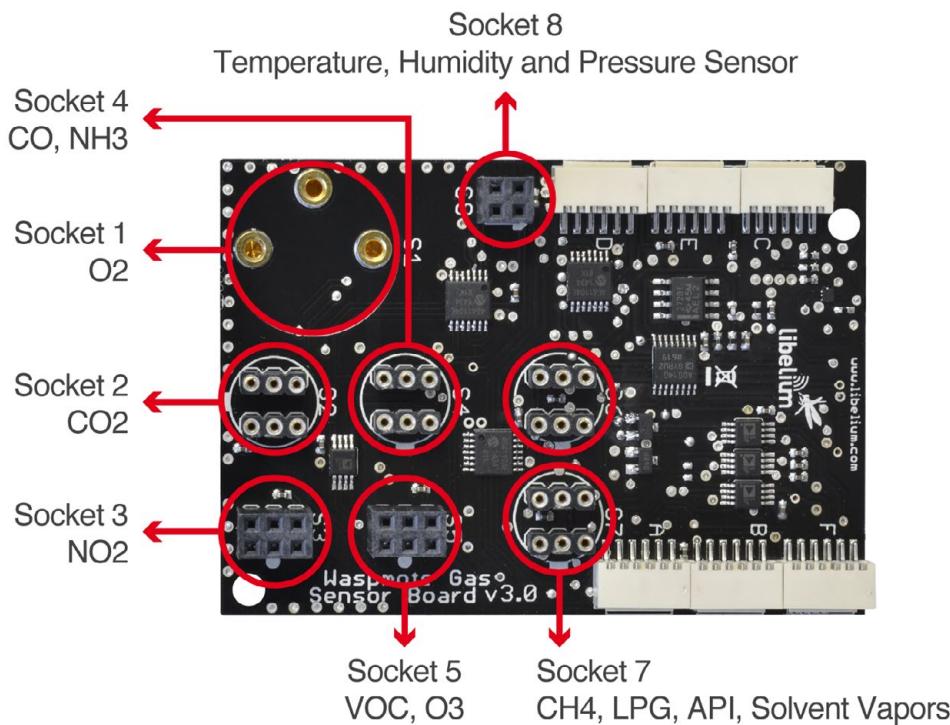


Figure : Upper side

The sensor are distributed in the next way by sockets:

**Socket1:**

- Molecular Oxygen – O<sub>2</sub>

**Socket2:**

- Carbon Dioxide – CO<sub>2</sub>

**Socket3:**

- Nitrogen Dioxide – NO<sub>2</sub>

**Socket4:**

- Ammonia – NH<sub>3</sub>
- Carbon Monoxide – CO

**Socket5:**

- Volatile Organic Compounds (VOC's)
- Ozone – O<sub>3</sub>

**Socket6 and Socket7:**

- Methane – CH<sub>4</sub>
- Isobutane – C<sub>4</sub>H<sub>10</sub>
- Ethanol – CH<sub>3</sub>CH<sub>2</sub>OH
- Toluene – C<sub>6</sub>H<sub>5</sub>CH<sub>3</sub>
- Hydrogen Sulphide – H<sub>2</sub>S
- Hydrocarbons

**Socket8:**

- Temperature, Humidity and Pressure sensor

## 6.3. Electrical Characteristics

**Board power voltage:** 3.3 V and 5 V

**Sensor power voltage:** 5 V

**Maximum admitted current (continuous):** 200 mA

**Maximum admitted current (peak):** 400 mA

## 7. Sensors

### 7.1. General considerations in the use of the sensors

A very similar structure has been installed for all the connectors for the gas sensors: a load resistance at the output of each sensor, except in sockets 1 and 2 where it is not necessary, combined with an amplification stage of maximum gain 101. All sockets share the same amplification stage and the same output to the microprocessor. For more details on the connectors, see the "Design and connections" section in this manual.

The choice of amplification stage gain and of the sensor's load resistance can be carried out according to two parameters: the specific sensor available, since there may be significant variations between two different sensors of the same model, and the value and range of concentrations of gas to be monitored.

**Important:** when selecting load resistance and amplification it must be remembered that, although the sensors must be powered by a voltage of 5 V to function appropriately, the WaspMote ADC allows input between 0 and 2.56 V, so it will be necessary to calculate the resistance, load and gain values to adapt the measurement range of the sensor to the WaspMote input.

The amplification stage gain and load resistance of a connector can be configured through a simple group of commands available in the library, created to facilitate handling of the board from the WaspMote mote. For more information on the library instructions and steps to follow for the configuration of the sensors, consult the "Library" section of this manual.

The accuracy which can be obtained in the sensor's output value will be dependent on the way in which it is supplied. This way, the longer the power time or duty cycle, as appropriate, the better accuracy will be obtained. The disadvantage of prolonged power is an increase in the mote's consumption, with the consequent decrease of the battery's life, so adjusting the power of each sensor to the requirements of the specific application being developed is recommended to optimize the equipment's performance.

The new version of the library implements new functions for reading the resistance and the concentration of the gas sensor. For more information on the library instructions and steps to follow for the configuration of the sensors, consult the "Library" section of this manual.

When a sensor remains without power for a prolonged period it is possible that it shows an unstable output. This stability is regained after spending time switched on or after many consecutive cycles of power supply.

Sensitivity of the sensor may vary when the device is subjected to large variations in temperature or humidity, for example in outdoor conditions. To compensate for these variations, use the tables and graphs used in the sensors' data sheets.

## 7.2. Starting with the gas sensors

In this section we are going to explain the first steps to start with the sensors used in the Gases Sensor Board, including the most common doubts, such as what parameters configure for the adaptation stage, which calibration process must be followed or how to transform the read voltage or resistance value into a concentration.

Take into account that developing a robust application for gases detection or measurement may take an important effort of testing and knowing the sensors.

### 7.2.1. Sensor calibration

When dealing with most of the sensors used in the Gases Sensor Board it is highly recommended to calibrate the sensor in order to get an accurate value, since, as explained in section "General considerations in the use of the sensors", its normal resistance and sensitivity may vary from one unit to another in a wide range. This calibration may not be necessary in all applications, for example if the sensor is going to be used in a gas detection, where monitoring the variation of the sensor output may be enough to have it working properly, and the normal operation conditions may be replicated without a specific equipment.

The calibration procedure requires the capture of the sensor response under different concentrations of gas in the target operation range (that should be comprised in the operation range of the sensor), and, depending on the conditions of the application to be implemented, under controlled temperature and humidity. The larger the number of calibration points in that range the more accurate the calibration will be, a logarithmic approximation to the response of the sensor for the intermediate values should be applied (in one or two axes, depending on the sensors, as shown in their respective response graphs). Take into account that gas sensors are sensitive, in different degree, to several target gases, which may affect the measurement both in the field and in the laboratory. The recreation of these conditions may require of specific equipment, so it may be necessary the help of a specialized laboratory.

### 7.2.2. Gain and load resistor configuration

The gain and load resistor configured for a given stage will depend on two main aspects: first of all, the sensor connected to the stage. Since in most cases there is a high variability in the parameters that rule the behavior of the sensor (sensor's initial resistance and sensitivity, taking this as the variability of the sensor resistance with the gas concentration), every single unit may show important differences in operation, so it will be necessary a specific configuration for each one. Secondly, the application conditions under which the sensor is supposed to be operating will also have to be taken into account, in order to maintain the output voltage of the adaptation stage in the voltage range of the input of the mote's microcontroller (between 0 and 2.56 V). It will also be recommended to set both parameters to have an output voltage of around 1.28 V (the middle of the voltage input range) when the gas concentration is in the middle of the desired operation range.

The most adequate way of configuring the adaptation stage is by determining the load resistance and gain necessary from the calibration data and the application requirements and test it, adjusting the definitive values with the results obtained. In other cases, in which these parameters cannot be that clearly defined owing to the lack of calibration or to the indeterminacy of the application conditions, for example when handling the sensors for the first time, you can follow the advice provided below:

As a general rule, gain will be fixed at 1 in almost every application, only in very specific situations, such as operation in the limits of the sensor range, it will be necessary a different value. There are two exceptions to this rule, that are sensors for CO<sub>2</sub> (TGS4161) and O<sub>2</sub> (SK-25) which output a voltage that has to be amplified to adjust it to the WaspMote's analog-to-digital converter. In the case of the O<sub>2</sub> sensor the extremely low voltage advises a gain of 100 in some cases, whilst for the CO<sub>2</sub> sensor, with a higher (and more variable) output, a gain between 7 and 10 will be adequate.

In the case of the load resistor it is not that simple, since, as said before, it will be different from one sensor to another. If calibration has not been performed it will be necessary a wide testing based on trial and error to achieve the correct resistance value. It is recommended in this case to start with the lowest value indicated as initial sensor resistance in the documentation, and adjust it to get the desired output voltage (usually 1.28 V, the middle of the analog-to-digital converter range, if it is not specified by another restriction). When carrying out this adjustment beware not to configure a resistance below the minimum load resistance indicated in the corresponding section of the documentation, since it could damage the sensor.

As indicated in section "General considerations in the use of the sensors", beware when operating with sensors that have not been used for a long time, since they may require of some time to get to work properly. Applying a "burning" process, having them on with a continuous reading for at least about 12 hours, is highly recommended before starting using the sensors.

### 7.2.3. Converting the read data

The gas concentration value in ppm or ppb, depending on the sensor, can be obtained from the ratio between the read resistance of the sensor (that can be obtained using function `readConcentration()`, described in section "Library") and the initial resistance of the sensor, except in the case of the O<sub>2</sub> (SK-25) and CO<sub>2</sub> (TGS4161), which as said before output a voltage proportional to the gas concentration.

Again, the most accurate response will be obtained if sensors are calibrated previously to their introduction in the definitive application. Constructing a conversion function or graph out of the calibration results and translating the read values into it will lead to the best performance.

In case the calibration process cannot be performed, the concentration may be estimated using the provided output graphs, which correspond to the typical response of the sensors. Even though, it will be necessary to calculate the initial resistance of the sensor, whose range is given in the specifications section of the sensor. In some cases it may be determined from a normal environment measurement, since information may be obtained from other sources such as detectors or studies about the gas presence in the operation environment. Anyway, take into account that the data obtained following this method will be highly inaccurate.

Finally, it is necessary to keep in mind that in the case of some sensors and applications the extraction of the concentration value may not be that straightforward, since other parameters may interfere in the behavior of the sensor. In some sensors other gases different from the target can cause a significant change in the resistance value, so it will be necessary a high knowledge of the application conditions to be able to determine precisely the gas concentration. The combination of several sensors is a suitable option to avoid these situations. Also, in applications in which high variations of temperature and humidity are present it may be required to perform a compensation. For this, a temperature and a humidity sensor have been integrated in the Gases Sensor boards, so all the necessary information can be obtained. Please refer to the sensors data sheet and application notes for more information about the temperature and humidity compensation of the sensors.

### 7.2.4. Calibrated sensors performance

In optimal working conditions the accuracy of the measurement has been calculated to be of the ±5% of the measurement, including the error owed to the electronics and the calibration. However, when operating with the calibrated sensors there are several additional issues regarding the accuracy of the measurement that will have to be taken into account when developing a new application:

First of all, as mentioned before, the accuracy will be highly dependent on the heating time applied to the sensor. For proper operation sensors must be powered until they reach their operating temperature, which may take several minutes depending on the sensor, the environmental temperature and the duty cycle of the application. If a completely stable measurement has not been achieved an additional error to that of the sensor and the electronics itself will be added.

Secondly, the environmental temperature will also affect the behavior of the sensor, being necessary a temperature correction to achieve the maximum accuracy. This correction should be performed from the temperature dependence graphs and tables provided by the manufacturer of the sensor in its data sheet or application note.

The last problem encountered with these sensors (which generally affects all chemical sensors) is recalibration. **After some operation time, the wear of the sensing layer owing to different effects will lower the accuracy of the sensor, so it will have to be replaced with a new calibrated probe.** The more accuracy required or the more harass environment sensors will operate in, the more frequent replacement will be required, but **as a general rule probes should be always replaced every six months.**

## 8. Sensors

### 8.1. Temperature, Humidity and Pressure Sensor

The BME280 is a digital temperature, humidity and pressure sensor developed

#### 8.1.1. Specifications

##### Electrical characteristics

Supply voltage: 3.3 V

Sleep current typical: 0.1  $\mu$ A

Sleep current maximum: 0.3  $\mu$ A



##### Temperature sensor

Operational range: -40 ~ +85 °C

Full accuracy range: 0 ~ +65 °C

Accuracy:  $\pm 1$  °C (range 0 °C ~ +65 °C)

Response time: 1.65 seconds (63% response from +30 to +125 °C).

Typical consumption: 1  $\mu$ A measuring

*Figure : Temperature, humidity and pressure sensor*

##### Humidity sensor

Measurement range: 0 ~ 100% of relative humidity (for temperatures < 0 °C and > 60 °C see figure below)

Accuracy: <  $\pm 3$ % RH (at 25 °C, range 20 ~ 80%)

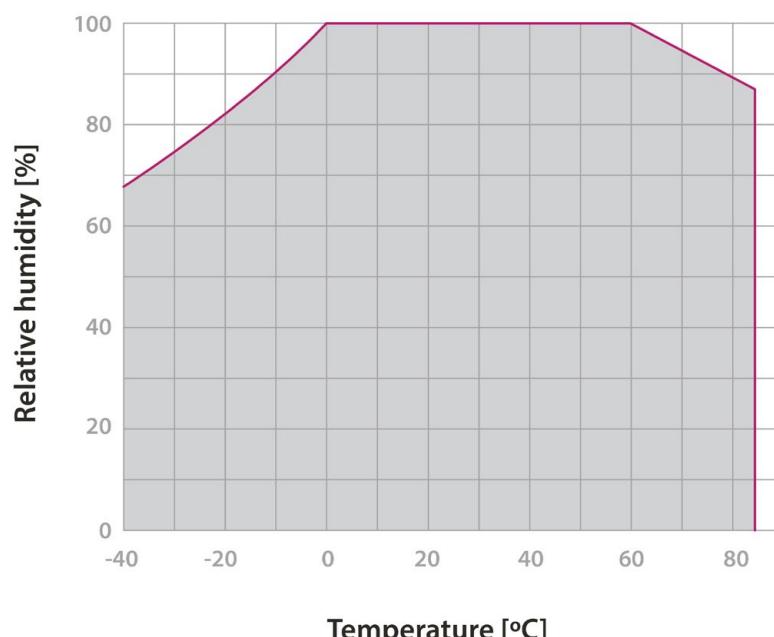
Hysteresis:  $\pm 1$ % RH

Operating temperature: -40 ~ +85 °C

Response time (63% of step 90% to 0% or 0% to 90%): 1 second

Typical consumption: 1.8  $\mu$ A measuring

Maximum consumption: 2.8  $\mu$ A measuring



*Figure : Humidity sensor operating range*

### Pressure sensor

Measurement range: 30 ~ 110 kPa  
Operational temperature range: -40 ~ +85 °C  
Full accuracy temperature range: 0 ~ +65 °C  
Absolute accuracy: ±0.1 kPa (0 ~ 65 °C)  
Typical consumption: 2.8 µA measuring  
Maximum consumption: 4.2 µA measuring

### 8.1.2. Measurement process

The BME280 is a combined digital humidity, pressure and temperature sensor based on proven sensing principles. The humidity sensor provides an extremely fast response time for fast context awareness applications and high overall accuracy over a wide temperature range.

The pressure sensor is an absolute barometric pressure sensor with extremely high accuracy and resolution and drastically lower noise.

The integrated temperature sensor has been optimized for lowest noise and highest resolution.

Its output is used for temperature compensation of the pressure and humidity sensors and can also be used for estimation of the ambient temperature.

The sensor provides both SPI and I2C interfaces and can be supplied using 1.71 to 3.6 V for the sensor supply VDD and 1.2 to 3.6 V for the interface supply VDDIO. Measurements can be triggered by the host or performed in regular intervals. When the sensor is disabled, current consumption drops to 0.1 µA. To read this sensor, you should use the following functions:

Reading code:

```
{  
    float temperature; // Stores the temperature in °C  
    float humidity; // Stores the relative humidity in %RH  
    float pressure; // Stores the pressure in Pa  
  
    // Switch ON and configure the Gases Board  
    Gases.ON();  
  
    // Read environmental variables  
    temperature = Gases.getTemperature();  
    humidity = Gases.getHumidity();  
    pressure = Gases.getPressure();  
}
```

You can find a complete example code for reading the BME280 sensor in the Gases Sensor Board, and connected to the socket8 in the following link:

<http://www.libelium.com/development/wasp mote/examples/ga-v30-01-bme280-sensor-reading>

## 8.2. Carbon Monoxide (CO) Sensor – TGS2442

### 8.2.1. Specifications

**Gases:** CO

**Measurement range:** 30 ~ 1000 ppm

**Resistance at 100 ppm:** 13.3 ~ 133 kΩ

**Sensitivity:** 0.13 ~ 0.31 (ratio between the resistance at 300 ppm and at 100 ppm)

**Supply voltage:** 5V ±0.2 V DC

**Operating temperature:** -10 ~ +50 °C

**Response time:** 1 second

**Minimum load resistance:** 10 kΩ

**Average consumption:** 3 mA (throughout the complete power supply cycle in one second)



Figure : Image of the TGS2442 sensor

### 8.2.2. Measurement Process

The CO sensor TGS2442 is a resistive sensor sensitive to the changes in concentration of Carbon Monoxide (CO) and, very slightly, Hydrogen (H<sub>2</sub>), which may be placed in the board's socket 4, and must be connected in the way indicated in the images below. The sensor's resistance would vary according to the graph in the figure below these paragraphs, which may present significant variations between two different sensors, so it is recommended to consult the sensor's documentation to choose the load resistance and amplification gain and calibrate it before finally inserting it into the application.

Reading this sensor requires a cycle of one second throughout which two power supply pulses are generated on heat resistance and sensor resistance of 14 ms and 5 ms each (average consumption throughout the power supply cycle is 3 mA). The execution of this cycle and the reading of the sensor can be done automatically using the functions of the library.

```
// CO Sensor must be connected physically in SOCKET4
COSensorClass COSensor;

// Switch ON and configure the Gases Board
Gases.ON();
// Switch ON the sensor socket
COSensor.ON();

// Voltage value of the sensor
float coVol = COSensor.readVoltage();
// Resistance of the sensor
float coRes = COSensor.readResistance(coVol);
// PPM value of CO
float coValPPM = COSensor.readConcentration(coRes);
}
```

You can find a complete example code for reading the CO sensor TGS2442 sensor placed on socket 4 in the following link:

<http://www.libelium.com/development/wasp mote/examples/ga-v30-05-co-sensor-reading>

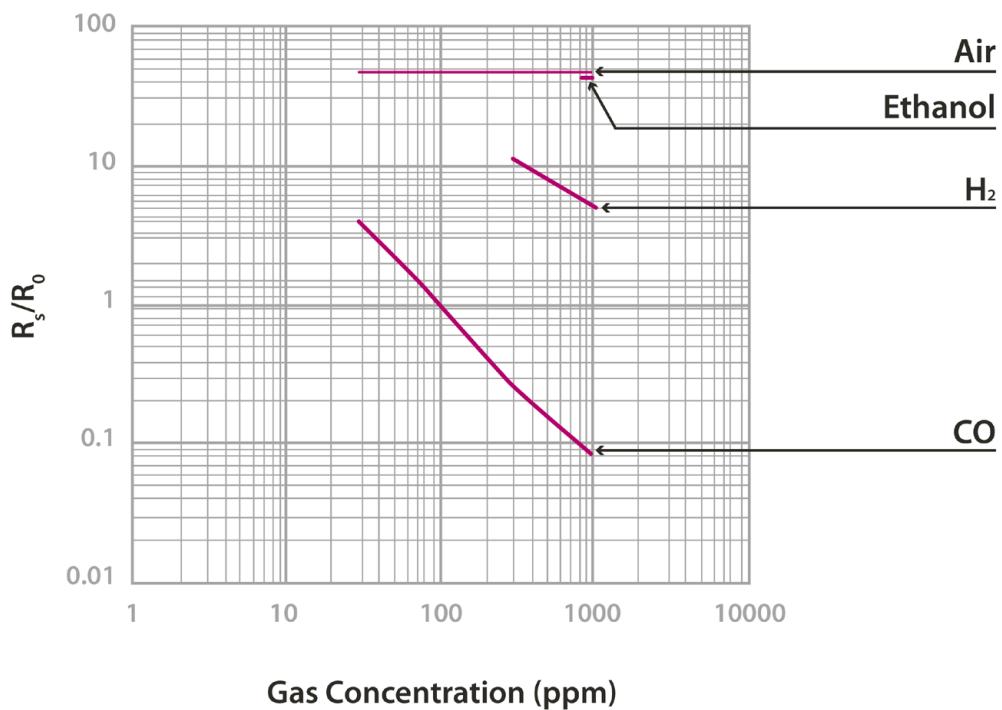


Figure : Graph of the sensitivity of the TGS2442 taken from the Figaro sensor's data sheet

## 8.3. Carbon Dioxide ( $CO_2$ ) Sensor – TGS4161

### 8.3.1. Specifications

**Gases:**  $CO_2$

**Measurement range:** 350 ~ 10000 ppm

**Voltage at 350 ppm:** 220 ~ 490 mV

**Sensitivity:** 44 ~ 72 mV (variation between the voltage at 350 ppm and at 3500 ppm)

**Supply voltage:** 5 V  $\pm 0.2$  V DC

**Operating temperature:** -10 ~ +50 °C

**Response time:** 1.5 minutes

**Average consumption:** 50 mA



Figure : Image of the TGS4161 sensor

### 8.3.2. Measurement Process

The TGS4161 sensor provides a voltage output proportional to the CO<sub>2</sub> concentration in the atmosphere. It shows a value between 220 and 490 mV for a concentration of 350 ppm (approximately the normal CO<sub>2</sub> concentration in the air) decreasing as the amount of gas increases. Different sensors may show a large variability in the initial voltage values at 350 ppm and sensitivity, so it is recommended to calibrate each sensor before including it in the application.

The accuracy that this sensor can offer will vary depending on the time it has remained powered before being measured. A time of 30 seconds is sufficient to detect significant changes in concentration, while a high accuracy measurement will require at least 10 minutes of power.

This sensor must be placed only in socket 2 as indicated in the figure below. To access the sensor's output value it is enough to execute library command which captures the sensor's analog value.

Reading code:

```
{
    // Switch ON and configure the Gases Board
    Gases.ON();
    // Switch ON the CO2 Sensor SOCKET2
    CO2Sensor.ON();

    // PPM value of CO2
    float co2Vol = CO2Sensor.readVoltage();
    float co2ValPPM = CO2Sensor.readConcentration();
}
```

You can find a complete example code for reading the TGS4161 sensor in the following link:

<http://www.libelium.com/development/wasp mote/examples/ga-v30-03-co2-sensor-reading>

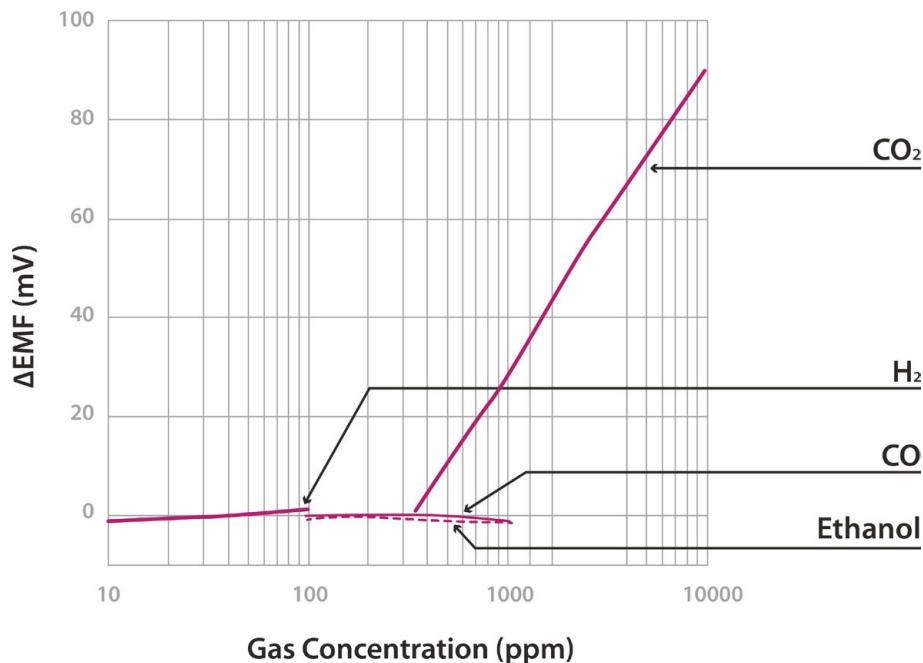


Figure : Graph of sensitivity of the TGS4161 sensor taken from the Figaro sensor's data sheet

## 8.4. Molecular Oxygen ( $O_2$ ) Sensor – SK-25

### 8.4.1. Specifications

**Gases:**  $O_2$

**Measurement range:** 0 ~ 30 %

**Output range:** Approximately 0 ~ 10 mV

**Initial voltage:** 5.5 ~ 8.8 mV

**Operating temperature:** 5 ~ +40 °C

**Response time:** 15 seconds

**Consumption:** 0  $\mu$ A



Figure : Image of the SK-25 sensor

### 8.4.2. Measurement Process

The SK-25 is an analog sensor which provides a voltage output proportional to the  $O_2$  concentration in the atmosphere, without needing power and therefore with zero consumption. It shows an output range between 0 and 10 mV, with voltage in standard conditions (approximately 21 %  $O_2$  concentration) of between 5.5 and 8.8 mV. The output response can vary from one sensor to another, so it is recommended to calibrate the sensor before finally inserting it into the application.

The function `readConcentration()` returns the value of the oxygen in % using the calibration values configured in the function `setCalibrationPoints()`. The corresponding example shows how to read and calibrate the sensor in an easy way:

Reading code:

```
{
    // O2 Sensor must be connected in SOCKET1
    O2SensorClass O2Sensor;

    // Switch ON and configure the Gases Board
    GasesBoard.ON();
    // O2 Sensor does not need power supply
    float o2Vol = O2Sensor.readVoltage();
    // Read the concentration value in %
    float o2Val = O2Sensor.readConcentration();
}
```

You can find a complete example code for reading the SK25 sensor in the following link:

<http://www.libelium.com/development/wasp mote/examples/ga-v30-02-o2-sensor-reading>

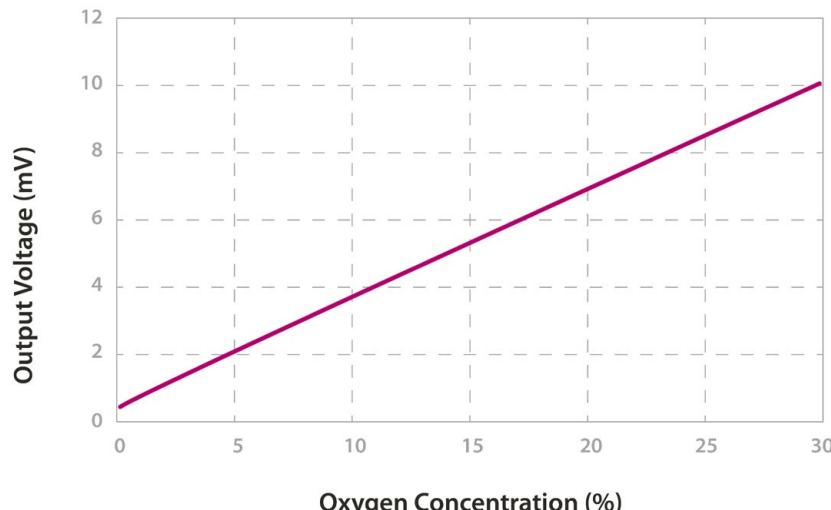


Figure : Graph of the sensitivity of the SK-25 extracted from the Figaro sensor's data sheet

## 8.5. Nitrogen Dioxide (NO<sub>2</sub>) Sensor - MiCS-2714

### 8.5.1. Specifications

This sensor is a version for the MiCS-2710 sensor.

**Gases:** NO<sub>2</sub>

**Measurement range:** 0.05 ~ 5 ppm

**Air resistance:** 0.8 ~ 8k Ω (typically 2.2 kΩ)

**Sensitivity:** 6 ~ 100 (typically 55, ratio between the resistance at 0.25 ppm and in air)

**Supply voltage:** 1.7 ~ 2.5 V DC

**Operating temperature:** -30 ~ +85 °C

**Response time:** 30 seconds

**Average consumption:** 26 mA (throughout the complete power supply cycle in one second)



Figure : Image of the MiCS-2714 sensor

### 8.5.2. Measurement process

The MiCS-2714 is a sensor whose resistance varies in the presence of small concentrations of NO<sub>2</sub>. This value varies between 2 kΩ and 2 MΩ approximately, providing high accuracy throughout the output range. Unlike the rest of the board's gas sensors, which operate at a voltage of 5 V, this sensor is powered through a 1.8 V voltage regulator, with consumption of approximately 26 mA. The sensor's resistance in air, as well as its sensitivity, can vary between different units, so it is recommended to calibrate each one of them before finally inserting them in the application.

This sensor must be connected in socket 3 of the board, and its output can be read through the capture commands of the library.

Reading code:

```
{
    // NO2 Sensor must be connected physically in SOCKET3
    NO2SensorClass NO2Sensor;
    // Switch ON and configure the Gases Board
    Gases.ON();
    // Switch ON the sensor socket
    NO2Sensor.ON();

    // PPM value of NH3
    float NO2Vol = NO2Sensor.readVoltage();
    float NO2Res = NO2Sensor.readResistance(NO2Vol);
    float NO2PPM = NO2Sensor.readConcentration(NO2Res);
}
```

You can find a complete example code for reading the MiCS-2714 sensor in the following link:

<http://www.libelium.com/development/wasp mote/examples/ga-v30-09-no2-sensor-reading>

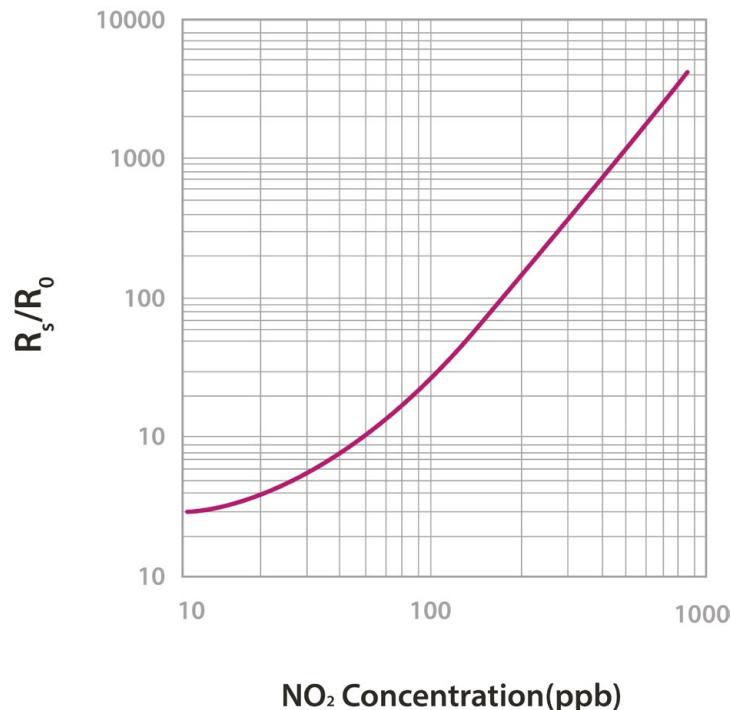


Figure : Graph of the sensitivity of the MiCS-2714 taken from the e2v's sensor data sheet.

## 8.6. Ammonia (NH<sub>3</sub>) sensor – TGS2444

### 8.6.1. Specifications

**Gases:** NH<sub>3</sub>, H<sub>2</sub>S

**Measurement range:** 10 ~ 100 ppm

**Resistance at 10 ppm:** 3.63 ~ 36.3 kΩ

**Sensitivity:** 0.063 ~ 0.63 (ratio between the resistance at 3000 and at 1000 ppm)

**Supply voltage:** 5 V ±0.2 V DC

**Operating temperature:** -10 ~ +50 °C

**Response time:** 250 ms

**Minimum load resistance:** 8 kΩ

**Average consumption:** 12 mA (throughout the complete power supply cycle in 250 ms)



Figure : Image of the TGS2444 sensor

### 8.6.2. Measurement Process

The TGS2444 sensor is a resistive sensor which is highly sensitive to variations in the concentration of Ammonia (NH<sub>3</sub>) and which shows slight sensitivity to hydrogen sulphide (H<sub>2</sub>S) and to a lesser extent, to Hydrogen (H<sub>2</sub>) and Ethanol (CH<sub>3</sub>CH<sub>2</sub>OH). Both the sensor's initial resistance (at 10 ppm) and its sensitivity vary widely between different sensors of the same model, so it is recommended to calibrate each one of them independently before finally including them in the application.

This sensor can be placed on socket 4 following the direction indicated in the silkscreen of the PCB. To read it, the necessary instructions are in the library:

Reading code:

```
{
    // NH3 Sensor must be connected physically in SOCKET4
    NH3SensorClass NH3Sensor;

    // Switch ON and configure the Gases Board
    Gases.ON();
    // Switch ON the sensor socket
    NH3Sensor.ON();

    // PPM value of NH3
    float NH3Vol = NH3Sensor.readVoltage();
    float NH3Res = NH3Sensor.readResistance(NH3Vol);
    float NH3PPM = NH3Sensor.readConcentration(NH3Res);
}
}
```

You can find a complete example code for reading the TGS2444 sensor placed on socket 4 in the following link:

<http://www.libelium.com/development/wasp mote/examples/ga-v30-08-nh3-sensor-reading>

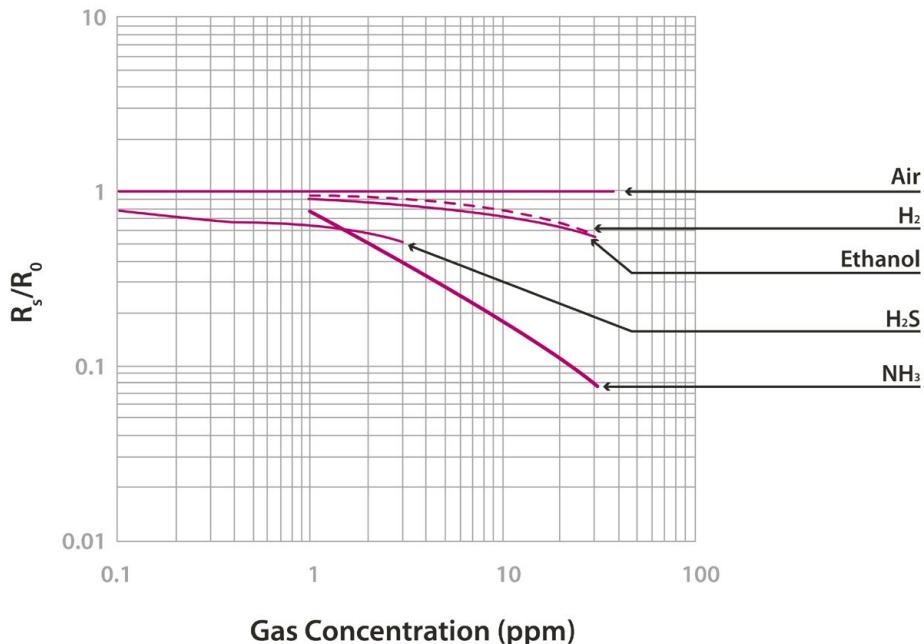


Figure : Graph of the sensitivity of the TGS2444 taken from the Figaro sensor data sheet

## 8.7. Methane ( $\text{CH}_4$ ) sensor – TGS2611

### 8.7.1. Specifications

**Gases:**  $\text{CH}_4, \text{H}_2$

**Measurement range:** 500 ~ 10000 ppm

**Resistance at 5000 ppm:** 0.68 ~ 6.8 k $\Omega$

**Sensitivity:**  $0.6 \pm 0.06$  (ratio between the resistance at 9000 and at 3000 ppm)

**Supply voltage:** 5 V  $\pm 0.2$  V DC

**Operating temperature:** -10 ~ +40 °C

**Response time:** 30 seconds

**Minimum load resistance:** 0.45 k $\Omega$

**Average consumption:** 61 mA



Figure : Image of the TGS2611 sensor

### 8.7.2. Measurement Process

The TGS2611 sensor shows a variable resistance with the concentration of  $\text{CH}_4$  and to a lesser extent with the concentration of  $\text{H}_2$ . The sensor's initial resistance (for 5000 ppm) and its sensitivity may show large variations between different sensors of the same model, so it is recommended to consult the manufacturer's documentation and calibrate it before finally inserting it in the application.

This sensor can be inserted in sockets 6 and 7, and must be placed following the direction indicated in the silkscreen of the PCB. To read it, the necessary instructions are in the library:

Reading code:

```
{
    // CH4 Sensor can be connected in SOCKET6 AND SOCKET7
    CH4SensorClass CH4Sensor(SOCKET_6); // ----- SOCKET6 Class used

    // Switch ON and configure the Gases Board
    Gases.ON();
    // Switch ON the sensor socket
    CH4Sensor.ON();

    // Voltage value of the sensor
    float ch4Vol = CH4Sensor.readVoltage();
    // Resistance of the sensor
    float ch4Res = CH4Sensor.readResistance(ch4Vol);
    // PPM value of CH4
    float ch4ValPPM = CH4Sensor.readConcentration(ch4ValPPM);
}
```

You can find a complete example code for reading the TGS2444 sensor placed on socket 6 in the following link:

<http://www.libelium.com/development/wasp mote/examples/ga-v30-07-ch4-sensor-reading>

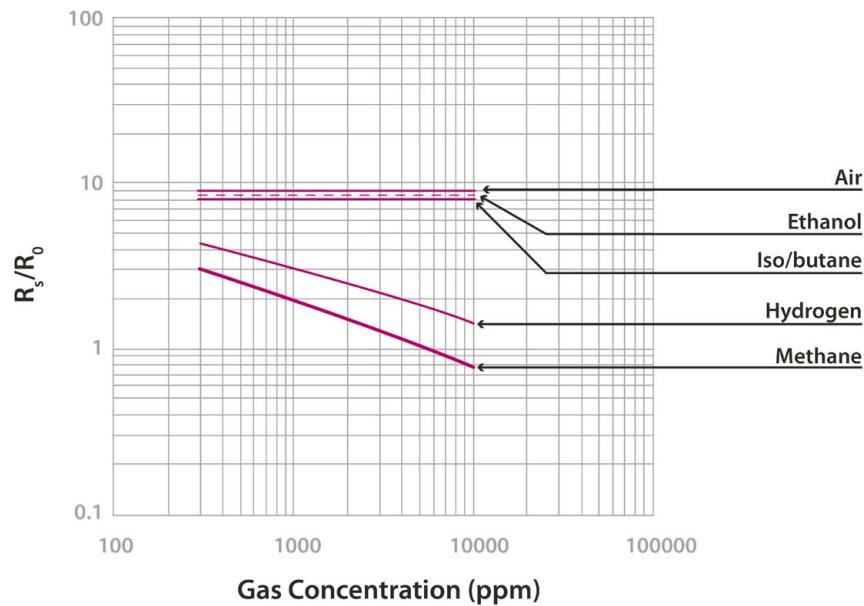


Figure : Graph of sensitivity of the TGS2611 taken from the Figaro sensor's data sheet

## 8.8. Liquefied Petroleum Gas Sensor - TGS2610

### 8.8.1. Specifications

**Gases:**  $\text{CH}_3\text{CH}_2\text{OH}$ ,  $\text{CH}_4$ ,  $\text{C}_4\text{H}_{10}$ ,  $\text{H}_2$

**Measurement range:** 500 ~ 10000 ppm

**Resistance at 1800 ppm (isobutane)** 0.68 ~ 6.8 k $\Omega$

**Sensitivity:**  $0.56 \pm 0.06$  (ratio between the resistance at 3000 and at 1000 ppm)

**Supply voltage:** 5 V  $\pm 0.2$  V DC

**Operating temperature:** -10 ~ +40 °C

**Response time:** 30 seconds

**Minimum load resistance:** 0.45 k $\Omega$

**Average consumption:** 61 mA



Figure : Image of the TGS2610 sensor

### 8.8.2. Measurement Process

The TGS2610 is a resistive sensor which shows sensitivity to combustible gases and derivatives. Especially reactive to Isobutane ( $\text{C}_4\text{H}_{10}$ ), it is also sensitive to Methane ( $\text{CH}_4$ ), Ethanol ( $\text{CH}_3\text{CH}_2\text{OH}$ ) and Hydrogen ( $\text{H}_2$ ). Because both its resistance and sensitivity show significant variations between different sensors of the same model, it is recommended to consult the manufacturer's documentation and carry out a process of calibration prior to its final inclusion in an application.

This sensor can be inserted in sockets 6 and 7, and must be placed following the direction indicated in the silkscreen of the PCB. To read it, the necessary instructions are in the library:

Reading code:

```
{
    // LPS Sensor can be connected in SOCKET6 AND SOCKET7
    LPSSensorClass LPSSensor(SOCKET_7); // ----- SOCKET7 Class used

    // Switch ON and configure the Gases Board
    Gases.ON();
    // Switch ON the sensor socket
    LPSSensor.ON();

    // Voltage value of the sensor
    float lpsVol = LPSSensor.readVoltage();
    // Resistance of the sensor
    float lpsRes = LPSSensor.readResistance(lpsVol);
    // PPM value of LPS
    float lpsValPPM = LPSSensor.readConcentration(lpsRes);
}
```

You can find a complete example code for reading the TGS2610 sensor placed on socket 7 in the following link:

<http://www.libelium.com/development/wasp mote/examples/ga-v30-11-lpg-sensor-reading>

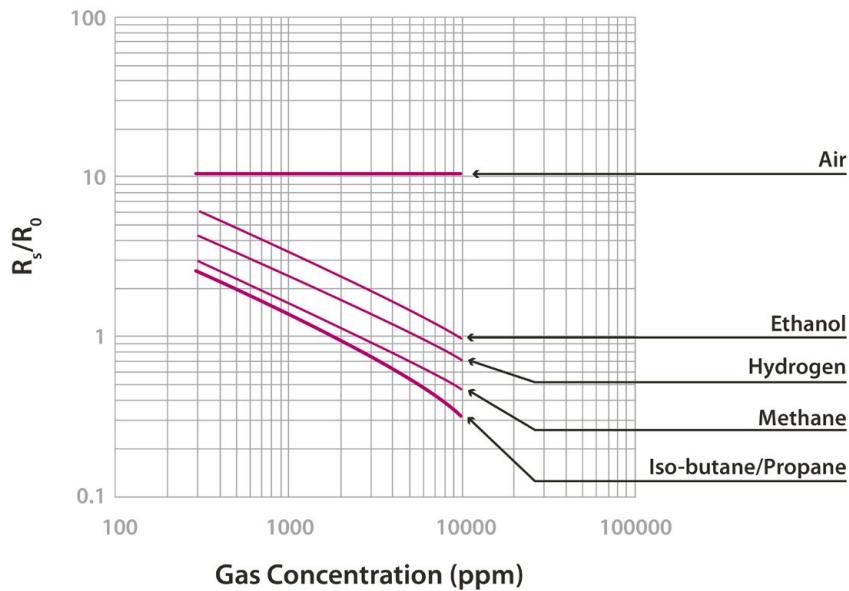


Figure : Graph of the sensitivity of the TGS2610 taken from the Figaro sensor's data sheet

## 8.9. Air Pollutants Sensor – TGS2600

### 8.9.1. Specifications

**Gases:** C<sub>4</sub>H<sub>10</sub>, CH<sub>3</sub>CH<sub>2</sub>OH, H<sub>2</sub>, CO, CH<sub>4</sub>

**Measurement range:** 1 ~ 100 ppm

**Air resistance:** 10 ~ 90 kΩ

**Sensitivity:** 0.3 ~ 0.6 (ratio between the resistance in 10 ppm of H<sub>2</sub> and in air)

**Supply voltage:** 5 V ±0.2 V DC

**Operating temperature:** -10 ~ +40 °C

**Response time:** 30 seconds

**Minimum load resistance:** 0.45 kΩ

**Average consumption:** 46 mA



Figure : Image of TGS2600 sensor

### 8.9.2. Measurement Process

The TGS2600 sensor shows sensitivity to the variation of the concentration of numerous gases that are not usually found in the composition of the atmosphere and which are considered contaminants. Amongst these would be mainly, Ethanol (CH<sub>3</sub>CH<sub>2</sub>OH) and Isobutane (C<sub>4</sub>H<sub>10</sub>) and, with less response, Carbon Monoxide (CO) and Methane (CH<sub>4</sub>). This sensor is also sensitive to variations in the concentration of Hydrogen (H<sub>2</sub>). The sensor's resistance in air would vary between 10 and 90 kΩ, with a ratio of sensitivity between 0.3 and 0.6 for an H<sub>2</sub> concentration of 10ppm. Because of this variability it is recommended to calibrate each one of the sensors prior to their use in a final application.

This sensor can be inserted in sockets 6 and 7, and must be placed following the direction indicated in the silkscreen of the PCB. To read it, the necessary instructions are in the library:

Reading code:

```
{
    // Air Pollutants Sensor can be connected in SOCKET6 AND SOCKET7
    APPSensorClass APPSensor(SOCKET_7); // ----- SOCKET7 Class used

    // Switch ON and configure the Gases Board
    Gases.ON();
    // Switch ON the sensor socket
    APPSensor.ON();

    // Voltage value of the sensor
    float APPVol = APPSensor.readVoltage();
    // Resistance of the sensor
    float APPRes = APPSensor.readResistance(APPVol);
    // PPM value of CO
    float APPVolPPM = APPSensor.readConcentration(APPRes);
}
```

You can find a complete example code for reading the TGS2600 sensor placed on socket 7 in the following link:

<http://www.libelium.com/development/wasp mote/examples/ga-v30-12-app-sensor-reading>

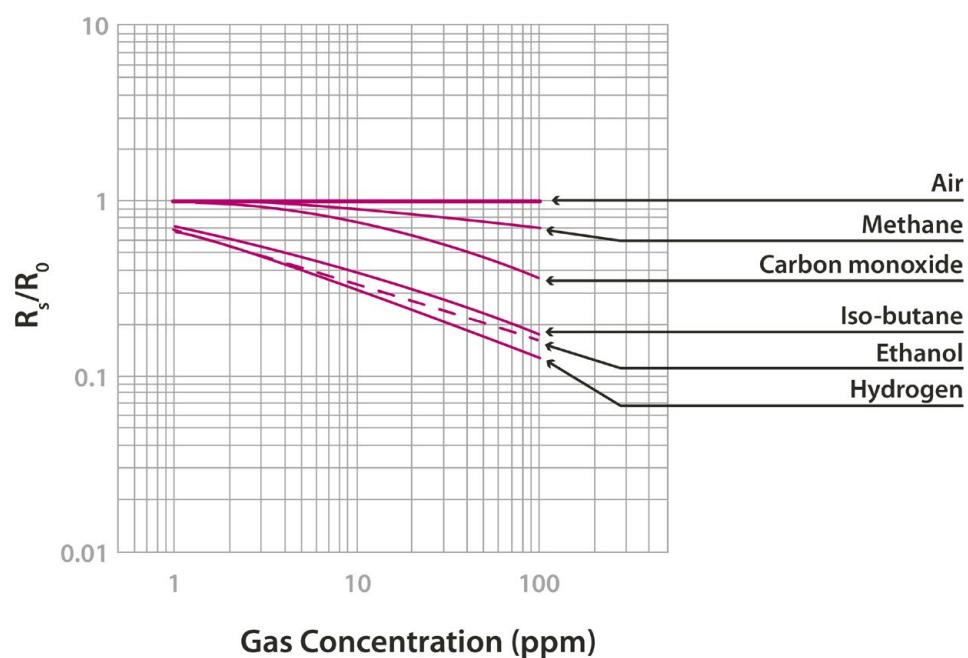


Figure : Graph of the sensitivity of the TGS2600 taken from the Figaro sensor's data sheet

## 8.10. Air Pollutants II sensor – TGS2602

### 8.10.1. Specifications

**Gases:** C<sub>6</sub>H<sub>5</sub>CH<sub>3</sub>, H<sub>2</sub>S, CH<sub>3</sub>CH<sub>2</sub>OH, NH<sub>3</sub>, H<sub>2</sub>

**Measurement range:** 1 ~ 30 ppm

**Air resistance:** 10 ~ 100 kΩ

**Sensitivity:** 0.15 ~ 0.5 (ratio between the resistance in 10 ppm of Ethanol and in air)

**Supply voltage:** 5 V ±0.2 V DC

**Operating temperature:** +10 ~ +50 °C

**Storage temperature:** -20 ~ +60 °C

**Response time:** 30 seconds

**Minimum load resistance:** 0.45 kΩ

**Average consumption:** 61 mA



Figure : Image of the TGS2602 sensor

### 8.10.2. Measurement Process

The TGS2602 is a sensor similar to the TGS2600 which reacts varying its resistance in the presence of contaminant gases, mainly Toluene (C<sub>6</sub>H<sub>5</sub>CH<sub>3</sub>), Hydrogen Sulphide (H<sub>2</sub>S), Ethanol (CH<sub>3</sub>CH<sub>2</sub>OH), Ammonia (NH<sub>3</sub>) and to a lesser extent, Hydrogen (H<sub>2</sub>). In air without contaminants the sensor shows a resistance between 10 and 100 kΩ with a variation ratio between 0.15 and 0.5 between the resistance in 10 ppm of CH<sub>3</sub>CH<sub>2</sub>OH and this one. This variability makes a calibration of the sensor necessary before using it in a final application.

The TGS2602 sensor can be inserted in sockets 6 and 7, and must be placed following the direction indicated in the silkscreen of the PCB. To read it, the necessary instructions are in the library:

Reading code:

```
{
    // Air Pollutants Sensor can be connected in SOCKET6 AND SOCKET7
    APPSensorClass APPSensor(SOCKET_7); // ----- SOCKET7 Class used

    // Switch ON and configure the Gases Board
    GasesBoard.ON();
    // Switch ON the sensor socket
    APPSensor.ON();

    // Voltage value of the sensor
    float APPVol = APPSensor.readVoltage();
    // Resistance of the sensor
    float APPRes = APPSensor.readResistance(APPVol);
    // PPM value of Air Pollutants
    float APPValPPM = APPSensor.readConcentration(APPRes);
}
```

You can find a complete example code for reading the TGS2602 sensor placed on socket 7 in the following link:

<http://www.libelium.com/development/wasp mote/examples/ga-v30-12-app-sensor-reading>

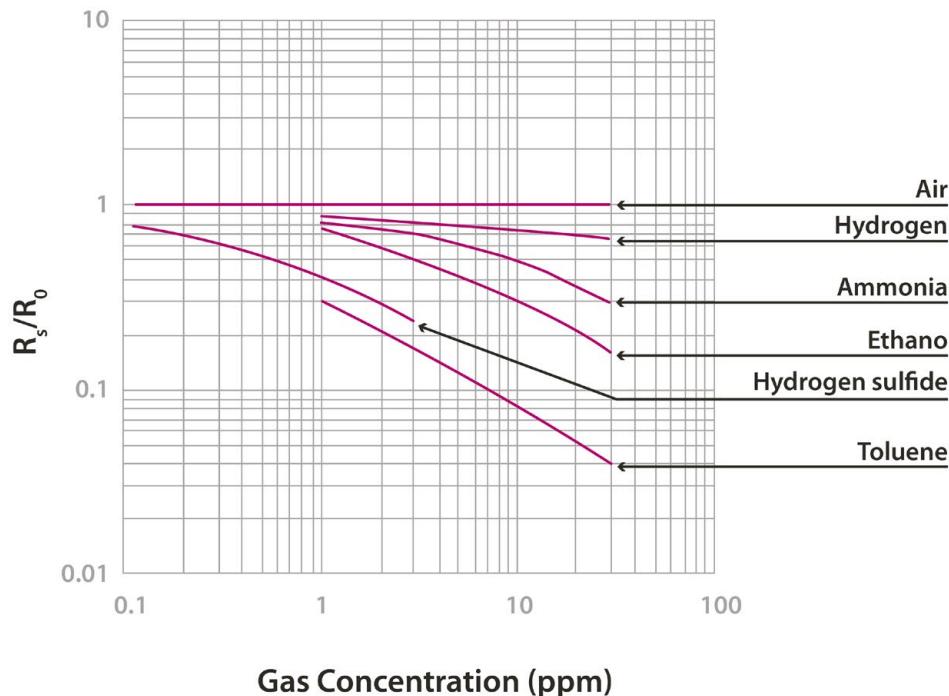


Figure : Graph of the sensitivity of the TGS2602 taken from the Figaro sensor's data sheet

## 8.11. Solvent Vapors Sensor - TGS2620

### 8.11.1. Specifications

**Gases:**  $\text{CH}_3\text{CH}_2\text{OH}$ ,  $\text{H}_2$ ,  $\text{C}_4\text{H}_{10}$ ,  $\text{CO}$ ,  $\text{CH}_4$

**Measurement range:** 50 ~ 5000 ppm

**Resistance to 300 ppm of Ethanol:** 1 ~ 5 k $\Omega$

**Sensitivity:** 0.3 ~ 0.5 (ratio between the resistance at 300 ppm and at 50 ppm)

**Supply voltage:** 5 V  $\pm$ 0.2 V DC

**Operating temperature:** -10 ~ +40 °C

**Response time:** 30 seconds

**Load minimum resistance:** 0.45 k $\Omega$

**Average consumption:** 46 mA (throughout the complete power supply cycle in 250 ms)



Figure : Image of the TGS2620 Sensor

### 8.11.2. Measurement Process

The TGS2620 sensor allows detection of alcohol and organic gases, mainly Ethanol ( $\text{CH}_3\text{CH}_2\text{OH}$ ), Hydrogen ( $\text{H}_2$ ), Isobutane ( $\text{C}_4\text{H}_{10}$ ), Carbon Monoxide ( $\text{CO}$ ) and Methane ( $\text{CH}_4$ ). The resistance the sensor shows in a 300 ppm concentration of Ethanol can vary between 1 and 5 k $\Omega$ , while the sensitivity ratio between this and the resistance in 50 ppm varies between 0.3 and 0.5. As a consequence of these variations it is necessary to calibrate each sensor before their insertion into a final application.

The TGS2620 sensor can be inserted in sockets 6 and 7, and must be placed following the direction indicated in the silkscreen of the PCB. To read it, the necessary instructions are in the library:

Reading code:

```
{
    // Solvent Vapors sensor can be connected in SOCKET6 AND SOCKET7
    SVSensorClass SVSensor(SOCKET_7); // <---- SOCKET7 Class used

    // Switch ON and configure the Gases Board
    Gases.ON();
    // Switch ON the sensor socket
    SVSensor.ON();

    // Voltage value of the sensor
    float SVVol = SVSensor.readVoltage();
    // Resistance of the sensor
    float SVRes = SVSensor.readResistance(SVVol);
    // PPM value of Solvent Vapors
    float SVPPM = SVSensor.readConcentration(SVRes);
}
```

You can find a complete example code for reading the TGS2620 sensor placed on socket 6 in the following link:

<http://www.libelium.com/development/wasp mote/examples/ga-v30-10-sv-sensor-reading>

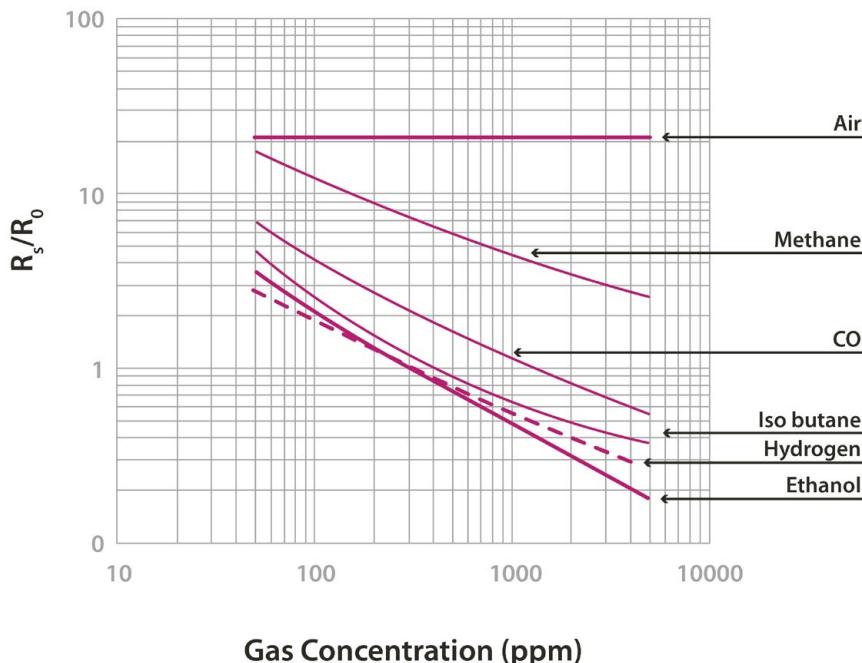


Figure : Graph of the sensitivity of the TGS2620 taken from the Figaro sensor's data sheet

## 8.12. Ozone ( $O_3$ ) Sensor - MiCS-2614

### 8.12.1. Specifications

This sensor is a version for the MiCS-2610 sensor.

**Gases:**  $O_3$

**Measurement range:** 10 ~ 1000 ppb

**Air resistance:** 3 ~ 60 k $\Omega$  (typically 11k $\Omega$ )

**Sensitivity:** 2 ~ 4 (typically 1.5, ratio between the resistance at 100ppb and at 50ppb)

**Supply voltage:** 1.95 ~ 5 V DC

**Operating temperature:** -30 ~ +85 °C

**Response time:** 30 seconds

**Average consumption:** 34 mA



Figure : Image of the MiCS-2614 sensor

### 8.12.2. Measurement process

The MiCS-2614 is a resistive sensor that allows to measure the variation of the  $O_3$  concentration between 10 ppb and 1000 ppb. Its resistance varies between 11 k $\Omega$  and 2 M $\Omega$  approximately. Unlike the MiCS-2714, this sensor is powered through a 2.5 V voltage regulator, with consumption of approximately 34 mA. The sensor's resistance in air, as well as its sensitivity, can vary between different units, so it is recommended to calibrate each one of them before finally inserting them in the application.

This sensor must be connected in socket 5 of the board, and must be placed following the direction indicated in the silkscreen of the PCB, and its output can be read through the capture commands of the library:

Reading code:

```
{
    // O3 Sensor can be connected in SOCKET5
    O3SensorClass O3Sensor;

    // Switch ON and configure the Gases Board
    Gases.ON();
    // Switch ON the sensor socket
    O3Sensor.ON();

    // Voltage value of the sensor
    float o3Vol = O3Sensor.readVoltage();
    // Resistance of the sensor
    float o3Res = O3Sensor.readResistance(o3Vol);
    // PPM value of O3
    float o3ValPPM = O3Sensor.readConcentration(o3Res);
}
```

You can find a complete example code for reading the MiCS-2614 sensor placed on socket 5 in the following link:

<http://www.libelium.com/development/wasp mote/examples/ga-v30-04-o3-sensor-reading>

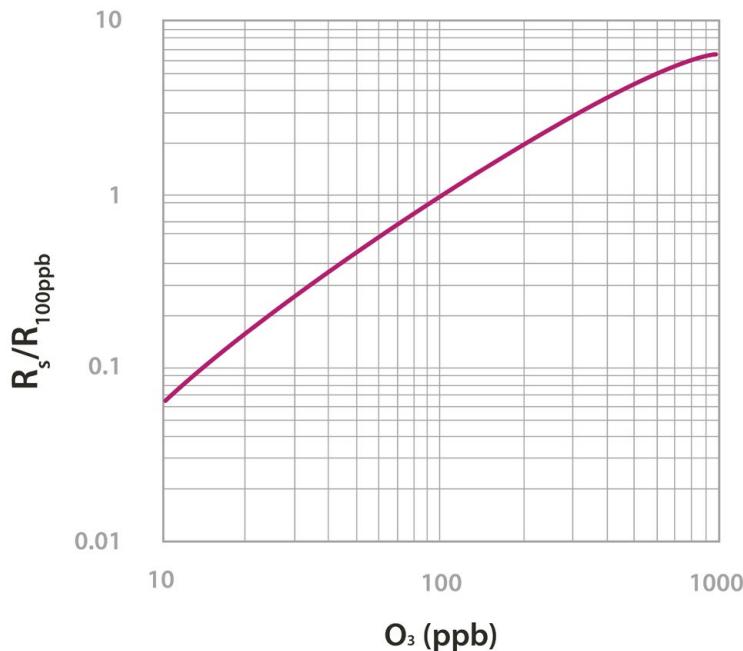


Figure : Graph of the sensitivity of the MiCS-2614 taken from the e2v's sensor data sheet

## 8.13. VOC's Sensor - MiCS-5524

### 8.13.1. Specifications

This sensor is a version for the MiCS-5521 sensor.

**Gases:** CO, Hydrocarbons, Volatile Organic Compounds \*

**Measurement range:** 30 ~ 400 ppm

**Air resistance:** 100 ~ 1500 kΩ

**Sensitivity:** 1.8 ~ 6 (typically 3, ratio between the resistance at 60 ppm and at 200 ppm of CO)

**Supply voltage:** 2.1 ~ 5 V DC

**Operating temperature:** -30 ~ +85 °C

**Response time:** 30 seconds

**Average consumption:** 32 mA



Figure : Image of the MiCS-5524 sensor

(\* ) Chlorinated hydrocarbons, aromatic hydrocarbons, aromatic alcohols, aliphatic alcohols, terpenes, glycols, aldehydes, esters and acids. Detailed list can be found at <http://www.libelium.com/downloads/voc-sensors.xls>

### 8.13.2. Measurement process

The MiCS-5524 is a resistive sensor that responds to a great variety of gases, such as Carbon Monoxide (CO), Hydrocarbons and Volatile Organic Compounds. It's resistance varies between 1000 kΩ and 2 kΩ approximately. Like the MiCS-2614, the MiCS-5524 is powered through a 2.5 V voltage regulator, with consumption of approximately 32 mA. The sensor's resistance in air, as well as its sensitivity, can vary between different units, so it is recommended to calibrate each one of them before finally inserting them in the application.

This sensor must be connected in socket 5 of the board, and must be placed following the direction indicated in the silkscreen of the PCB, and its output can be read through the capture commands of the library:

Reading code:

```
{
    // VOC Sensor must be connected in SOCKET5
    VOCSensorClass VOCSensor;

    // Switch ON and configure the Gases Board
    Gases.ON();
    // Switch ON the sensor socket
    VOCSensor.ON();

    // Voltage value of the sensor
    float vocVol = VOCSensor.readVoltage();
    // Resistance of the sensor
    float vocRes = VOCSensor.readResistance(vocVol);
    // PPM value of VOC
    float vocValPPM = VOCSensor.readConcentration(vocRes);
}
```

You can find a complete example code for reading the MiCS-5524 sensor placed on socket 5 in the following link:

<http://www.libelium.com/development/wasp mote/examples/ga-v30-06-voc-sensor-reading>

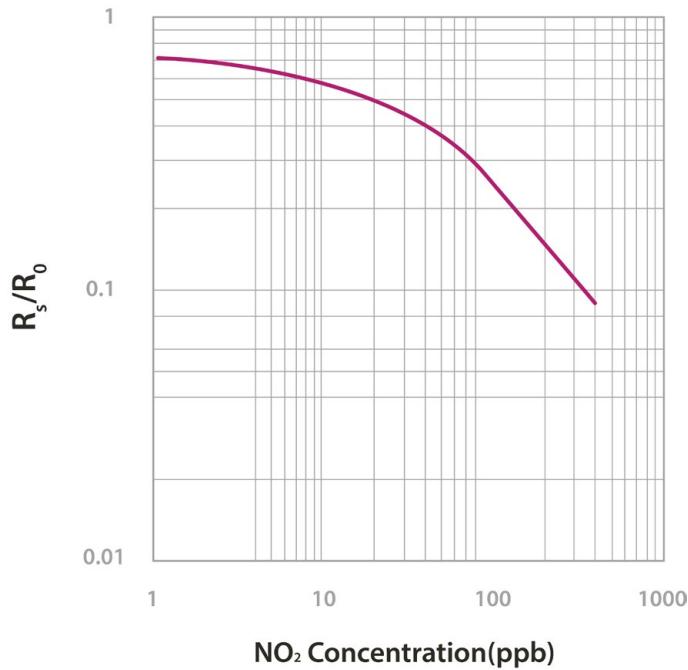


Figure : Graph of the sensitivity of the MiCS-5524 taken from the e2v's sensor data sheet

## 8.14. Ultrasound sensor probe (MaxSonar® from MaxBotix™)

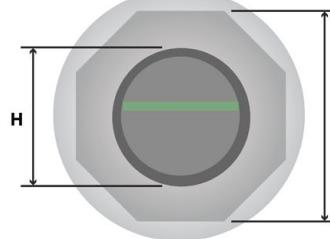
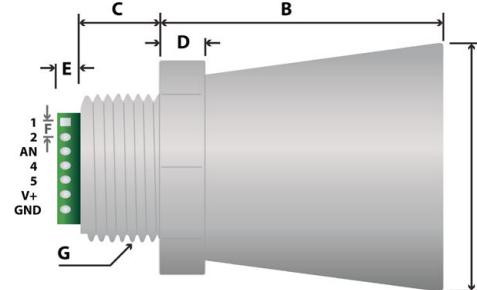
### 8.14.1. Specifications

#### I2CXL-MaxSonar®-MB7040™

- Operation frequency:** 42 kHz
- Maximum detection distance:** 765 cm
- Interface:** Digital bus
- Power supply:** 3.3 V ~ 5 V
- Consumption (average):** 2.1 mA (powered at 3.3 V) – 3.2 mA (powered at 5 V)
- Consumption (peak):** 50 mA (powered at 3.3 V) – 100 mA (powered at 5 V)
- Usage: Indoors and outdoors (IP-67)**



Figure : Ultrasonic I2CXL-MaxSonar®-MB7040 from MaxBotix™ sensor



A	1.72" dia.	43.8 mm dia.
B	2.00"	50.7 mm
C	0.58"	14.4 mm
D	0.31"	7.9 mm
E	0.18"	4.6 mm
F	0.1"	2.54 mm
G	3/4" National Pipe Thread Straight	
H	1.032" dia.	26.2 dia.
I	1.37"	34.8 mm

weight: 1.76 oz. ; 50 grams

Figure : Ultrasonic I2CXL-MaxSonar®-MB7040 sensor dimensions

In the figure below we can see a diagram of the detection range of the sensor developed using different detection patterns (a 0.63 cm diameter dowel for diagram A, a 2.54 cm diameter dowel for diagram B, an 8.25cm diameter rod for diagram C and a 28 cm wide board for diagram D):

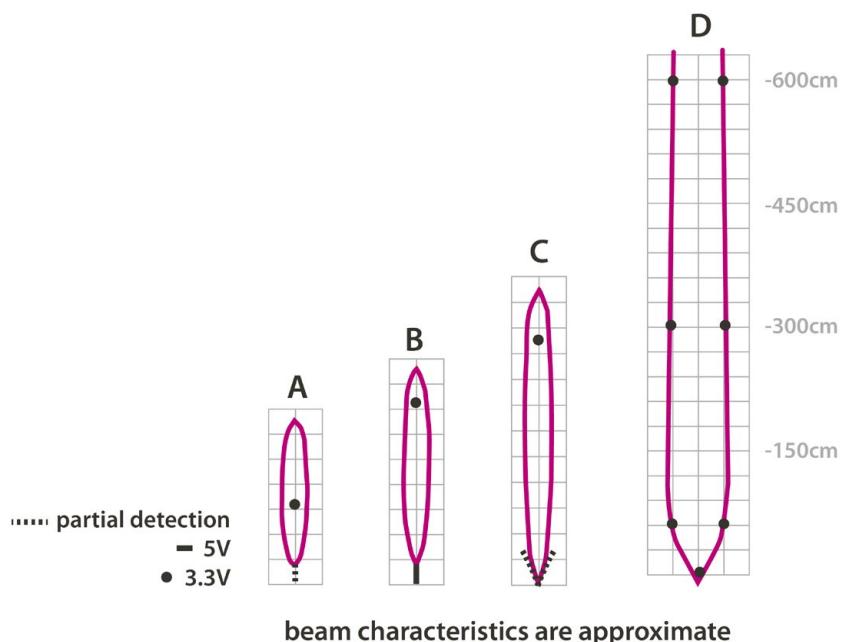


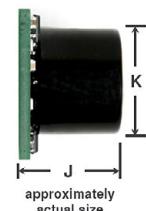
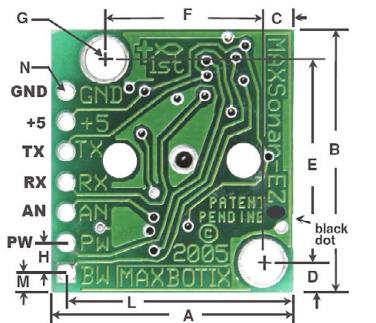
Figure : Diagram of the sensor beam extracted from the data sheet of the XL-MaxSonar®-WRA1™ sensor from MaxBotix

### I2CXL-MaxSonar®-MB1202™:

- Operation frequency:** 42 kHz
- Maximum detection distance:** 645 cm
- Sensitivity (analog output):** 2.5 mV/cm (powered at 3.3 V) – 3.8 mV/cm (powered at 5 V)
- Power supply:** 3.3 ~ 5 V
- Consumption (average):** 2 mA (powered at 3.3 V) – 3 mA (powered at 5 V)
- Usage:** Indoors



Figure : Ultrasonic I2CXL-MaxSonar®-MB1202 from MaxBotix™ Sensor



A	0.785"	19.9 mm	H	0.100"	2.54 mm
B	0.870"	22.1 mm	J	0.645"	16.4 mm
C	0.100"	2.54 mm	K	0.610"	15.5 mm
D	0.100"	2.54 mm	L	0.735"	18.7 mm
E	0.670"	17.0 mm	M	0.065"	1.7 mm
F	0.510"	12.6 mm	N	0.038" dia.	1.0 mm dia.
G	0.124" dia.	3.1 mm dia.			weight: 4.3 grams

Figure : Ultrasonic I2CXL-MaxSonar®-MB1202 Sensor dimensions

**Note:** Only one MB1202 sensor is supported.

In the figure below we can see a diagram of the detection range of the sensor developed using different detection patterns (a 0.63 cm diameter dowel for diagram A, a 2.54 cm diameter dowel for diagram B, an 8.25 cm diameter rod for diagram C and a 28 cm wide board for diagram D):

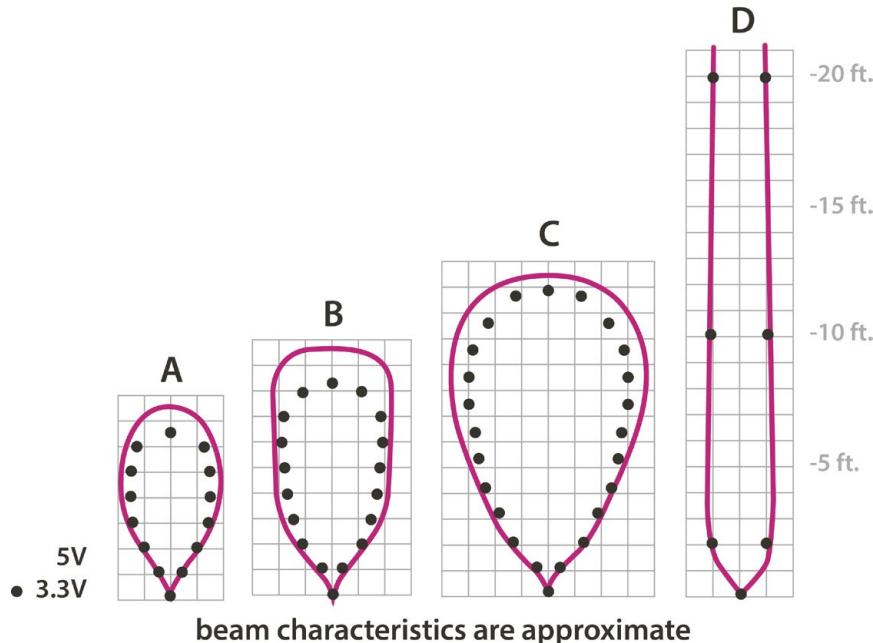


Figure : Diagram of the sensor beam extracted from the data sheet of the Ultrasonic I2CXL-MaxSonar®-MB1202 sensor from MaxBotix

### 8.14.2. Measurement process

The MaxSonar® sensors from MaxBotix can be connected through the digital bus interface. In the next figure, we can see a drawing of two example applications for the ultrasonic sensors, such as liquid level monitoring or presence detection.

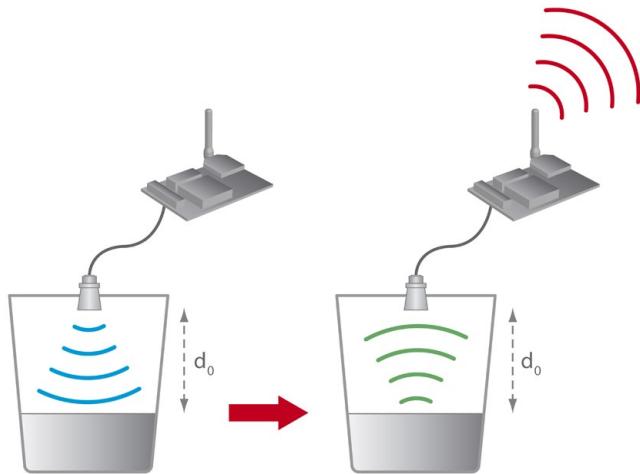


Figure : Examples of application for the MaxSonar® sensors

The XL-MaxSonar®-WRA1TM sensor is endowed with an IP-67 casing, so it can be used in outdoors applications, such as liquid level monitoring in storage tanks. Below a sample code to measure one of the ultrasound sensors (the XL-MaxSonar®-WRA1) is shown:

Reading code:

```
{
    uint16_t distance;
    Gases.ON();
    distance = Gases.getDistance();
}
```

### 8.14.3. Socket

These sensors share the sockets with the BME280 temperature, humidity and pressure sensor. The pin correspondence, highlighted in the figure below, is the same for both.

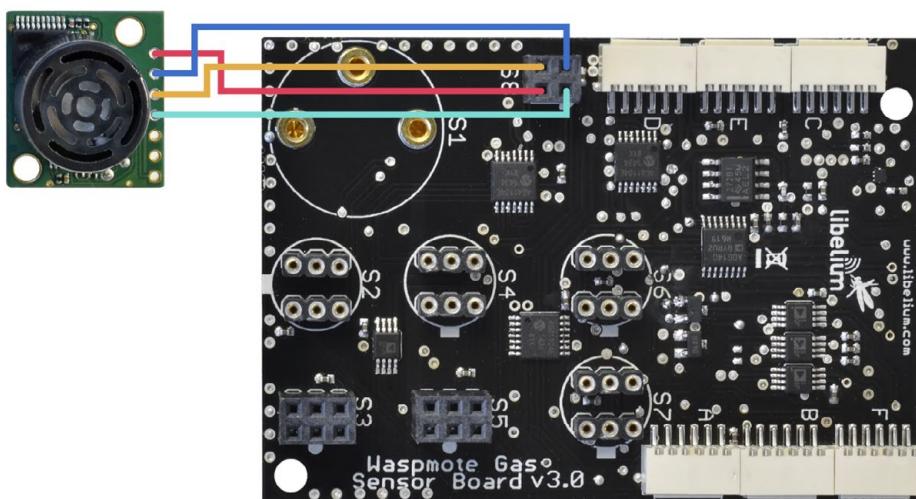


Figure : Image of the sockets for connecting the MaxSonar® Sensors

## 8.15. Luminosity Sensor

### 8.15.1. Specifications

#### Electrical characteristics

Dynamic range: 0.1 to 40000 lux

Spectral range: 300 ~ 1100 nm

Voltage range: 2.7 ~ 3.6 V

Supply current typical: 0.24 mA

Sleep current maximum: 0.3 µA

Operating temperature: -30 ~ 70 °C



Figure : Image of the Luminosity sensor

**Note:** Only one TSL2561 sensor is supported

### 8.15.2. Measurement process

This is a light-to-digital converter that transforms light intensity into a digital signal output. This device combines one broadband photo-diode (visible plus infrared) and one infrared-responding photo-diode on a single CMOS integrated circuit capable of providing a near-photopic response over an effective 20-bit dynamic range (16-bit resolution). Two integrating ADCs convert the photo-diode currents to a digital output that represents the irradiance measured on each channel. This digital output in lux is derived using an empirical formula to approximate the human eye response.

Reading code:

```
{
    // Switch ON and configure the Gases Board
    Gases.ON();
    // Read luxes (INDOOR or OUTDOOR)
    luxes = Gases.getLuxes(INDOOR);
}
```

You can find a complete example code for reading the luminosity in the following link:

<http://www.libelium.com/development/wasp mote/examples/ga-v30-14-luxes-sensor-reading>

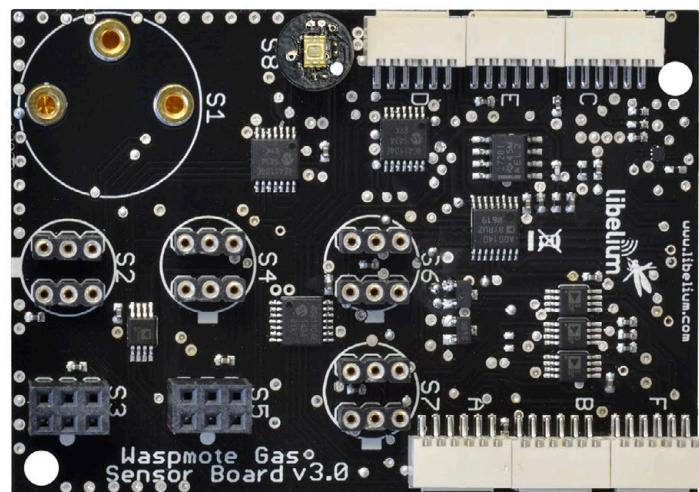


Figure : Image of the socket for the luxes sensor

In the image above we can see highlighted the four pins of the terminal block where the sensor must be connected to the board. The white dot luxes board, must match the mark of the Gases sensor board.

## 8.16. Design and connections

The different connectors used for the sensors' connection can be used for the integration of different sensors to those previously planned, provided that the organization of the pins is followed, as well as the defined electrical specifications in the WaspMote manual.

### 8.16.1. Socket 1

Socket 1 has been designed to connect SK-25 ( $O_2$ ). Three isolated pins have been assigned to the  $O_2$  sensor, one connected to ground, another to the sensor's output and another without connections assigned to the sensor's reference pin. This socket has no power supply, and the signal of the sensor can be read directly.

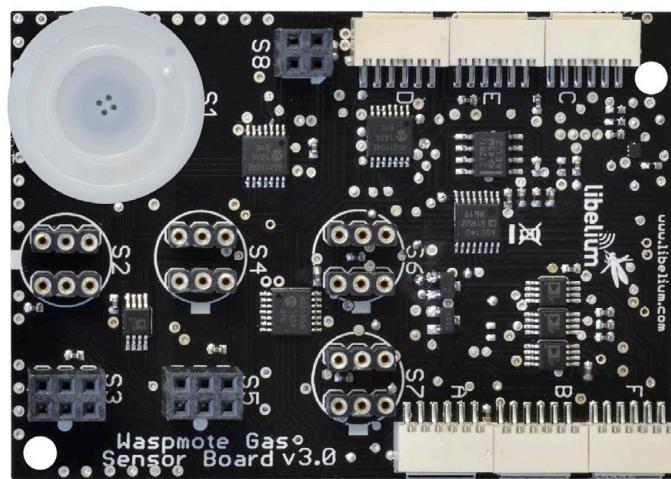


Figure : Image of socket 1 with a connected sensor

### 8.16.2. Socket 2

Socket 2 has been designed to connect TGS4161 (CO<sub>2</sub>). In this way, the connection designed for the CO<sub>2</sub> sensor has two strips of three 2.54 mm pitch female pins, of which two are without connection (the two in the center), one drives the sensor output to the amplification stage, the other the 5 V supply controlled by a switch and two are for the circuit's connection to ground.

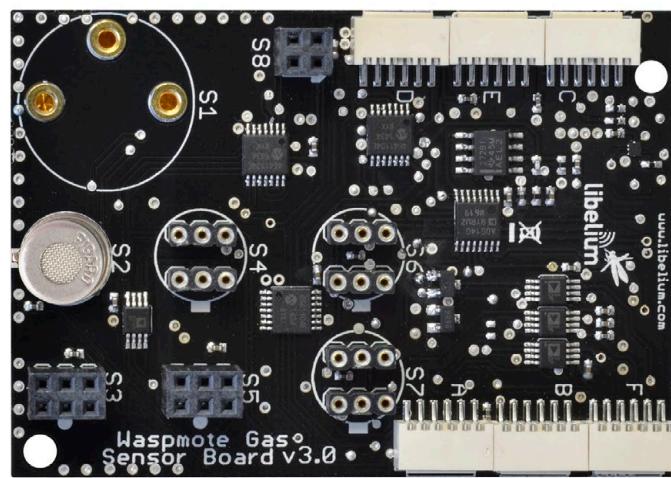


Figure : Image of socket 2 with a connected sensor

### 8.16.3. Socket 3

Socket 3 is endowed with a 1.8 V regulator in order to provide connection to the MiCS-2714 sensors for NO<sub>2</sub>.

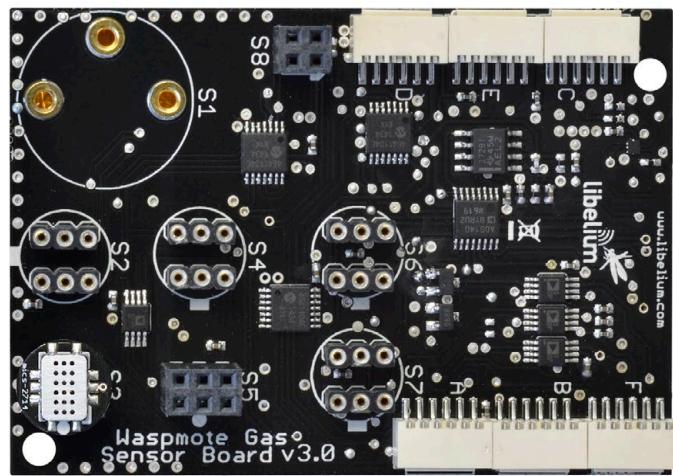


Figure : Image of socket 3 with a connected sensor

### 8.16.4. Socket 4

Socket 4 has been designed to connect TGS2442 for CO, TGS2444 for NH<sub>3</sub> sensors. It is composed of two strips of three 2.54 mm pitch female pins that connect the sensor to power supply, ground and output to a 100 kΩ load variable resistor followed by an amplification stage of maximum gain 101, both controlled by a digital potentiometer.



Figure : Image of socket 4 with a connected sensor

### 8.16.5. Socket 5

Socket 5 is endowed with a 2.5 V regulator in order to provide connection to the MiCS-2614 sensors for O<sub>3</sub> and the MiCS-5524 sensors for VOC's.

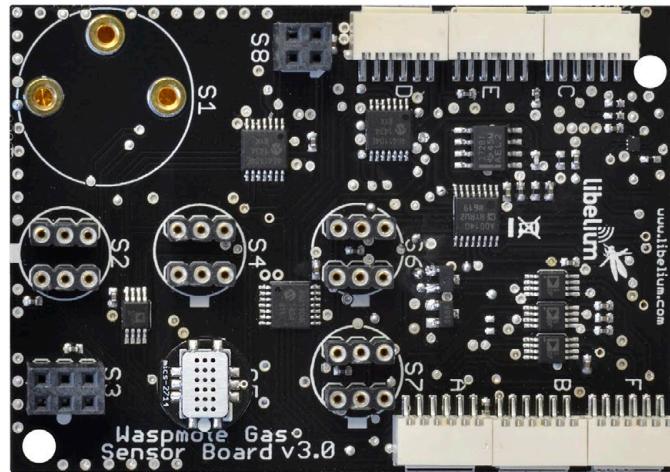


Figure : Image of socket 5 with a connected sensor

### 8.16.6. Socket 6 and socket 7

Socket 6 and socket 7 have been designed to connect TGS2611 resistive sensors for CH<sub>4</sub>, TGS2600 and TGS2602 for air contaminants, TGS2610 for LPG gases and TGS2620 for solvent vapors.

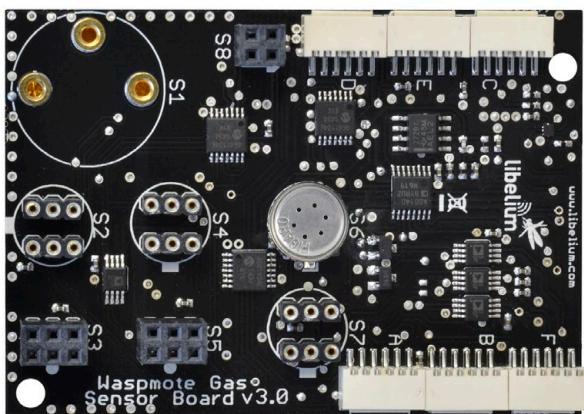


Figure : Image of socket 6 with a connected sensor

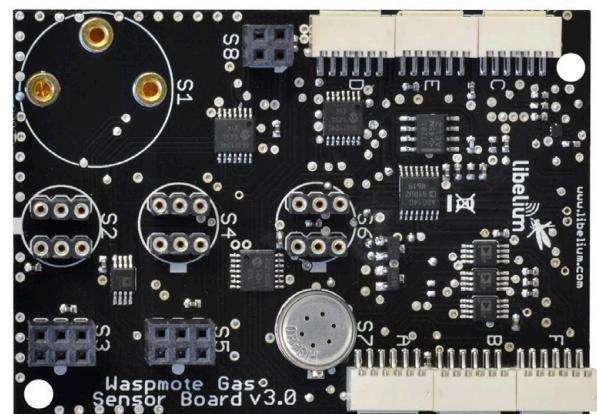


Figure : Image of socket 7 with a connected sensor

### 8.16.7. Socket 8

In the image below we can see how to connect the BME280 sensor in the socket 8 of the Gases Sensor Board. Is important to connect the sensor with the correct orientation. Please see the reference point of the PCB.

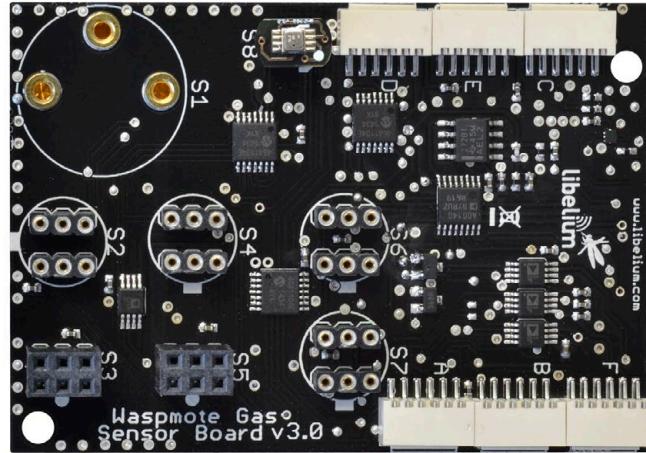


Figure : Image of the socket for the BME280 sensor

## 9. Board configuration and programming

### 9.1. Hardware configuration

The Gases Sensor Board does not require any handling of the hardware by the user except for placing the sensors in their corresponding position. In the section dedicated to each connector we can see an image of each of the sensors inserted into the corresponding socket with the reference to the sensor's direction highlighted.

### 9.2. Library

The Gases Sensor Board for WaspMote has its own library which contains the set of necessary instructions to easily and intuitively configure and read each one of the sensors which connect to the board. Next each one of the functions is described and the process of configuration detailed for each sensor. The specific configuration which must be applied to each one of the sensors is explained in the specific sensor's section.

When using the Gases Sensor Board v30 on WaspMote PRO, remember it is mandatory to include the library by introducing the next line at the beginning of the code:

```
#include <WaspSensorGas_v30.h>
```

Next, the different functions that make up the library are described:

#### Object creation

Each sensor can be managed using the corresponding class by creating an instance of the class. If no arguments in the constructor, the socket will be configured with the default parameters, normally, gain 1, and load resistor at 12.5 kΩ:

- O<sub>2</sub> sensor object creation: `O2SensorClass O2Sensor;`
- CO<sub>2</sub> sensor object creation: `CO2SensorClass CO2Sensor;`
- NO<sub>2</sub> sensor object creation: `NO2SensorClass NO2Sensor;`
- NH<sub>3</sub> sensor object creation: `NH3SensorClass NH3Sensor;`
- CO sensor object creation: `COSensorClass COSensor;`
- VOC sensor object creation: `VOCSensorClass VOCSensor;`
- O<sub>3</sub> sensor object creation: `O3SensorClass O3Sensor;`

Sensors for socket 6 and 7 must be referred to the corresponding socket:

#### Object creation in the socket 6

- CH<sub>4</sub> sensor object creation in socket 6: `CH4SensorClass CH4Sensor(SOCKET_6);`
- APP sensor object creation in socket 6: `APPSensorClass APPSensor(SOCKET_6);`
- SV sensor object creation in socket 6: `SVSensorClass SVSensor(SOCKET_6);`
- LPG sensor object creation in socket 6: `LPGSensorClass LPGSensor(SOCKET_6);`

#### Same sensors in the socket 7:

- CH<sub>4</sub> sensor object creation in socket 7: `CH4SensorClass CH4Sensor(SOCKET_7);`
- APP sensor object creation in socket 7: `APPSensorClass APPSensor(SOCKET_7);`
- SV sensor object creation in socket 7: `SVSensorClass SVSensor(SOCKET_7);`
- LPG sensor object creation in socket 7: `LPGSensorClass LPGSensor(SOCKET_7);`

The gain and the load resistor can be configured as shown below. In the next line we have configured a gain of 5, and a load resistor of 15 kΩ:

```
NO2SensorClass NO2Sensor(5.0, 15.0);
```

O<sub>2</sub> sensor and CO<sub>2</sub> sensors do not use load resistance and only the gain can be configured.

```
CO2SensorClass CO2Sensor(2);
```

In the case of the O<sub>2</sub> sensor, for configuring the gain, you must also define the socket where the sensor is going to be connected:

```
O2SensorClass O2Sensor(SOCKET_1, 50.0);
```

### Power control functions

Turns on the sensor board by activating the 3.3V and 5V supply lines:

```
Gases.ON();
```

Turns off the sensor board by cutting the 3.3V and 5V supply lines:

```
Gases.OFF();
```

Switch on the different sensors using the corresponding object:

```
O2Sensor.ON();
CO2Sensor.ON();
NO2Sensor.ON();
```

Switch off the different sensor using the corresponding object:

```
O2Sensor.OFF();
CO2Sensor.OFF();
NO2Sensor.OFF();
```

The sensors share the same amplifier stage, so it is necessary to switch them on before reading and switch them off after the reading process.

### Calibration functions

For measuring ppm values of gases, is necessary to configure some parameters. In the case of the NO<sub>2</sub> sensor we are going to use three points. Each point has a value in ppm's and the resistance for this concentration, and the number of points to be used:

```
// Concentrations used in calibration process
#define POINT1_PPM_NO2 10.0 // PPM VALUE <-- Normal concentration in air
#define POINT2_PPM_NO2 50.0 // PPM VALUE
#define POINT3_PPM_NO2 100.0 // PPM VALUE

// Calibration voltages obtained during calibration process (in KOHMs)
#define POINT1_RES_NO2 45.25 // <-- Rs at normal concentration in air
#define POINT2_RES_NO2 25.50
#define POINT3_RES_NO2 3.55
```

```
// Define the number of calibration points
#define numPoints 3
```

These calibration values will be stored in two arrays:

```
float concentrations[] = {POINT1_PPM_NO2, POINT2_PPM_NO2, POINT3_PPM_NO2};
float voltages[] = {POINT1_RES_NO2, POINT2_RES_NO2, POINT3_RES_NO2};
```

Finally, it is necessary to call the function that calculates the approximation curve:

```
N02Sensor.setCalibrationPoints(voltages, concentrations, numPoints);
```

### Reading functions

All sensor have functions for reading the voltage generated by the sensor, the resistance and the concentration, except he O<sub>2</sub> (SK-25) sensor and the CO<sub>2</sub> (TGS4161) sensor, that only returns voltage and concentration values:

```
// Voltage value of the sensor
float N02Vol = N02Sensor.readVoltage();
// Resistance of the sensor
float N02Res = N02Sensor.readResistance();
// PPM value of N02
float N02PPM = N02Sensor.readConcentration();
```

The files of the sensor board library are: WaspSensorGas\_v30.cpp, WaspSensorGas\_v30.h

They can be downloaded from: [http://www.libelium.com/development/wasp mote/sdk\\_and\\_applications](http://www.libelium.com/development/wasp mote/sdk_and_applications)

## 10. Consumption

### 10.1. Power control

On one side, the control of the Gases Sensor Board power can be carried out using the WaspMote's general on/off system for the 3.3 V and 5 V supply lines, which allows the board to be totally switched on and off (0 µA).

On the other hand, specific control mechanisms have been installed inside the sensor board using a system of solid state switches, allowing the independent digital control of each sensor power without the need to physically access the circuit, except for the Temperature, Humidity and Pressure sensor, which is powered always the board is on. This way, activation and reading of each sensor can be programmed at the same main code, controlled by the microcontroller. In section "Library" where the API libraries related to this board are presented and the use of each of these switches is clearly and precisely described, as well as the correct way to read each sensor.

### 10.2. Consumption table

In the following table the consumption shown by the board when active is detailed, from minimum consumption (constant, fixed by the permanently active components, such as the adaptation electronics and temperature and humidity sensors).

To find out the total consumption of the board with sensors integrated to the consumption of each connector, the consumption of each chosen sensor must be added together. This consumption can be consulted in the section for the sensor itself when all its characteristics are described.

	Consumption
<b>Gases Sensor Board v30</b>	8 mA
<b>O2 Sensor - SK-25</b>	0 mA
<b>CO2 Sensor - TGS4161</b>	50 mA
<b>NO2 Sensor - MiCS-2714</b>	26 mA
<b>CO Sensor - TGS2442</b>	3 mA (throughout the complete power supply cycle in 1 s)
<b>NH3 Sensor - TGS2444</b>	12 mA (throughout the complete power supply cycle in 250 ms)
<b>VOC's Sensor - MiCS-5524</b>	32 mA
<b>O3 Sensor - MiCS-2614</b>	34 mA
<b>CH4 Sensor - TGS2611</b>	61 mA
<b>Air Pollutants I Sensor - TGS2600</b>	46 mA
<b>Air Pollutants II Sensor - TGS2602</b>	61 mA
<b>Liquid Petroleum Sensor - TGS2610</b>	61 mA
<b>Solvent Vapors Sensor - TGS2620</b>	46 mA

### 10.3. Low Consumption Mode

From the point of view of optimizing WaspMote resources when the Gases Sensor Board is used, it is recommended that the following instructions are followed:

- Keep the board switched off while no measurement is being taken. This is the most efficient method of lowering consumption when none of the parameters are being continually monitored.
- To completely disconnect the board's power, disable the switches that allow passage of the 3.3 V supply, the 5 V supply from WaspMote (using the Gases.OFF() library function).
- Optimize the time the sensors are switched on depending on your application. The accuracy of each sensor's measurement which can be obtained will vary depending on the time that it remains switched on or on the power supply cycles which are continually applied, depending on the type of sensor. Knowing the time required to take a measurement in a determined application will allow saving of consumption without losing resolution in the sampled value.
- Simultaneously activate the minimum number of sensors possible. Given that the current allowed in the digital switches' output is limited (about 200mA), it is recommended to not overload them by activating a number of sensors at the same time which in total may surpass this current.

## 11. API Changelog

Keep track of the software changes on this link:

[www.libelium.com/development/waspmote/documentation/changelog/#Gases](http://www.libelium.com/development/waspmote/documentation/changelog/#Gases)

## 12. Maintenance

- In this section, the term "Waspmote" encompasses both the Waspmote device itself as well as its modules and sensor boards.
- Take care with the handling of Waspmote, do not drop it, bang it or move it sharply.
- Avoid putting the devices in areas of high temperatures since the electronic components may be damaged.
- The antennas are lightly threaded to the connector; do not force them as this could damage the connectors.
- Do not use any type of paint for the device, which may damage the functioning of the connections and closure mechanisms.

## 13. Disposal and Recycle

- In this section, the term "Wasp mote" encompasses both the Wasp mote device itself as well as its modules and sensor boards.
- When Wasp mote reaches the end of its useful life, it must be taken to a recycling point for electronic equipment.
- The equipment has to be disposed on a selective waste collection system, different to that of urban solid waste. Please, dispose it properly.
- Your distributor will inform you about the most appropriate and environmentally friendly waste process for the used product and its packaging.



## 14. Certifications

Libelium offers 2 types of IoT sensor platforms, WaspMote OEM and Plug & Sense!:

- **WaspMote OEM** is intended to be used for research purposes or as part of a major product so it needs final certification on the client side. More info at: [www.libelium.com/products/waspmote](http://www.libelium.com/products/waspmote)
- **Plug & Sense!** is the line ready to be used out-of-the-box. It includes market certifications. See below the specific list of regulations passed. More info at: [www.libelium.com/products/plug-sense](http://www.libelium.com/products/plug-sense)

Besides, Meshlium, our multiprotocol router for the IoT, is also certified with the certifications below. Get more info at:

[www.libelium.com/products/meshlium](http://www.libelium.com/products/meshlium)

List of certifications for Plug & Sense! and Meshlium:

- CE (Europe)
- FCC (US)
- IC (Canada)
- ANATEL (Brazil)
- RCM (Australia)
- PTCRB (cellular certification for the US)
- AT&T (cellular certification for the US)



Figure : Certifications of the Plug & Sense! product line

You can find all the certification documents at:

[www.libelium.com/certifications](http://www.libelium.com/certifications)