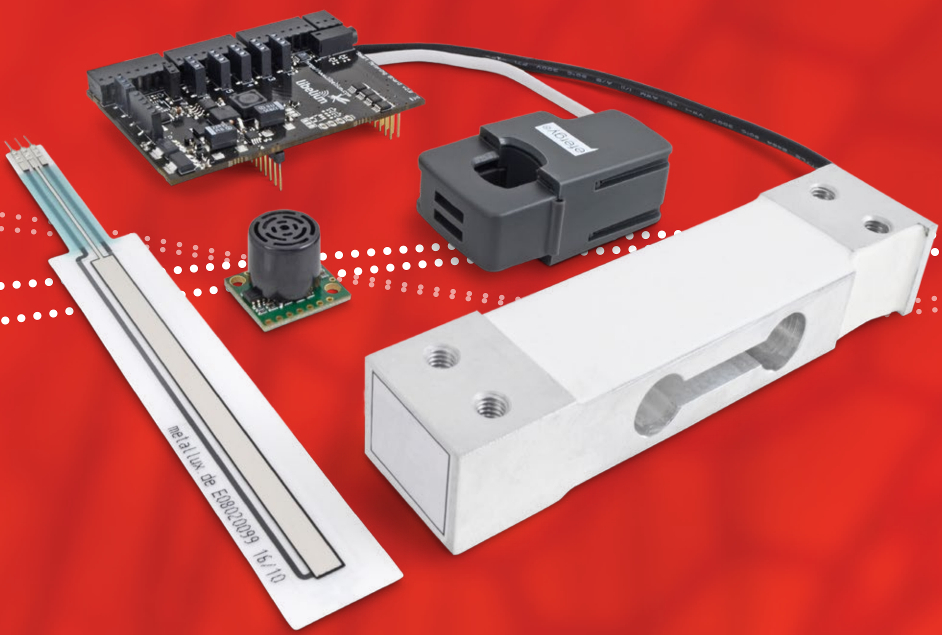
Smart Metering 2.0

Technical Guide



Document version: v4.2 - 04/2013

INDEX

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

2. Waspmote Plug & Sense!

The new Waspmote Plug & Sense! line allows you to easily deploy wireless sensor networks in an easy and scalable way ensuring minimum maintenance costs. The new 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.***

2.1. Features

- Robust waterproof IP65 enclosure
- Add or change a sensor probe in seconds
- Solar powered with internal and external panel options
- Radios available: Zigbee, 802.15.4, Wifi, 868MHz, 900MHz and 3G/GPRS
- Over the air programming (OTAP) of multiple nodes at once
- Special holders and brackets ready for installation in street lights and building fronts
- Graphical and intuitive programming interface

2.2. 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 1: Connecting a sensor probe to Waspmote Plug & Sense!

2.3. Solar Powered

Battery can be recharged using the internal or external solar panel options.
The external solar panel is mounted on a 45º holder which ensures the maximum performance of each outdoor installation.

Figure 2: 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 3: Internal solar panel

Figure 4: Waspmote Plug & Sense! powered by an internal solar panel

2.4. 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 5: Programming a node

Over the Air Programming is also possible once the node has been installed. With this technique you can reprogram wirelessly one or more Waspmote sensor nodes at the same time by using a laptop and the Waspmote Gateway.

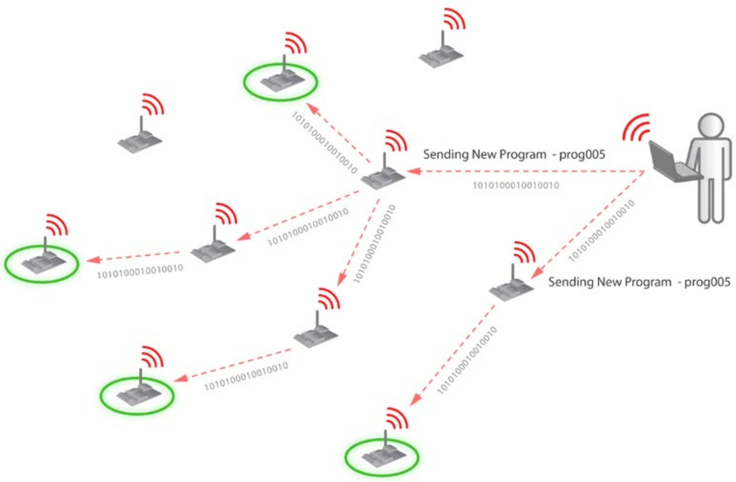


Figure 6: Typical OTAP process

2.5. Radio Interfaces

Model	Protocol	Frequency	txPower	Sensitivity	Range *
XBee-802.15.4-Pro	802.15.4	2.4GHz	100mW	-100dBm	7000m
XBee-ZB-Pro	ZigBee-Pro	2.4GHz	50mW	-102dBm	7000m
XBee-868	RF	868MHz	315mW	-112dBm	12km
XBee-900	RF	900MHz	50mW	-100dBm	10Km
Wifi	802.11b/g	2.4GHz	0dBm - 12dBm	-83dBm	50m-500m
GPRS	-	850MHz/900MHz/ 1800MHz/1900MHz	2W(Class4) 850MHz/900MHz, 1W(Class1) 1800MHz/1900MHz	-109dBm	
3G/GPRS	-	Tri-Band UMTS 2100/1900/900MHz Quad-Band GSM/EDGE, 850/900/1800/1900 MHz	UMTS 900/1900/2100 0,25W GSM 850MHz/900MHz 2W DCS1800MHz/PCS1900MHz 1W	-106dBm	

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

2.6. Program in minutes

In order to program the nodes an intuitive graphic interface has been developed. Developers just need to fill a web form in order to obtain the complete source code for the sensor nodes. This means the complete program for an specific application can be generated just in minutes. Check the Code Generator to see how easy it is at:
http://www.libelium.com/development/plug_&_sense/sdk_and_applications/code_generator

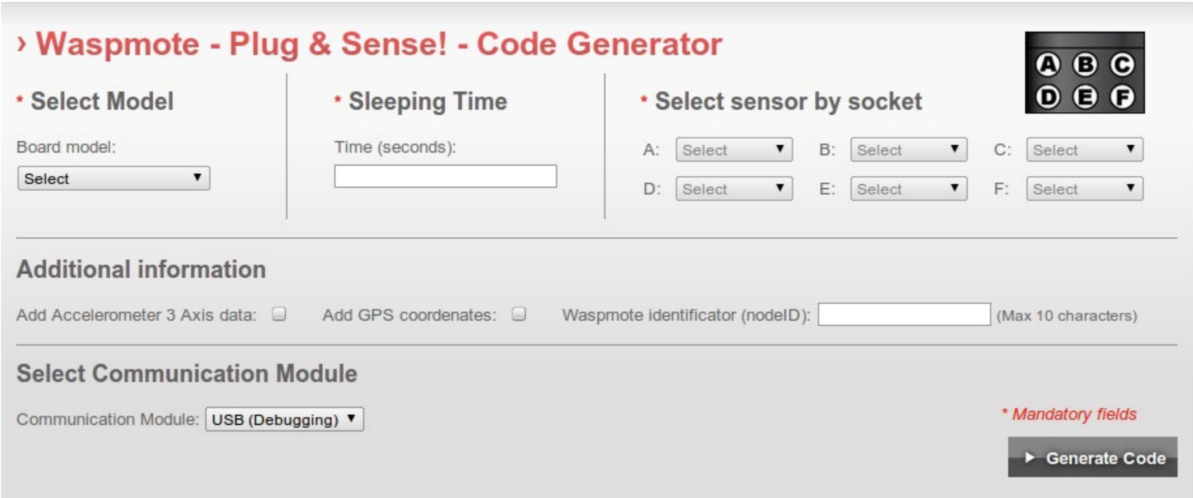


Figure 7: Code Generator

2.7. Data to the Cloud

The Sensor data gathered by the Waspmote Plug & Sense! nodes is sent to the Cloud by **Meshlium**, the Gateway router specially designed to connect Waspmote sensor networks to the Internet via Ethernet, Wifi and 3G interfaces.

Thanks to Meshlium's new feature, the Sensor Parser, now it is easier to receive any frame, parse it and store the data into a local or external Data Base.

Figure 8: Meshlium

2.8. Meshlium Storage Options

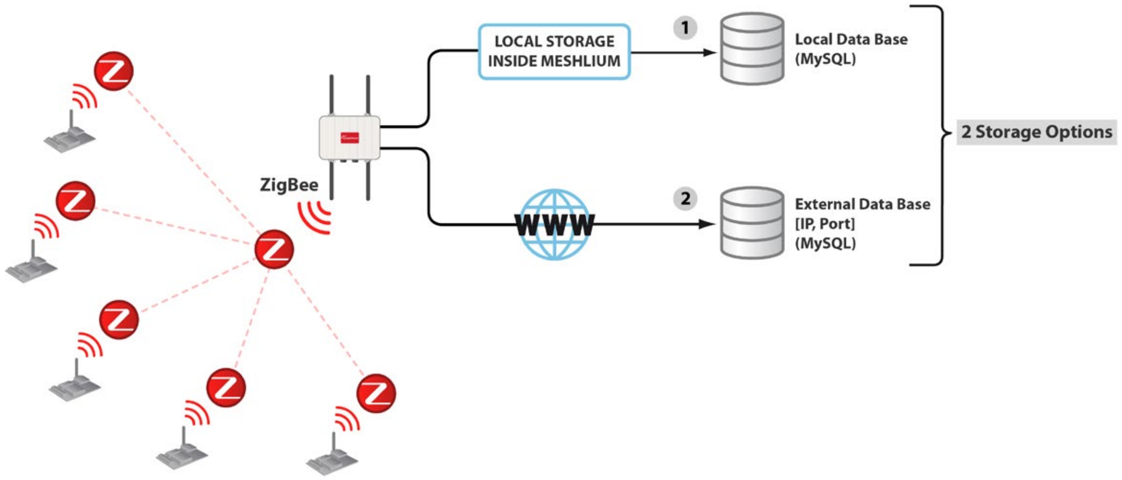


Figure 9: Meshlium Storage Options

- Local Data Base
- External Data Base

2.9. Meshlium Connection Options

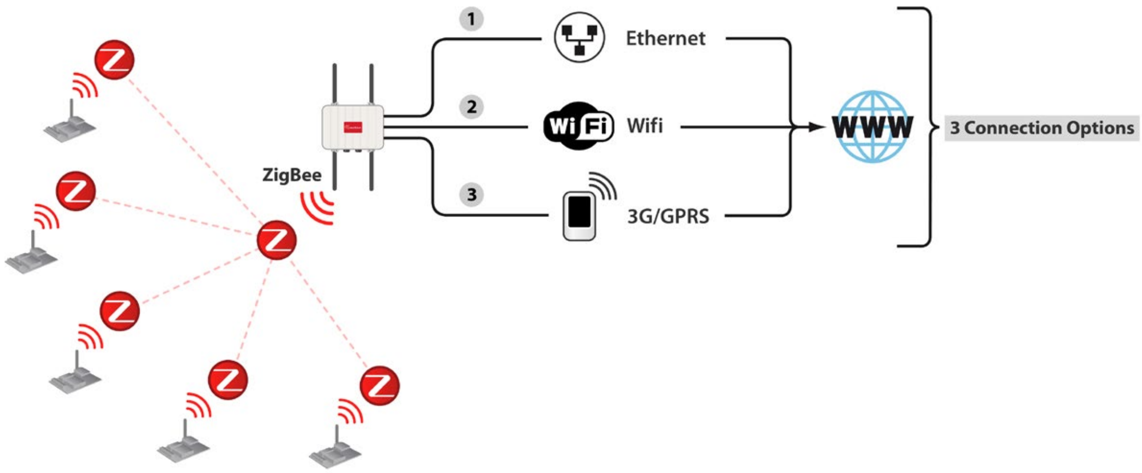


Figure 10: Meshlium Connection Options

- ZigBee → Ethernet
- ZigBee → Wifi
- ZigBee → 3G/GPRS

2.10. 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.

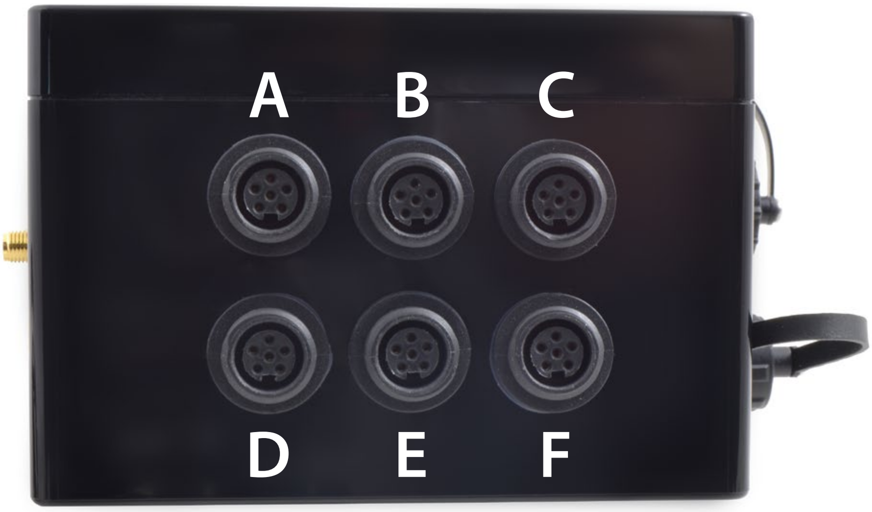


Figure 11: Identification of sensor sockets

2.10.1. Smart Metering

The main applications for this Waspmote Plug & Sense! model are energy measurement, water consumption, pipe leakage detection, liquid storage management, tanks and silos level control, supplies control in manufacturing, industrial automation, agricultural irrigation, etc. Go to the application section in the **Libelium website** for a complete list of services.

Figure 12: Smart Metering 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	Temperature	9203
	Soil temperature	86949*
B	Humidity	9204
C	Ultrasound (distance measurement)	9246
	Liquid flow	9296, 9297, 9298
D	Current sensor	9266
E	Ultrasound (distance measurement)	9246
	Liquid flow	9296, 9297, 9298
F	Luminosity	9205

* Ask Libelium **Sales Department** for more information.

Figure 13: Sensor sockets configuration for Smart Metering model

As we see in the figure below, thanks to the directionable probe, the ultrasound sensor probe may be placed in different positions. The sensor can be focused directly to the point we want to measure.

Figure 14: Configurations of the ultrasound sensor probe

***Note:** For more technical information about each sensor probe go to the **Development section** in Libelium website.*

3. Hardware

3.1. General Description

The Smart Metering 2.0 Board for Waspmote has been conceived to monitor those parameters that may require to be controlled in a domestic environment. It includes sensors for power and water consumption control, displacement, luminosity and environmental humidity. In the normal version of the board, up to 8 sensors, powered independently, can be connected at the same time, while existing a PRO board version with the necessary electronics for load cell control and adaptation.

3.2. Specifications

Weight: 20gr
Dimensions: 73.5 x 51 x 1.3 mm
Temperature Range: [-20ºC, 65ºC]

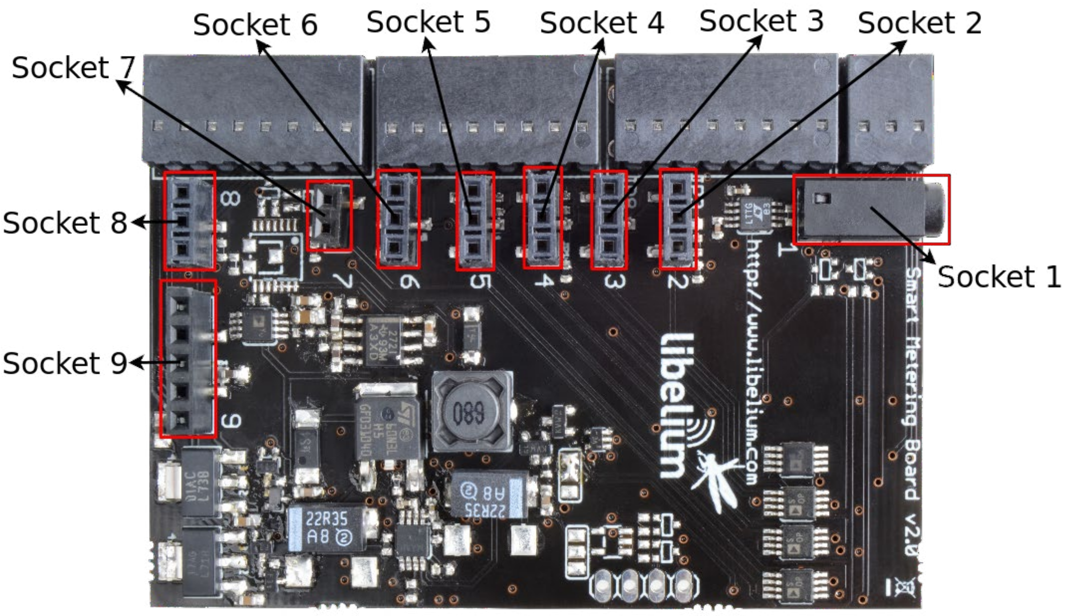


Figure 15: Upper side

3.3. Electrical Characteristics

- **Board Power Voltages:** 3.3V y 5V
- **Sensor Power Voltages:** 3.3V, 5V, 10V and 15V
- **Maximum admitted current (continuous):** 200mA
- **Maximum admitted current (peak):** 400mA

4. Sensors

4.1. Current Sensor

4.1.1. Specifications

Maximum primary current: 100A
Turns ratio: 1:2000 approximately
Minimum resolution: 130mA approximately
Measurement range: 500mA ~ 40A

Figure 16: Current clamp

4.1.2. Measurement Process

The current clamp is a low cost sensor that outputs a current proportional to the current in the primary circuit. That current (related with the primary current through a 1:2000 ratio) is converted into voltage through a load resistor obtaining a signal readable by the mote's analog-to-digital converter. The functions implemented to read the output value of the clamp, expressed in RMS current amperes, are described in section "API".

Below, a small sample of a code for reading the sensor is shown:

```
{
	SensorSmartv20.ON();
	SensorSmartv20.setSensorMode(SENS_ON, SENS_SMART_CURRENT);
	delay(50);
	float value_current;
	value_current = SensorSmartv20.readValue(SENS_SMART_CURRENT);
}
```

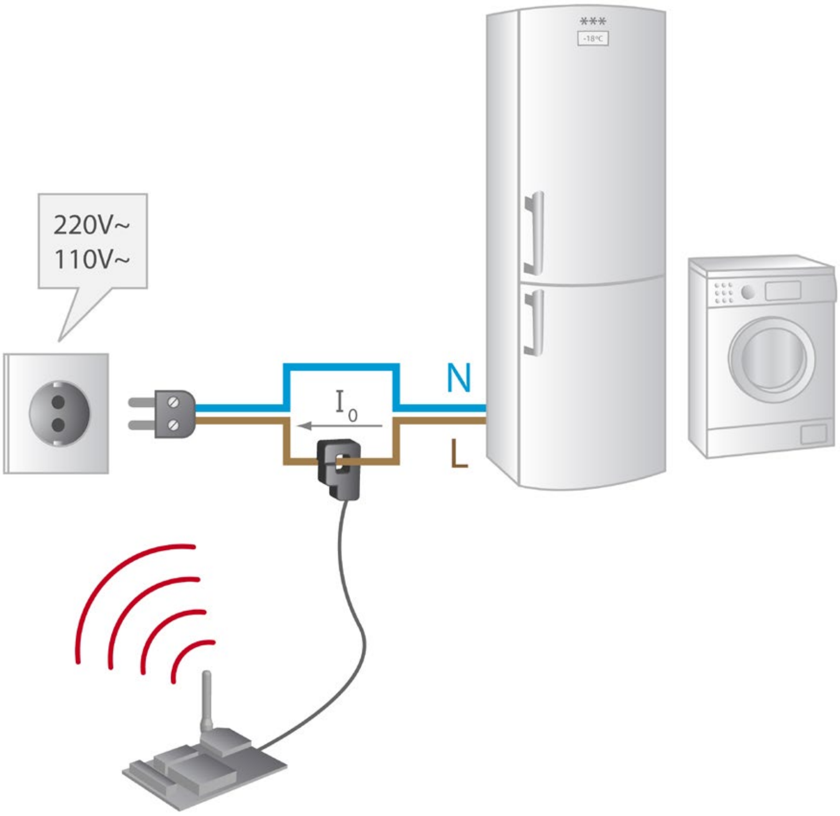


Figure 17: Example of application with the current clamp sensor

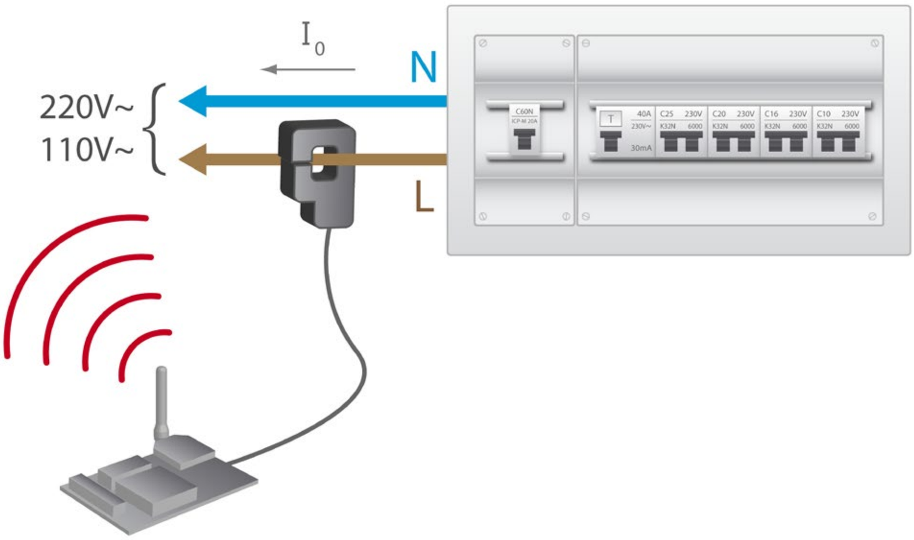


Figure 18: Example of application with the current clamp sensor

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

www.libelium.com/development/waspmote/examples/sm-4-current-sensor-reading

4.1.3. Socket

The clamp should be connected to the board through the power jack connector in socket 1.

We can see an image of socket 1 in figure 18.

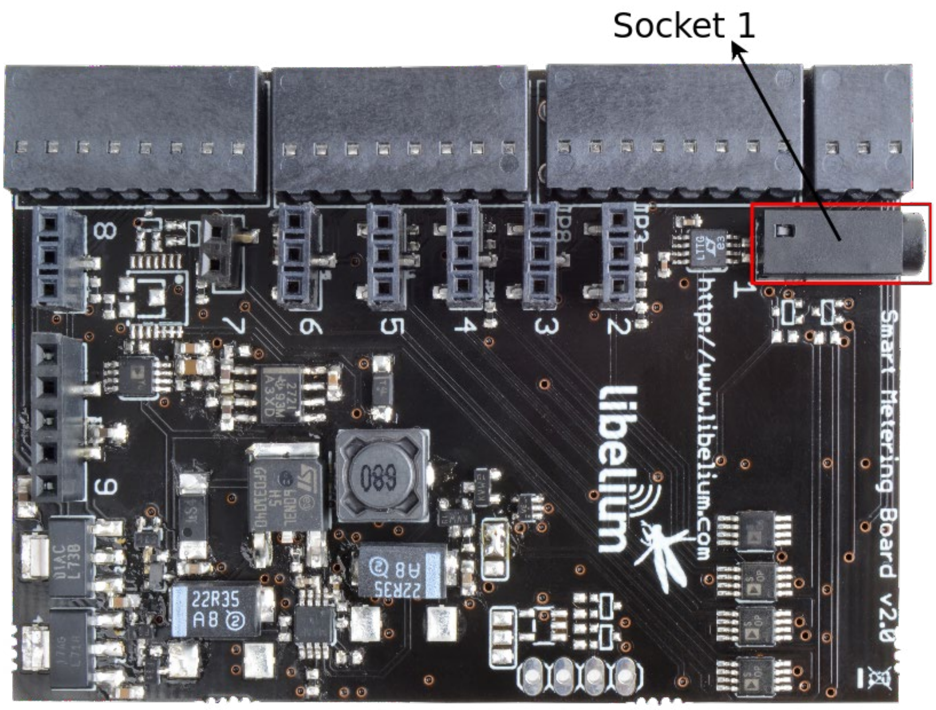


Figure 19: Picture of the Smart Metering 2.0 Board with socket 1 highlighted

4.2. Load Cell (AME, AMT and AMS from Hanyu)

4.2.1. Specifications

AMT:

Rate load: 300, 600, 1500, 3000g
Sensitivity: 2.0±0.1mv/V
Accuracy grade: 0.02%F.S
Nonlinearity: ±0.02%F.S
Recommended excitation voltage: +5V
Maximum excitation voltage: +15V
Operation temperature: -20ºC ~ +60ºC

Figure 20: Image of the load cell AMT

AME:

Rate load: 3, 5, 6, 8, 10, 15, 20, 30, 35, 40, 50kg
Output sensitivity: 2.0±0.1mv/V
Accuracy grade: 0.02%F.S
Nonlinearity: ±0.02%F.S
Recommended excitation voltage: +9V ~ +12V
Operation temperature: -20ºC ~ +60ºC
Protection class: IP-65

Figure 21: Image of the load cell AME

AMS:

Rate load: 50, 100, 150, 200, 250, 300, 500, 600kg
Sensitivity: 2.0±0.1mv/V
Accuracy grade: 0.02%F.S
Nonlinearity: ±0.02%F.S
Recommended excitation voltage: +9V ~ +12V
Operation temperature: -20ºC ~ +60ºC
Protection: IP-65

Figure 22: Image of the load cell AMS

4.2.2. Measurement Process

The load cells used in the Smart Metering 2.0 Board, included in the PRO version of the board, are single point cells with a Wheatstone bridge output in a full bridge configuration that requires connection to power supply, ground, positive output and negative output (red, black green and white wires respectively). The result is a differential voltage that is amplified and filtered to get an analog voltage proportional to the weight on the cell. In figure 23 the output voltage of the cell related to the load on it has been represented for three different models powered at 10V.

A sample code for reading the load cells is shown below:

```
{
 SensorSmartv20.ON();
 SensorSmartv20.setSensorMode(SENS_ON, SENS_SMART_LCELLS);
 delay(50);
 float value_lcell;
 value_lcell = SensorSmartv20.readValue(SENS_SMART_LCELLS);
}
```

You can find a complete example code for reading the load cells in the following links:

www.libelium.com/development/waspmote/examples/sm-11-5v-load-cell-reading

www.libelium.com/development/waspmote/examples/sm-12-10v-load-cell-reading

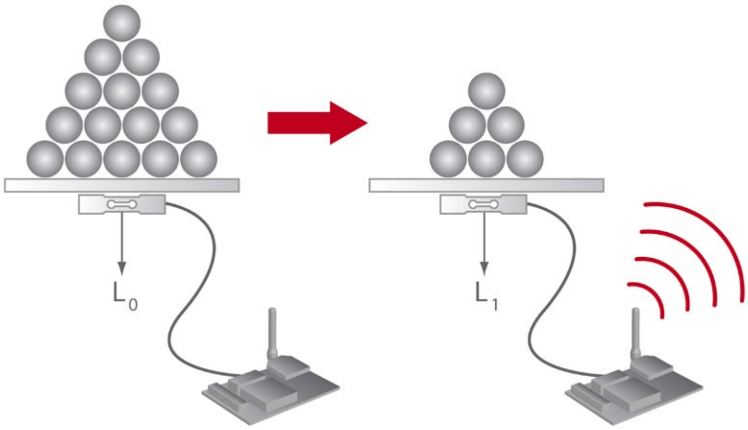


Figure 23: Example of application with the load cell

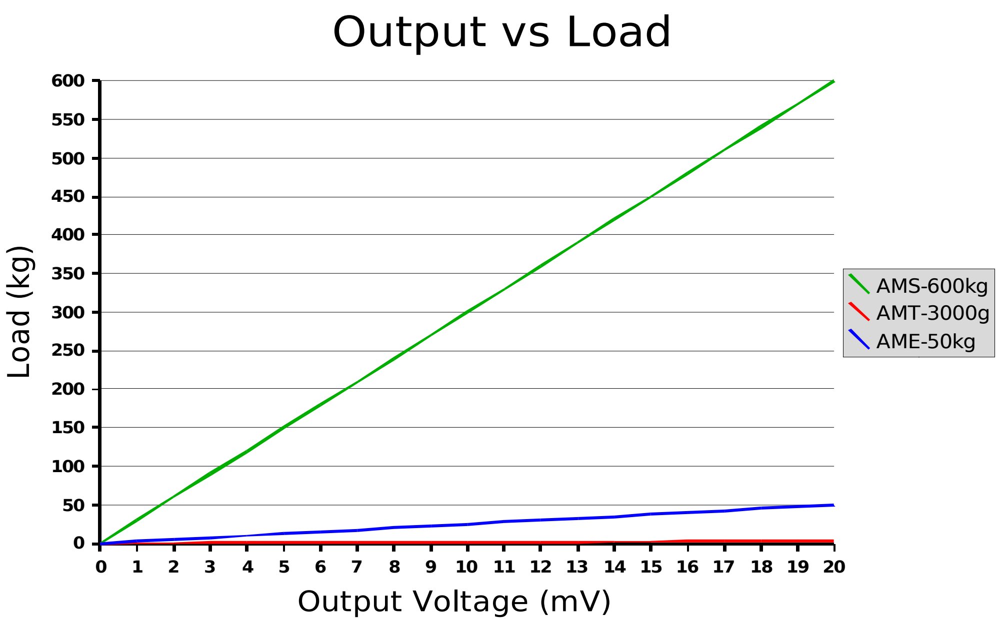


Figure 24: Graph of the output of three models of load cells

4.2.3. Socket

Any of the load cell models can be connected to socket 9 on the Smart Metering 2.0 Board. We can see in image 24 the pin correspondence between the input and output wires of the cell and the socket, in function of the supply voltage required.

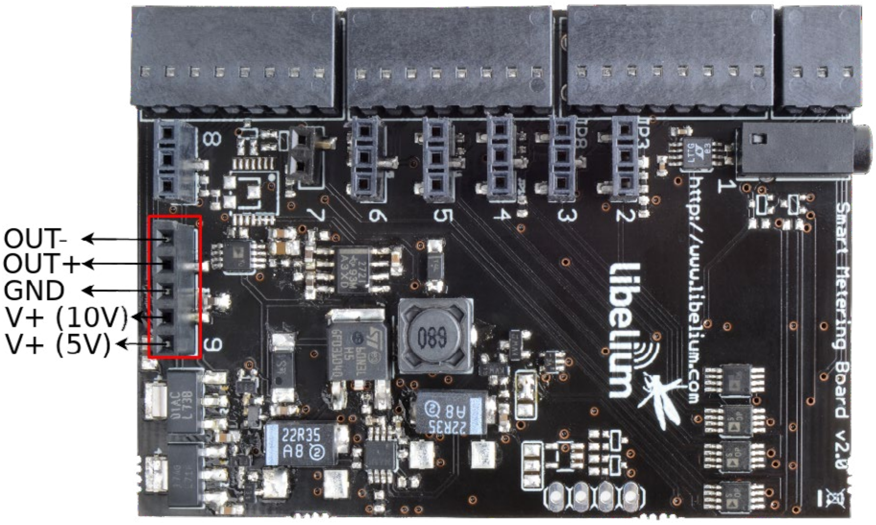


Figure 25: Image of socket 9 for load cells

Note: *The load cells can only be integrated in the Smart Metering PRO sensor board.*

4.3. Liquid Flow Sensor (FS100A, FS200A and FS400 from Broiltech)

4.3.1. Specifications

FS100A:

Flow rate: 0.15 ~ 2.5L/Min
Working voltage: +3.3V ~ +24V
Working temperature: -10ºC ~ 120ºC
Pulse number: 3900 pulses/liter
Inlet pipe size: 2mm
Outlet pipe size: 4mm
Accuracy: ±0.5%
Max rated current: 8mA

FS200A:

Flow rate: 0.5 ~ 25L/Min
Working voltage: +3.3V ~ +24V
Working temperature: -10ºC ~ 120ºC
Pulse number: 450 pulses/liter
Pipe connection: ½"
Accuracy: ±1%
Max rated current: 8mA

Figure 26: Image of the Liquid FS200A Flow sensor

FS400:

Flow rate: 1 ~ 60L/Min
Working voltage: +3.3V ~ +24V
Working temperature: -10ºC ~ 120ºC
Pulse number: 390 pulses/liter
Pipe connection: 1”
Accuracy: ±2%
Max rated current: 8mA

Figure 27: Image of the Liquid FS400 Flow sensor

4.3.2. Measurement Process

The liquid flow sensors included in the Smart Metering 2.0 Board output a signal that consists of a series of digital pulses whose frequency is proportional to the flow rate of the liquid through the sensor. That digital signal, whose frequency is in the range between 0Hz and 100Hz, is directly read through one of the digital input/output pins of the microcontroller.

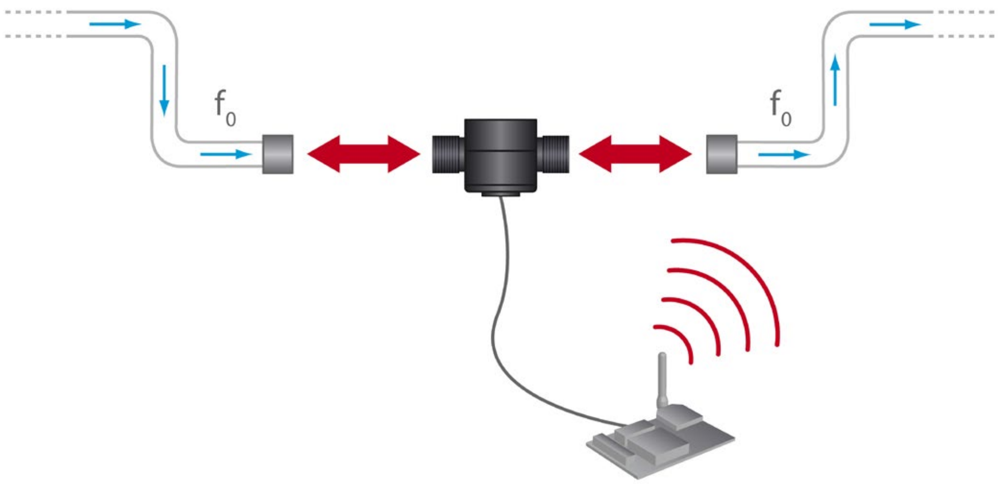


Figure 28: Example of application with the liquid flow sensor

The flow sensors may be connected on sockets 5, for a power supply of 3.3V, or 4, for a power supply of 5V.

Below, a small sample of a code for reading the flow sensors is shown:

```
{
  SensorSmartv20.ON();
  SensorSmartv20.setSensorMode(SENS_ON, SENS_SMART_FLOW);
  delay(50);
  float value_flow;
  value_flow = SensorSmartv20.readValue(SENS_SMART_FLOW, SENS_FLOW_FS100);
}
```

You can find a complete example code for reading the flow sensors in the following links:

www.libelium.com/development/waspmote/examples/sm-5-flow-sensor-on-3v3-socket-reading

www.libelium.com/development/waspmote/examples/sm-6-flow-sensor-on-5v-socket-reading

4.3.3. Socket

Flow sensors can be connected to sockets 5, for a 3.3V power supply, and 4, for a 5V power supply, as shown in figure 28.

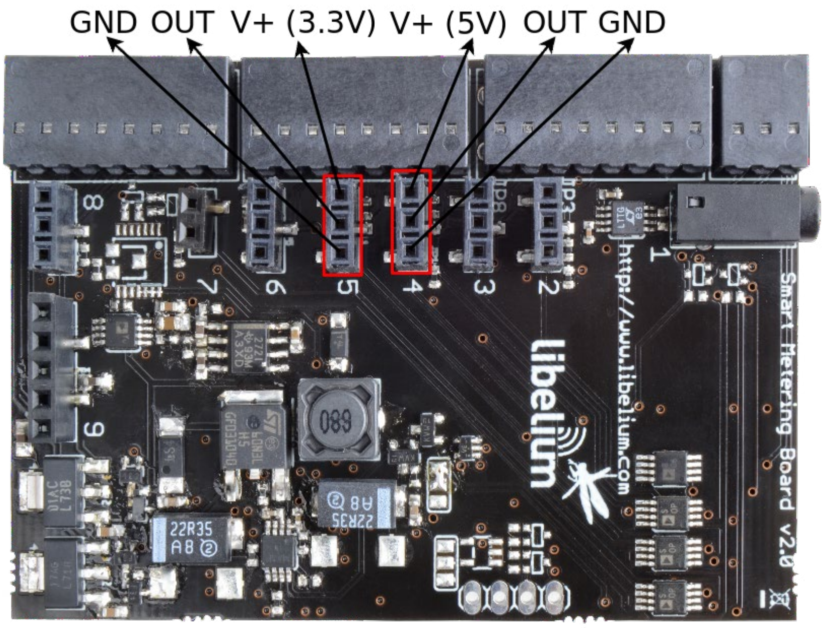


Figure 29: Picture of the sockets for the flow sensors

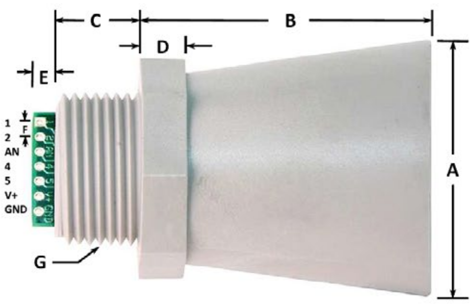
4.4. Ultrasonic Sensor (MaxSonar® from MaxBotix™)

4.4.1. Specifications

Figure 30: Ultrasonic XL-MaxSonar®-WRA1 from MaxBotix™ sensor

<u>**XL-MaxSonar®-WRA1™**</u>

Operation frequency: 42kHz
Maximum detection distance: 765cm
Maximum detection distance (analog output): 600cm (powered at 3.3V) - 700cm (powered at 5V)
Sensitivity (analog output): 3.2mV/cm (powered at 3.3V) – 4.9mV/cm (powered at 5V)
Power supply: 3.3 ~ 5V
Consumption (average): 2.1mA (powered at 3.3V) – 3.2mA (powered at 5V)
Consumption (peak): 50mA (powered at 3.3V) – 100mA (powered at 5V)
Usage: Indoors and outdoors (IP-67)



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 mm dia.
I	1.37"	34.8 mm
weight, 1.76 oz., 50 grams		

values are nominal

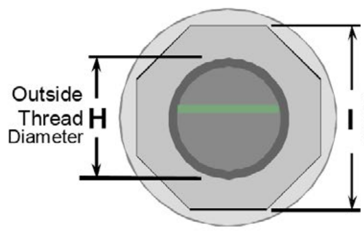
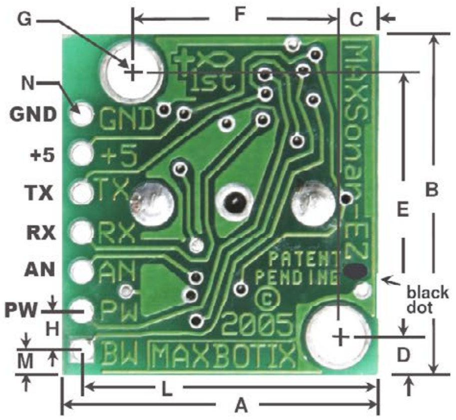


Figure 31: Ultrasonic XL-MaxSonar®-WRA1 sensor dimensions

LV-MaxSonar®-EZ0™

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

Figure 32: Ultrasonic LV-MaxSonar®-EZ0 from MaxBotix™ sensor



A	0.785"	19.9 mm
B	0.870"	22.1 mm
C	0.100"	2.54 mm
D	0.100"	2.54 mm
E	0.670"	17.0 mm
F	0.510"	12.6 mm
G	0.124" dia.	3.1 mm dia.

H	0.100"	2.54 mm
J	0.645"	16.4 mm
K	0.610"	15.5 mm
L	0.735"	18.7 mm
M	0.065"	1.7 mm
N	0.038" dia.	1.0 mm dia.
weight, 4.3 grams		

values are nominal

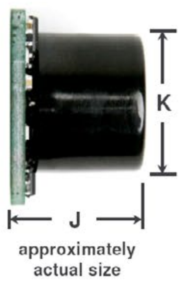


Figure 33: Ultrasonic LV-MaxSonar®-EZ0 sensor dimensions

4.4.2. Measurement Process

The MaxSonar® sensors from MaxBotix output an analog voltage proportional to the distance to the object detected. This sensor can be powered at both 3.3V or 5V, although the detection range will be wider for the last one.

A sample of code for reading both sensors is shown below, with the XL-MaxSonar®-WRA1™ on socket 3 and the LV-MaxSonar®-EZ0™ on socket 2:

```
{
 SensorSmartv20.ON();
 SensorSmartv20.setSensorMode(SENS_ON, SENS_SMART_US_3V3);
 delay(2000);
 float value_ultrasound_wra1;
 float value_ultrasound_ez0;
 value_ultrasound_wra1 = SensorSmartv20.readValue(SENS_SMART_US_3V3, SENS_US_WRA1);
 value_ultrasound_ez0 = SensorSmartv20.readValue(SENS_SMART_US_5V, SENS_US_EZ0);
}
```

You can find a complete example code for reading the ultrasound sensors in the following links:

www.libelium.com/development/waspmote/examples/sm-7-ultrasound-sensor-on-3v3-socket-reading

www.libelium.com/development/waspmote/examples/sm-8-ultrasound-sensor-on-5v-socket-reading

In figure 33 we can see a drawing of two example applications for the ultrasonic sensors, such as liquid level monitoring or presence detection.

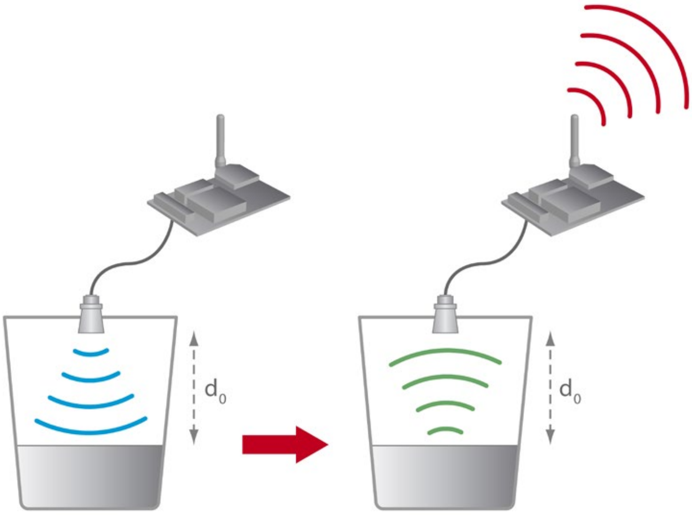


Figure 34: Example of application for the MaxSonar® sensors

The XL-MaxSonar®-WRA1™ sensor is endowed with an IP-67 casing, so it can be used in outdoors applications, such as liquid level monitoring in storage tanks.

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

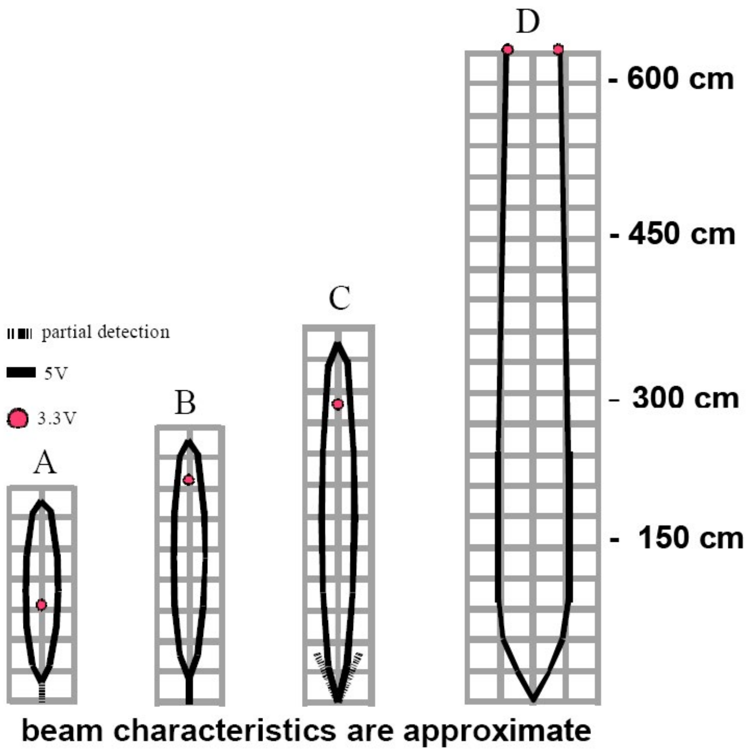


Figure 35: Diagram of the sensor beam extracted from the data sheet of the XL-MaxSonar®-WRA1™ sensor from MaxBotix

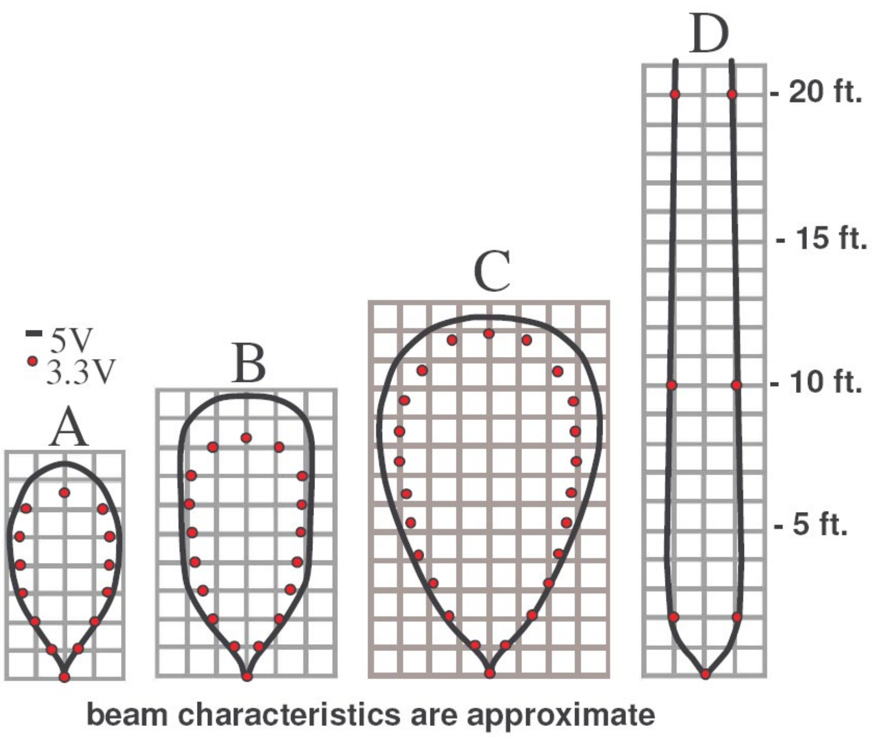


Figure 36: Diagram of the sensor beam extracted from the data sheet of the LV-MaxSonar®-EZ0™ sensor from MaxBotix

4.4.3. Socket

Since this sensor may be powered at 3.3V or 5V, it can be placed on any of the sockets 2 and 8 (5V) or 3 and 6 (3,3V). In figure 36 sockets 2 and 3 are shown.

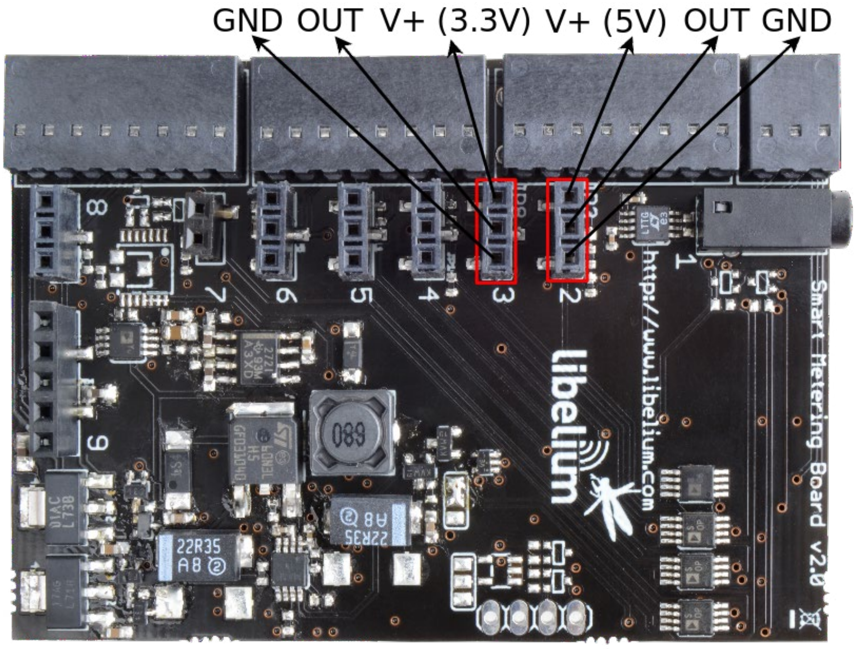


Figure 37: Picture of sockets for ultrasonic sensors

4.5. Humidity Sensor (808H5V5)

4.5.1. Specifications

Measurement range: 0 ~ 100%RH
Output signal: 0,8 ~ 3.9V (25ºC)
Accuracy: <±4%RH (a 25ºC, range 30 ~ 80%), <±6%RH (range 0 ~ 100)
Typical consumption: 0.38mA
Maximum consumption: 0.5mA
Power supply: 5VDC ±5%
Operation temperature: -40 ~ +85ºC
Storage temperature: -55 ~ +125ºC
Response time: <15 seconds

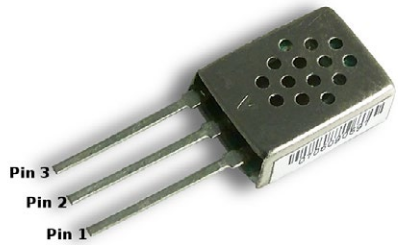


Figure 38: Image of the 808H5V5 sensor

4.5.2. Measurement Process

This is an analog sensor which provides a voltage output proportional to the relative humidity in the atmosphere. As the sensor's signal is outside of that permitted at the input of the microcontroller's analog-to-digital converter, it's output voltage has been adapted to a range of values between 0.48V and 2.34V.

A sample of code for reading this sensor is shown below:

```
{
  SensorSmartv20.ON();
  SensorSmartv20.setSensorMode(SENS_ON, SENS_SMART_HUMIDITY);
  delay(15000);
  float value_humidity;
  value_humidity= SensorSmartv20.readValue(SENS_SMART_HUMIDITY);
}
```

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

www.libelium.com/development/waspmote/examples/sm-2-humidity-sensor-reading

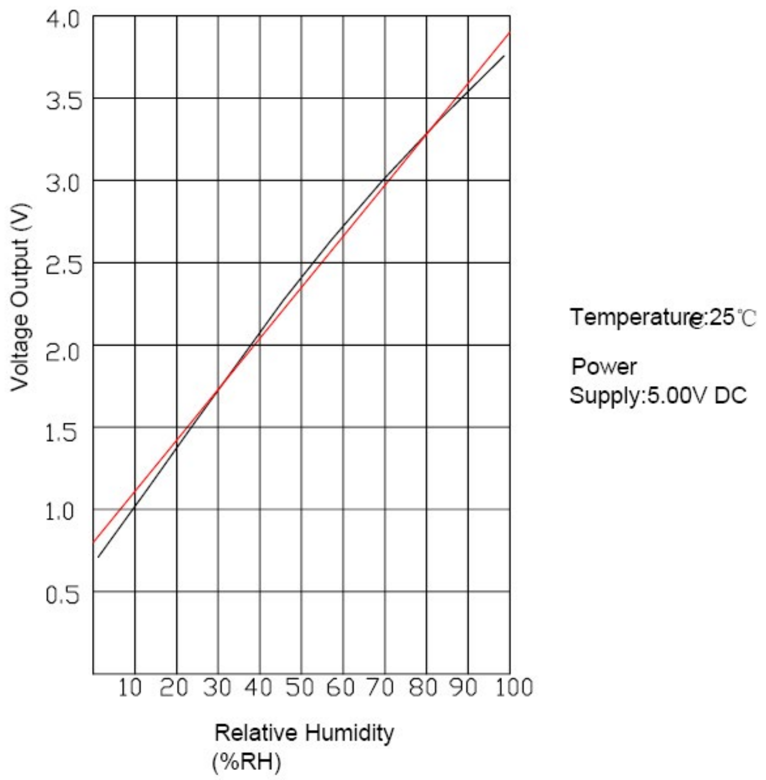


Figure 39: 808H5V5 humidity sensor output taken from the Sencera Co. Ltd sensor data sheet

4.5.3. Socket

This sensor may be connected to any of the sockets prepared for analog sensors powered at 5V, i.e. sockets2 and 8, shown in figure 39, where the function of each pin has been indicated.

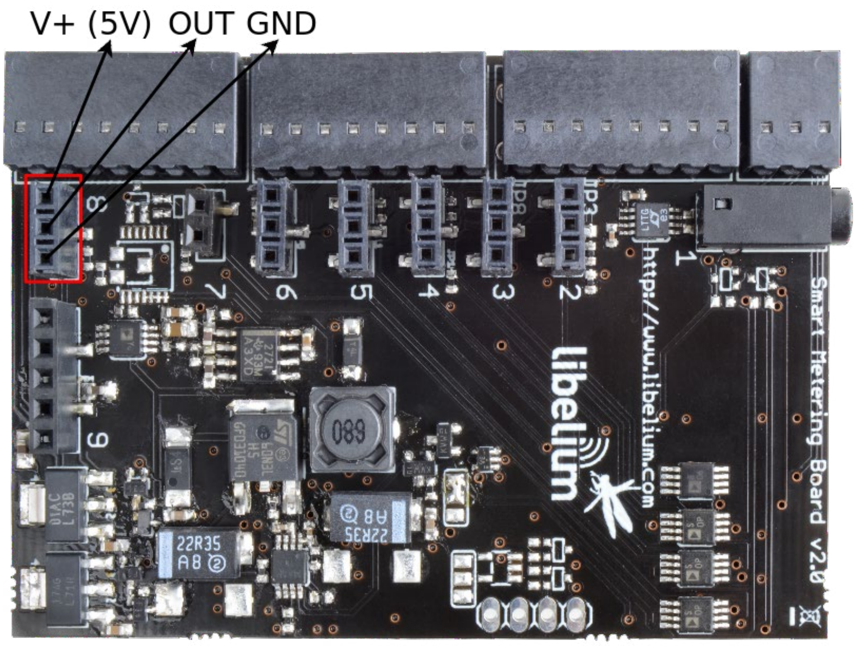


Figure 40: Picture of the sockets for the humidity sensor

4.6. Temperature Sensor (MCP9700A)

4.6.1. Specifications

Measurement range: -40ºC ~ +125ºC
Output voltage (0ºC): 500mV
Sensitivity: 10mV/ºC
Accuracy: ±2ºC (range 0ºC ~ +70ºC), ±4ºC (range -40 ~ +125ºC)
Typical consumption: 6μA
Maximum consumption: 12μA
Power supply: 2.3 ~ 5.5V
Operation temperature: -40 ~ +125ºC
Storage temperature: -65 ~ 150ºC
Response time: 1.65 seconds (63% of the response for a range from +30 to +125ºC)

Figure 41: Image of the MCP9700A temperature sensor

4.6.2. Measurement Process

The MCP9700A is an analog sensor which converts a temperature value into a proportional analog voltage. The range of output voltages is between 100mV (-40ºC) and 1.75V (125ºC), resulting in a variation of 10mV/Cº, with 500mV of output for 0ºC. The voltage output may be directly captured by the analog-to-digital converter of the microcontroller.

Below we show a code for reading the temperature sensor:

```
{
  SensorSmartv20.ON();
  SensorSmartv20.setSensorMode(SENS_ON, SENS_SMART_TEMPERATURE);
  delay(50);
  float value_temperature;
  value_temperature = SensorSmartv20.readValue(SENS_SMART_TEMPERATURE);
}
```

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

www.libelium.com/development/waspmote/examples/sm-1-temperature-sensor-reading

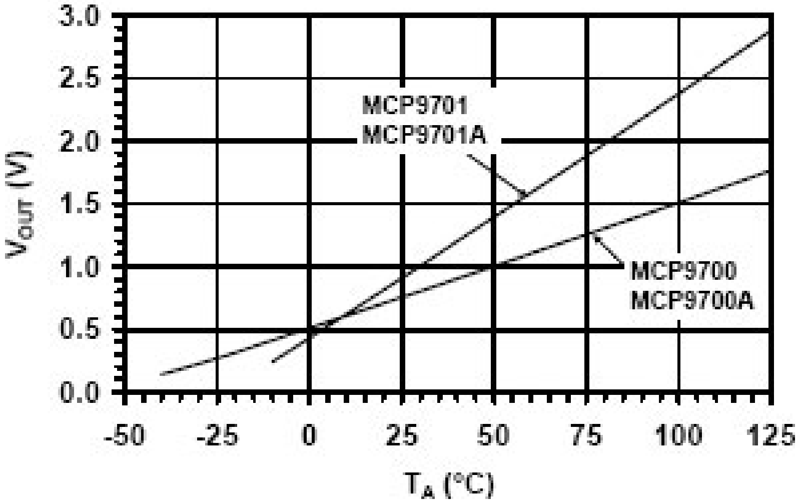


Figure 42: Graph of the MCP9700A sensor output voltage with respect to temperature, taken from the Microchip sensor's data sheet

4.6.3. Socket

Placing the MCP9700A sensor on socket 6 is recommended. Even though, since the sensor may be powered with 3.3V or 5V, it can be placed on any of the sockets for analog sensors. We have an image of socket 6 in figure 42, where the function of each pin has been indicated.

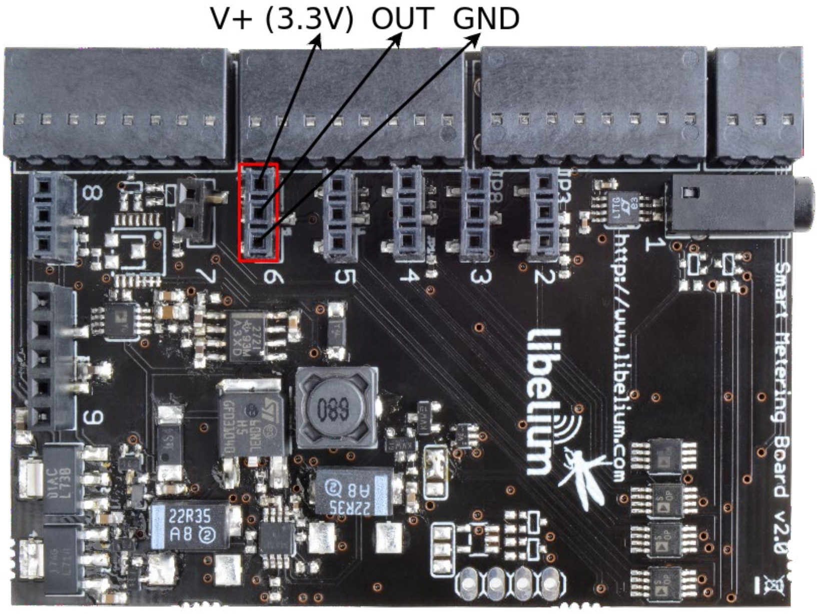


Figure 43: Picture of the sockets for the temperature sensor

4.7. Luminosity Sensor (LDR)

4.7.1. Especificaciones

Resistance in darkness: 20MΩ

Resistance in light(10lux): 5 ~ 20kΩ

Spectral range: 400 ~ 700nm

Operating temperature: -30ºC ~ +75ºC

Minimum consumption: 0μA approximately

Figure 44: Image of the LDR luminosity sensor

4.7.2. Measurement Process

This is a resistive sensor whose conductivity varies depending on the intensity of light received on its photosensitive part. The measurement of the sensor is carried out through the analog-to-digital converter of the microcontroller, reading the resulting voltage out of a voltage divider formed by the sensor itself and the load resistor of the socket upon which it has been connected.

The measurable spectral range (400nm – 700nm) coincides with the human visible spectrum so it can be used to detect light/ darkness in the same way that a human eye would detect it.

A sample code for reading this sensor is shown below:

```
{
 SensorSmartv20.ON();
 SensorSmartv20.setSensorMode(SENS_ON, SENS_SMART_LDR);
 delay(50);
 float value_ldr;
 value_ldr= SensorSmartv20.readValue(SENS_SMART_LDR);
}
```

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

www.libelium.com/development/waspmote/examples/sm-3-ldr-sensor-reading

4.7.3. Socket

This sensor has been thought to be placed on socket 7, specifically configured for it.

We have an image of the sockets in figure 44, where the function of each pin has been indicated.

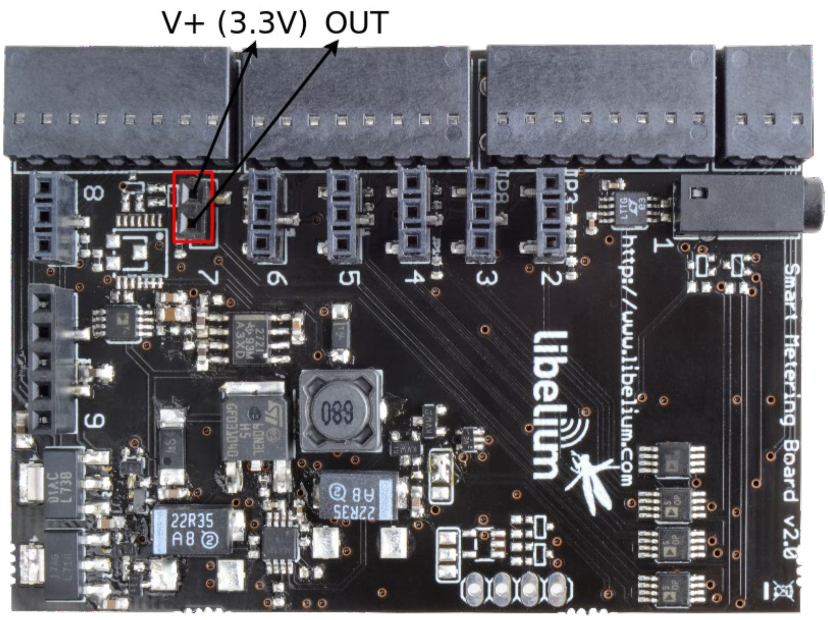


Figure 45: Picture of the LDR socket

4.8. Displacement Foil Sensor (MTP sensor from Metallux and MagnetoPots from Spectra Symbol)

4.8.1. Specifications

Figure 46: Image of the displacement sensor

Length: 200mm
Resistance range: 0 ~ 10kΩ
Minimum resolution: 1mm approximately
Consumption: 0.33mA (socket 3) ~ 1.6mA (socket 2)

4.8.2. Measurement Process

The displacement sensors MagnetoPot from Spectra Symbol and MTP from Metallux are two potentiometers whose resistance changes in function of the position of a magnet (case of the MagnetoPot sensor) or the pressure exerted along its surface, so the value of its output voltage, that can be read through the analog-to-digital converter of the microcontroller, varies between 0V and the supply voltage.

A sample code for reading this sensors on socket 3 is shown below:

```
{
  SensorSmartv20.ON();
  SensorSmartv20.setSensorMode(SENS_ON, SENS_SMART_LD_3V3);
  delay(50);
  float value_ld;
  value_ld = SensorSmartv20.readValue(SENS_SMART_LD_3V3);
}
```

You can find a complete example code for reading the displacement foil sensors in the following link:

www.libelium.com/development/waspmote/examples/sm-9-linear-displacement-sensor-on-3v3-socket-reading

www.libelium.com/development/waspmote/examples/sm-10-linear-displacement-sensor-on-5v-socket-reading

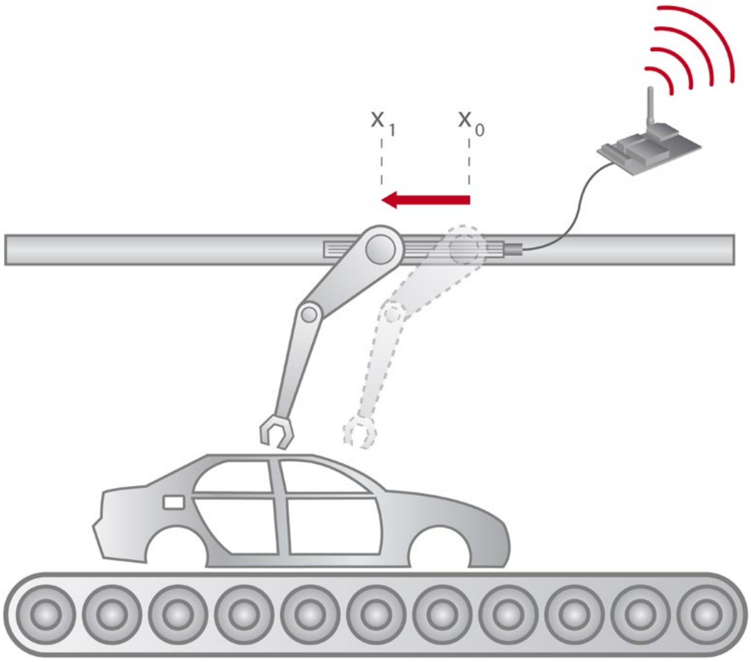


Figure 47: Example of application for the displacement foil sensors

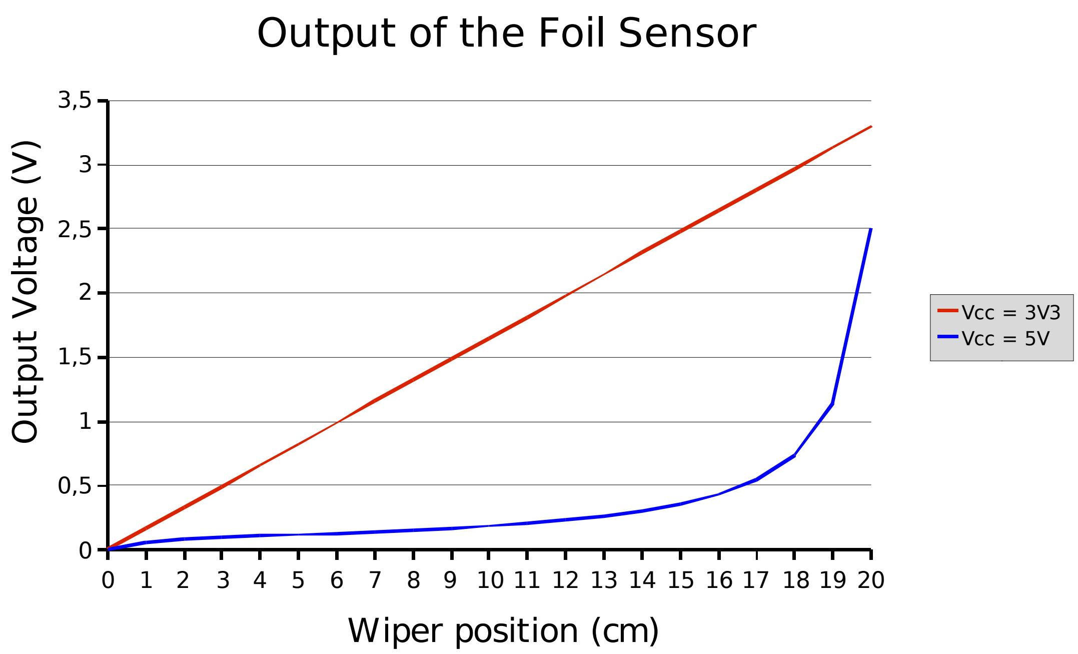


Figure 48: Graph of the output of the sensor for the 3.3V sockets (3 and 6) and the 5V sockets (2 and 8)

4.8.3. Socket

Like the ultrasonic sensors, both kind of sensors can be placed on any of the sockets 2 and 8 (5V) or 3 and 6 (3,3V). In figure 48 sockets 2 and 3 are shown.

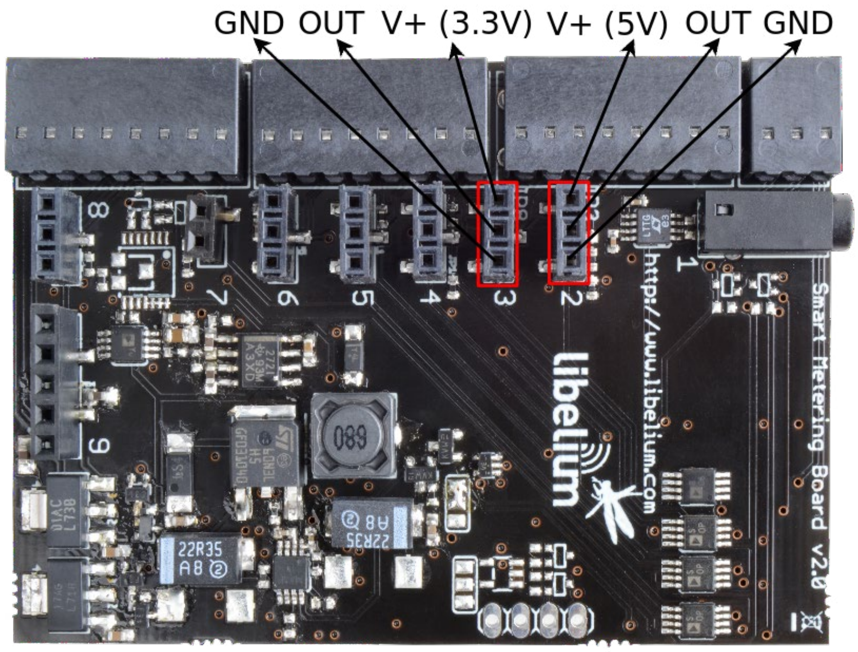


Figure 49: Picture of sockets for displacement foil sensors

4.9. Sockets for casing

In case the Smart Metering 2.0 board is going to be used in an application that requires the use of a casing, such as an outdoors application, a series of sockets to facilitate the connection of the sensors through a probe has been disposed.

These sockets (PTSM from Phoenix Contact) allow to assemble the wires of the probe simply by pressing them into it. To remove the wire press the slot the input pin and pull off the wire softly.

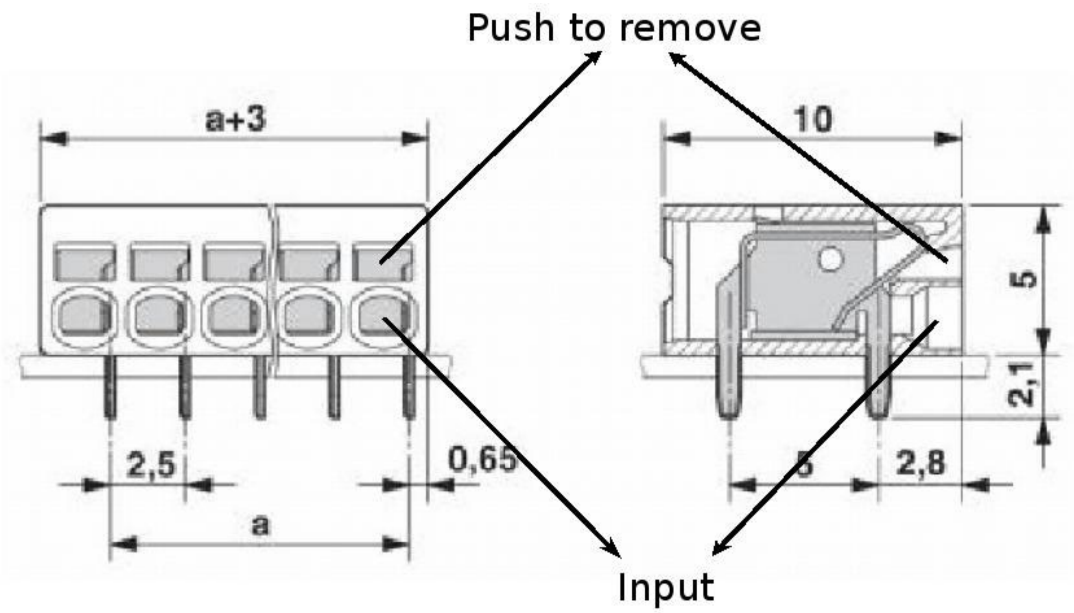


Figure 50: Diagram of a socket extracted from the Phoenix Contact data sheet

In the figure below an image of the board with the sockets in it and the correspondence between its inputs and the sensor's pins is shown.

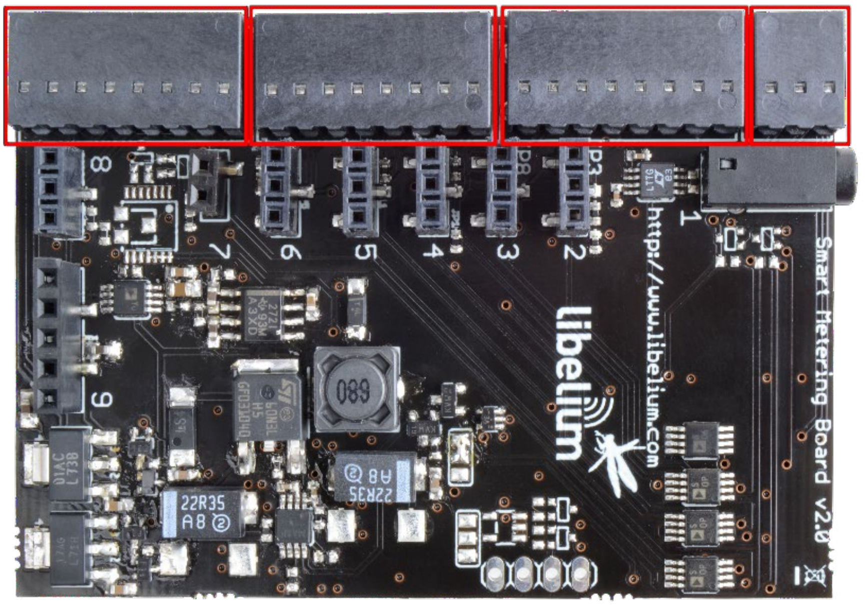


Figure 51: Image of the sockets for casing applications

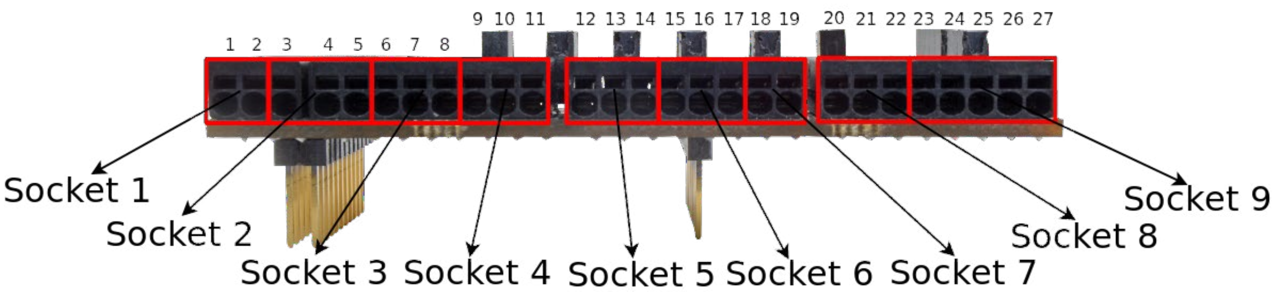


Figure 52: Image of the pin correspondence between the sockets and the sensors

Sensor	Pin	Function
Socket 1	1	Input 1 (no polarity)
	2	Input 2 (no polarity)
Socket 2	3	GND
	4	Output
	5	Vcc (5V)
Socket 3	6	GND
	7	Vcc (3.3V)
	8	Output
Socket 4	9	GND
	10	Output
	11	Vcc (5V)
Socket 5	12	GND
	13	Output
	14	Vcc (3.3V)
Socket 6	15	GND
	16	Output
	17	Vcc (3.3V)
Socket 7	18	Vcc (3.3V)
	19	Output
Socket 8	20	GND
	21	Output
	22	Vcc (5V)
Socket 9	23	Output -
	24	Output +
	25	GND
	26	Vcc (10V)
	27	Vcc (5V)

5. Board configuration and programming

5.1. Hardware configuration

The only hardware configuration needed by the user is that related to the correct sensor connection. The right way to connect the sensors to their respective socket is shown in sections "Socket", dedicated to described each of the sockets of the board.

5.2. API

When using this board remember it is mandatory to include the SensorSmartv20 library by introducing the next line at the beginning of the code:

```
#include <WaspSensorSmart_v20.h>
```

The functions to handle all the features of the board, included in the WaspSensorSmart_v20 API library, are detailed below:

SensorSmartv20.ON()

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

SensorSmartv20.OFF()

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

SensorSmartv20.setBoardMode(MODE)

This function is used to manage the power supply applied to the Smart Metering 2.0 board. Assigning the value SENS_ON to the variable MODE activates the Waspmote's switches which allow the passage of the 3.3V and 5V voltages, while assigning SENS_OFF disconnects both switches cutting the power.

SensorSmartv20.setSensorMode(MODE, SENSOR)

This function, similar to setBoardMode, allows to activate or deactivate the power of each sensor independently.

The state on which the sensor should be set is defined through the variable MODE, which can take the values SENS_ON, to connect the power of the sensor, or SENS_OFF, to disconnect it.

The sensor, circuit or group of sensors that we are going to manage is stored in the variable SENSOR, that can take the following values:

- SENS_SMART_US_3V3, to activate the ultrasound sensor on socket 3.
- SENS_SMART_US_5V, to activate the ultrasound sensor on socket 2.
- SENS_SMART_FLOW_3V3, to activate the flow sensor on socket 5.
- SENS_SMART_FLOW_5V, to activate the flow sensor on socket 4.
- SENS_SMART_DFS_3V3, to activate the displacement foil sensor on socket 3.
- SENS_SMART_DFS_5V, to activate the displacement foil sensor on socket 2.
- SENS_SMART_LCELLS_10V, to activate the load cells' 10V supply line.
- SENS_SMART_LCELLS_5V, to activate the load cells' 5V supply line.
- SENS_SMART_CURRENT, to activate the current sensor.
- SENS_SMART_TEMPERATURE, to activate the temperature sensor on socket 6.
- SENS_SMART_HUMIDITY, to activate the humidity sensor on socket 8.
- SENS_SMART_LDR, to read the activate sensor on socket 7.

As said in section "Power control", the LDR, TEMPERATURE and HUMIDITY sensors, the two FLOW sensors and the two ultrasound sensors are powered through the same switch, so when turning on or off one of them you will be acting on the whole group.

SensorSmartv20.readValue(SENSOR, TYPE)

The function readValue may be used to execute the configuration, conversion and reading process of any of the sensors on the board through the analog-to-digital converter. In the variable SENSOR the sensor to be read is introduced. The values that can be assigned to this variable are:

- SENS_SMART_US_3V3, to read the ultrasound sensor on socket 3.
- SENS_SMART_US_5V, to read the ultrasound sensor on socket 2.
- SENS_SMART_FLOW_3V3, to read the flow sensor on socket 5.
- SENS_SMART_FLOW_5V, to read the flow sensor on socket 4.
- SENS_SMART_LCELLS_10V, to read the load cells.
- SENS_SMART_LCELLS_5V, to read the load cells.
- SENS_SMART_DFS_3V3, to read the displacement foil sensor on socket 3.
- SENS_SMART_DFS_5V, to read the displacement foil sensor on socket 2.
- SENS_SMART_CURRENT, to read the current sensor.
- SENS_SMART_TEMPERATURE, to read the temperature sensor on socket 6.
- SENS_SMART_HUMIDITY, to read the humidity sensor on socket 8.
- SENS_SMART_LDR, to read the luminosity sensor on socket 7.

The TYPE parameter, used when reading ultrasound or flow sensors, indicates the model of the sensor to be read. It may take the values:

- SENS_US_WRA1, to refer to the XL-MaxSonar®-WRA1™ ultrasound sensor.
- SENS_US_EZ0, to refer to the LV-MaxSonar®-EZ0™ ultrasound sensor.
- SENS_FLOW_FS100, to refer to the FS100A flow sensor.
- SENS_FLOW_FS200, to refer to the FS200A flow sensor.
- SENS_FLOW_FS400, to refer to the FS400 flow sensor.

A basic program to detect events from the board will present a similar structure to the following, subject to changes in dependence of the application:

1. The board is switched on using the function **SensorSmartv20.setBoardMode**.

2. Turn on those inactive sensors to be read using function **SensorSmartv20.setSensorMode**.

3. Take the measurements needed using function **SensorSmartv20.readValue**.

4. Turn off the sensors with function **SensorSmartv20.setSensorMode**.

Store or send via a radio module the gathered information.

Put the mote to sleep with the functions **PWR.sleep** or **PWR.deepSleep**.

In the code below a complete example to read the temperature, current and liquid flow and transmit the data through the XBee 802.15.4 module is given:

```
/* ------------Smart Metering v20 board example---------------

   www.Libelium.com
*/

// Inclusion of the Smart Metering v20 Board library
#include <WaspSensorSmart_v20.h>

// Inclusion of the Frame library
#include <WaspFrame.h>

// Inclusion of the XBee 802.15.4 library
#include <WaspXBee802.h>

float current_value = 0;
float temperature_value = 0;
float flow_value = 0;

// Pointer to an XBee packet structure
packetXBee* packet;

void setup()
{


}

void loop()
{

  //Switch on the board
  SensorSmartv20.ON();
  delay(100);
  // Init RTC
  RTC.ON();
  delay(100);

  //Turn on the current sensor
  SensorSmartv20.setSensorMode(SENS_ON, SENS_SMART_CURRENT);
  delay(50);
  //Read the current value from the Current sensor
  current_value = SensorSmartv20.readValue(SENS_SMART_CURRENT);
  //Turn off the current sensor
  SensorSmartv20.setSensorMode(SENS_OFF, SENS_SMART_CURRENT);

  //Turn on the temperature sensor on socket 6
  SensorSmartv20.setSensorMode(SENS_ON, SENS_SMART_TEMPERATURE);
  delay(50);
  //Read the temperature value from the MCP9700A sensor
  temperature_value = SensorSmartv20.readValue(SENS_SMART_TEMPERATURE);
  //Turn off the temperature sensor on socket 6
  SensorSmartv20.setSensorMode(SENS_OFF, SENS_SMART_TEMPERATURE);

  //Turn on the flow sensor on socket 5
  SensorSmartv20.setSensorMode(SENS_ON, SENS_SMART_FLOW_3V3);
  delay(50);
  //Read the flow value from the FS100A sensor
  flow_value = SensorSmartv20.readValue(SENS_SMART_FLOW_3V3, SENS_FLOW_FS100);
  //Turn off the flow sensor on socket 5
  SensorSmartv20.setSensorMode(SENS_OFF, SENS_SMART_FLOW_3V3);
```

```
    // Create new frame (ASCII)
    frame.createFrame(ASCII,"Waspmote_Pro");
    // Add the values read to the frame composition
    frame.addSensor(SENSOR_TCA, temperature_value);
    frame.addSensor(SENSOR_CU, current_value);
    frame.addSensor(SENSOR_WF, flow_value);


    // Init XBee
    xbee802.ON();
    // Set parameters to packet:
    packet=(packetXBee*) calloc(1,sizeof(packetXBee));
    packet->mode=BROADCAST;

    // Set destination XBee parameters to packet
    xbee802.setDestinationParams( packet, "000000000000FFFF", frame.buffer, frame.length);

    // Send XBee packet
    xbee802.sendXBee(packet);

    // Turn off the XBee Module
    xbee802.OFF();
    delay(100);

    //Put the mote to sleep
    PWR.deepSleep("00:00:05:00", RTC_OFFSET, RTC_ALM1_MODE1, ALL_OFF);

  }
```

6. Consumption

6.1. Power control

The Smart Metering 2.0 Board for Waspmote requires both the 3.3V and 5V power supplies output from the mote. In the PRO version of the board a DC-DC converter is used to get the 15V voltage needed by the 5V and 10V voltage references for load cells.

A combination of transistors and solid state switches is used to control independently the power supply of the components in the board and the sensors connected to it:

First of all, the load cells, that have to be powered at 15V are controlled independently through a series of MOSFET transistors that may handle higher currents and voltages than those permitted at the solid state switches. This way, the 10V and 5V voltage references' supply can be controlled through signals `DIGITAL3` and `DIGITAL4` respectively. The activation of any of those signals will also activate the DC-DC converter.

The remaining sensors are controlled using solid state switches. The supply voltage of the electronics used to adapt the current clamp is controlled through signal `DIGITAL5`, the humidity, luminosity and temperature sensors (sockets 6, 7 and 8), that require both 5V and 3.3V supplies, are controlled via signal `DIGITAL8`, ultrasound sensors (sockets 2 and 3) are both controlled via signal `DIGITAL2` and flow sensors (sockets 4 and 5) via signal `DIGITAL6`.

All these switches may be controlled through the `SensorSmartv20.setBoardMode` and `SensorSmartv20.setSensorMode` functions implemented in the API. You can find more information about it in section "API".

6.2. Tables of consumption

In the following table the consumption of the board is shown, the constant minimum consumption (fixed by the permanently active components) and the consumption of each of the independent blocks that may be powered independently (in the case of the Humidity, temperature and LDR block, the flow sensors block and the ultrasound sensors block, the consumption of the sensors connected must be summed to the consumption of the electronics). The board's power can be completely disconnected, reducing the consumption to zero, using the 3.3V and the 5V main switches disconnection `SensorSmartv20.setBoardMode` command included in the library.

	Consumption
Minimum (Constant)	5µA
Humidity, Temperature and LDR sensors	15µA
Flow Sensors (3.3V and 5V)	30µA
Ultrasonic sensors (3.3V and 5V)	15µA
Current sensor	135µA
Load Cell – 10V (only PRO version)	150mA
Load Cell – 5V (only PRO version)	80mA

6.3. Low consumption mode

The Smart Metering 2.0 Board has been designed to minimize the consumption of the mote in operation conditions as long as in low consumption modes.

- **Avoid activating all the sensors at the same time**
  Owing to the high consumption of some of the sensors of the Smart Metering 2.0 Board, it is highly advisable, with the purpose of avoiding current peaks that exceed the maximum supported by the switches, to avoid turning on many sensors at the same time, specially the load cells, that require the 15V DC-DC converter.
- **Use the Waspmote low consumption mode**
  As the other sensor boards for Waspmote, the library of the Smart Metering 2.0 Board includes all the functions needed to deactivate the sensors which are not being used and put the mote in low consumption mode.
- **Do not connect sensors that are not going to be used**
  Since several sensors share the same power line, a sensor that is not going to be used connected to the board will entail an additional consumption, and so a shorter life of the battery.

7. API Changelog

Function / File	Changelog	Version
`#include`	Remember to include the WaspSensorSmart_v20 library in the top of your pde	v.31 → v1.0
`SensorSmartv20.ON()`	New function to turn on the board	v.31 → v1.0
`SensorSmartv20.OFF()`	New function to turn off the board	v.31 → v1.0
`SENS_SMART_LCELLS`	Variable SENS_SMART_LCELLS disappears and is replace by SENS_SMART_LCELLS_5V and SENS_SMART_LCELLS_10V, now it will not be necessary to modify the API file to select which kind of load cell is held	v.31 → v1.0

8. Documentation changelog

Added references to 3G/GPRS Board in section: Radio Interfaces

9. 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.

10. Disposal and recycling

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