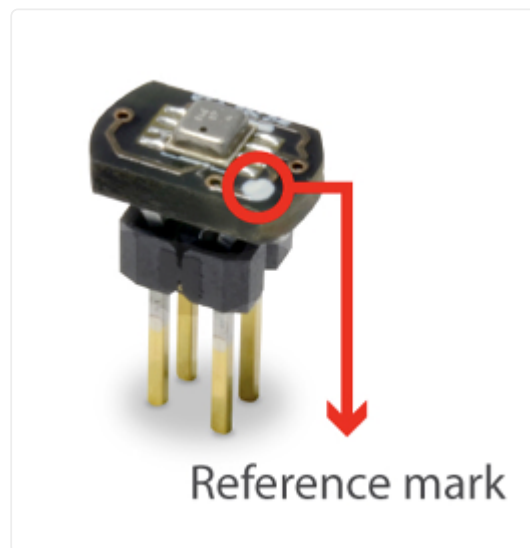


Sensors

Temperature, humidity and pressure sensor (BME280)



i Only one BME280 sensor is supported at the same time.

Specifications

Electrical characteristics

- Supply voltage: 3.3 V
- Sleep current typical: 0.1 μ A
- Sleep current maximum: 0.3 μ A

Temperature sensor

- Operational range: -40 ~ +85 °C
- Full accuracy range: 0 ~ +65 °C
- Accuracy: ± 1 °C (range 0 °C ~ +65 °C)
- Response time: 1.65 seconds (63% response from +30 to +125 °C).

- Typical consumption: 1 μA measuring

Humidity sensor

- Measurement range: 0 ~ 100% of Relative Humidity (for temperatures $\leq 0\text{ }^\circ\text{C}$ and $> 60\text{ }^\circ\text{C}$ see figure below)
- Accuracy: $\leq \pm 3\%$ RH (at $25\text{ }^\circ\text{C}$, range 20 ~ 80%)
- Hysteresis: $\pm 1\%$ RH
- Operating temperature: $-40\text{ }^\circ\text{C}$ ~ $+85\text{ }^\circ\text{C}$
- Response time (63% of step 90% to 0% or 0% to 90%): 1 second
- Typical consumption: 1.8 μA measuring
- Maximum consumption: 2.8 μA measuring

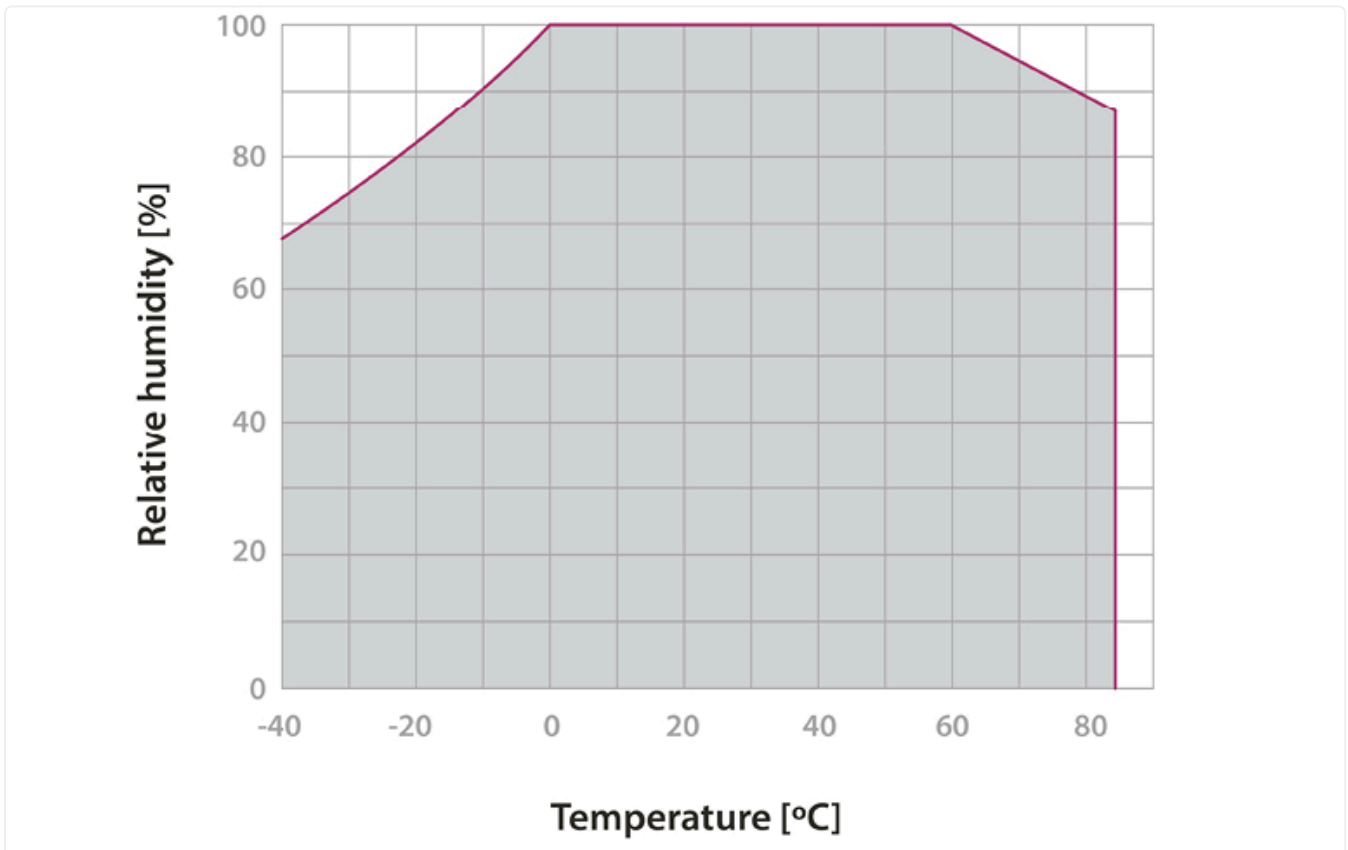


Figure: Humidity sensor operating range

Pressure sensor

- Measurement range: 30 ~ 110 kPa
- Operational temperature range: $-40\text{ }^\circ\text{C}$ ~ $+85\text{ }^\circ\text{C}$
- Full accuracy temperature range: $0\text{ }^\circ\text{C}$ ~ $+65\text{ }^\circ\text{C}$
- Absolute accuracy: $\pm 0.1\text{ kPa}$ ($0\text{ }^\circ\text{C}$ ~ $65\text{ }^\circ\text{C}$)
- Typical consumption: 2.8 μA measuring

- Maximum consumption: 4.2 μ A measuring

Measurement process

The BME280 is as 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.

When the sensor is disabled, current consumption drops to 0.1 μ A.

To read this sensor, you should use the following functions:

Temperature

Reading code:

```
{
    float temp;
    Agriculture.ON();
    // Reads the temperature from BME280 sensor
    temp = Agriculture.getTemperature();
}
```

Humidity

Reading code:

```
{
    float humd;
    Agriculture.ON();
    // Reads the humidity from BME280 sensor
    humd = Agriculture.getHumidity();
}
```

Pressure

Reading code:

```
{
  float pres;
  Agriculture.ON();
  // Reads the pressure from BME280 sensor
  humd = Agriculture.getPressure();
}
```

You can find a complete example code for reading the BME280 sensor in the following link:

<https://development.libelium.com/ag-v30-01-temperature-sensor/>

Socket

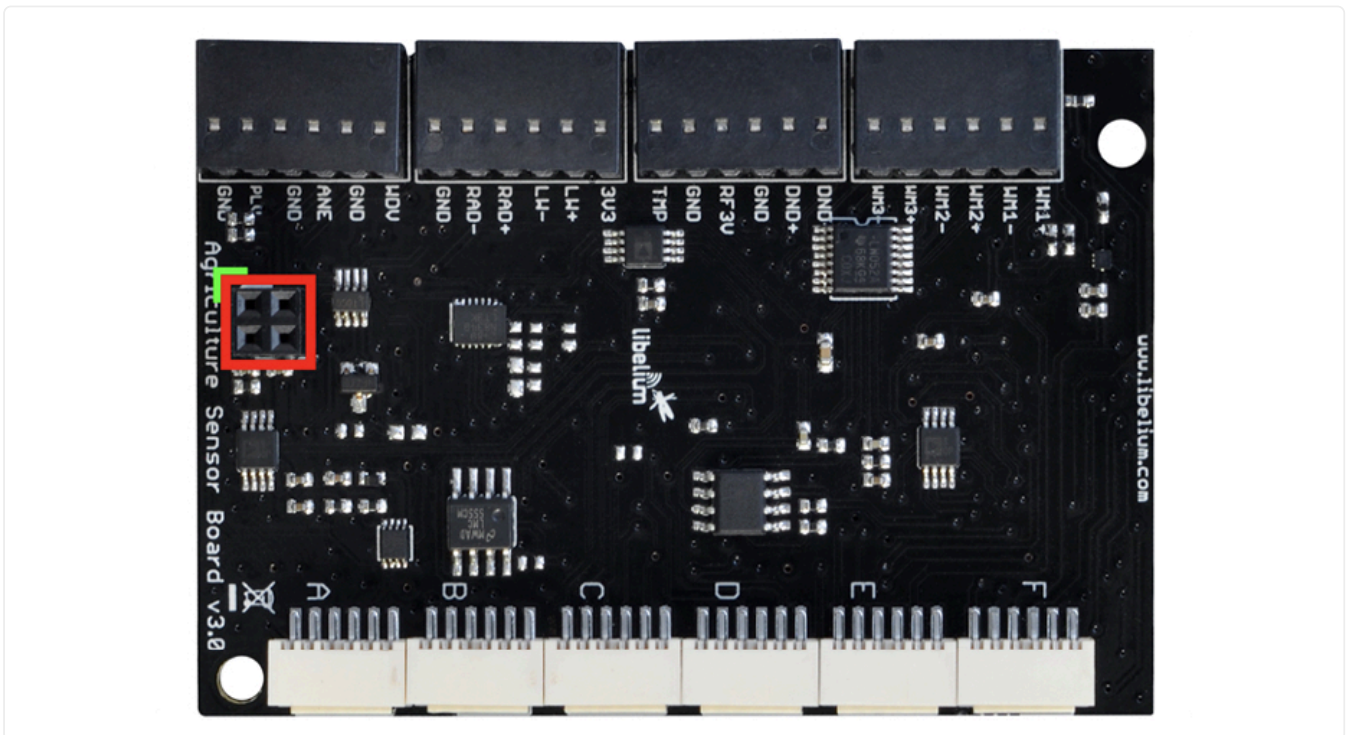


Figure: Image of the socket for the BME280 sensor

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 on the BME280 must match the mark on the Agriculture sensor board.

Leaf Wetness sensor (LWS)



Figure: Leaf Wetness sensor

Specifications

Resistance Range: 5 k Ω ~ >2 M Ω

Output Voltage Range: 1 V ~ 3.3 V

Length: 3.95 cm

Width: 1.95 cm

Measurement process

The leaf wetness sensor behaves as a resistance of a very high value (infinite, for practical purposes) in absence of condensation in the conductive combs that make it up, and that may fall down when it is completely submerged in water. The voltage at its output is inversely proportional to the humidity condensed on the sensor, and can be read at the analog input of Waspote `ANALOG3` when selected the proper output of a multiplexer that connects this sensor and one of the Watermark soil humidity sensors to that analog pin. The value returned by the reading function of the library corresponds to the percentage of condensation present on the sensor. The power supply of the sensor (3.3 V) can be cut off or connected through the switched controlled by the digital pin `ANALOG7`. It is highly recommended to switch off this sensor in order to minimize the global consumption of the board (you can find more information about the consumption of the board and how to handle it in chapter "Board configuration and programming").

Reading code:

```

{
  // Variable to store the read value
  float value_lw;
  // Instance sensor object
  leafWetnessClass lwSensor;
  Agriculture.ON();
  // Read the leaf wetness sensor
  value_lw = lwSensor.getLeafWetness();
}

```

You can find a complete example code for reading the leaf wetness sensor in the following link: <https://development.libelium.com/ag-v30-02-leaf-wetness/>

Socket

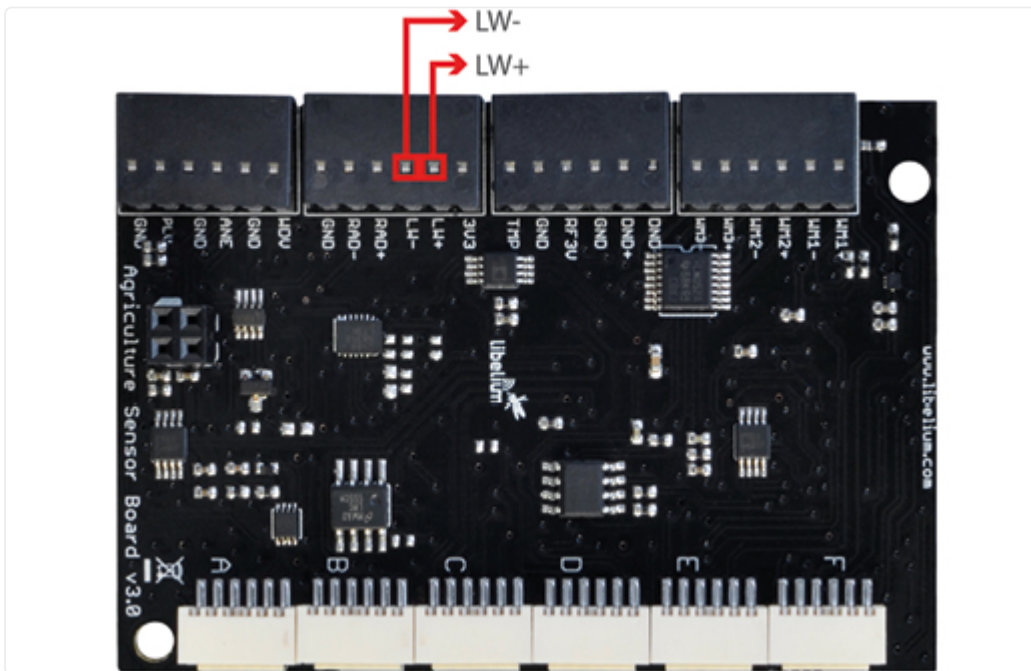


Figure: Image of the socket for the Leaf Wetness sensor

In the image above we can see highlighted the two pins of the terminal block where the sensor must be connected to the board.

Soil moisture sensor (Watermark)



Figure: Watermark sensor

Specifications

Measurement range: 0 ~ 200 cb

Frequency Range: 50 ~ 10000 Hz approximately

Diameter: 22 mm

Length: 76 mm

Terminals: AWG 20

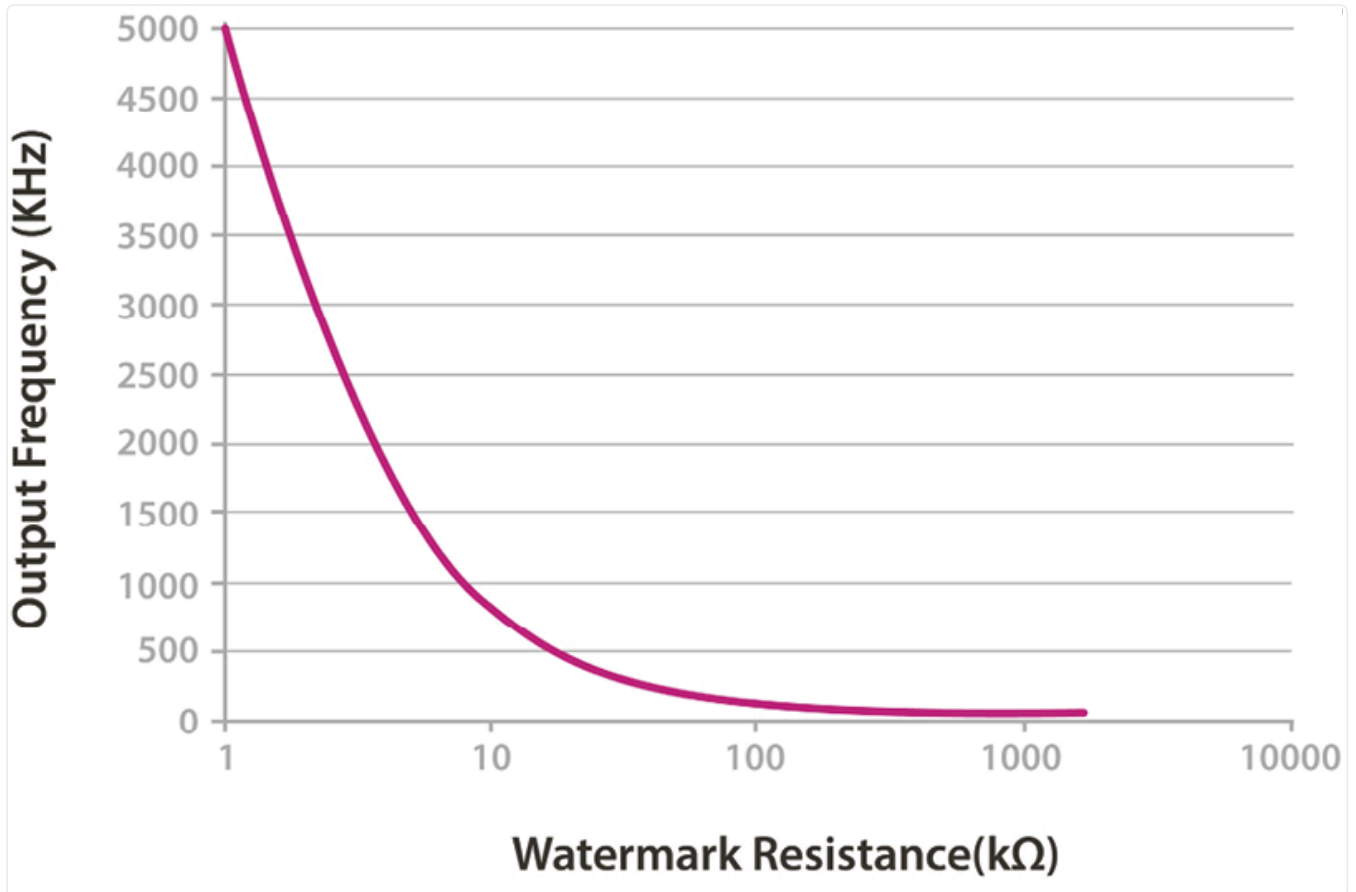


Figure: Output frequency of the Watermark sensor circuit with respect to the resistance of the sensor

Measurement process

The Watermark sensor by Irrrometer is a resistive type sensor consisting of two electrodes highly resistant to corrosion embedded in a granular matrix below a gypsum wafer. The resistance value of the sensor is proportional to the soil water tension, a parameter dependent on moisture that reflects the pressure needed to extract the water from the ground. The function of the library `readWatermark()` returns the frequency output of the sensor's adaptation circuit in Herzs (Hz), for more information about the conversion into soil water tension look at Appendix 1 in this technical guide. It is highly recommended to switch off this sensor in order to minimize the global consumption of the board (you can find more information about the consumption of the board and how to handle it in chapter "Board configuration and programming").

Reading code:

```
{  
  // Variable to store the read value  
  float watermark1, watermark2, watermark3;  
  //Instance objects  
  watermarkClass wmSensor1(SOCKET_1);  
  watermarkClass wmSensor2(SOCKET_2);  
  watermarkClass wmSensor3(SOCKET_3);  
  Agriculture.ON()  
  
  // Read the Watermarks sensors one by one  
  USB.println(F(\ "Wait for Watermark 1...\"));  
  watermark1 = wmSensor1.readWatermark();  
  USB.println(F("Wait for Watermark 2..."));  
  watermark2 = wmSensor2.readWatermark();  
  USB.println(F("Wait for Watermark 3..."));  
  watermark3 = wmSensor3.readWatermark();  
  
  // Print the watermark measures  
  USB.print(F("Watermark 1 - Frequency: "));  
  USB.print(watermark1);  
  USB.println(F(" Hz"));  
  USB.print(F("Watermark 2 - Frequency: "));  
  USB.print(watermark2);  
  USB.println(F(" Hz"));  
  USB.print(F("Watermark 3 - Frequency: "));  
  USB.print(watermark3);  
  USB.println(F(" Hz"));  
}
```

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

<https://development.libelium.com/ag-v30-07-watermark-sensor/>

Socket

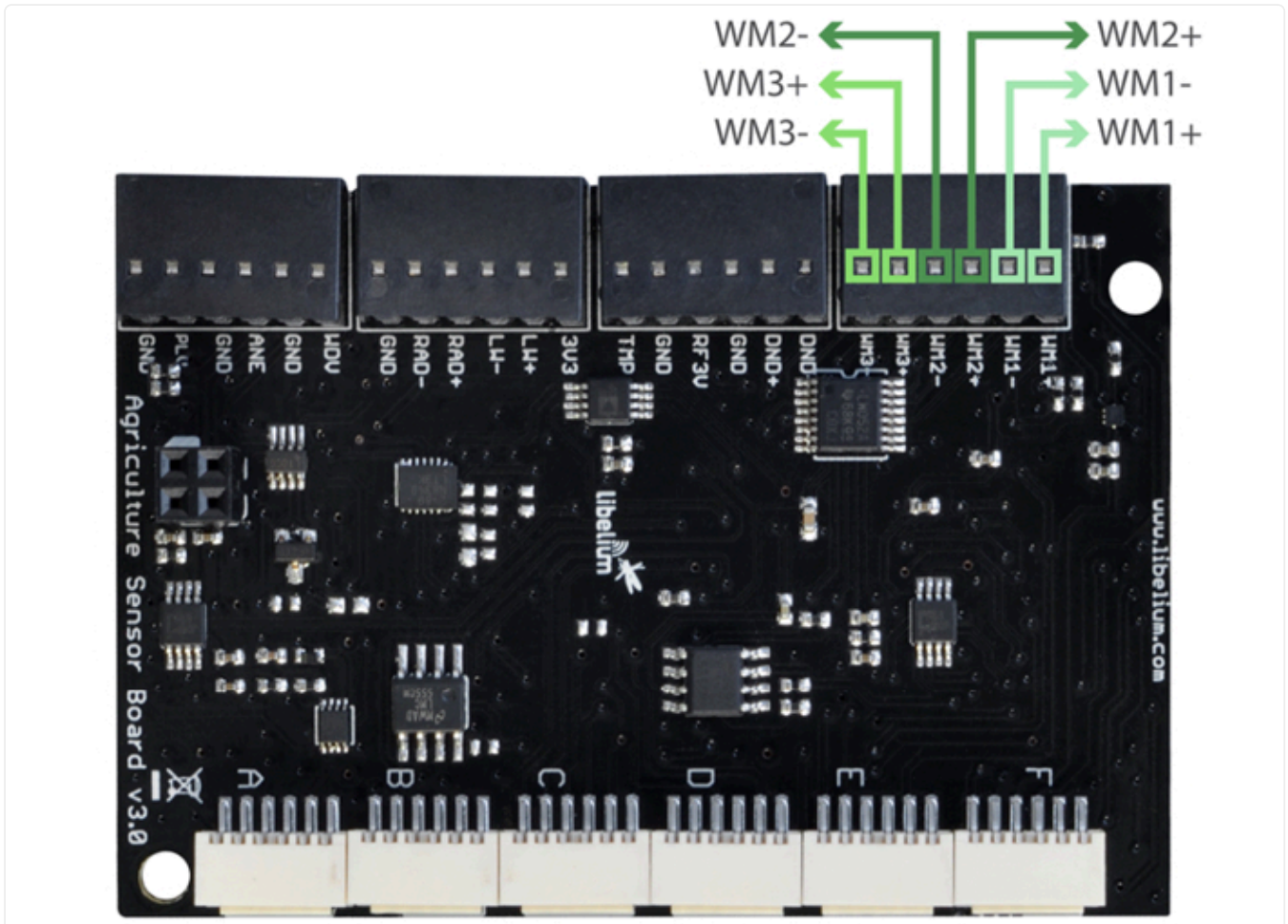


Figure: Image of the socket for the Watermark sensor

Three sockets for Watermark sensors have been placed in the agriculture board (marked in the image in the figure above) and the electronics necessary for powering and signal conditioning, so that the soil moisture can be measured at three different depths.

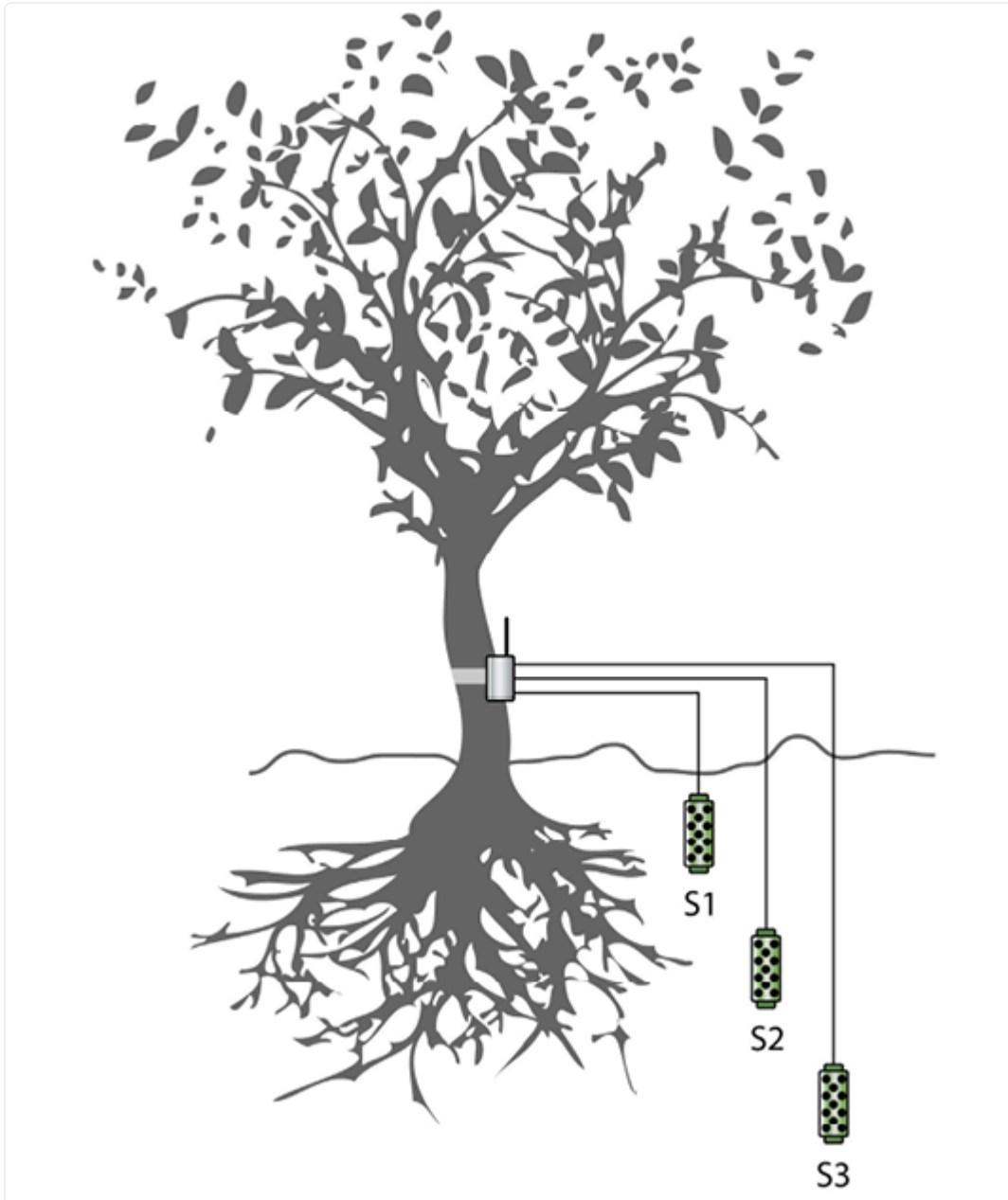


Figure: Illustration of the three Watermark sensors placed at different depths

Figure: Illustration of the three Watermark sensors placed at different depths

Soil temperature sensor (DS18B20)



Figure: DS18B20 sensor

Specifications

Measurement range: -55 ~ 125 °C

Accuracy: ±0.5 °C accuracy from -10 °C to +85 °C

Diameter: 7 mm

Length: 26 mm

Cable: 1.8 m

Waterproof

1-Wire interface

Measurement Process

The DS18B20 digital thermometer provides 9-bit Celsius temperature measurements. The DS18B20 communicates over a 1-Wire bus that by definition requires only one data line (and ground) for communication with Waspote.

This sensor only is included in the Smart Agriculture Normal version.

Reading code:


```

{
  // Variable to store the read value
  float temp;
  // Instance object
  ds18b20Class ds18b20;
  Agriculture.ON();
  // Read the temperature sensor
  temp = ds18b20.readDS18b20();
}

```

You can find a complete example code for reading the DS18B20 sensor in the following link:

<https://development.libelium.com/ag-v30-12-ds18b20-sensor/>

Socket

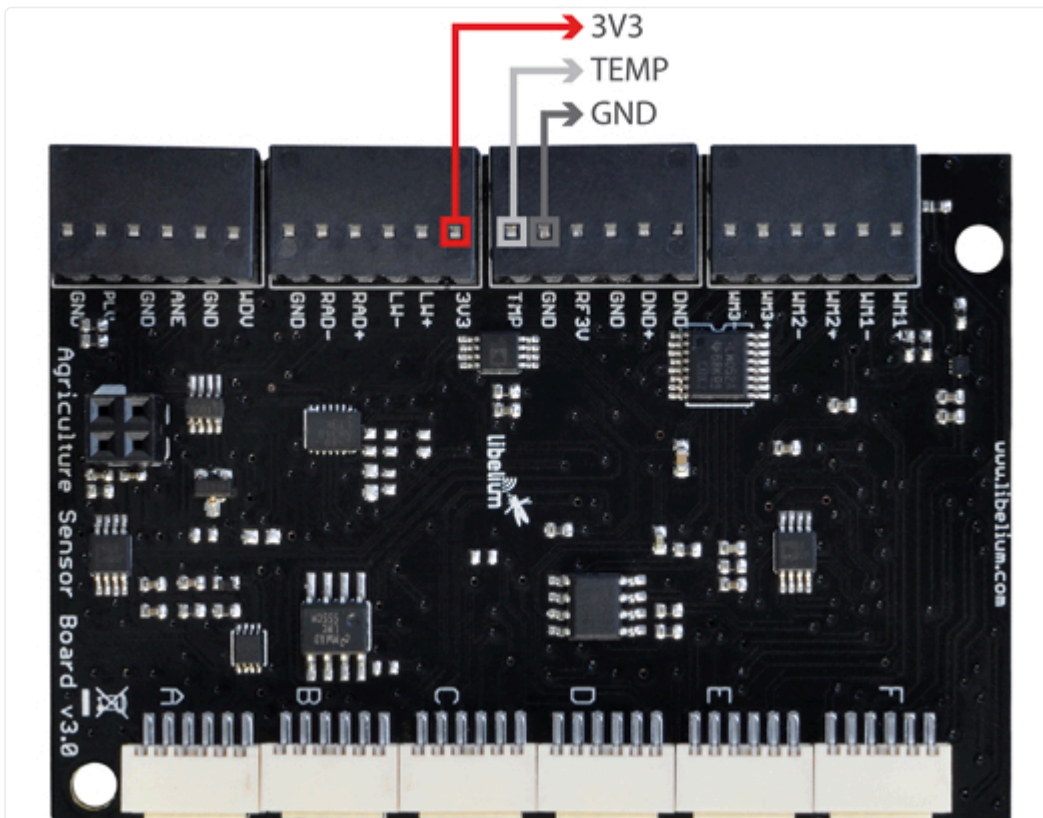


Figure: Image of the socket for the DS18B20 sensor

The sensor must be connected to its adaptation stage through a 2.54 mm pitch. We can see an image with the 3 pins of this socket corresponding to the sensor in the previous figure. The function of the library `readDS18b20()` returns the temperature value in Celsius degree (°C). The power supplies required by the sensor is 3.3 V.

Soil temperature sensor (Pt-1000)



Figure: Pt-1000 sensor

Figure: Pt-1000 sensor

Specifications

Measurement range: -50 ~ 300 °C

Accuracy: DIN EN 60751

Resistance (0 °C): 1000 Ω

Diameter: 6 mm

Length: 40 mm

Cable: 5 m

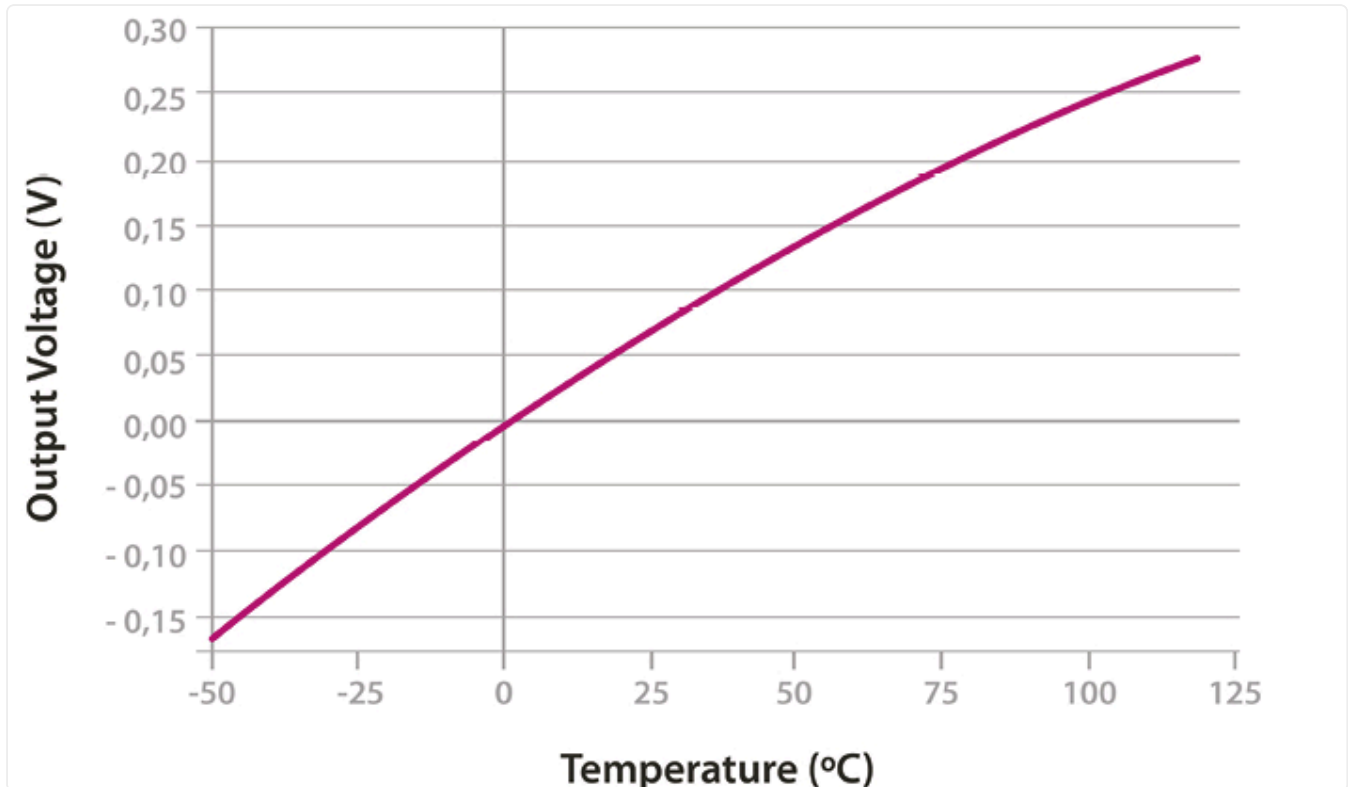


Figure: Output voltage of the Pt-1000 sensor with respect to temperature

Measurement process

The resistance of the Pt-1000 sensor varies between approximately 920 Ω and 1200 Ω in the range considered useful in agriculture applications (-20 ~ 50 °C approximately), which results in too low variations of voltage at significant changes of temperature for the resolution of the Waspote's analog-to-digital converter. The function of the library `readPT1000()` returns the temperature value in Celsius degree (°C). The power supplies required by the sensor, both 3.3 V and 5 V, are controlled internally by the library.

This sensor can only be read by Smart Agriculture PRO version.

Reading code:

```
{  
  // Variable to store the read value  
  float value_PT1000;  
  // Instance object  
  pt1000Class pt1000Sensor;  
  Agriculture.ON();  
  // Read PT1000 sensor  
  value_PT1000 = pt1000Sensor.readPT1000();  
}
```

You can find a complete example code for reading the Pt-1000 sensor in the following link:

<https://development.libelium.com/ag-v30-06-pt1000-sensor/>

Socket

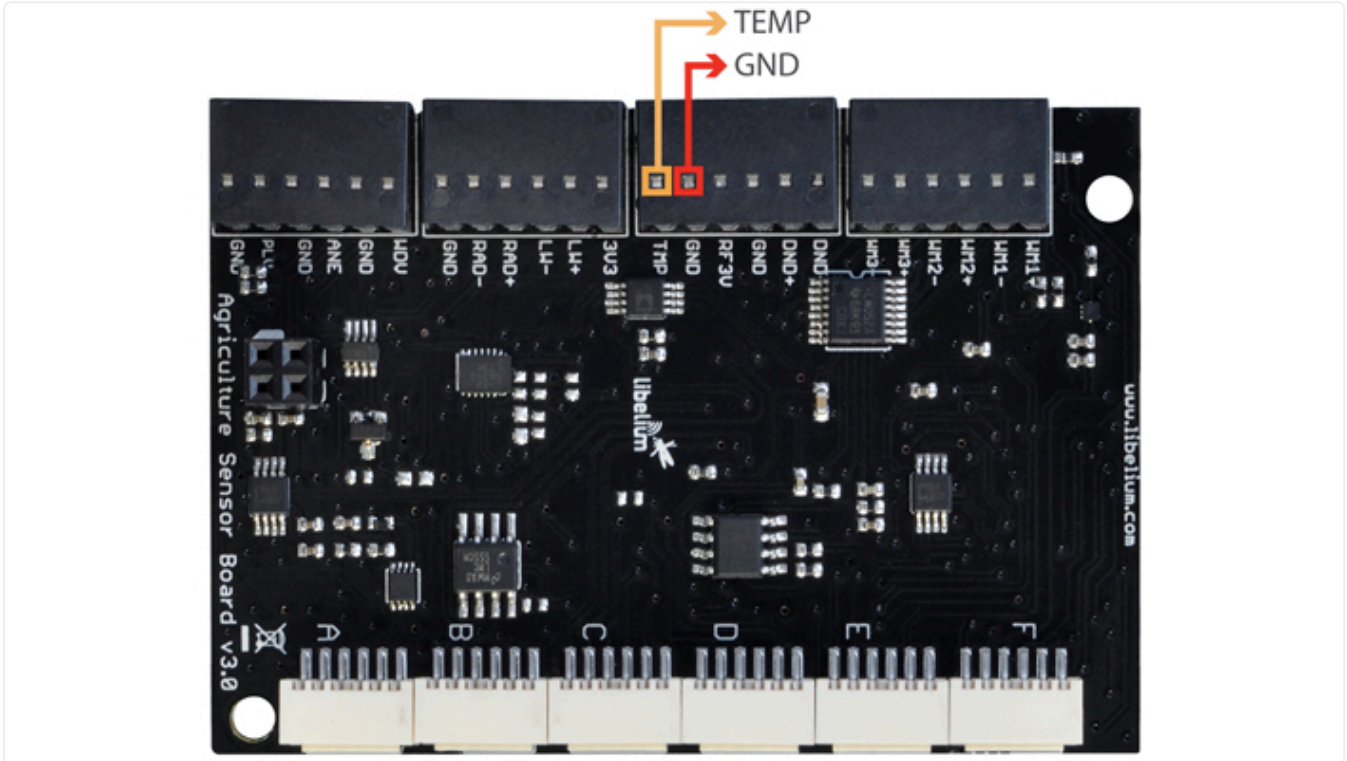


Figure: Image of the socket for the Pt-1000 sensor

The sensor must be connected to its adaptation stage through a 2.54 mm pitch. We can see an image with the two pins of this socket corresponding to the sensor in the previous figure. Both pins of the sensor can be connected to any of the two ways, since there is no polarity to be respected.

Trunk diameter dendrometer (Ecomatik DC3)

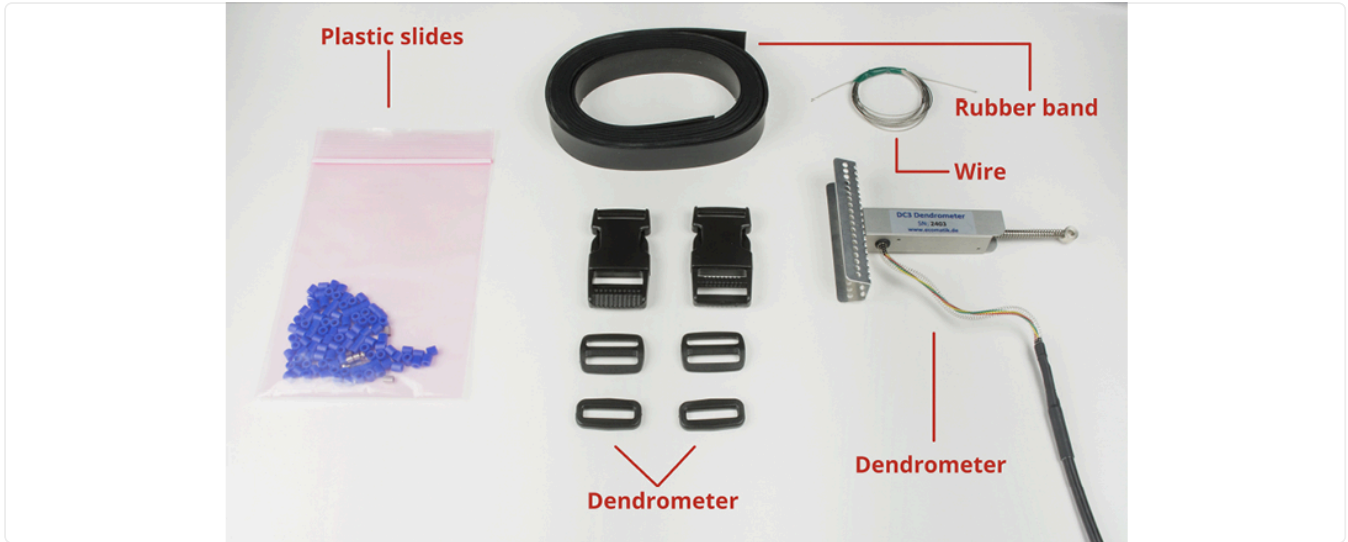


Figure: Ecomatik DC3 sensor

Specifications

Operation temperature: -30 ~ 40 °C

Operation humidity: 0 ~ 100% RH

Trunk/branch diameter: from 5 cm

Accuracy: $\pm 3.3 \mu\text{m}$

Temperature coefficient: $< 1.4 \mu\text{m}/\text{K}$

Linearity: 0.7%

Output range: 0 ~ 20 k Ω

Range of the sensor: function of the size of the tree:

Tree diameter (cm)	Measuring range in diameter (mm)
10	16.2
50	11.3
100	9.0

i The previous version of the DC3 Trunk diameter (the DC2) has been discontinued, but it is still available for replacements on demand. Contact your sales agent for more information.

Installation process

Cut the rubber band into 2 pieces as long as the tree circumference and bond them with the plug locks together. Unroll the wire from the reel, turn the thread rods of the turnbuckle to the middle position. Thread a certain number (dependent on the tree circumference) of the plastic slides provided on the wire. This plastic slides will avoid the wire to be swallowed by the tree cortex. Then insert the end of the wire through the sensor hole. Finally, pass the prepared wire around the tree trunk, and fix it on the turnbuckle by the adjusting screw.

In the pictures below can be seen how the sensor should look after the installation process.

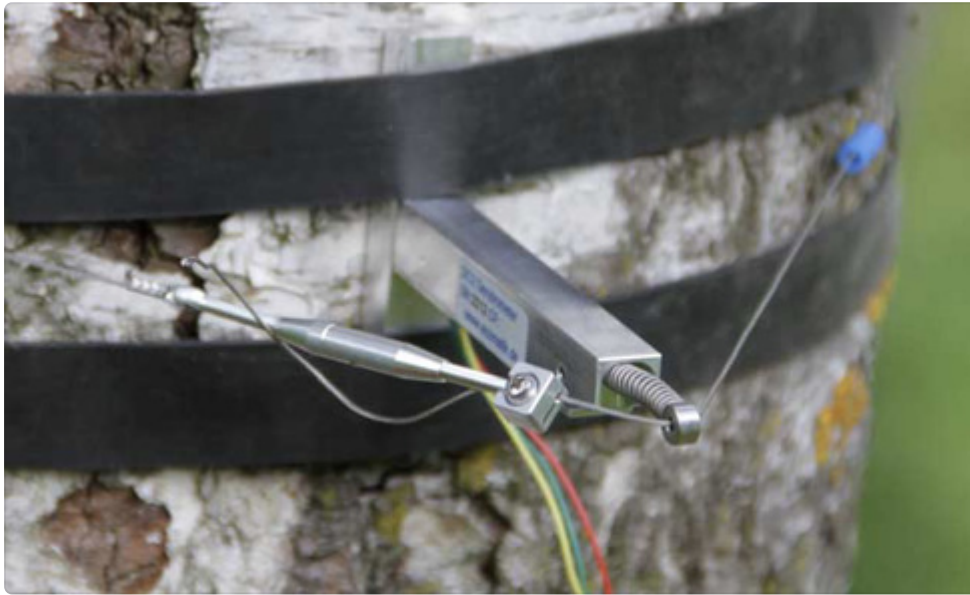


Figure: Ecomatik DC2 sensor (previous version)



Figure: Ecomatik DC3 sensor

Ensure that the wires lies flat around the trunk, and move the plastic slides along the wire so that the wire itself does not touch the bark. Then turn the turnbuckle slowly so that the sensor rod is pushed in by about 2-3 mm. Fix the cable onto the tree trunk/branch so that the sensor is protected from any accidental pull/drag of the entire cable length. This can be done using a

rope or cable straps. In addition, there should be no tension between the sensor and cable. Ensure that no rain water can run along the cable into the sensor casing.

Measurement process

The operation of the 3 Ecomatik dendrometers, DC3, DD-S and DF, is based on the variation of an internal resistance with the pressure that the growing of the trunk, stem, branch or fruit exerts on the sensor. The circuit permits the reading of that resistance in a full bridge configuration through a 16 bits analog-to-digital converter whose reference is provided by a high precision 3V voltage reference in order to acquire the most accurate and stable measurements possible. The reading of the converter, shared with the PT-1000 temperature sensor, will be carried out through the Digital Bus using the functions of the library `WaspSensorAgr_v30` explained in the chapter "API" about this API library, returning the read value in millimeters (mm). The 3.3 V and 5 V power supplies of the devices may be cut off or connected via the digital switch controlled internally by the library.

This sensor can only be read by Smart Agriculture PRO version.

Reading code:

```
{
  // Variable to store the read value
  float value_dendrometer;
  // Instance object
  dendrometerClass dendSensor r(SENS_SA_DC3);
  Agriculture.ON();
  // Read the dendrometer sensor
  value_dendrometer = dendSensor.readDendrometer();
}
```

You can find a complete example code for reading the dendrometer sensor in the following link: <https://development.libelium.com/ag-v30-05a-dendrometer-sensor/>

There is another possibility to take measures using the dendrometer. First, a reference is stored at the beginning of the code with the `setReference()` function (it should be run inside the `setup()`). Then, the value returned is calculated with the current value of the dendrometer minus the reference stored, as if it was an offset. So the measurement will be the upgrowth of the fruit using the `readGrowth()` function.

```
{  
  // Variable to store the read value  
  float value_dendrometer;  
  
  // Instance object  
  dendrometerClass dendSensor(SENS_SA_DC3);  
  
  Agriculture.ON();  
  
  // This function reads the dendrometer value  
  // and sets it as zero millimeters reference  
  dendSensor.setReference();  
  
  // Read the dendrometer sensor and return the  
  // the difference between this value and reference  
  value_dendrometer = dendSensor.readGrowth();  
}
```

You can find a complete example code for reading the dendrometer sensor in the following link: <https://development.libelium.com/ag-v30-05b-dendrometer-sensor-with-reference/>

Socket

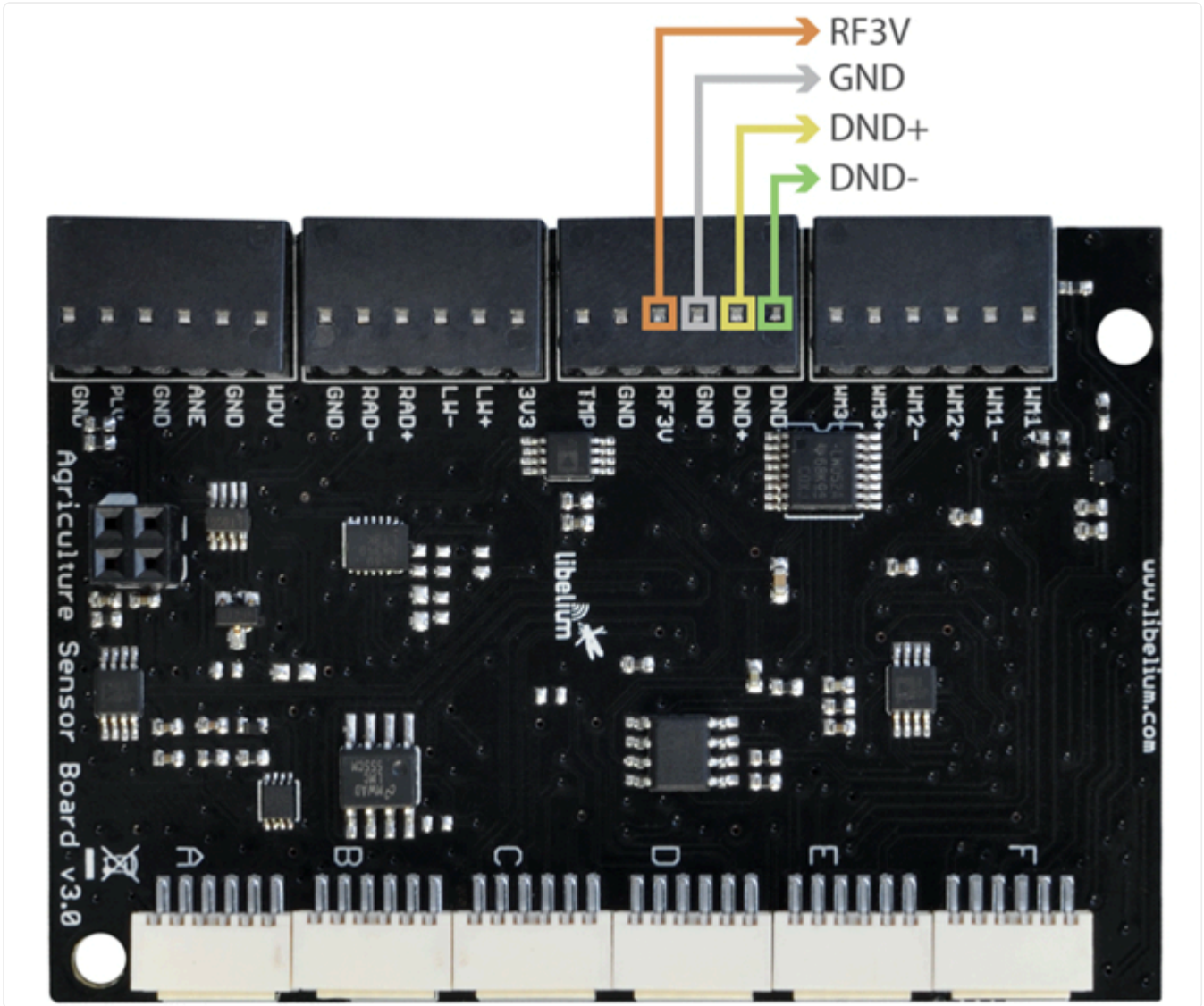


Figure: Image of the socket for the dendrometers

Any of the three dendrometers available may be connected to Waspnote through the two 2.54 mm pitch terminal blocks marked in the image in the figure above. The pins corresponding to the sensor in this terminal block (highlighted in the figure) provide the sensor with connection to ground, power supply and to the analog-to-digital converter differential input.

Stem diameter dendrometer (Ecomatik DD-S)



Figure: Ecomatik DD-S sensor

Specifications

Stem/branch diameter: 0 ~ 5 cm

Range of the sensor: 5.5 mm

Output range: 0 ~ 20 k Ω

Accuracy: $\pm 2 \mu\text{m}$

Temperature coefficient: $< 0.1 \mu\text{m/K}$

Operation temperature: $-30 \sim 40 \text{ }^\circ\text{C}$

Operation humidity: 0 ~ 100% RH

Cable length: 2 m

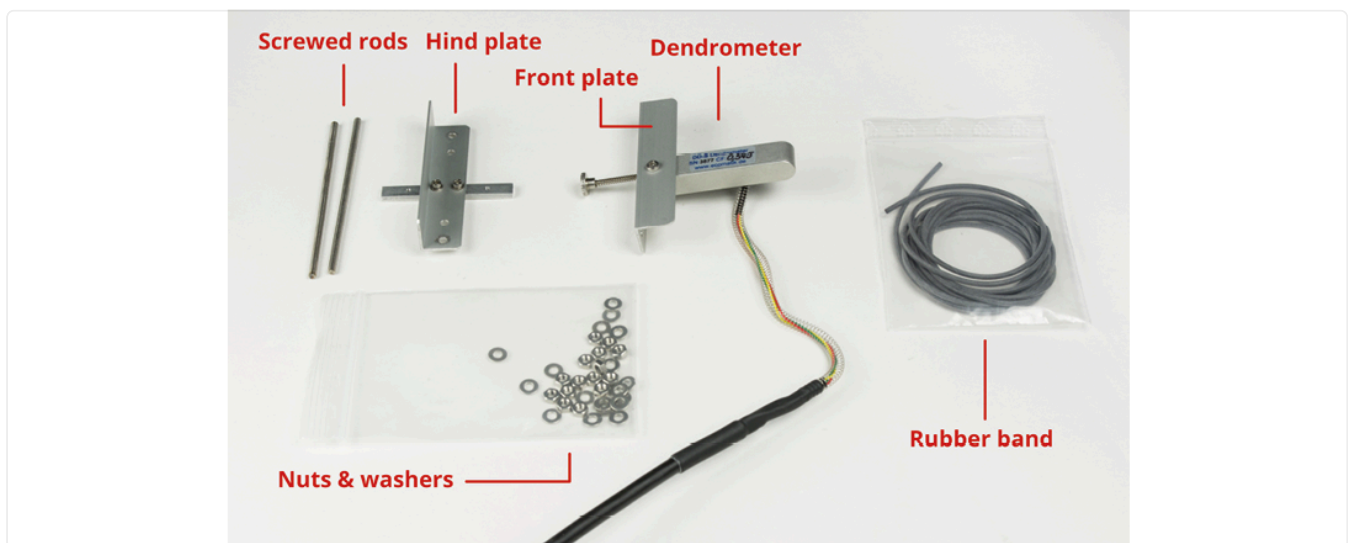


Figure: Ecomatik DD-S sensor

Installation process

Detach the front frame of the dendrometer to allow you fix the stem/branch to be measured. Fix the hind plate and hind frame on the stem/branch using a rubber band so that the hind plate is firmly held on the stem/branch. Replace the front part of the frame and fix it with screws. Turn the screws slowly to push in the rod for about 2 mm. Fix the cable onto the tree stem/branch so that the sensor is protected from any accidental pull/drag on the entire cable length. This can be done using a rope or cable straps. Ensure the suspension rope/strap is not so tight as to interfere with normal tree growth and expansion during the entire measurement period. Also, there should be no tension between the sensor and cable. Ensure that no rain water can run along the cable into the sensor casing.

In the pictures below can be seen how the sensor should look after the installation process.



Figure: Ecomatik DD-S sensor (stem)



Figure: Ecomatik DD-S sensor (branch)

For very small plants, the dendrometer can be hung or fixed to a wood stick.

Measurement process

The operation of the three Ecomatik dendrometers, DC3, DD-S and DF, is based on the variation of an internal resistance with the pressure that the growing of the trunk, stem, branch or fruit exerts on the sensor. The circuit permits the reading of that resistance in a full bridge configuration through a 16 bits analog-to-digital converter whose reference is provided by a high precision 3V voltage reference in order to acquire the most accurate and stable measurements possible. The reading of the converter, shared with the Pt-1000 temperature sensor, will be carried out through the Digital Bus using the functions of the library `WaspSensorAgr_v30` explained in the chapter "API" about this API library, returning the read value in millimeters (mm). The 3.3 V and 5 V power supplies of the devices may be cut off or connected via the digital switch controlled internally by the library.

This sensor can only be read by Smart Agriculture PRO version.

Reading code:


```
{
  // Variable to store the read value
  float value_dendrometer;
  // Instance object
  dendrometerClass dendSensor(SENS_SA_DD);
  Agriculture.ON();
  // Read the dendrometer sensor
  value_dendrometer = dendSensor.readDendrometer();
}
```

You can find a complete example code for reading the dendrometer sensor in the following link: <https://development.libelium.com/ag-v30-05a-dendrometer-sensor/>

There is another possibility to take measures using the dendrometer. First, a reference is stored at the beginning of the code with the `setReference()` function (it should be run inside the `setup()`). Then, the value returned is calculated with the current value of the dendrometer minus the reference stored, as if it was an offset. So the measurement will be the upgrowth of the fruit using the `readGrowth()` function.

```
{
  // Variable to store the read value
  float value_dendrometer;

  // Instance object
  dendrometerClass dendSensor(SENS_SA_DC3);

  Agriculture.ON();

  // This function reads the dendrometer value
  // and sets it as zero millimeters reference
  dendSensor.setReference();

  // Read the dendrometer sensor and return the
  // difference between this value and reference
  value_dendrometer = dendSensor.readGrowth();
}
```

You can find a complete example code for reading the dendrometer sensor in the following link: <https://development.libelium.com/ag-v30-05b-dendrometer-sensor-with-reference/>

Socket

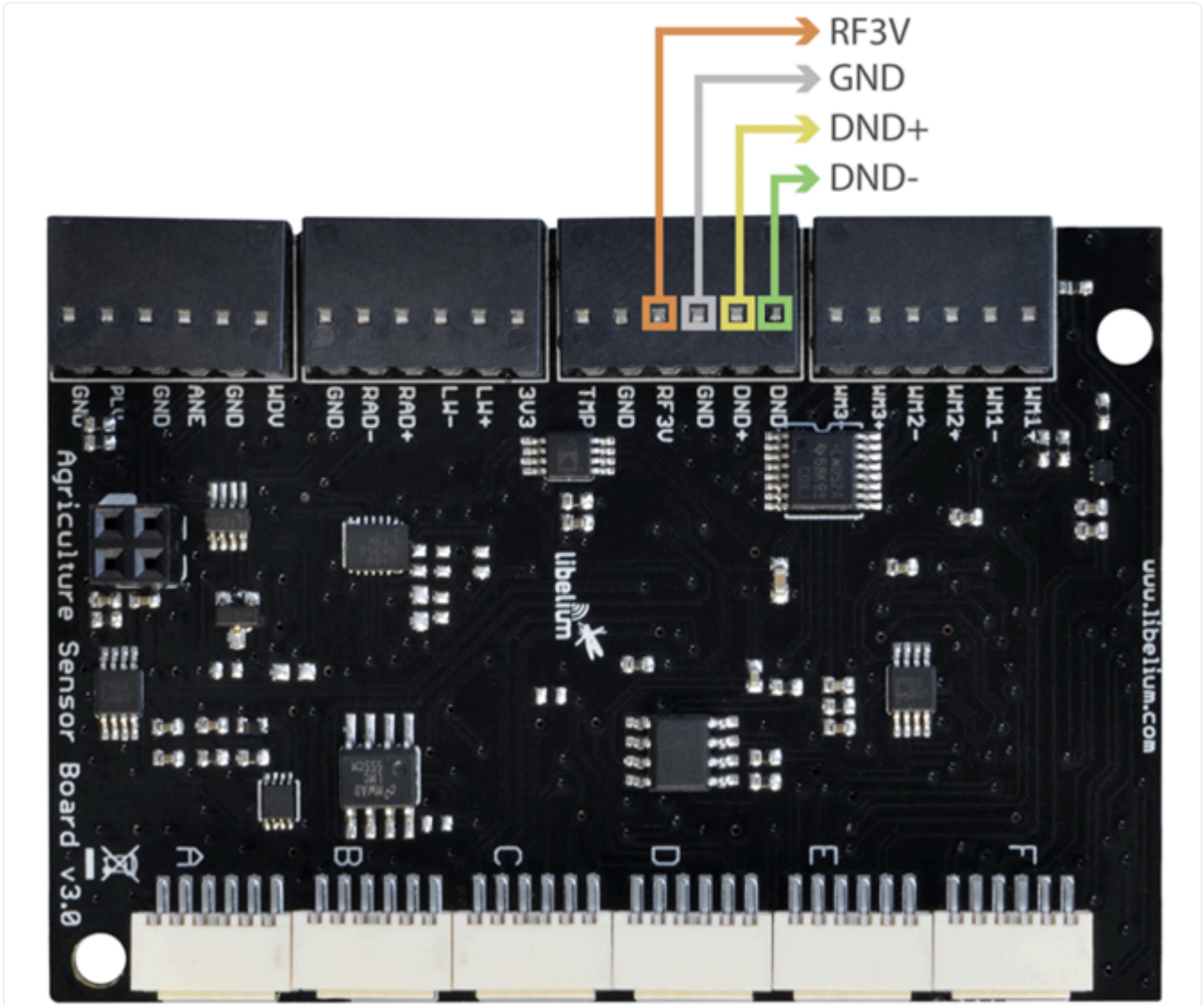


Figure: Image of the socket for the dendrometers

Any of the three dendrometers available may be connected to Waspote through the two 2.54 mm pitch terminal blocks marked in the image in the figure above. The pins corresponding to the sensor in this terminal block (highlighted in the figure) provide the sensor with connection to ground, power supply and to the analog-to-digital converter differential input.

Fruit diameter dendrometer (Ecomatik DF)



Figure: Ecomatik DF sensor

Specifications

Fruit diameter: 0 ~ 11 cm

Range of the sensor: 7.5 mm

Output range: 0 ~ 20 k Ω

Accuracy: $\pm 2 \mu\text{m}$

Temperature coefficient: $< 0.1 \mu\text{m}/\text{K}$

Operation temperature: $-30 \sim 40 \text{ }^\circ\text{C}$

Operation humidity: 0 ~ 100% RH

Cable length: 2 m

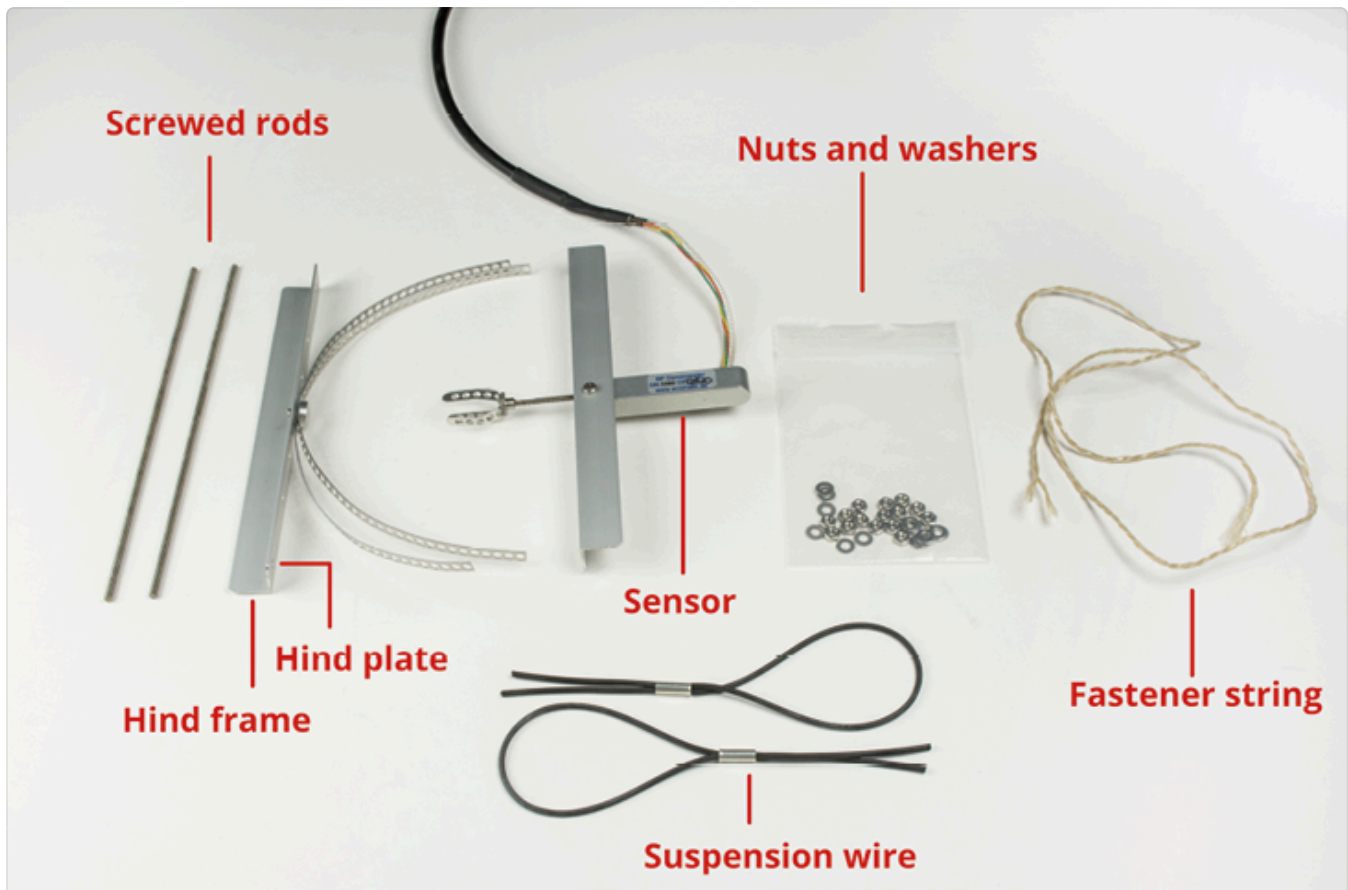


Figure: Ecomatik DF sensor

Installation process

Detach the front frame of the dendrometer to allow you fix the fruit to be measured. Insert the fruit carefully between the fixing wires until it rests onto the hind plate. Curve the fixing wires accordingly so as to fit the shape of the fruit. Ensure that the fruit rests on the hind plate. Fix the wires on the fruit using a rubber band so that the fruit is firmly held between the fixing wires and the hind plate. Replace the front part of the frame and fix it with screws. The complete set-up should be suspended freely from a tree branch using a suspension wire. Ensure that the fruit hangs freely and that the branch from which it grows is not strained by the dendrometer. Turn the screws slowly to push in the rod about 2 mm. Fix the cable onto the tree branch so that the sensor is protected from any accidental pull/drag on the entire cable length. This can be done using a rope or cable straps. Ensure the suspension rope/strap is not so tight as to interfere with normal tree growth and expansion during the entire measurement period. Also, there should be no tension between the sensor and cable.

In the pictures below can be seen how the sensor should look after the installation process.



Figure: Ecomatik DF sensor with apple



Figure: Ecomatik DF sensor with grape



Figure: Ecomatik DF sensor with melon



Figure: Ecomatik DF sensor with cucumber

Ensure that no rain water can run along the cable into the sensor casing.

It may not be necessary but, in case the fruit is too small to withstand the dendrometer's frame weight, you can adapt the instrument: the frames and the screwed rods can be shortened by cutting.

Measurement process

The operation of the three Ecomatik dendrometers, DC3, DD-S and DF, is based on the variation of an internal resistance with the pressure that the growing of the trunk, stem, branch or fruit exerts on the sensor. The circuit permits the reading of that resistance in a full bridge configuration through a 16 bits analog-to-digital converter whose reference is provided by a high precision 3 V voltage reference in order to acquire the most accurate and stable measurements possible. The reading of the converter, shared with the Pt-1000 temperature sensor, will be carried out through the Digital Bus using the functions of the library `WaspSensorAgr_v30` explained in chapter "API" about this API library, returning the read value in millimeters (mm). The 3.3 V and 5 V power supplies of the devices may be cut off or connected via the digital switch controlled internally by the library. It is highly recommended to switch off this sensor in order to minimize the global consumption of the board (you can find more information about the consumption of the board and how to handle it in chapter "Board configuration and programming").

This sensor can only be read by Smart Agriculture PRO version.

Reading code:

```
{
  // Variable to store the read value
  float value_dendrometer;
  // Instance object
  dendrometerClass dendSensor(SENS_SA_DF);
  Agriculture.ON();
  // Read the dendrometer sensor
  value_dendrometer = dendSensor.readDendrometer();
}
```

You can find a complete example code for reading the dendrometer sensor in the following link: <https://development.libelium.com/ag-v30-05a-dendrometer-sensor/>

There is another possibility to take measures using the dendrometer. First, a reference is stored at the beginning of the code with the `setReference()` function (it should be run inside

the `setup()`). Then, the value returned is calculated with the current value of the dendrometer minus the reference stored, as if it was an offset. So the measurement will be the upgrowth of the fruit using the `readGrowth()` function.

```
{
  // Variable to store the read value
  float value_dendrometer;

  // Instance object
  dendrometerClass dendSensor(SENS_SA_DC3);

  Agriculture.ON();

  // This function reads the dendrometer value
  // and sets it as zero millimeters reference
  dendSensor.setReference();

  // Read the dendrometer sensor and return the
  // difference between this value and reference
  value_dendrometer = dendSensor.readGrowth();
}
```

You can find a complete example code for reading the dendrometer sensor in the following link: <https://development.libelium.com/ag-v30-05b-dendrometer-sensor-with-reference/>

Socket

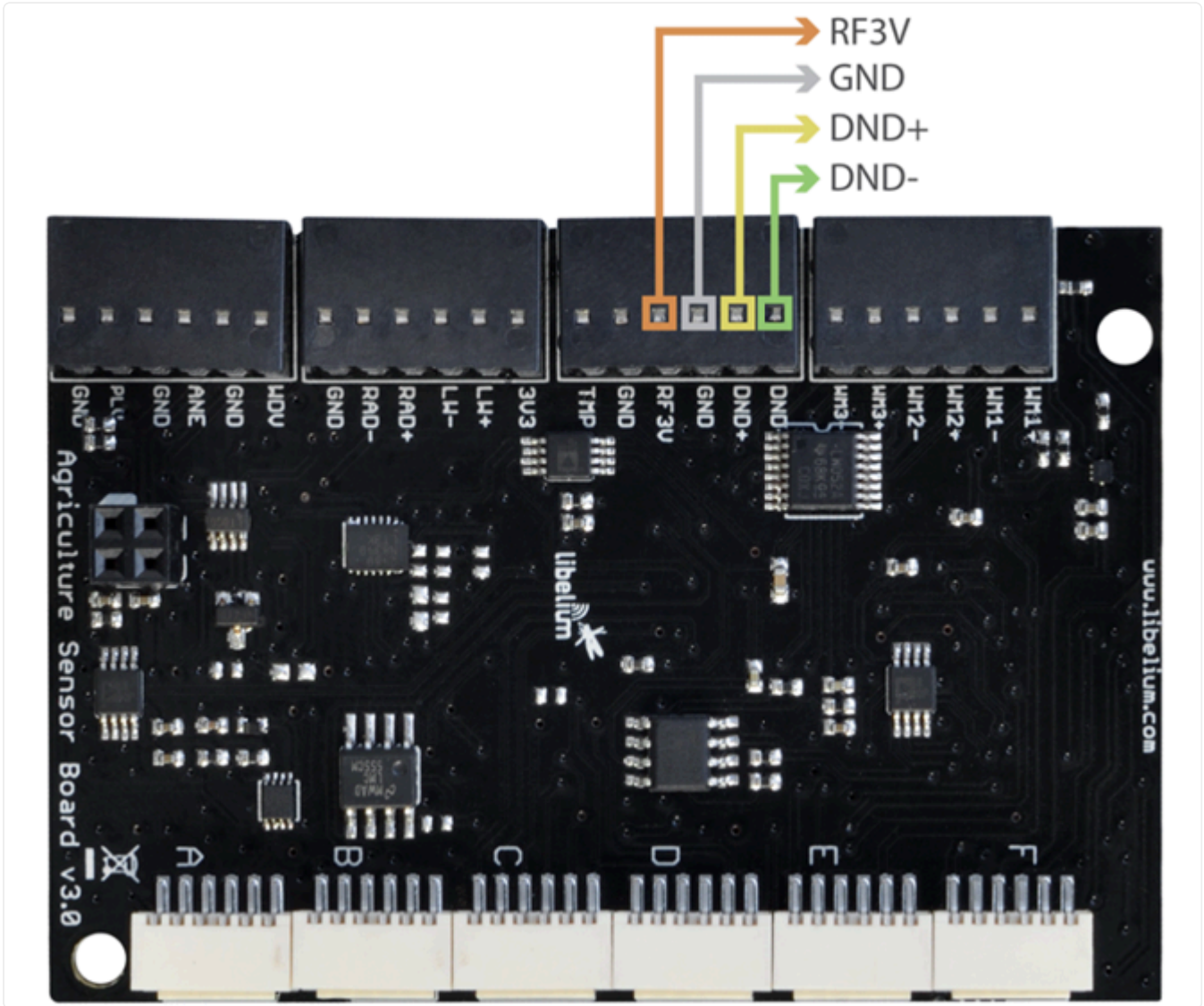


Figure: Image of the socket for the dendrometers

Any of the three dendrometers available may be connected to Waspote through the 2.54 mm pitch terminal blocks marked in the image in the figure above. The pins corresponding to the sensor in this terminal block (highlighted in the figure) provide the sensor with connection to ground, power supply and to the analog-to-digital converter differential input.

Fruit diameter dendrometer (Ecomatik DF4)



Ecomatik DF4 sensor

Specifications

Fruit diameter: ~ 13 cm

Range of the sensor: 120 mm

Output range: 0 ~ 20 k Ω

Accuracy: $\pm 4.4 \mu\text{m}$

Temperature coefficient: $< 0.2 \mu\text{m}/\text{K}$

Operation temperature: $-30 \sim 70 \text{ }^\circ\text{C}$

Operation humidity: 0 ~ 100% RH

Cable length: 5 m



The sensor is protected from rainwater, but it is not sealed. Please do not immerse the sensor in water. Max. fruit diameter 135 mm, exceedance may cause sensor

damage



Figure: Ecomatik DF4 sensor

Installation process

- a) Select a representative fruit for instrumentation.
- b) Use the included rubber cord to strain relief the sensor by fixing the sensor cable on the fruit carrying branch. Please leave enough cable between the fixation point on the branch and the sensor. In the final installation position, the fruit sensor should be able to move freely with the instrumented fruit, without tension on the cable.
- c) Pull apart the sensor clamping system and carefully insert the fruit into the fruit gripper. Make sure that the sensor is attached firmly to the fruit and that the fruit grippers are in firm contact to the fruit surface.
- d) Fix the cable onto the stem or on a ground stake for strain relief of the sensor cable between instrumented plant and the data logger. This can be done using a rope or cable

straps. There should be no tension between the sensor, fixation point at the fruit carrying branch and the remaining sensor cable.

!! IMPORTANT !! Fix the cable onto the tree stem/branch so that the sensor is protected from any accidental pull/ drag of the entire cable length. This can be done using a rope or cable straps. In addition, there should be no tension between the sensor and cable. Ensure that no rainwater can run along the cable, or the sensor rod and enter the sensor casing. Rod entrance, as well as wire outlet, should hence always be inclined downwards.

Measurement process

The operation of the three Ecomatik dendrometers, DC3, DD-S, DF and DF4 is based on the variation of an internal resistance with the pressure that the growing of the trunk, stem, branch or fruit exerts on the sensor. The circuit permits the reading of that resistance in a full bridge configuration through a 16 bits analog-to-digital converter whose reference is provided by a high precision 3 V voltage reference in order to acquire the most accurate and stable measurements possible. The reading of the converter, shared with the Pt-1000 temperature sensor, will be carried out through the Digital Bus using the functions of the library `WaspSensorAgr_v30` explained in chapter "API" about this API library, returning the read value in millimetres (mm). The 3.3 V and 5 V power supplies of the devices may be cut off or connected via the digital switch controlled internally by the library. It is highly recommended to switch off this sensor in order to minimize the global consumption of the board (you can find more information about the consumption of the board and how to handle it in chapter "Board configuration and programming").

In order to take measures using the dendrometer, the user can load the below example. In this example, first, a reference is stored at the beginning of the code with the `setReference()` function (it should be run inside the `setup()`). Then, the value returned is calculated with the current value of the dendrometer minus the reference stored, as if it was an offset. So the measurement will be the upgrowth of the fruit using the `readGrowth()` function.


```
{  
  // Variable to store the read value  
  float value_dendrometer;  
  
  // Instance object  
  dendrometerClass dendSensor(SENS_SA_DF4);  
  
  Agriculture.ON();  
  
  // This function reads the dendrometer value  
  // and sets it as zero millimeters reference  
  dendSensor.setReference();  
  
  // Read the dendrometer sensor and return the  
  // difference between this value and reference  
  value_dendrometer = dendSensor.readGrowth();  
}
```

You can find a complete example code for reading the dendrometer sensor in the following link: <https://development.libelium.com/ag-v30-05b-dendrometer-sensor-with-reference/>

Socket

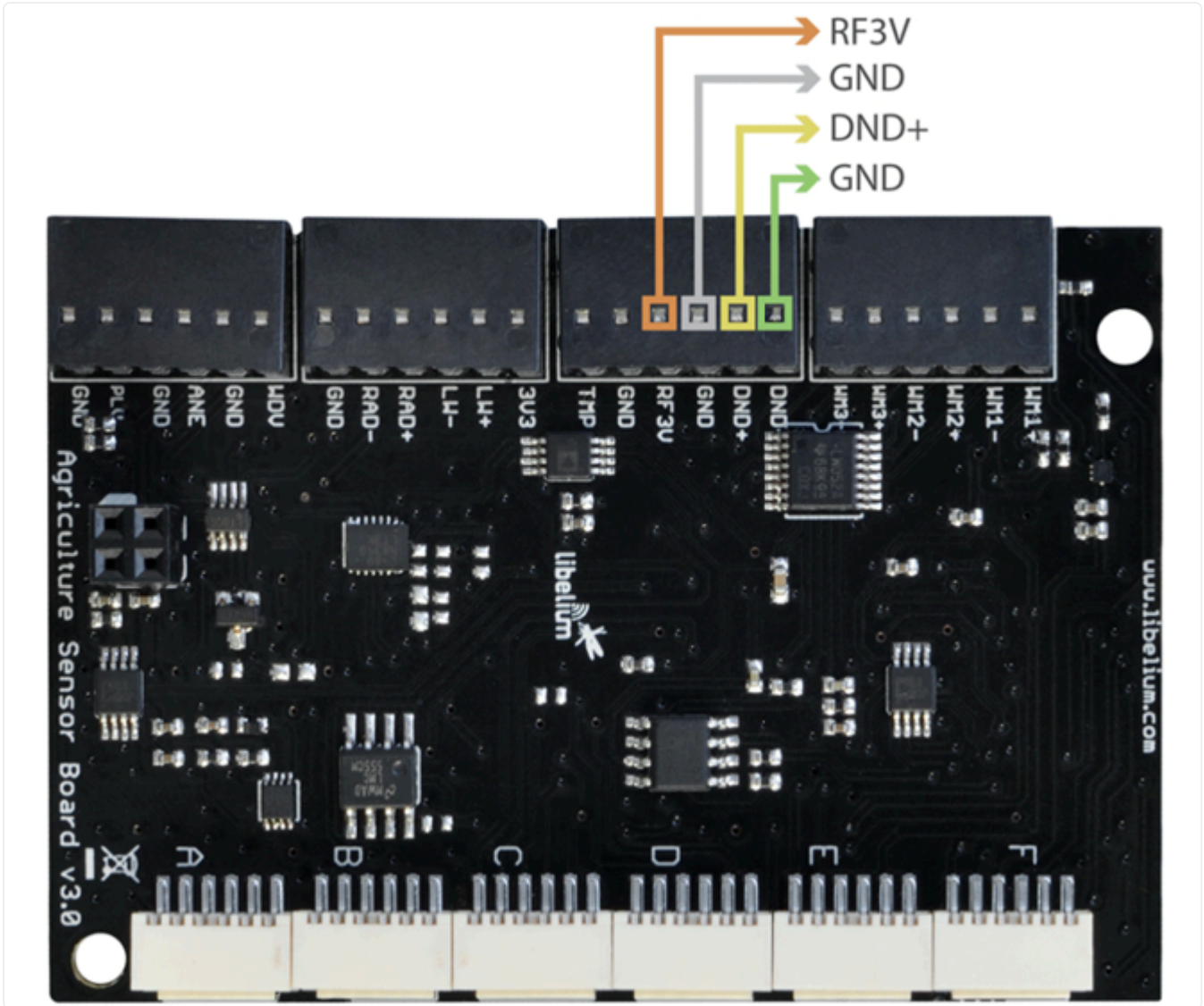


Figure: Image of the socket for the dendrometers

Any of the three dendrometers available may be connected to Waspote through the 2.54 mm pitch terminal blocks marked in the image in the figure above. The pins corresponding to the sensor in this terminal block (highlighted in the figure) provide the sensor with connection to ground, power supply and to the analog-to-digital converter differential input.

Solar radiation sensor - PAR (SQ-110)



Figure: SQ-110 sensor

i Until July 2021, Libelium distributed this sensor, but it was discontinued by Apogee.

From July 2021, Libelium only offers the SQ-100x sensor for PAR metering, which replaces the SQ-110 and has a slightly better performance.

Specifications

Sensitivity: 0.200 mV / $\mu\text{mol}\cdot\text{m}^{-2}\cdot\text{s}^{-1}$

Calibration factor: 5 $\mu\text{mol}\cdot\text{m}^{-2}\cdot\text{s}^{-1}/\text{mV}$

Non-linearity: < 1% (up to 4000 $\mu\text{mol}\cdot\text{m}^{-2}\cdot\text{s}^{-1}$)

Non-stability (long-term drift): <2% per year

Spectral range: 410 ~ 655 nm

Accuracy: $\pm 5\%$

Repeatability: <1%

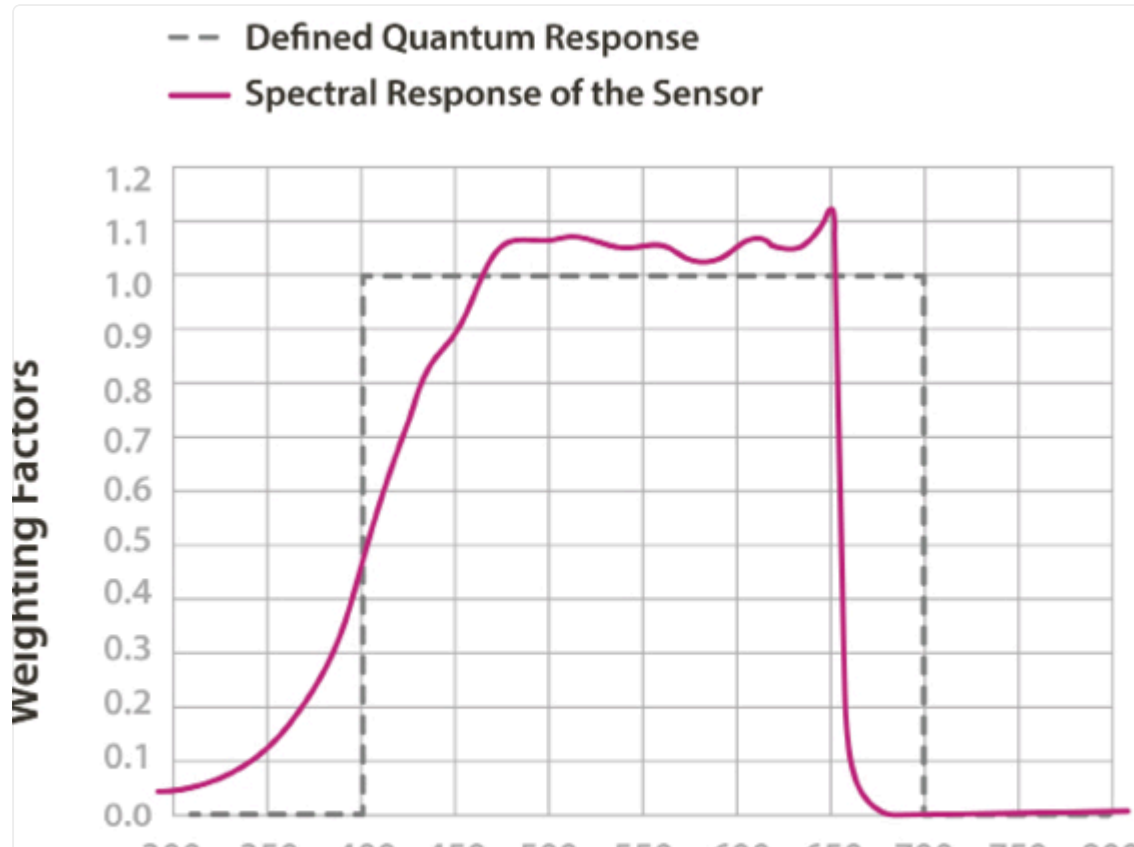
Diameter: 2.4 cm

Height: 2.8 cm

Cable length: 5 m of shielded, twisted-pair wire

Operation temperature: -40 ~ 70 °C

Operation humidity: 0 ~ 100% RH



**Agriculture Sensor
Guide**

Development website



Figure: Graph of the spectral response of the SQ-110 sensor compared to the photosynthetic response of a plant

Measurement process

The SQ-110 sensor, specifically calibrated for the detection of solar radiation, provides at its output a voltage proportional to the intensity of the light in the visible range of the spectrum, a key parameter in photosynthesis processes. It presents a maximum output of 400mV under maximum radiation conditions and a sensitivity of $5.00\mu\text{mol}\cdot\text{m}^{-2}\cdot\text{s}^{-1}/\text{mV}$. In order to improve the accuracy of the reading, this is carried out through a 16 bits analog-to-digital converter that communicates with the microcontroller of the mote through the Digital Bus. It can be configured or read using the functions implemented in the API library

`WaspSensorAgr_v30` for the Agriculture v3.0 Board. The 5 V power supply of this stage is controlled through a digital switch that can be activated and deactivated using the digital pin `DIGITAL7`.

This sensor can only be read by Smart Agriculture PRO version.

For a professional installation, we advise the use the Solar sensors mounting accessory (see Appendix in this Guide). It works for both OEM sensor and Plug & Sense! probe versions.

Reading code:

```

{
  // Instance object
  radiationClass radSensor;
  // Variable to store the radiation conversion value
  float radiation,value;
  Agriculture.ON();
  // Read radiation sensor
  value = radSensor.readRadiation();
  // Conversion from voltage into  $\mu\text{mol}\cdot\text{m}^{-2}\cdot\text{s}^{-1}$ 
  radiation = value / 0.0002;
}

```

You can find a complete example code for reading the SQ-110 sensor in the following link:
<https://development.libelium.com/ag-v30-04-photosynthetic-solar-radiation/>

Socket

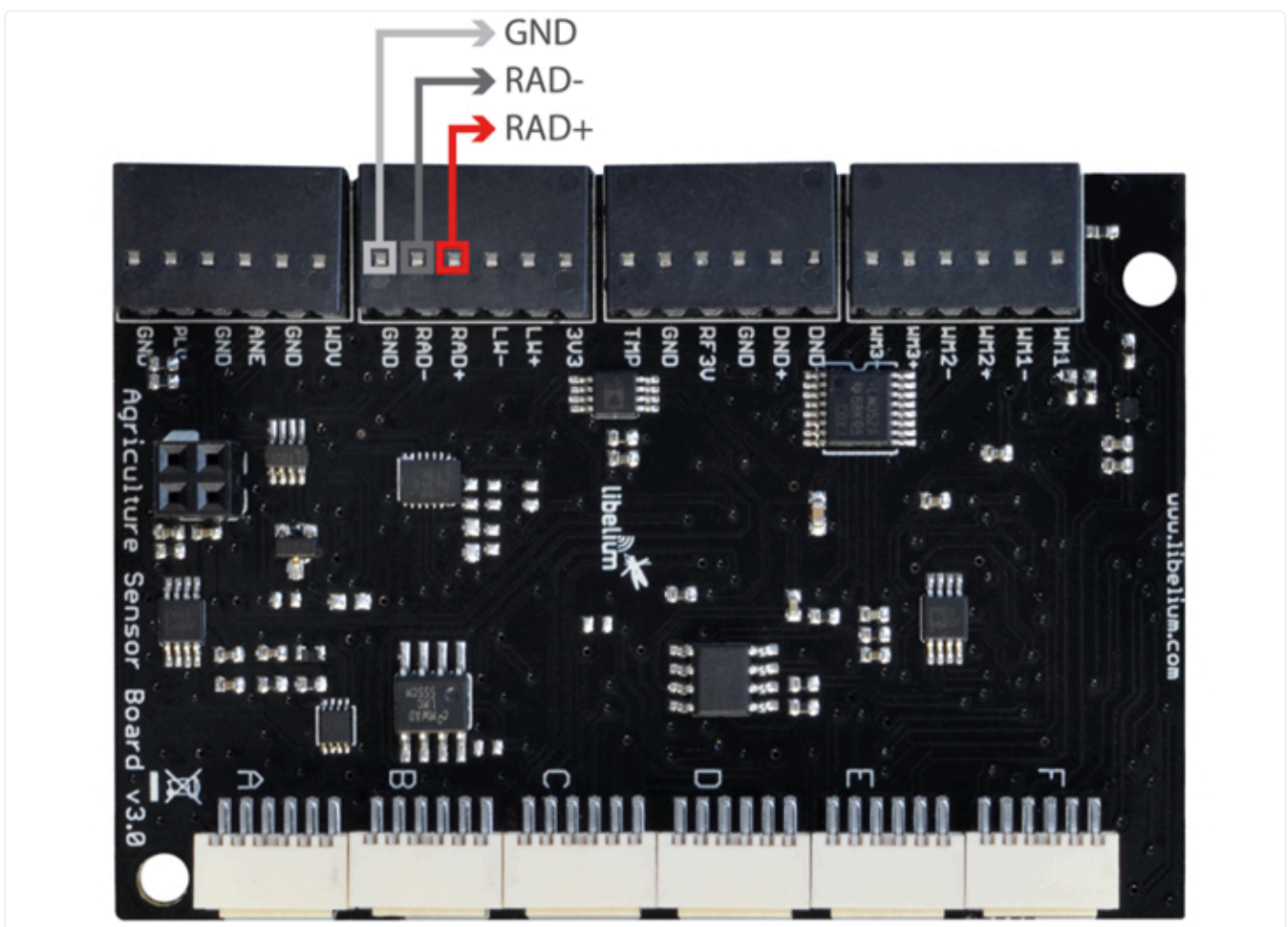


Figure: Image of the socket for the SQ-110 sensor

This sensor is connected to the analog-to-digital converter through the 3 pins of the 2.54 mm pitch terminal block marked in the image in the figure above.

Solar radiation sensor - PAR (Apogee SQ-100x)

Photosynthetically active radiation (PAR) is the radiation that drives photosynthesis and is typically defined as total radiation across a range from 400 to 700 nm. PAR is often expressed as photosynthetic photon flux density (PPFD): photon flux in units of micromoles per square meter per second ($\mu\text{mol}\cdot\text{m}^{-2}\cdot\text{s}^{-1}$).



Figure: SQ-100x sensor

Specifications

- **Operation temperature:** -40 ~ 60 °C
- **Operation humidity:** 0 ~ 100% RH
- **Sensitivity:** 0.1 mV / $\mu\text{mol}\cdot\text{m}^{-2}\cdot\text{s}^{-1}$
- **Calibration factor** (reciprocal of sensitivity): 10 $\mu\text{mol}\cdot\text{m}^{-2}\cdot\text{s}^{-1}$ / mV
- **Non-linearity:** < 1% (up to 4000 $\mu\text{mol}\cdot\text{m}^{-2}\cdot\text{s}^{-1}$ / mV)
- **Non-stability** (long-term drift): <2% per year
- **Spectral range:** 370 ~ 650 nm
- **Repeatability:** <0.5%

- **Diameter:** 2.4 cm
- **Height:** 3.3 cm
- **Cable length:** 5 m

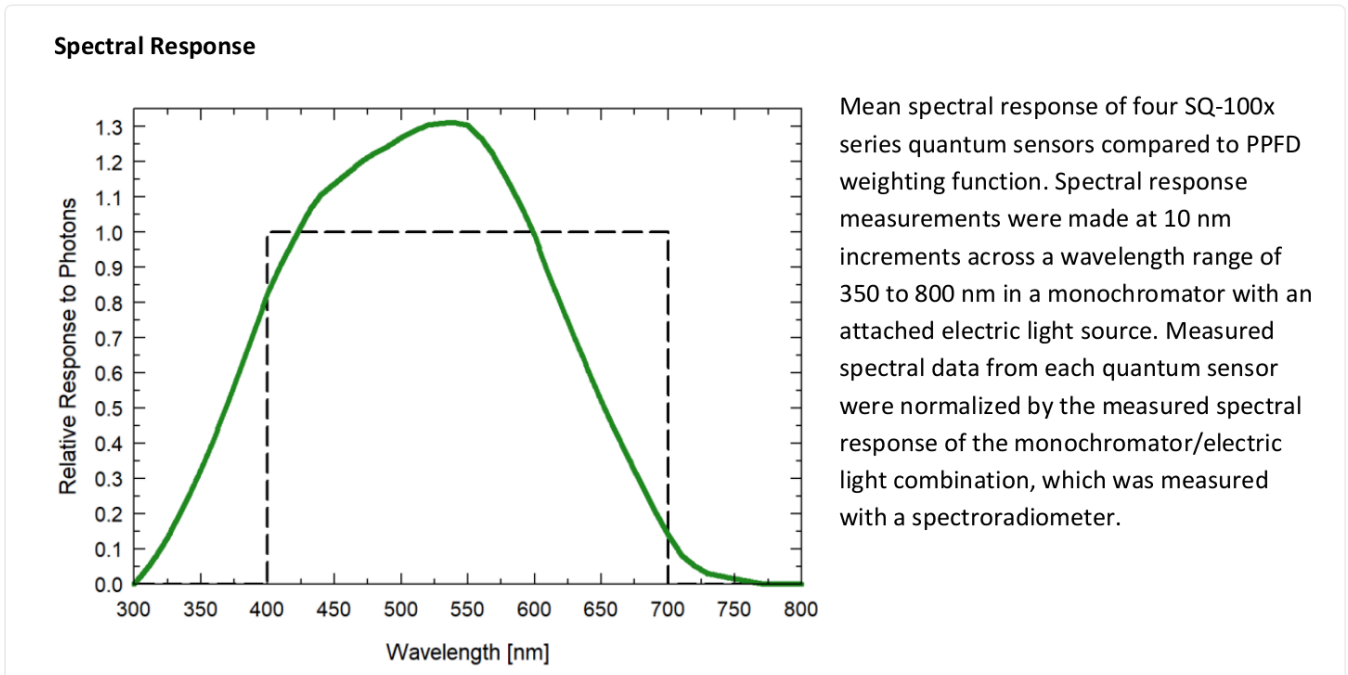


Figure: Graph of the spectral response of the PAR sensor (Apogee SQ-100x) compared to the photosynthetic response of a plant

Measurement process

The SQ-100x sensor, specifically calibrated for the detection of solar radiation, provides at its output a voltage proportional to the intensity of the light in the visible range of the spectrum, a key parameter in photosynthesis processes. It presents a maximum output of 400mV under maximum radiation conditions and a sensitivity of $10.00 \mu\text{mol}\cdot\text{m}^{-2}\cdot\text{s}^{-1}$ per mV. In order to improve the accuracy of the reading, this is carried out through a 16 bits analog-to-digital converter that communicates with the microcontroller of the mote through the Digital Bus. It can be configured or read using the functions implemented in the API library

`WaspSensorAgr_v30` for the Agriculture v3.0 Board. The 5 V power supply of this stage is controlled through a digital switch that can be activated and deactivated using the digital pin `DIGITAL7`.

This sensor can only be read by Smart Agriculture PRO version.

For a professional installation, we advise the use the Solar sensors mounting accessory (see Appendix in this Guide). It works for both OEM sensor and Plug & Sense! probe versions.

Reading code:

```

{
  // Instance object
  radiationClass radSensor;
  // Variable to store the radiation conversion value
  float radiation,value;
  Agriculture.ON();
  // Read radiation sensor
  value = radSensor.readRadiation();
  // Conversion from voltage into  $\text{umol}\cdot\text{m}^{-2}\cdot\text{s}^{-1}$ 
  radiation = value / 0.0001;
}

```

You can find a complete example code for reading the SQ-100x sensor in the following link:
<https://development.libelium.com/ag-v30-04b-photosynthetic-solar-radiation/>

Socket

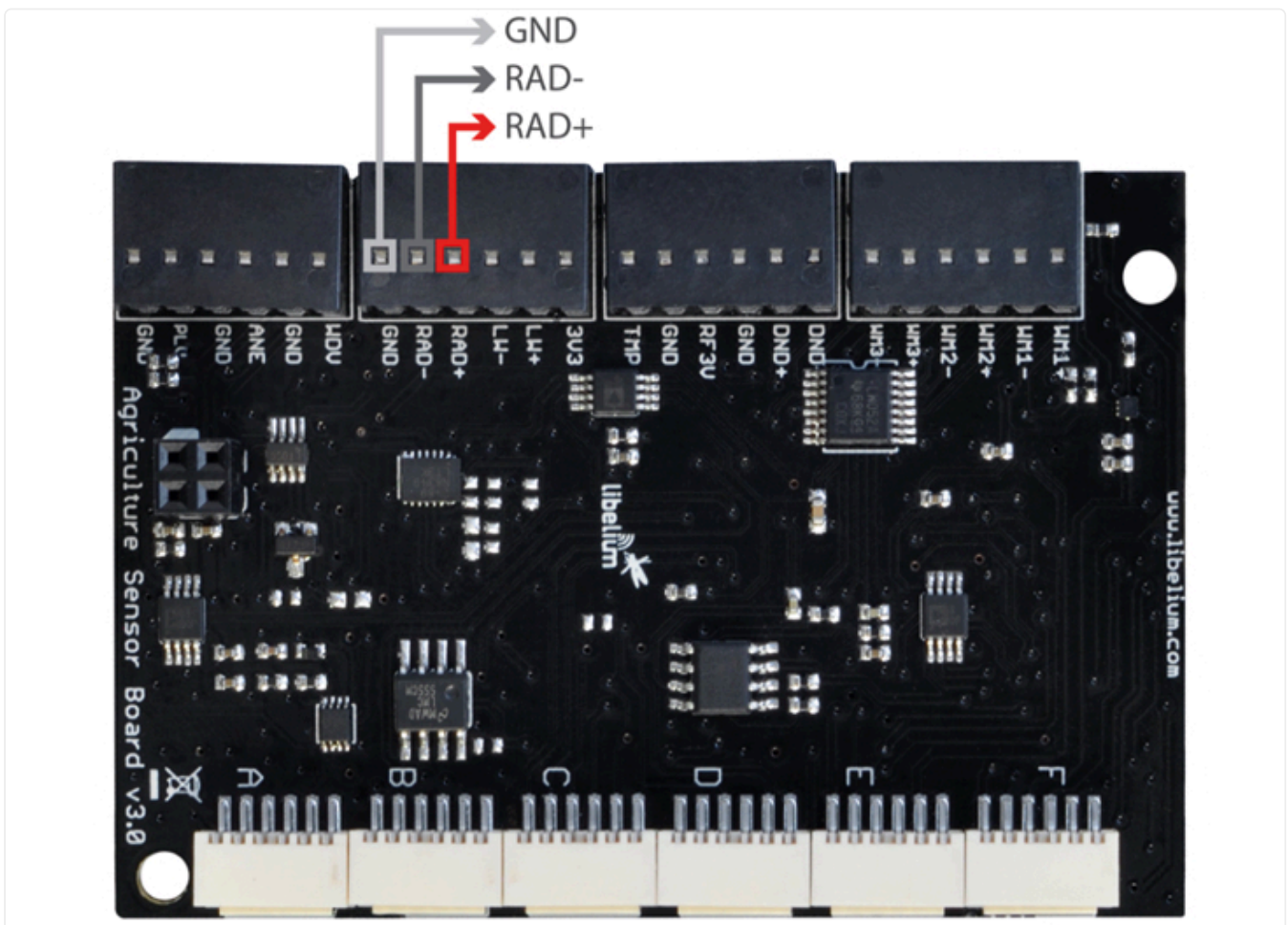


Figure: Image of the socket for the SQ-110 sensor

This sensor is connected to the analog-to-digital converter through the 3 pins of the 2.54 mm pitch terminal block marked in the image in the figure above.

Ultraviolet Radiation sensor (SU-100)



Figure: SU-100 sensor

Specifications

Sensitivity: $0.2 \text{ mV} / \mu\text{mol}\cdot\text{m}^{-2}\cdot\text{s}^{-1}$

Calibration factor: $5.0 \mu\text{mol}\cdot\text{m}^{-2}\cdot\text{s}^{-1}/\text{mV}$

Non-stability (Long-term drift): <3% per year

Non-linearity: <1% (up to $300 \mu\text{mol}\cdot\text{m}^{-2}\cdot\text{s}^{-1}$)

Spectral range: 250 ~ 400 nm

Accuracy: $\pm 10\%$

Repeatability: <1%

Diameter: 2.4 cm

Height: 2.8 cm

Cable length: 5 m shielded, twisted-pair wire

Operation temperature: -40 to 70 °C

Operation humidity: 0 ~ 100 %RH

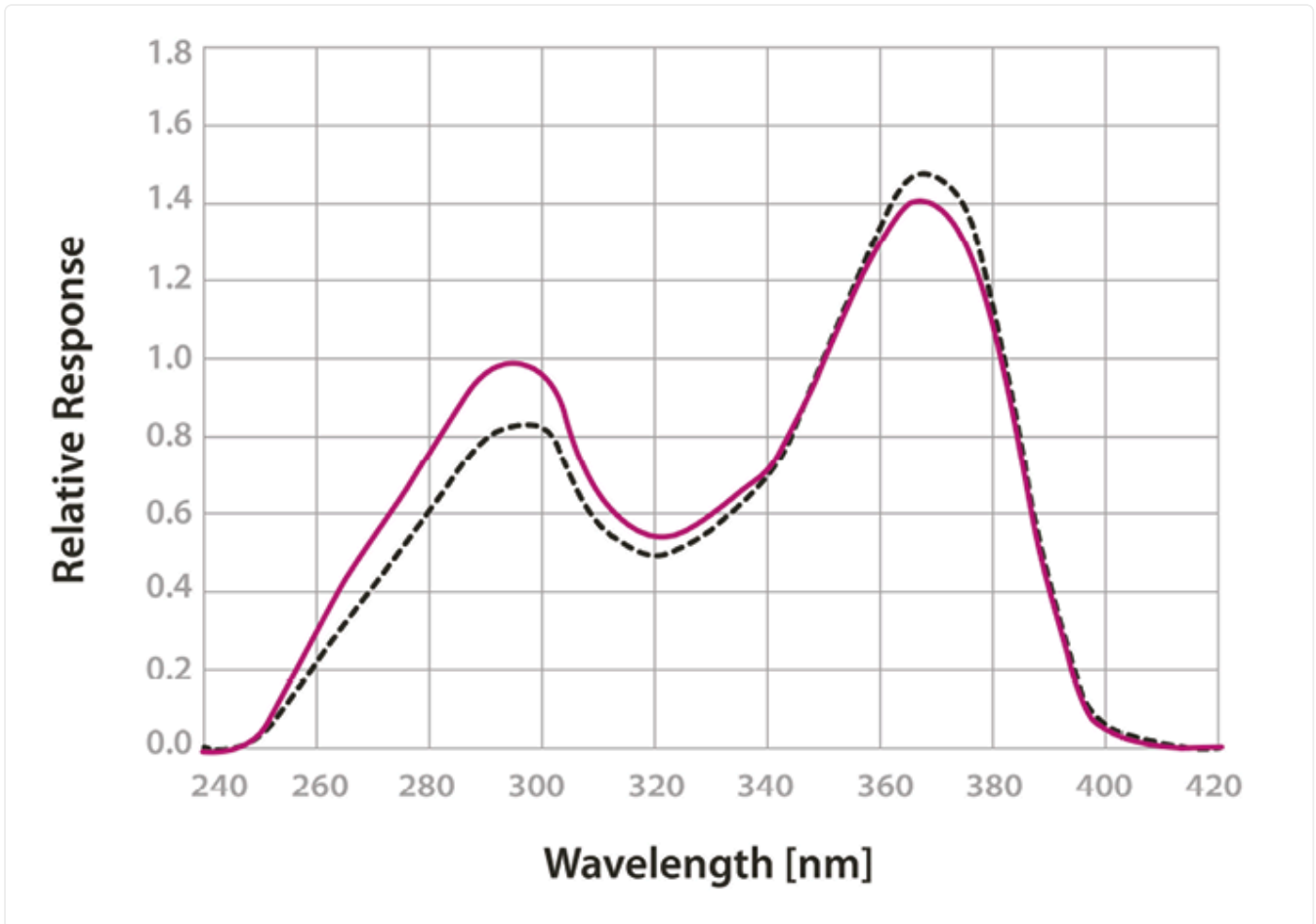


Figure: Graph of the spectral response of the SU-100 sensor compared to the photosynthetic response of a plant

Measurement process

The SU-100 sensor, complementary to the SQ-110 sensor, provides at its output a voltage proportional to the intensity of the light in the ultraviolet range of the spectrum. It presents a maximum output of 26mV under maximum radiation conditions and a sensitivity of $0.2\text{mV} / \mu\text{mol}\cdot\text{m}^{-2}\cdot\text{s}^{-1}$. This sensor is read by the mote through the same 16 bits analog-to-digital converter used with the SQ-110 sensor. It can be configured or read using the functions implemented in the API library `WaspSensorAgr_v30` for the Agriculture v3.0 Board. The 5 V power supply of this stage is controlled through a digital switch that can be activated and deactivated internally by the library.

This sensor can only be read by Smart Agriculture PRO version.

For a professional installation, we advise the use the Solar sensors mounting accessory (see Appendix in this Guide). It works for both OEM sensor and Plug & Sense! probe versions.

Reading code:


```

{
  // Instance object
  radiationClass radSensor;
  // Variable to store the radiation conversion value
  float radiation,value;
  Agriculture.ON();
  // Read radiation sensor
  value = radSensor.readRadiation();
  // Conversion from voltage into umol·m-2·s-1
  radiation = value / 0.0002;
}

```

You can find a complete example code for reading the SU-100 sensor in the following link:
<https://development.libelium.com/ag-v30-03-uv-solar-radiation/>

i The function delivers the SU-100 measure for the photon flux, in $\text{umol}\cdot\text{m}^{-2}\cdot\text{s}^{-1}$ units ($\text{umol}/\text{m}^2/\text{s}$ = number of photons which cross a m^2 , each second). This value can be transformed to UV irradiance value, in W/m^2 units, multiplying by 0.327.

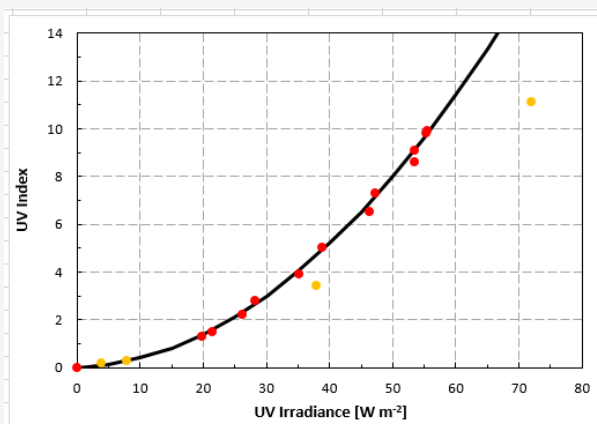
The UV Index is based on a weighted measurement of UV radiation, using the Erythemal action spectrum as the weights. This means that the W/m^2 measurements of total UV irradiance cannot be simply related to the UV Index scale.

However you can estimate the UV Index from UV irradiance. Apogee estimated the following polynomial relationship between total UV irradiance as measured by the SU-100-SS sensor and the UV Index:

$$\text{UV Index} \approx 0.003 * (\text{UV_Wm-2})^2 + (0.01 * \text{UV_Wm-2})$$

(Where UV_Wm-2 is the UV irradiance measurement in W/m^2)

This renders a nearly linear relationship that looks like this:



Socket

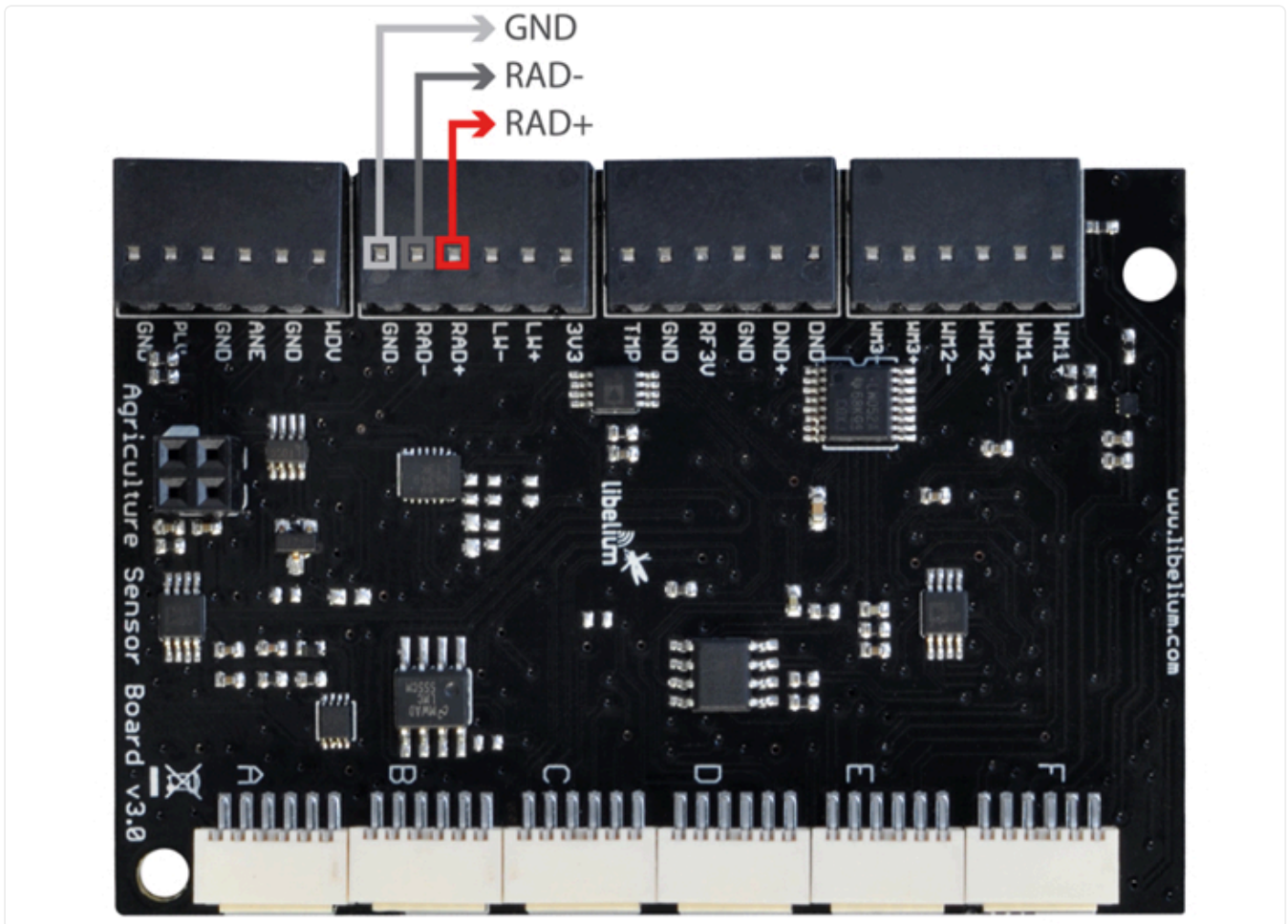


Figure: Image of the socket for the SU-100 sensor

This sensor is connected to the analog-to-digital converter through the three pins of the 2.54 mm pitch terminal block marked in the image in the figure above.

Weather station (WS-3000)

This weather station consists of three different sensors, described in detail below: a wind vane, an anemometer and a pluviometer. It connects to Waspote through six wires that must be connected to the terminal block shown in the corresponding figures, being the anemometer connected to the vane through an RJ11 socket.



Figure: Image of the Weather station WS-3000

Anemometer



Specifications

Sensitivity: 2.4km/h / turn

Wind Speed Range: 0 ~ 240km/h

Height: 7.1 cm

Arm length: 8.9 cm

Connector: RJ11

Measurement process

The anemometer chosen for Waspnote consists of a reed switch normally open that closes for a short period of time when the arms of the anemometer complete a 180° angle, so the output is a digital signal whose frequency will be proportional to the wind speed. There will be 2 events in each complete turn. That signal can be read from the digital pin `DIGITAL2` of the Waspnote. The function of the library `readAnemometer()` returns the wind speed value in kilometers per hour (km/h). The power of the sensor and of the electronics around it may be connected or disconnected using a digital switch controlled internally by the library.

Reading code:

```
{  
  // Variable to store the anemometer value  
  float anemometer;  
  // Instance objects  
  weatherStationClass anemSensor;  
  Agriculture.ON();  
  // Read Anemometer sensor  
  anemometer = anemSensor.readAnemometer();  
}
```

You can find a complete example code for reading the whole Weather Station WS-3000 data sensor in the following link: <https://development.libelium.com/ag-v30-08-weather-station/>

Socket

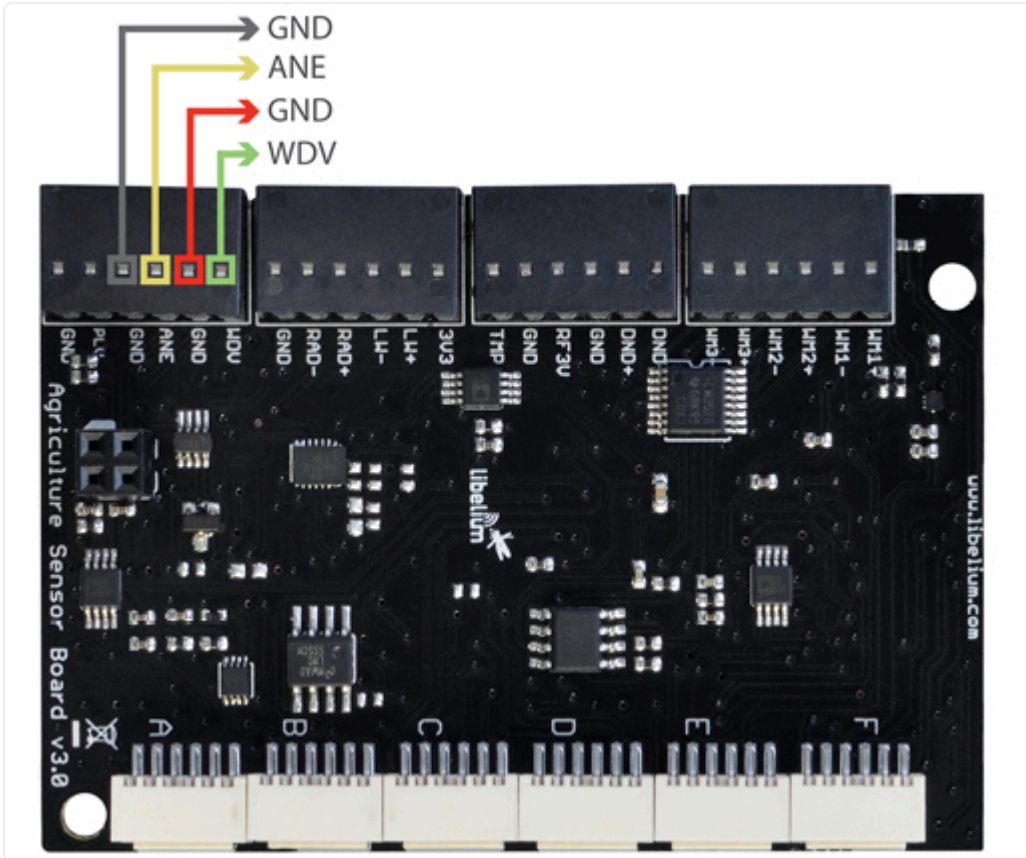


Figure: Image of the connector for the anemometer

The way to connect the anemometer to the Agriculture v3.0 Board is through the vane: the anemometer's cable must be plugged into the socket that can be found on the base of the vane.

Wind vane



Figure: Wind vane

Specifications

Height: 8.9 cm

Length: 17.8 cm

Maximum accuracy: 22.5°

Resistance range: 688 Ω ~ 120 k Ω

Measurement process

The wind vane consists of a basement that turns freely on a platform endowed with a net of eight resistances connected to eight switches that are normally and are closed (one or two) when a magnet in the basement acts on them, which permits us to distinguish up to 16 different positions (the equivalent to a resolution of 22.5°). The equivalent resistance of the wind vane, along with a 10 k Ω resistance, form a voltage divider, powered at 3.3 V through a digital switch controlled by the `ANALOG1` pin, whose output can be measured in the analog input `ANALOG5`. The function of the library `readVaneDirection()` also stores in variable `vane_direction` an 8 bits value which corresponds with an identifier of the pointing direction.

Below, a table with the different values that the equivalent resistance of the wind vane may take is shown, along with the direction corresponding to each value:

Direction (degrees)	Resistance (k Ω)	Voltage (V)	Identifier
0	33	2.53	<code>SENS_AGR_VANE_N</code>
22.5	6.57	1.31	<code>SENS_AGR_VANE_NNE</code>
45	8.2	1.49	<code>SENS_AGR_VANE_NE</code>
67.5	0.891	0.27	<code>SENS_AGR_VANE_ENE</code>
90	1	0.3	<code>SENS_AGR_VANE_E</code>
112.5	0.688	0.21	<code>SENS_AGR_VANE_ESE</code>
135	2.2	0.59	<code>SENS_AGR_VANE_SE</code>
157.5	1.41	0.41	<code>SENS_AGR_VANE_SSE</code>
180	3.9	0.92	<code>SENS_AGR_VANE_S</code>
202.5	3.14	0.79	<code>SENS_AGR_VANE_SSW</code>
225	16	2.03	<code>SENS_AGR_VANE_SW</code>
247.5	14.12	1.93	<code>SENS_AGR_VANE_WSW</code>

270	120	3.05	SENS_AGR_VANE_W
292.5	42.12	2.67	SENS_AGR_VANE_WNW
315	64.9	2.86	SENS_AGR_VANE_NW
337.5	21.88	2.26	SENS_AGR_VANE_NNW

Reading code:

```
{
// Variable to store the vane value
int vane;
// Instance objects
weatherStationClass vaneSensor;
Agriculture.ON();
// Read Vane sensor
vane = vaneSensor.readVaneDirection();
}
```

You can find a complete example code for reading the whole Weather Station WS-3000 data sensor in the following link: <https://development.libelium.com/ag-v30-08-weather-station/>

Besides, it is recommended to use the function `getVaneFiltered()` in order to perform a mean filtered measurement during a specified period of time. Thus, mechanical fluctuations will be avoided and a more accurate measurement will be done. See example here:

<https://development.libelium.com/ag-v30-09-wind-vane/>

Socket

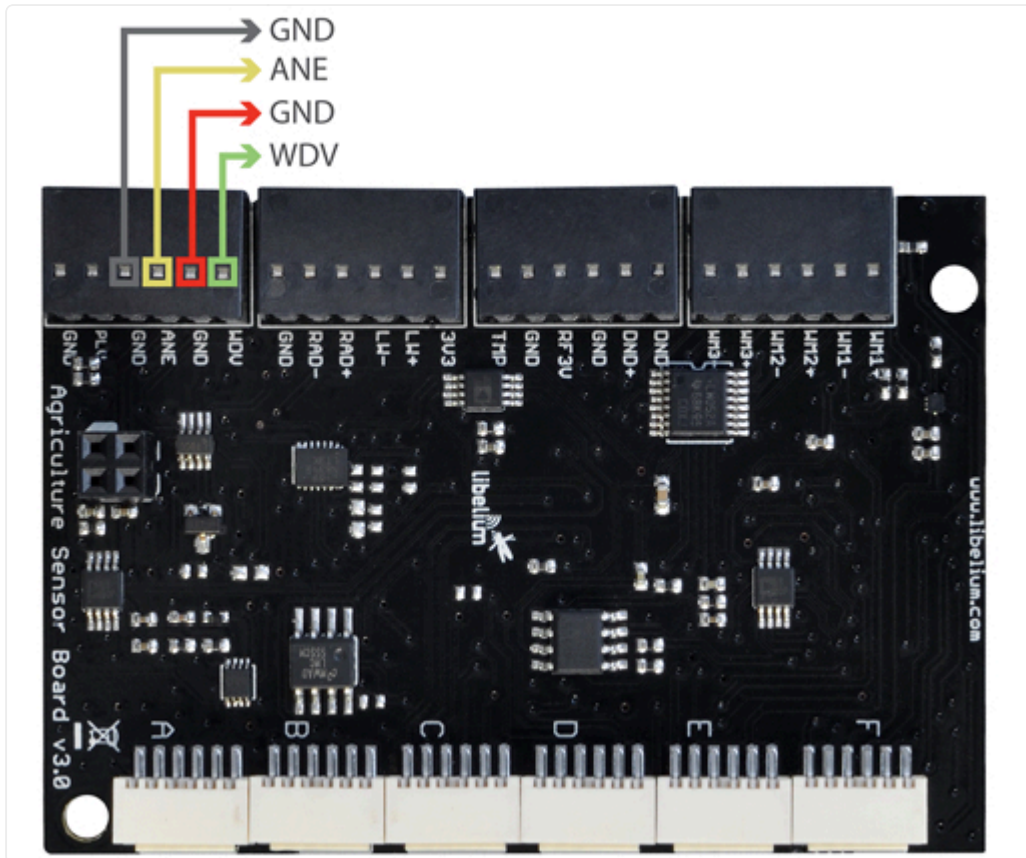


Figure: Image of the connector for the wind vane

The wind vane is connected to the board through four pins of a terminal block whose correspondence with the sensor's wire is shown in the figure above.

Pluviometer



Figure: Pluviometer

Specifications

Height: 9.05 cm

Length: 23 cm

Bucket capacity: 0.28 mm of rain

Measurement process

The pluviometer consists of a small bucket that, once completely filled (0.28 mm of water approximately), closes a switch, emptying automatically afterwards. The sensor is connected directly to the Waspote `DIGITAL4` digital input through a pull-up resistance and to the interruption pin `RXD1`, allowing the triggering of an interruption of the microcontroller when the rainfall provokes a bucket emptying event. Since the consumption of this sensor, in absence of rain, is null, no elements of power control have been added for it.

The library defines several functions to measure from the pluviometer. An inner structure is being used in order to store the number of pluviometer interruptions that took place for the last 24 hours (regarding the RTC settings). The function used for storing a new pulse is called `storePulse()` and it should be called every time a new pluviometer interruption is generated. On the other hand, it is possible to know three different values given by the following functions:

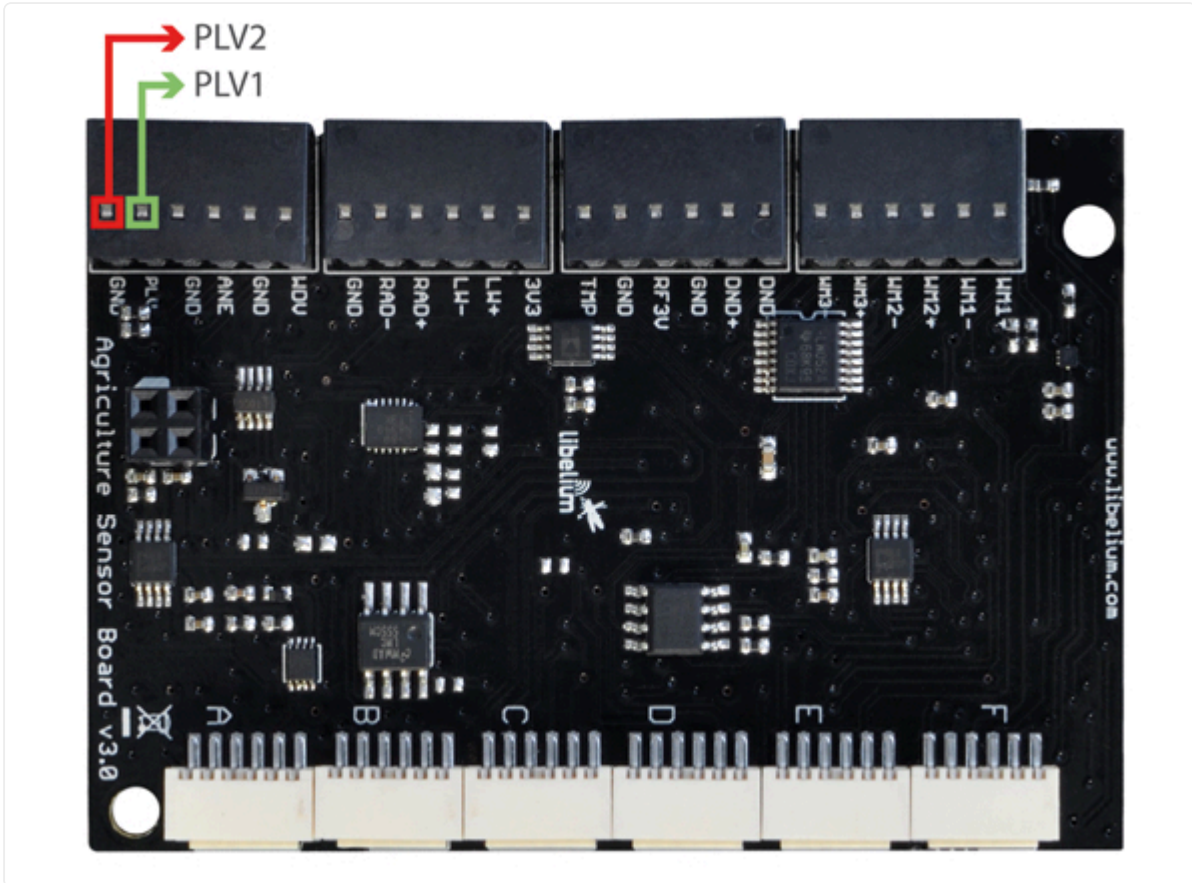
- `readPluviometerCurrent()` calculates the precipitations in mm for the current period. For example, it is 10:42 am, so this function returns the mm of rainfall that took place in the last 42 minutes.
- `readPluviometerHour()` calculates the precipitations in the previous one-hour period. For example, it is 10:42 am, so this function returns the mm of rainfall between 9 am and 10 am.
- `readPluviometerDay()` calculates the accumulated precipitation in the last 24 hours.

Tip: the user can apply a little of paraffin on the pluviometer's upper surface in order to help the rain drops to flow down to the inside of the sensor.

You can find a complete example code for reading the whole Weather Station WS-3000 data sensor in the following link: <https://development.libelium.com/ag-v30-08-weather-station/>

There is an example of how to use the pluviometer sensor separately from the Weather Station: <https://development.libelium.com/ag-v30-10-pluviometer/>

Socket



In the image in the figure above we can see marked the two pins in the terminal block that correspond to the pluviometer (no polarity is required).

Ultrasonic Sensor (MaxSonar® from MaxBotix™)

Specifications

I2CXL-MaxSonar®-MB7040™

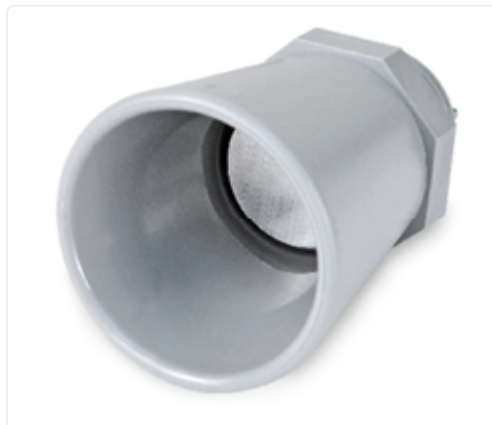


Figure: Ultrasonic I2CXL- MaxSonar®-MB7040 from Max- Botix™ sensor

Operation frequency: 42 kHz

Maximum detection distance: 765 cm

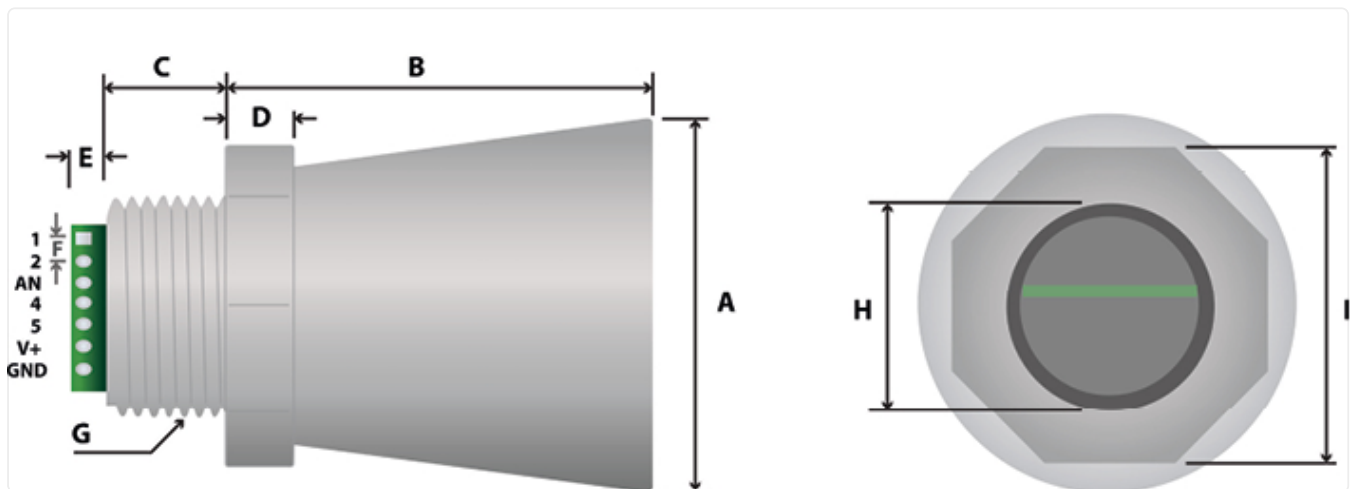
Interface: Digital Bus

Power supply: 3.3 ~ 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)



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	<i>3/4" National Pipe Thread Straight</i>	
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

 Only one MB7040 sensor is supported at the same time.

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):

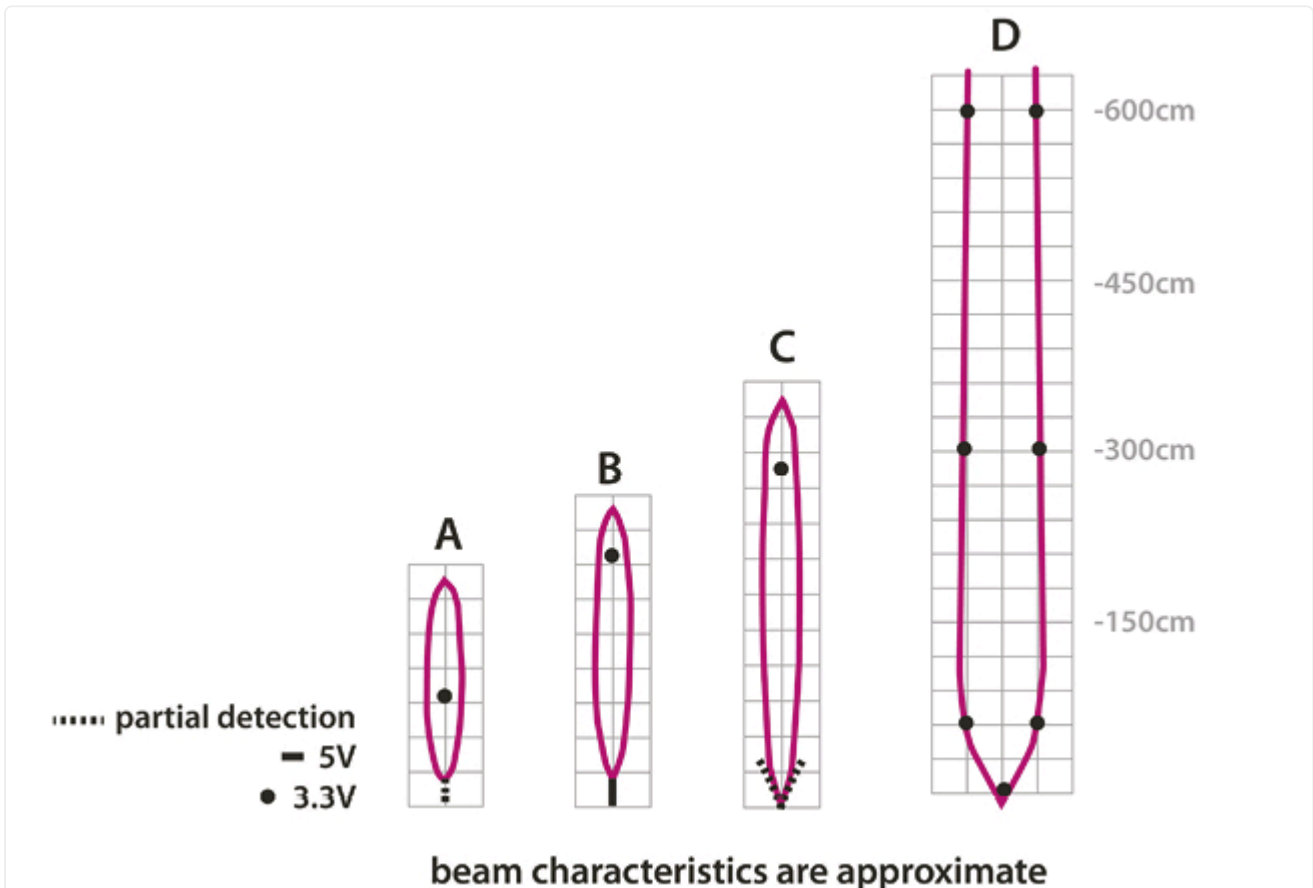


Figure: Diagram of the sensor beam extracted from the data sheet of the XL-MaxSonar®-WRA1™ sensor from MaxBotix

I2CXL-MaxSonar®-MB1202™



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

Operation frequency: 42 kHz

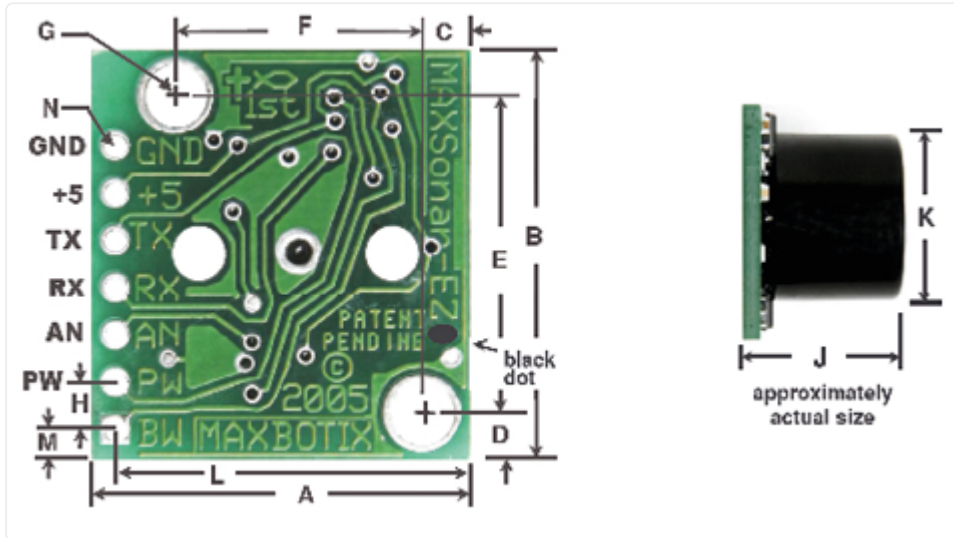
Maximum detection distance: 645 cm

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) - 3mA (powered at 5 V)

Usage: Indoors



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

i Only one MB1202 sensor is supported at the same time.

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):

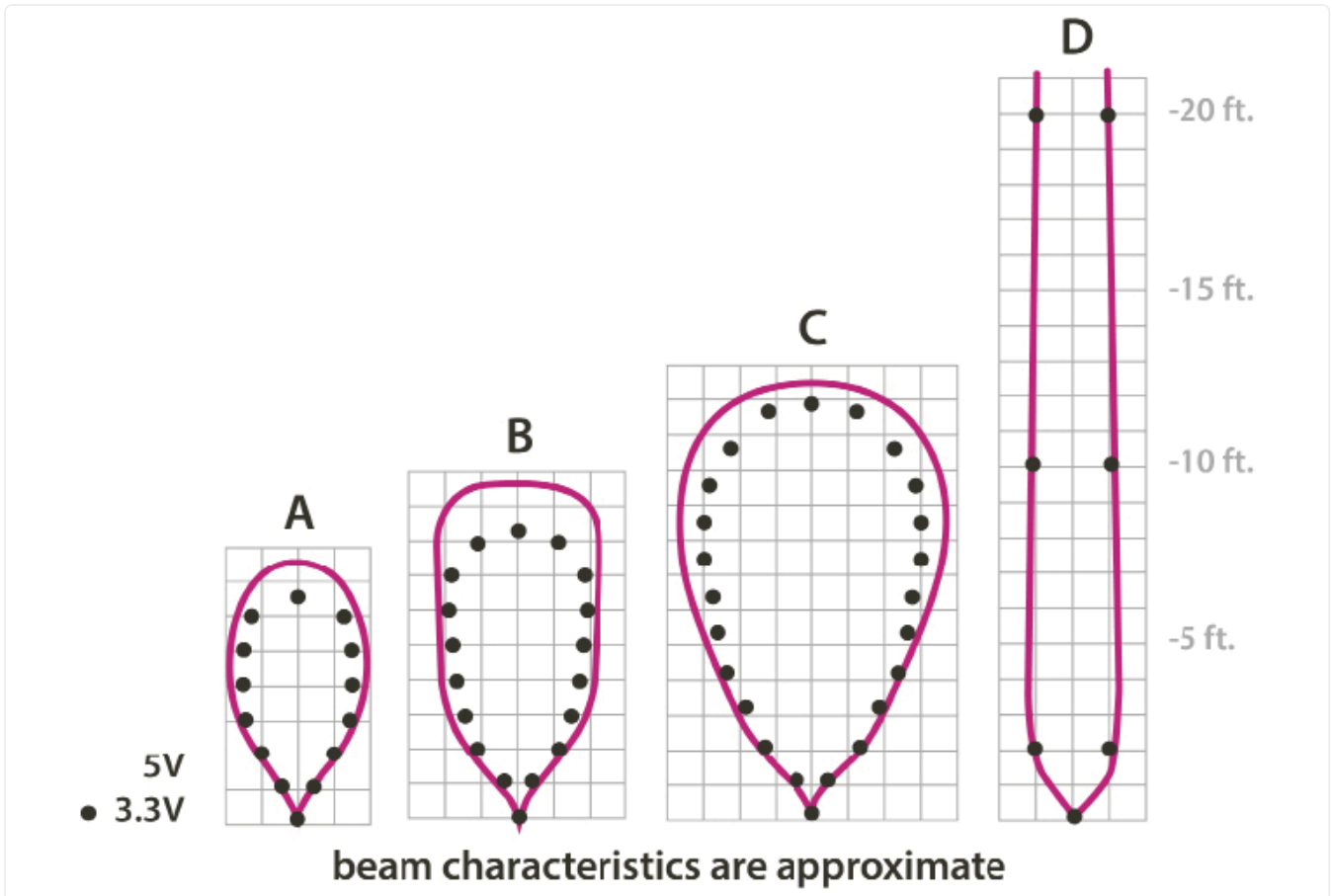


Figure: Diagram of the sensor beam extracted from the data sheet of the Ultrasonic I2CXL-MaxSonar®-MB1202 sensor from MaxBotix

Measurement Process

The MaxSonar® sensors from MaxBotix connects through 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.

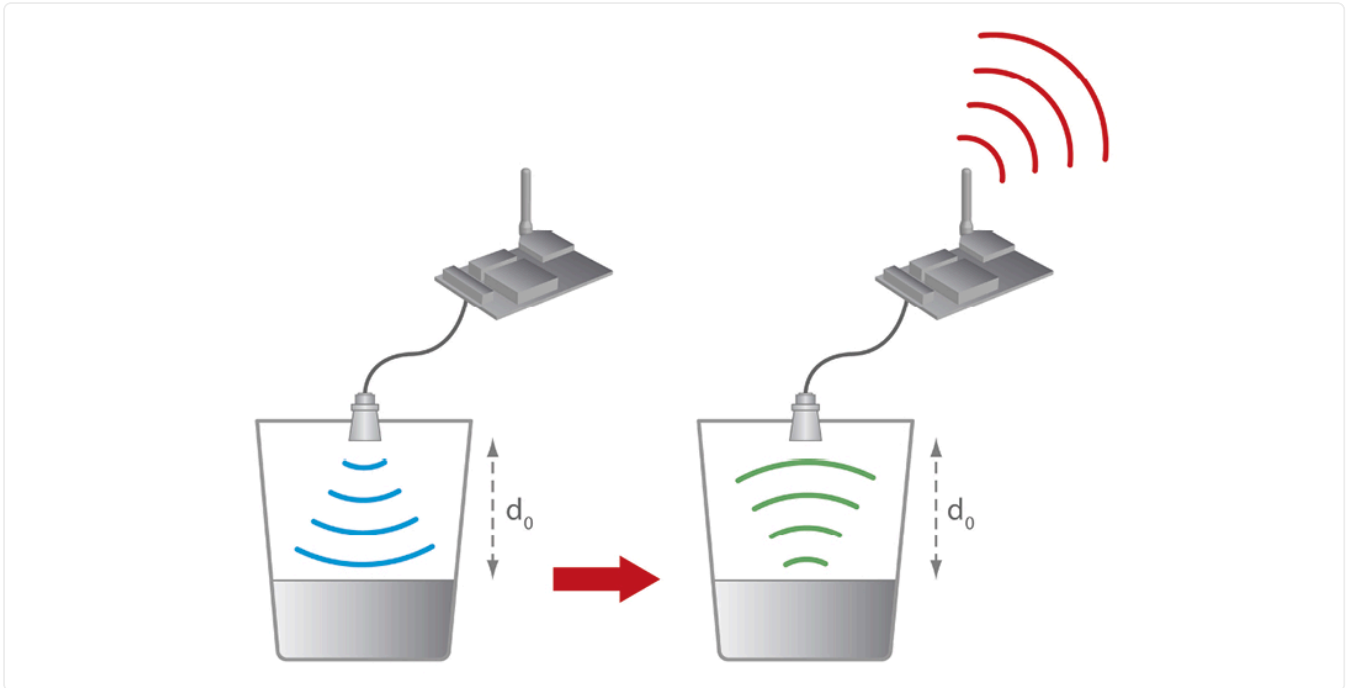


Figure: Examples of application for the MaxSonar® sensors

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;
  Agriculture.ON();
  distance = Agriculture.getDistance();
}
```

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

<https://development.libelium.com/ag-v30-13-ultrasound-sensor/>

Socket

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

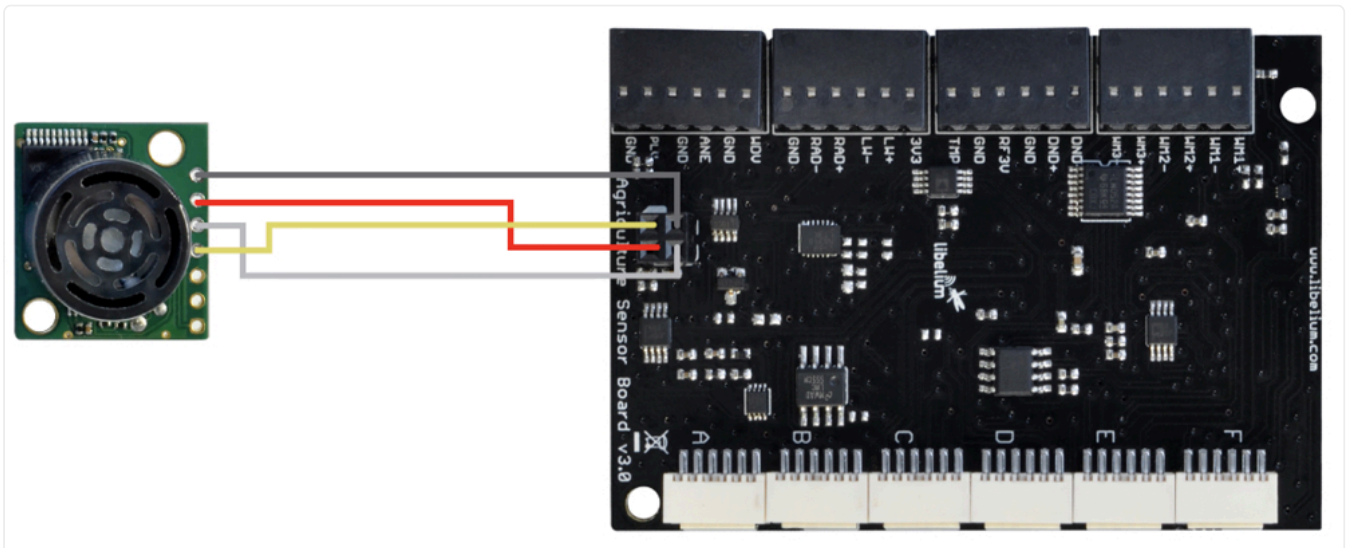
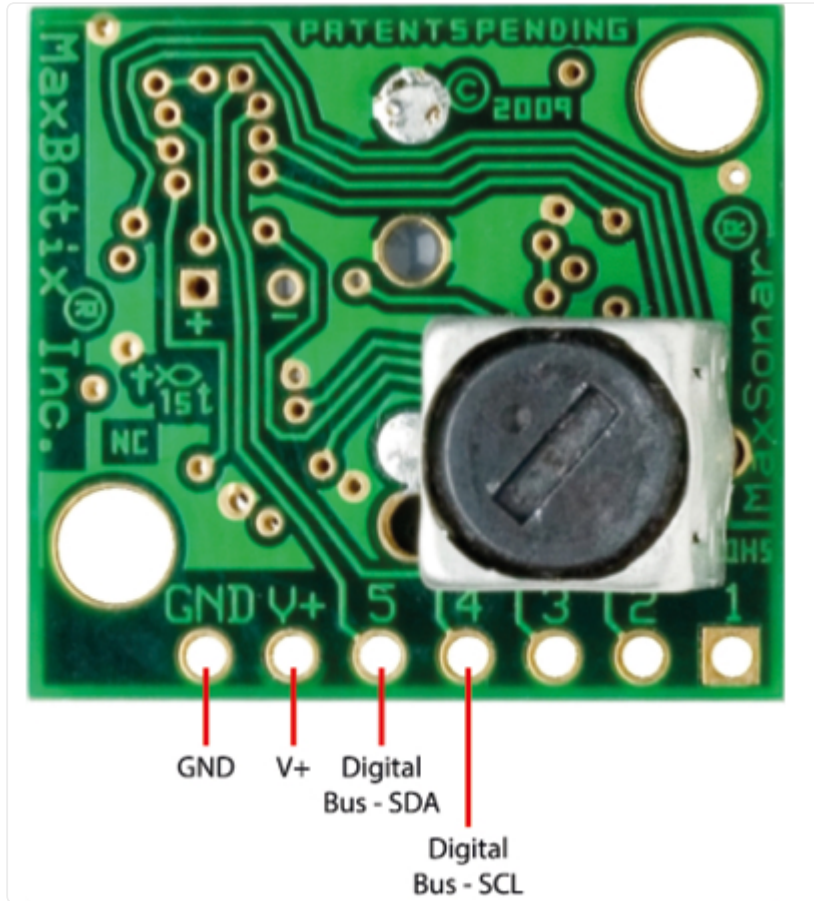


Figure: Image of the socket for connecting the MaxSonar® Sensors

Luminosity sensor (TSL2561)

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 μ A

Operating temperature: -30 ~ +70 °C



Figure: Image of the Luminosity sensor

i Only one TSL2561 sensor is supported at the same time.

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:

```

{
  uint32_t luxes = 0;
  Agriculture.ON();
  // Reads the luxes sensor
  // Select the option of measure:
  //   - OUTDOOR
  //   - INDOOR
  luxes = Agriculture.getLuxes(INDOOR);
}

```

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

<https://development.libelium.com/ag-v30-14-luxes-sensor/>

Socket

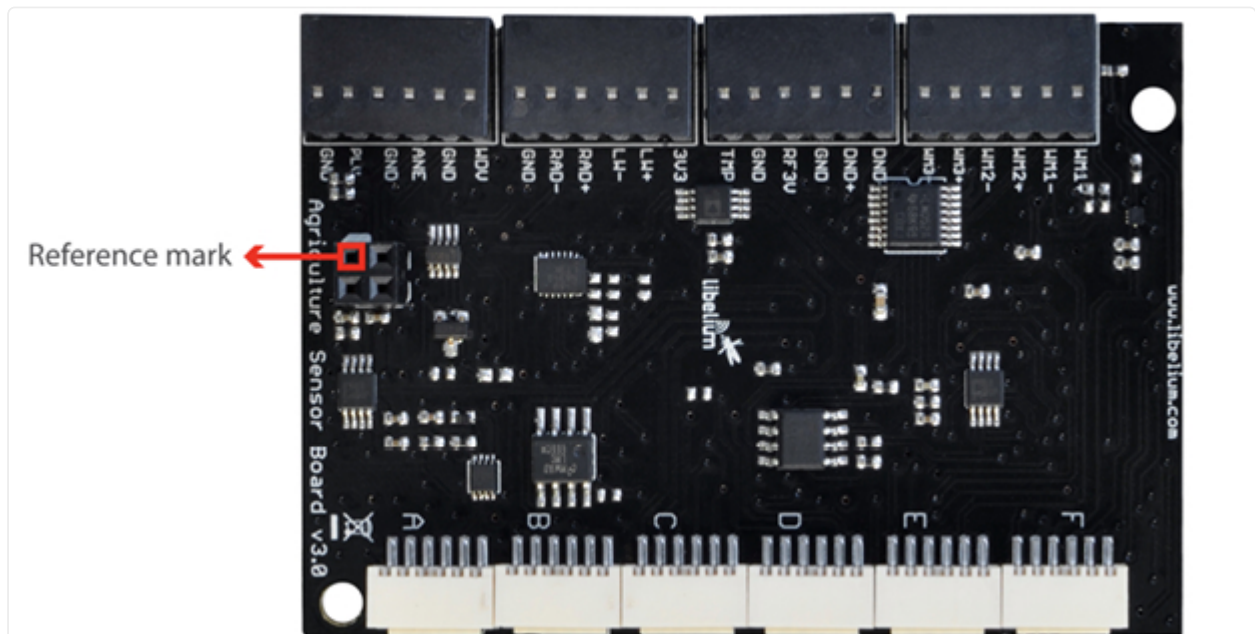


Figure: Image of the socket for the luxes sensor

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 Agriculture sensor board.

Sockets for casing

In case the Agriculture v3.0 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 above the input pin and pull off the wire softly.

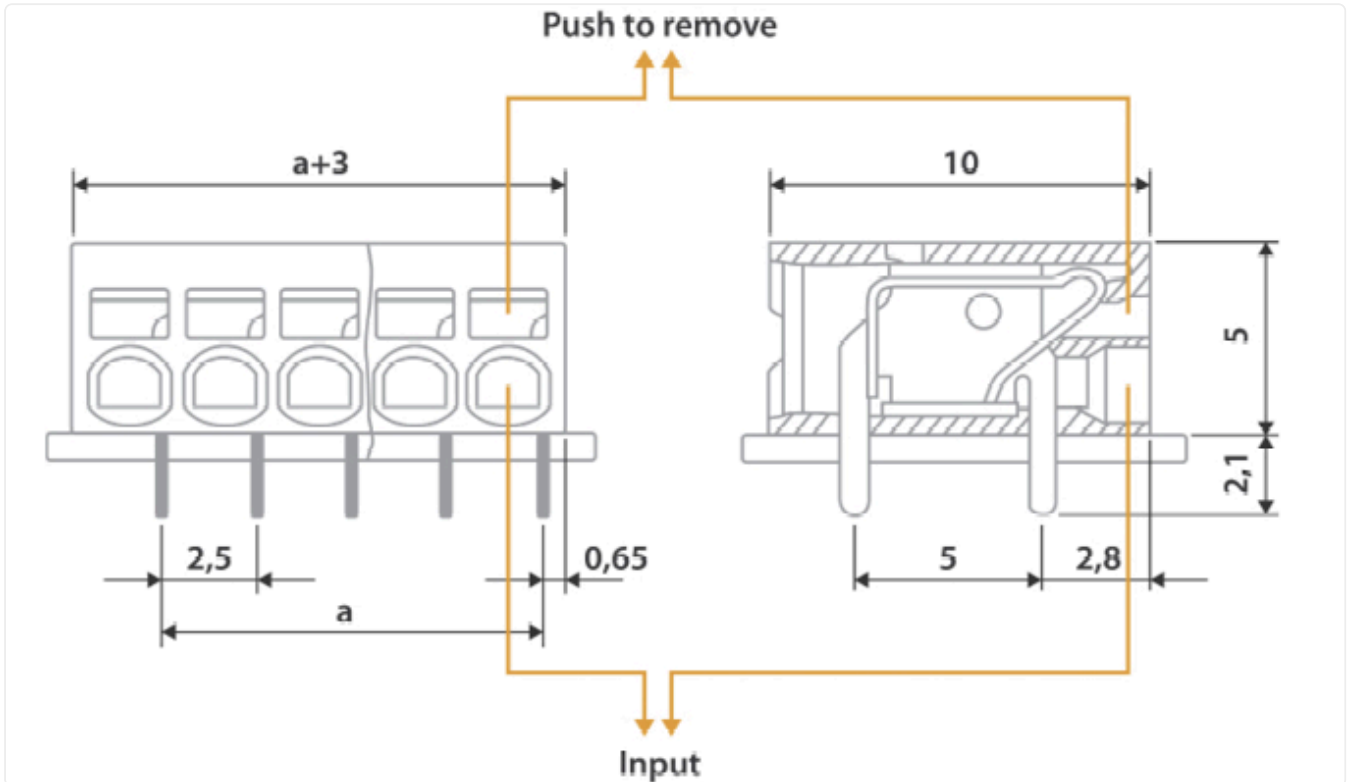


Figure: Diagram of a socket extracted from the Phoenix Contact data sheet

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.

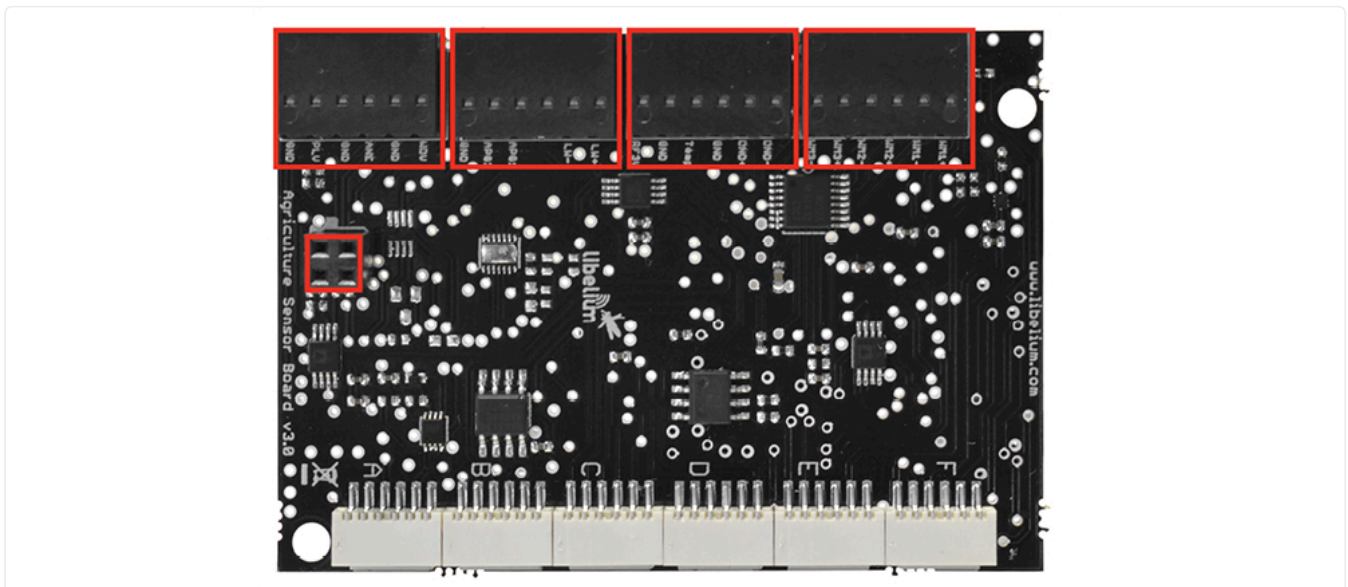


Figure: Image of the sockets for casing applications

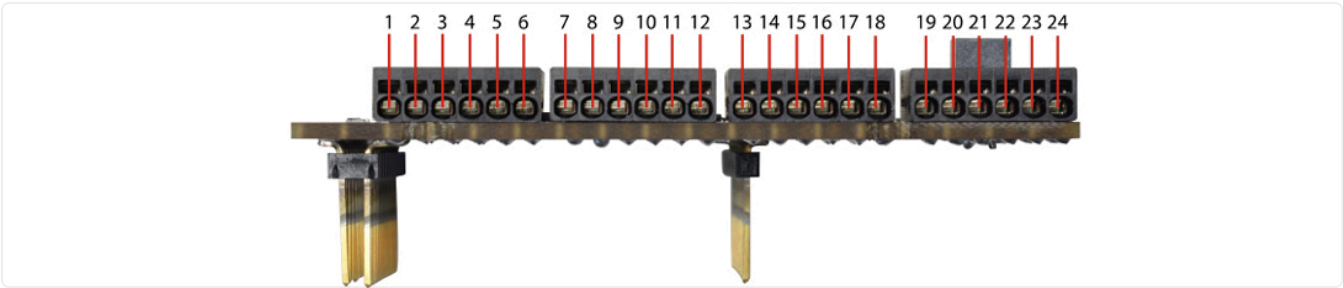


Figure: Image of the pin correspondence between the sockets and the sensors

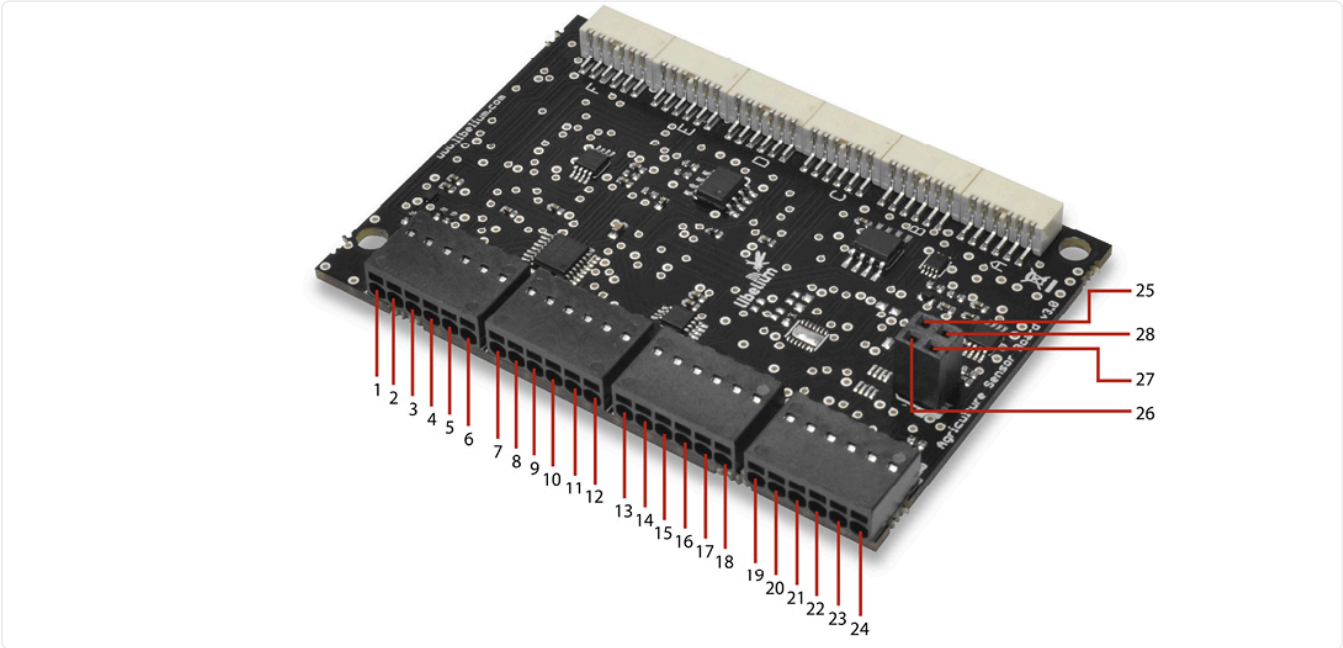


Figure: Image of the pin correspondence between the sockets and the sensors

Sensor	Pin	Function
Watermark	1	Watermark 1 +
Watermark	2	Watermark 1 -
Watermark	3	Watermark 2 +
Watermark	4	Watermark 2 -
Watermark	5	Watermark 3 +
Watermark	6	Watermark 3 -
Dendrometer	7	Dendrometer -
Dendrometer	8	Dendrometer +
Dendrometer	9	GND
Dendrometer	10	REF3V
Pt-1000/DS18B20	11	GND

Pt-1000/DS18B20	12	Pt-1000/DS18B20
Pt-1000/DS18B20	13	+3V3
Leaf Wetness	14	Leaf Wetness +
Leaf Wetness	15	Leaf Wetness -
Solar Radiation	16	Radiation +
Solar Radiation	17	Radiation -
Solar Radiation	18	GND
Weather Station WS-3000 (Vane)	19	Vane Output (Yellow)
Weather Station WS-3000 (Vane)	20	GND (Black)
Weather Station WS-3000 (Anemometer)	21	Anemometer Output (Green)
Weather Station WS-3000 (Anemometer)	22	GND (Red)
Weather Station WS-3000 (Pluviometer)	23	Pluviometer
Weather Station WS-3000 (Pluviometer)	24	Pluviometer
Digital Bus	25	Digital Bus (SCL)
Digital Bus	26	GND
Digital Bus	27	Digital Bus (SDA)
Digital Bus	28	VCC

[Previous
Hardware](#)

[Next
Board configuration and programming](#)

Last updated 2 years ago