GROLAB®

EN

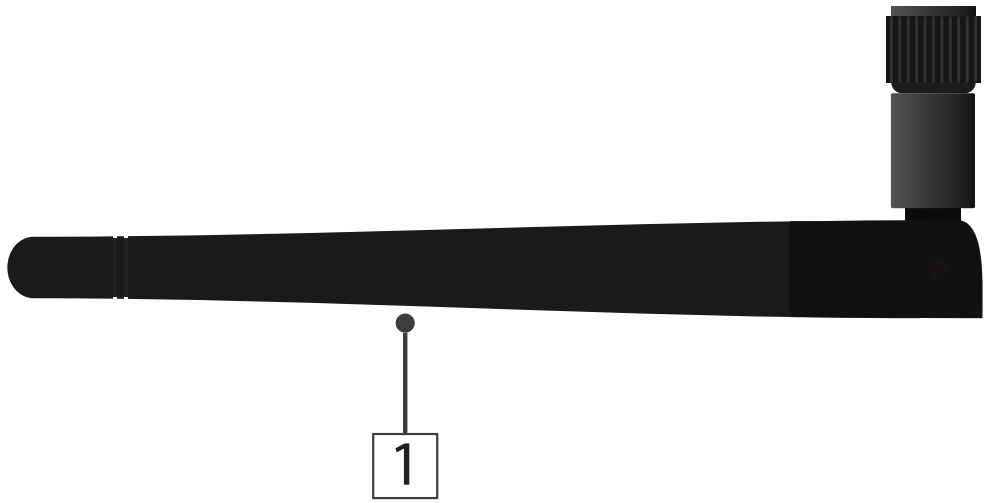
USERBOT

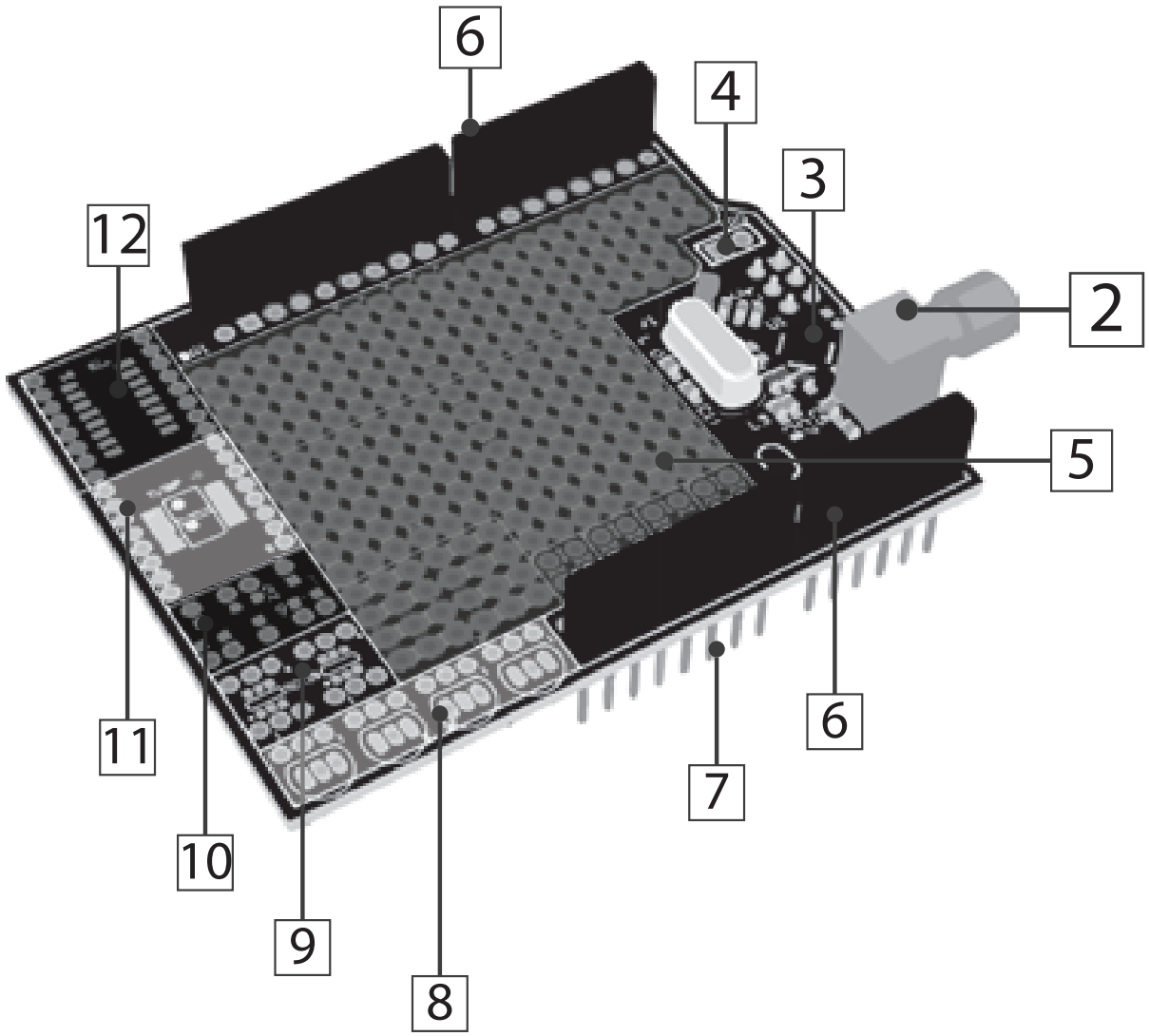
(HW01)

INSTALLATION MANUAL

www.opengrow.pt

GL-MN-001A-02


1


6
4
3
12
2
5
11
6
7
10
9
8

1 RF communication antenna

2 RF antenna outlet

3 RF module

4 RF interrupt pin

5 Universal breakout board

6 Female pin headers

7 Arduino connection pins

8 4 x TO-92

9 2 x SOT26

10 3 x SSOT223

11 2 x TSSOP-8 (=) 1 x TSSOP-16

12 SOIC16

SPECIFICATIONS TABLE

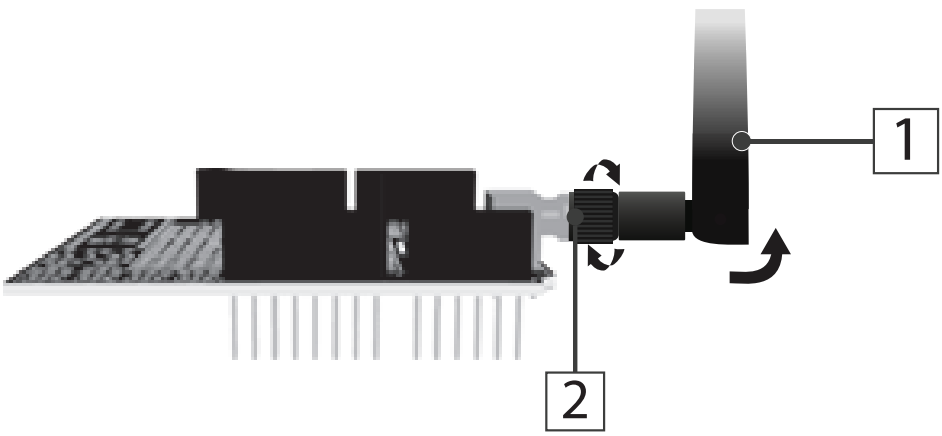
Hardware Version	HW01
Dimensions	68.6mm x 53.4mm x20.01mm
Net Weight	~20 grams (-0.71 oz)
Exterior	Material: FR-4 Color: Green
Operating Voltage	3.3 VDC
Power Consumption	15mA - 0.05W
Connections	SMA Female Female Pin Headers (Arduino connection/extension)
SMD Sockets	1 x SOIC16 2 x TSSOP 8 (=) 1 x TSSOP 16 3 x SSOT223 2 x SOT26
TH Sockets	4 x TO-92
BreakOut Board Spacing	2.54mm - 100mil DIP/1206/0805/0603
Includes	Antenna
Inter-Module Communication	Radio Frequency - 2.4GHz
Expected Service Life	>5-years
Warranty	3-year limited hardware warranty

INSTALLATION ①

Base connections

Screw the communication antenna (1) into the module's RF antenna outlet (2).

For correct communication, turn the communications antenna (1) to position it vertically in the direction of the module.



At this point, the UserBot shield is ready to be assembled into the Arduino.

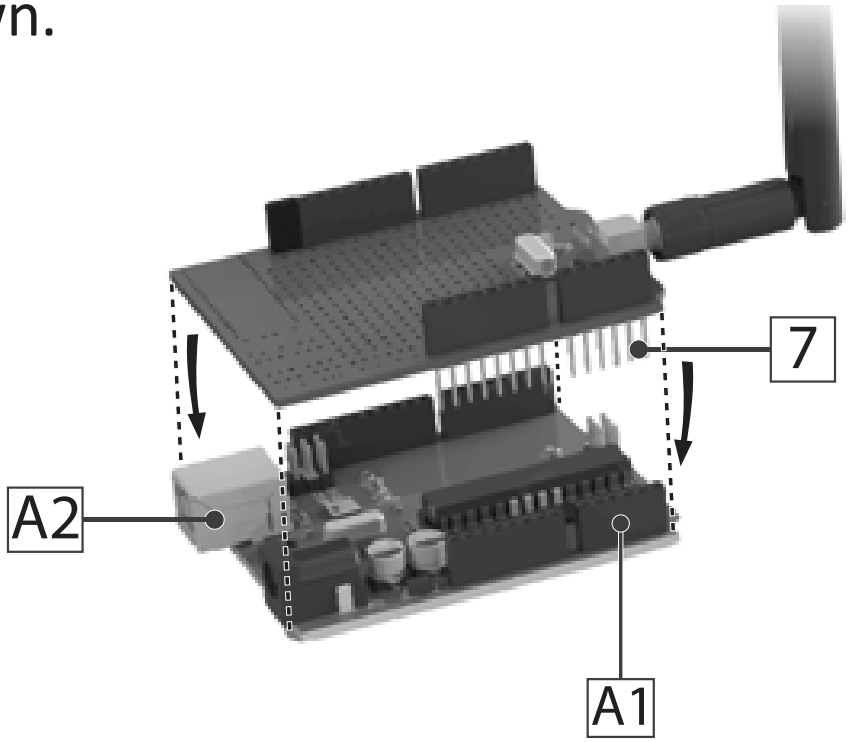
Connection to Arduino

First, check if your Arduino model is compatible with UserBot shield. Compatible models:

- **Arduino UNO;**
- **Arduino Duemilanove;**
- **Arduino Leonardo with headers;**
- **Genuino Zero.**

Before connecting the UserBot shield to the Arduino, make sure the Arduino is correctly assembled and ready to be used (follow the official Arduino instructions manual). Also, make sure your Arduino is not connected to the power supply.

Place the Arduino on a firm surface, far from flood-prone areas. Make sure the Arduino female pin headers (A1) are faced up and the UserBot shield connection pins (7) are faced down.



Insert the UserBot connection pins (7) into the Arduino female header (A1) from top-down as shown in the above picture.

⚠ Avoid the contact between the UserBot shield and the Arduino USB power entry (A2).

Peripherals connection

Since UserBot is an extension for Arduino, it means that any device or sensor that can interface with Arduino is also compatible with UserBot shield.

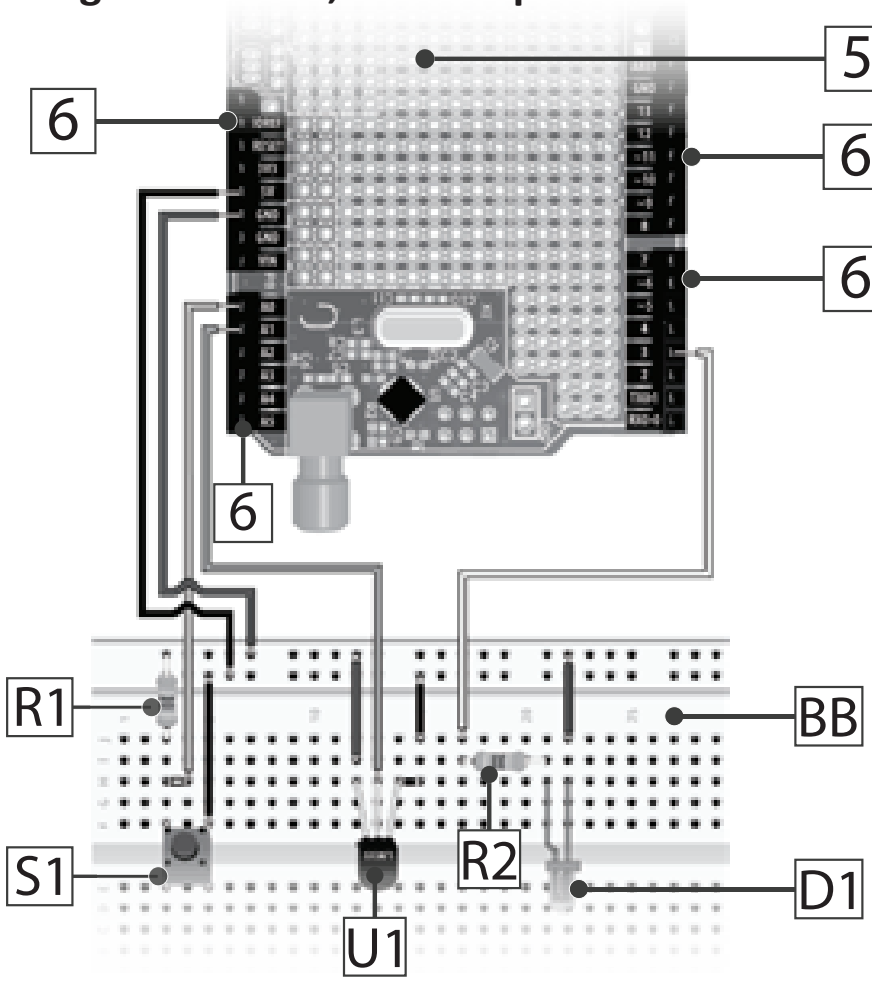
In this way, the list of compatible devices and sensors is extremely long, being impossible to list and explain how to connect all of them.

We will provide two examples: the first one shows how to connect/sample a button and a temperature sensor, while also act on a LED; the second one explains how to act on a DC motor.

⚠ Before assembling any sensor or device, make sure the Arduino is not connected to the power supply. Moreover, it is advisable to carry out the assembly on a firm surface, far from flood-prone areas. Also, use the proper tools for handling electronic components.

Example 1

Connecting one button, one temperature sensor and one LED.



5	Universal breakout board
6	Female pin headers
R1	Pull down resistor 10kΩ
R2	LED resistor 240Ω
S1	Tactile Button SPST
U1	LM35 temperature sensor
D1	LED
BB	External breadboard

The diagram on the previous page shows the connection of some different peripherals to the UserBot shield through its female pin headers (6).

The tactile button SPST (S1) is sampled on Arduino **analog pin A0**; the LM35 temperature sensor (U1) is sampled on Arduino **analog pin A1**; the LED (D1) is controlled through the Arduino **digital pin 3**. To control the 'speed/intensity' of a device, it is required to use a PWM Arduino digital pin (e.g. 3 and 5).

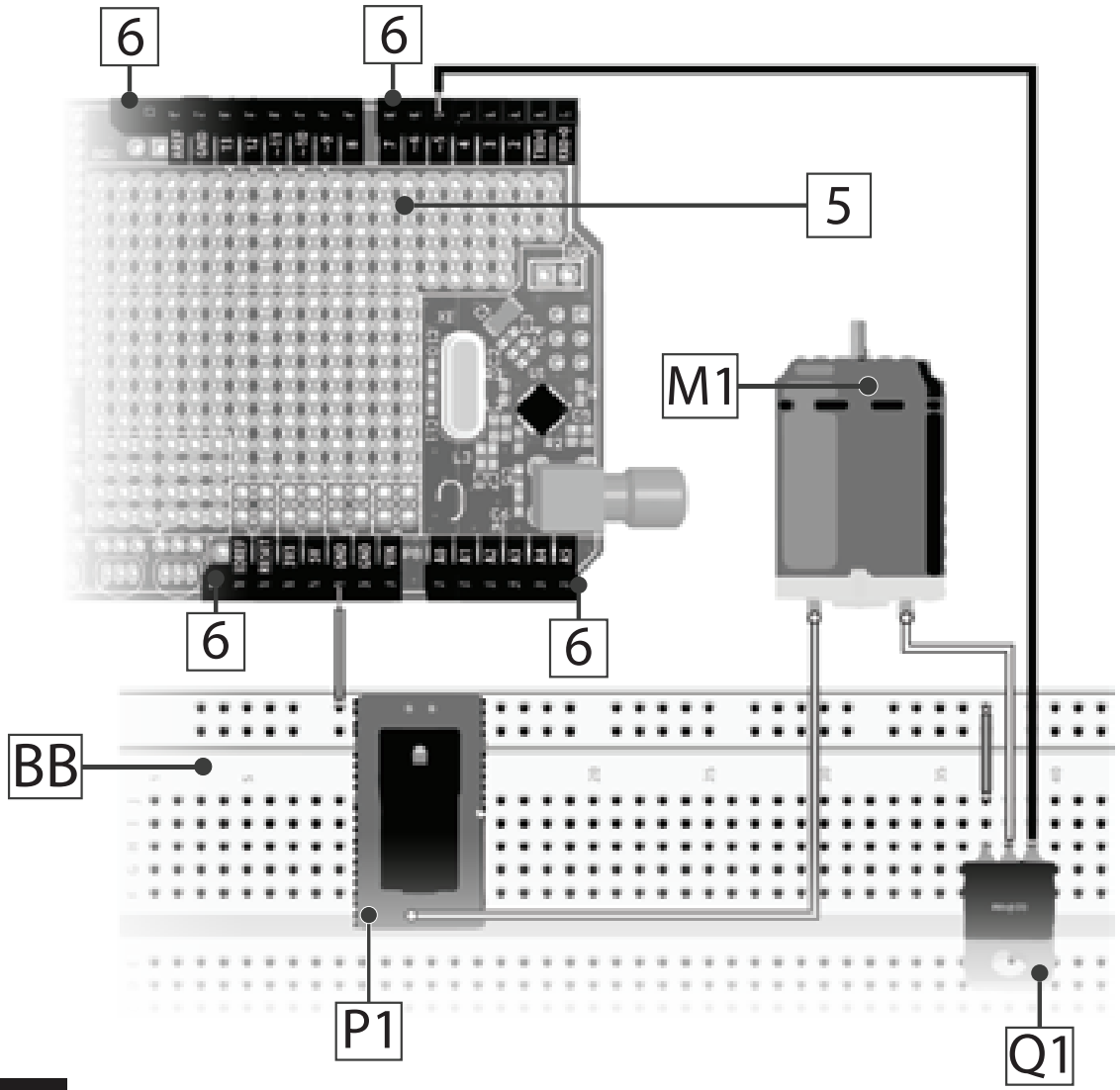
Use the **+5V**, **+3V3** and the **GND pins** on Arduino to power your sensors and/or devices. Please refer to the official Arduino documentation regarding power constraints on these pins.

The UserBot shield includes an onboard breadboard (5) that you are free to use. However, for the diagram, we used an external breadboard (BB) for a better understanding.

⚠ It is crucial to pay attention to the polarity of the different components. A wrong connection can damage your components and Arduino/UserBot.

Example 2

Connecting a DC motor (peristaltic pump, water pump...).



5 Universal breakout board

6 Female pin headers

M1 DC motor

BB External breadboard

P1 12VDC power entry

Q1 N-type MOSFET (e.g. IRLZ14)

The diagram on the previous page shows an N-type MOSFET (Q1) being used to control the supply voltage to a DC motor (M1).

The DC motor (M1) is powered by an external 12VDC (P1) and has PWM control via the Arduino **digital pin 5**. To control the "speed or intensity" of a device, it is required to use a PWM Arduino digital pin (e.g. 3 and 5).

The 12VDC power entry (P1) and the MOSFET (Q1) also need to be connected to the Arduino **GND pin**.

Use the UserBot female pin headers (6) to connect the 12VDC power entry (P1) and the MOSFET (Q1) to the respective Arduino pins.

The UserBot shield includes an onboard breadboard (5) that you are free to use. However, for the diagram, we used an external breadboard (BB) for a better understanding.

⚠ It is crucial to pay attention to the polarity of the different components. A wrong connection can damage your components and Arduino/UserBot.

Prepare the source code

After assembling all the sensors and devices, it is time to prepare the source code to handle them.

Open Grow provides the base firmware code in an open-source format. It is available to download on the GitHub at the following page:

github.com/OpenWeGrow/UserBot

In this way, the first task is to download the base code. Since there are several ways to do it (e.g. using apps to handle repositories or directly download from the GitHub web page), we will not detail any specific method in this manual. However, if you need assistance, please contact our technical support:

www.opengrow.pt @ support@opengrow.pt

NOTE: The examples provided by this manual are based on the code available on August 14, 2019.

Since the essence of the UserBot shield is to integrate the Arduino with the GroLab system, then to prepare the code we will use the Arduino IDE. If at any moment, you find any issues or obstacles with Arduino IDE, please refer to the Arduino official guides.

After downloading the base code, copy/move the contents of the folders **'libraries'** and **'examples'**, to the respective folders (same names) inside the Arduino installation folder on your PC.

Ensure you have the correct Arduino board selected in the Arduino IDE. For that, open the Arduino IDE and navigate to **Tools → Board** and choose the correct Arduino board based on your physical Arduino.

Next step is to open the code with the Arduino IDE, for that navigate to **File → Examples → OpenGrow → UserBot** and click to open it.

Before proceeding to change the code, we will list the file structure and explain it a little:

→ libraries
→ OpenGrow

• ComsTask.cpp • ComsTask.h	**Communication state machine.**
• CRC16.cpp • CRC16.h	**CRC calculator from Tim W. Shilling.**
• EEPROM_Utils.cpp • EEPROM_Utils.h	**NVM handling functions.**
• GroBot_Variables.h	**UserBot inputs/outputs definitions.**
• nRF24L01.h • RF24.cpp • RF24.h • RF24_config.h	**NRF24L01 driver from J. Coliz <maniacbug@ymail.com> with tweaks by Open Grow.**
• OpenBus.cpp • OpenBus.h	**Communication commands handler.**
• SerialTask.cpp • SerialTask.h	**State machine to handle UART commands.**

→ examples
→ OpenGrow
→ UserBot

• SensorsTask.cpp • SensorsTask.h	**A dedicated state machine for sensors sampling and output control.**
• UserBot.ino	**Arduino (.ino) file and base to run the program. This is the main UserBot configuration file. All of your inputs and outputs should be configured in this file.**

→ examples
→ OpenGrow
→ UserBot-DCMotor

(Same file structure as **examples → OpenGrow → UserBot**.)

Changing the code

Now everything is set, so it is time to start changing the code to match the user needs.

In summary, **for the majority of the cases (if not all)**, the **user only needs to change or duplicate the following files**:

→ examples
  → OpenGrow
    → UserBot
      - SensorsTask.cpp
      - SensorsTask.h
      - UserBot.ino

Since the required code changes are dependent on the assembled electronics by the user and the possible combination of sensors and devices is endless, it is not possible to provide the exact solution.

However, we will show the code required to achieve the two example diagrams previously presented.

Example 1

Connecting one button, one temperature sensor, and one LED.

To interface the button, the temperature sensor and the LED, open the **UserBot.ino** (**examples → OpenGrow → UserBot**) and navigate to the ***setup function***. This ***function*** contains the configuration for all the inputs and outputs. You should adjust the code to match the electronics you assembled on the board. For the example diagram, the code should be the following:

INPUTS CONFIGURATION

```
inputs[INPUT_INDEX0].arduinoPin = A0;    //Button Pin
inputs[INPUT_INDEX1].arduinoPin = A1;    //LM35 Pin
inputs[INPUT_INDEX2].arduinoPin = 0;     //Unused Input
inputs[INPUT_INDEX3].arduinoPin = 0;     //Unused Input
inputs[INPUT_INDEX4].arduinoPin = 0;     //Unused Input
inputs[INPUT_INDEX5].arduinoPin = 0;     //Unused Input
inputs[INPUT_INDEX6].arduinoPin = 0;     //Unused Input
inputs[INPUT_INDEX7].arduinoPin = 0;     //Unused Input
inputs[INPUT_INDEX8].arduinoPin = 0;     //Unused Input
inputs[INPUT_INDEX9].arduinoPin = 0;     //Unused Input

inputs[INPUT_INDEX0].type = BUTTON;
inputs[INPUT_INDEX1].type = DIG_TEMPERATURE;
inputs[INPUT_INDEX2].type = OPEN_DEFAULT;    //Unused Input
inputs[INPUT_INDEX3].type = OPEN_DEFAULT;    //Unused Input
inputs[INPUT_INDEX4].type = OPEN_DEFAULT;    //Unused Input
inputs[INPUT_INDEX5].type = OPEN_DEFAULT;    //Unused Input
inputs[INPUT_INDEX6].type = OPEN_DEFAULT;    //Unused Input
inputs[INPUT_INDEX7].type = OPEN_DEFAULT;    //Unused Input
inputs[INPUT_INDEX8].type = OPEN_DEFAULT;    //Unused Input
inputs[INPUT_INDEX9].type = OPEN_DEFAULT;    //Unused Input
```

OUTPUTS CONFIGURATION

```
outputs[OUTPUT_INDEX0].arduinoPin = 3;  //LED Pin
outputs[OUTPUT_INDEX1].arduinoPin = 0;  //Unused Output
outputs[OUTPUT_INDEX2].arduinoPin = 0;  //Unused Output
outputs[OUTPUT_INDEX3].arduinoPin = 0;  //Unused Output
outputs[OUTPUT_INDEX4].arduinoPin = 0;  //Unused Output
outputs[OUTPUT_INDEX5].arduinoPin = 0;  //Unused Output
outputs[OUTPUT_INDEX6].arduinoPin = 0;  //Unused Output
outputs[OUTPUT_INDEX7].arduinoPin = 0;  //Unused Output
outputs[OUTPUT_INDEX8].arduinoPin = 0;  //Unused Output
outputs[OUTPUT_INDEX9].arduinoPin = 0;  //Unused Output

outputs[OUTPUT_INDEX0].type = LED;
outputs[OUTPUT_INDEX1].type = OPEN_DEFAULT; //Unused Output
outputs[OUTPUT_INDEX2].type = OPEN_DEFAULT; //Unused Output
outputs[OUTPUT_INDEX3].type = OPEN_DEFAULT; //Unused Output
outputs[OUTPUT_INDEX4].type = OPEN_DEFAULT; //Unused Output
outputs[OUTPUT_INDEX5].type = OPEN_DEFAULT; //Unused Output
outputs[OUTPUT_INDEX6].type = OPEN_DEFAULT; //Unused Output
outputs[OUTPUT_INDEX7].type = OPEN_DEFAULT; //Unused Output
outputs[OUTPUT_INDEX8].type = OPEN_DEFAULT; //Unused Output
outputs[OUTPUT_INDEX9].type = OPEN_DEFAULT; //Unused Output
```

The different types of inputs and outputs are declared in the **GroBot_Variables.h** (**libraries → OpenGrow**) below the comments **'Possible Input Types'** and **'Possible Output Types'**.

The changes in this first file are complete and it is time to open the **SensorsTask.cpp** (**examples → OpenGrow → UserBot**).

NOTE: UserBot shield supports up to 10 inputs and 10 outputs.

After opening the **SensorsTask.cpp**, navigate to the ***SensorsTask function*** and search for the comment that says **'* IO Config *'.** Below the comment, you will find an example pin configuration that you should adjust to match your components. For the example diagram the code should be the following:

```
SensorsTask::SensorsTask(void)
{
    snsState = INIT_SENSORS;

    //************************  IO Config  ********************
    /*Here you need to set your used pins as inputs or outputs*/
    pinMode(inputs[INPUT_INDEX0].arduinoPin, INPUT);
    pinMode(inputs[INPUT_INDEX1].arduinoPin, INPUT);

    pinMode(outputs[OUTPUT_INDEX0].arduinoPin, OUTPUT);
}
```

Now, navigate to the ***GoSensorsTask function*** and look at the ***switch***. Inside the ***case GET_IOS***, you should add the code to sample your inputs and update its respective values in the ***inputs array***. For example:

```
if (digitalRead(inputs[INPUT_INDEX0].arduinoPin))
  inputs[INPUT_INDEX0].value = 255;
else
  inputs[INPUT_INDEX0].value = 0;
```

Or if it's an output, inside the ***case ACT_ON_IOS***, you should add the code to act on the outputs. For example:

```
if (outputs[OUTPUT_INDEX0].value > 0)
  digitalWrite(outputs[OUTPUT_INDEX0].arduinoPin, HIGH);
else
  digitalWrite(outputs[OUTPUT_INDEX0].arduinoPin, LOW);
```

You can add more ***cases*** based on your needs. For the example schematics, the code inside the ***switch*** should be the following:

```
case INIT_SENSORS:
  //Use this state in the machine to initialize any sensor you may need
  snsState = GET_TEMP;
  break;
case GET_TEMP:
  //Sampling Temperature Sensor
  time = getAnalogRead(inputs[INPUT_INDEX1].arduinoPin);
  inputs[INPUT_INDEX1].value += calcTemp(time);
  inputs[INPUT_INDEX1].value = inputs[INPUT_INDEX1].value/2;
  snsState = GET_IOS;
  break;
case GET_IOS:
  //Polling button
  if(digitalRead(inputs[INPUT_INDEX0].arduinoPin))
    inputs[INPUT_INDEX0].value = 255;
  else
    inputs[INPUT_INDEX0].value = 0x00;

  snsState= ACT_ON_IOS;
  break;
case ACT_ON_IOS:
  //Act on LED pin
  if(outputs[OUTPUT_INDEX0].value>0)
  {
    if((millis() - ticksOut1) > (MILLIS_PER_MINUTE * minutes2BackOffOut1))
    {
      if(outputs[OUTPUT_INDEX0].speed == 0)
        digitalWrite(outputs[OUTPUT_INDEX0].arduinoPin, HIGH);
      else
        analogWrite(outputs[OUTPUT_INDEX0].arduinoPin, outputs[OUTPUT_INDEX0].speed);
    }
  }
  else
  {
    digitalWrite(outputs[OUTPUT_INDEX0].arduinoPin, LOW);
  }
  snsState = GET_TEMP;
  break;
```

Note that we added some more code related to the output to support the variable **‘speed’** (e.g to control the LED’s intensity) and other functionalities already implemented in the GroLab System (such as the security cooldown time for sensitive devices).

NOTE: The inputs and outputs arrays, stores the sample values from sensors, as well as the state of the outputs. Those arrays are crucial to exchange info with the GroLab system.

If you need to change the ***cases*** from the ***switch*** it is also required to change the ***enum*** that contains those ***cases***. The ***enum*** is declared in the **SensorsTask.h** (**examples → OpenGrow → UserBot**).

At this point, the code is prepared to handle the physical peripherals from the diagram shown in example 1. Now it is time to compile and upload the code to your Arduino.

If you need assistance to compile and upload the code, please refer to the official Arduino guides or contact the Arduino support center.

After uploading the code to your Arduino, there is still one step required to make it fully operational and ready to communicate with GroLab system: the **factory settings need to be set** (serial number and communication channel). Please refer to the instructions on **pages 25 and 26**.

Example 2

Connecting a DC motor (peristaltic pump, water pump...).

To interface the DC motor, open the **UserBot.ino** (**examples → OpenGrow → UserBot-DCMotor**) and navigate to the ***setup function***. This ***function*** contains the configuration for all the inputs and outputs. You should adjust the code to match the electronics you assembled on the board. For the example diagram the code should be the following:

INPUTS CONFIGURATION

```
inputs[INPUT_INDEX0].arduinoPin = 0;     //Unused Input
inputs[INPUT_INDEX1].arduinoPin = 0;     //Unused Input
inputs[INPUT_INDEX2].arduinoPin = 0;     //Unused Input
inputs[INPUT_INDEX3].arduinoPin = 0;     //Unused Input
inputs[INPUT_INDEX4].arduinoPin = 0;     //Unused Input
inputs[INPUT_INDEX5].arduinoPin = 0;     //Unused Input
inputs[INPUT_INDEX6].arduinoPin = 0;     //Unused Input
inputs[INPUT_INDEX7].arduinoPin = 0;     //Unused Input
inputs[INPUT_INDEX8].arduinoPin = 0;     //Unused Input
inputs[INPUT_INDEX9].arduinoPin = 0;     //Unused Input

inputs[INPUT_INDEX0].type = OPEN_DEFAULT;    //Unused Input
inputs[INPUT_INDEX1].type = OPEN_DEFAULT;    //Unused Input
inputs[INPUT_INDEX2].type = OPEN_DEFAULT;    //Unused Input
inputs[INPUT_INDEX3].type = OPEN_DEFAULT;    //Unused Input
inputs[INPUT_INDEX4].type = OPEN_DEFAULT;    //Unused Input
inputs[INPUT_INDEX5].type = OPEN_DEFAULT;    //Unused Input
inputs[INPUT_INDEX6].type = OPEN_DEFAULT;    //Unused Input
inputs[INPUT_INDEX7].type = OPEN_DEFAULT;    //Unused Input
inputs[INPUT_INDEX8].type = OPEN_DEFAULT;    //Unused Input
inputs[INPUT_INDEX9].type = OPEN_DEFAULT;    //Unused Input
```

OUTPUTS CONFIGURATION

```
outputs[OUTPUT_INDEX0].arduinoPin = 5;  //DC Motor as Perista
outputs[OUTPUT_INDEX1].arduinoPin = 0;  //Unused Output
outputs[OUTPUT_INDEX2].arduinoPin = 0;  //Unused Output
outputs[OUTPUT_INDEX3].arduinoPin = 0;  //Unused Output
outputs[OUTPUT_INDEX4].arduinoPin = 0;  //Unused Output
outputs[OUTPUT_INDEX5].arduinoPin = 0;  //Unused Output
outputs[OUTPUT_INDEX6].arduinoPin = 0;  //Unused Output
outputs[OUTPUT_INDEX7].arduinoPin = 0;  //Unused Output
outputs[OUTPUT_INDEX8].arduinoPin = 0;  //Unused Output
outputs[OUTPUT_INDEX9].arduinoPin = 0;  //Unused Output

outputs[OUTPUT_INDEX0].type = PERISTALTIC_PUMP;  //DC Motor a
outputs[OUTPUT_INDEX1].type = OPEN_DEFAULT; //Unused Output
outputs[OUTPUT_INDEX2].type = OPEN_DEFAULT; //Unused Output
outputs[OUTPUT_INDEX3].type = OPEN_DEFAULT; //Unused Output
outputs[OUTPUT_INDEX4].type = OPEN_DEFAULT; //Unused Output
outputs[OUTPUT_INDEX5].type = OPEN_DEFAULT; //Unused Output
outputs[OUTPUT_INDEX6].type = OPEN_DEFAULT; //Unused Output
outputs[OUTPUT_INDEX7].type = OPEN_DEFAULT; //Unused Output
outputs[OUTPUT_INDEX8].type = OPEN_DEFAULT; //Unused Output
outputs[OUTPUT_INDEX9].type = OPEN_DEFAULT; //Unused Output
```

The different types of inputs and outputs are declared in the **GroBot_Variables.h** (**libraries → OpenGrow**) below the comments **'Possible Input Types'** and **'Possible Output Types'**.

The changes in this first file are complete and it is time to open the **SensorsTask.cpp** (**examples → OpenGrow → UserBot-DCMotor**).

NOTE: UserBot shield supports up to 10 inputs and 10 outputs.

After opening the **SensorsTask.cpp**, navigate to the ***SensorsTask function*** and search for the comment that says **'* IO Config *'.** Below the comment, you will find an example pin configuration that you should adjust to match your components. For the example diagram the code should be the following:

```
SensorsTask::SensorsTask(void)
{
    snsState = INIT_SENSORS;

    //************************  IO Config  ********************
    /*Here you need to set your used pins as inputs or output
    pinMode(outputs[OUTPUT_INDEX0].arduinoPin, OUTPUT);
}
```

Now, navigate to the ***GoSensorsTask function*** and look at the ***switch***. Inside the ***case ACT_ON_IOS***, you should add the code to act on the outputs. For example:

```
if (outputs[OUTPUT_INDEX0].value > 0)
  digitalWrite(outputs[OUTPUT_INDEX0].arduinoPin, HIGH);
else
  digitalWrite(outputs[OUTPUT_INDEX0].arduinoPin, LOW);
```

NOTE: The output array stores the state of the outputs. This array is crucial to exchange info with the GroLab system.

You can add more ***cases*** based on your needs. For the example schematics, the code inside the ***switch*** should be the following:

```
case INIT_SENSORS:
  //Use this state in the machine to initialize any sensor you may need
  snsState = ACT_ON_IOS;
  break;
case ACT_ON_IOS:
  //Act on DC Motor
  if(outputs[OUTPUT_INDEX0].value>0)
  {
    if((millis() - ticksOut1) > (MILLIS_PER_MINUTE * minutes2BackOffOut1) )
    {
      if(outputs[OUTPUT_INDEX0].speed == 0)
        digitalWrite(outputs[OUTPUT_INDEX0].arduinoPin, HIGH);
      else
        analogWrite(outputs[OUTPUT_INDEX0].arduinoPin, outputs[OUTPUT_INDEX0].speed);
    }
  }
  else
  {
    digitalWrite(outputs[OUTPUT_INDEX0].arduinoPin, LOW);
  }
  snsState = INIT_SENSORS;
  break;
```

Note that we added some more code related to the output to support the variable **'speed'** and other functionalities already implemented in the GroLab system (such as the security cooldown time for sensitive devices).

If you need to change the ***cases*** from the ***switch***, it is also required to change the ***enum*** that contains those ***cases***. The ***enum*** is declared in the **SensorsTask.h** (**examples → OpenGrow → UserBot-DCMotor**).

At this point, the code is prepared to handle the physical peripherals from the diagram shown in example 2. Now it is time to compile and upload the code to your Arduino.

If you need assistance to compile and upload the code, please refer to the official Arduino guides or contact the Arduino support center.

Set the factory settings

After uploading the code to your Arduino, there is still one step required to make it fully operational and ready to communicate with GroLab system: the **factory settings need to be set** (serial number and communication channel).

Make sure your Arduino is connected through USB to the PC, and that the serial port is not currently in use. Open the Arduino IDE and select the respective serial port. Access the **'Serial Monitor'** and set the **baud rate to 230400**. Send the **'R'** command with the option **'No line ending'**. Wait a moment until you see **RST** printed in the 'Serial Monitor'. Reboot your Arduino (power interruption or reset button if available).

Open the 'Serial Monitor' again, then send the **'M'** command with the option **'No line ending'**. Then send the communication channel that is currently being used by your GroNode* with the option **'Both NL & CR'**.

***There are 5 communication channels (1 to 5), the default channel is 5.**

Send the **‘S’** command with the option **‘No line ending’**, type the desired serial number (sending **digit by digit**) and when you reach the last number send it with the option **‘Both NL & CR’.** Note the serial number should be composed of 10 numbers and start by the number ‘5’, for example, ‘5123456789’.*

To check if everything is correct, send the **‘D’** command with the option **‘No line ending’**. You should see something similar to the following image through the ‘Serial Monitor’:

```
UserBot! Ready
N:      New Module
S:      5123456789
MID:    2
SAdd:   FF
FW:     1.1.0.8
```

After completing all the previous steps, Arduino is already set up and ready to communicate with GroNode through the UserBot shield.

***The serial number ‘5999999999’ is reserved as the default one and should not be used.**

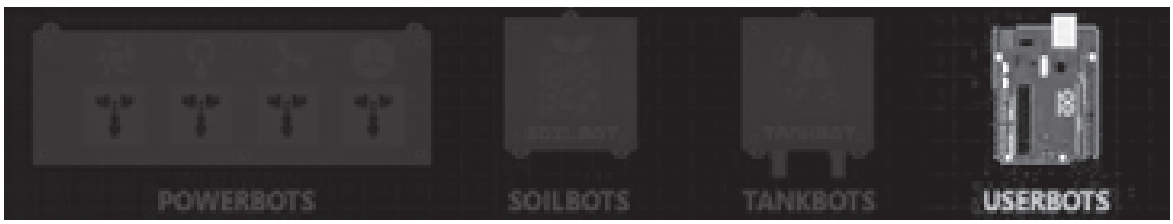
COMMUNICATIONS ②

Connection to GroNode

Make sure the GroNode is correctly installed and accessible through the GroLab Software. If not, follow the instructions provided inside the GroNode's manual.

After completing all the UserBot shield installation steps, open the GroLab Software, establish a connection with the GroNode and then, access **Modules** through the main menu.

Usually, 2 or 3 minutes are enough for GroNode to detect a newly installed module. Once it is detected, it will automatically appear in the **Modules** menu under the respective module type section.



You can check the list of modules of the same type on the right side of this menu. Note that one GroNode supports a maximum of 4 modules of each type.

Problems or losses in communication

GroNode was designed to communicate with the other modules through radio frequency signals. The **range of action** is **25 meters (82 feet) indoors** and **100 meters (328 feet) outdoors**, depending on space conditions.

Access the **Modules** menu on the GroLab Software and check if your modules are available. If they aren't available or they are experiencing communication losses, you may have exceeded the distance between your modules and GroNode.

Inside the **Modules** menu, you can find a wireless signal icon that indicates if the selected module is connected to the GroNode.

⚠ NOTICE: Some load-bearing walls and electronic devices may interfere with the signal.

If the problem persists, perform a test run by placing the module next to the GroNode and check the module's communication status through the software. If communication has been successfully re-established, repeat this operation at different distances until you detect the maximum communication distance.

If the module is not found at all, try to cycle the GroNode through all the available communication channels until you find your modules, allowing 3 minutes between channel changes.

To change the communication channel, open the GroLab Software and connect to the GroNode. Inside the **Main Menu (Configurations)**, go to **Settings (GroNode) → General Settings Tab**. Click the button in the lower right corner to enable editing and change the communication channel to the desired channel. To apply the changes, click the green button in the lower right corner.

If after following the previous steps your modules still don't show up on the GroLab Software, please contact our technical support.

Firmware Updates

Keep in mind that some software updates may require that you update the firmware of your GroLab modules.

All updates provide crucial improvements that ensure optimal system performance.

However, as UserBot firmware is compiled and applied by the user, it is required to keep monitoring the UserBot code repository on GitHub for any code updates. In case some code updates are submitted, the user must apply those changes to their code, re-compile and re-upload it to the Arduino. It is also necessary to reapply the factory settings.

SAFETY INSTRUCTIONS ④

The following general safety guidelines are provided to help ensure your own personal safety and protect your device from potential damage.

•Do not attempt to service the device and never disassemble the device. For some devices with a user-replaceable battery, please read and follow the instructions provided by the installation manual.
•Keep your device away from radiators and heat sources.
•Keep your device away from extremely hot or cold temperatures to ensure that it is used within the specified operating range.
•Do not spill food or liquids on your device.
•Before you clean your device, disconnect it from the electrical outlet. Clean your device with a dry soft cloth. Do not use liquids.
•If your device does not operate normally - in particular, if there are any unusual sounds or smells coming from it - unplug it immediately and contact an authorized dealer or the Open Grow support center.
•To help avoid the potential hazard of electric shock, do not connect or disconnect any cables, or perform maintenance or reconfiguration of your device during an electrical storm.
•Check the voltage rating before you connect the device to an electrical outlet to ensure that the required voltage and frequency match the available power source.
•Also, ensure that your GroLab modules and attached devices are electrically rated to operate with the AC power available in your location.
•Do not plug the power cables into an electrical outlet if the power cables are damaged.
•To prevent electric shock, plug the power cables into properly grounded electrical outlets.
•If you use an extension power cable, ensure that the total ampere rating of the devices plugged into the extension power cable does not exceed the ampere rating of the extension cable.

WARRANTY ⑤

General terms

The warranty set forth below is given by Open Grow, Lda. (from now on, will be referred to as OPEN GROW). This warranty is only effective upon presentation of the proof of purchase.

EXCEPT AS EXPRESSLY SET FORTH IN THIS WARRANTY, OPEN GROW MAKES NO OTHER WARRANTIES, EXPRESS OR IMPLIED, INCLUDING ANY IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. OPEN GROW EXPRESSLY DISCLAIMS ALL WARRANTIES NOT STATED IN THIS WARRANTY. ANY IMPLIED WARRANTIES THAT MAY BE IMPOSED BY LAW ARE LIMITED IN DURATION TO THE WARRANTY PERIOD.

TO THE EXTENT ALLOWED BY LOCAL LAW, THE REMEDIES IN THIS WARRANTY ARE YOUR SOLE AND EXCLUSIVE REMEDIES AGAINST OPEN GROW. THEY DO NOT, HOWEVER, AFFECT OR RESTRICT THE RIGHTS YOU HAVE AGAINST THE BUSINESS YOU BOUGHT A OPEN GROW PRODUCT FROM. IN NO EVENT SHALL OPEN GROW BE LIABLE FOR LOSS OF DATA OR FOR INDIRECT, SPECIAL, INCIDENTAL, CONSEQUENTIAL (INCLUDING LOST PROFIT OR DATA), OR OTHER DAMAGE, WHETHER BASED IN CONTRACT, TORT, OR OTHERWISE. HOWEVER, NOTHING IN THIS AGREEMENT LIMITS OPEN GROW'S LIABILITY TO YOU (I) IN THE EVENT OF DEATH OR PERSONAL INJURY TO THE EXTENT RESULTING FROM OPEN GROW'S NEGLIGENCE, OR (II) TO THE EXTENT RESULTING FROM ANY FRAUDULENT MISREPRESENTATION ON THE PART OF OPEN GROW, OR (III) TO THE EXTENT ARISING UNDER RELATED CONSUMER PROTECTION REGULATIONS. SOME COUNTRIES DO NOT ALLOW: (1) A DISCLAIMER OF IMPLIED WARRANTIES; (2) A LIMITATION ON HOW LONG AN IMPLIED WARRANTY LASTS OR THE EXCLUSION; OR (3) LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES FOR CONSUMER PRODUCTS. IN SUCH COUNTRIES, SOME EXCLUSIONS OR LIMITATIONS OF THIS LIMITED WARRANTY MAY NOT APPLY TO YOU. THIS WARRANTY GIVES YOU SPECIFIC LEGAL RIGHTS. YOU MAY ALSO HAVE OTHER RIGHTS THAT MAY VARY FROM COUNTRY TO COUNTRY. YOU ARE ADVISED TO CONSULT APPLICABLE COUNTRY LAWS FOR A FULL DETERMINATION OF YOUR RIGHTS.

This warranty applies to any OPEN GROW branded hardware product sold by OPEN GROW, its subsidiaries, affiliates, authorized resellers, or distributors (from now on, will be referred to as PARTNERS), with this warranty.

Limitation of warranty

OPEN GROW warrants that the product under normal use is free from material defects in materials and workmanship during the warranty period if the product is used and serviced in accordance with the user manual and other documentation provided to the purchaser at the time of purchase (or as amended from time to time). OPEN GROW does not warrant that the product will operate uninterrupted or error-free or that all deficiencies, errors, defects or non-conformities will be corrected.

This warranty shall not apply to, and OPEN GROW shall not be responsible for, problems or damages resulting from: (a) unauthorized alterations or attachments; (b) negligence, abuse or misuse, including failure to operate the product in accordance with specifications, instruction, or interface requirements; (c) improper handling, use or storage; (d) failure of goods or services not obtained from OPEN GROW or not subject to a then-effective OPEN GROW warranty or maintenance agreement; or (e) fire, water, acts of God or other catastrophic events. This warranty shall also not apply if any OPEN GROW serial number on the product has been removed or defaced in any way.

Product warranty period

The product warranty period starts on the date of purchase from OPEN GROW or PARTNERS. Your dated sales or delivery receipt, showing the date of purchase of the product, is your proof of the purchase date.

You may be required to provide proof of purchase as a condition of receiving warranty service. You are entitled to warranty service according to the terms and conditions of this document if a repair to your OPEN GROW product is required within the product warranty period.

This product warranty extends only to the original end-user purchaser of this OPEN GROW product and is not transferable to anyone who obtains ownership of the product from the original end-user purchaser.

All OPEN GROW products have a 2-year guarantee except for consumables (calibration solutions, sensors and/or actuators of any type) and under normal use.

Performance of the warranty

If any defect occurs, OPEN GROW's sole obligation shall be to repair or replace the product free of charge provided it is returned to OPEN GROW after-sales department during the warranty period. Such repair or replacement will only be rendered by OPEN GROW at an authorized OPEN GROW repair station. All components or products that are replaced under this warranty become the property of OPEN GROW. The replacement takes on the REMAINING warranty period of the product. The replacement need not be new or of an identical make, model, or part; OPEN GROW may in its discretion make the replacement with any reconditioned equivalents.

Warrantor

Open Grow, Lda.,
Parque Industrial de Coimbrões,
Edifício Expobeiras,
3500-618, Viseu,
Portugal

(+351) 232 458 475

www.opengrow.pt

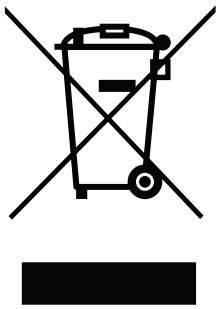
Do not ship directly to OPEN GROW without first consulting with us to verify the procedure that should be followed.

Contact our technical services at **support@opengrow.pt**.

DISCLAIMER:

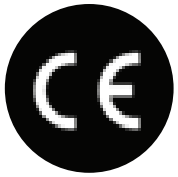
Application of the guarantee is excluded should the breakdown of the defective part or parts be a result of the product's inadequate and/or negligent use. Understood as inadequate and/or negligent use is any use other than the one for which the product is meant and/or that is recommended in the instructions manual, not executing the maintenance operations recommended in the instructions manual, carrying out operations that are different from those mentioned and that compromise the quality of the product, modifications that are not performed by authorized repairers and/or with non-original or non-approved parts.

COMPLIANCE ⑥



This symbol on the product or packaging means that according to local laws and regulations this product should not be disposed of in the household waste but sent for recycling. Please take it to a collection point designated by your local authorities once it has reached the end of its life, some will accept products for free. By recycling the product and its packaging in this manner you help to conserve the environment and protect human health.

This symbol on the product or packaging means that this product is compliant with RoHS Regulations of the European Parliament and Council Directive on the Restrictions of the Use of Certain Hazardous Substances in Electrical and Electronic Equipment (2011/65/EU).



This symbol on the product or packaging means that this product is in compliance with the following directives and regulations:

- (2014/53/EU) Radio Equipment Directive.
- (2011/65/EU) RoHS Directive.
- (2014/35/EU) Low Voltage Directive.
- (2014/30/EU) EMC.

	Frequency Band(s)	Max. Output Power (EIRP)
2.4 G	2.4 - 2.4835 GHz	100 mW

Open Grow, LDA, reserves the right to update and/or modify the content of its products at any time without prior warning. Check out our Terms & Conditions at www.opengrow.pt.

OPENGROW.

NOTES

EN Specifications are subject to change without notice. GroLab is a registered trademark of Open Grow, LDA. Other brands and product names are trademarks or registered trademarks of their respective holders.
No part of the specifications may be reproduced in any form or by any means or used to make any derivative such as translation, transformation, or adaptation without permission from Open Grow, LDA.

ES Las especificaciones están sujetas a cambios sin previo aviso. GroLab es una marca registrada de Open Grow, LDA. Otras marcas y nombres de productos son marcas comerciales o marcas comerciales registradas de sus respectivos propietarios.
Ninguna parte de las especificaciones se puede reproducir de ninguna forma ni por ningún medio, ni se puede utilizar para obtener ningún derivado, como traducción, transformación o adaptación sin el permiso de Open Grow, LDA.

FR Les spécifications sont sujettes à changement sans préavis. GroLab est une marque déposée de Open Grow, LDA. Les autres marques et noms de produits sont des marques commerciales ou des marques déposées de leurs propriétaires respectifs.
Aucune partie des spécifications ne peut être reproduite sous quelque forme ou par quelque moyen que ce soit, ni être utilisée pour obtenir des dérivés tels que la traduction, la transformation ou l'adaptation sans l'autorisation de Open Grow, LDA.

PT As especificações estão sujeitas a alterações sem aviso prévio. GroLab é uma marca registada da Open Grow, LDA. Outras marcas e nomes de produtos são marcas comerciais ou marcas registadas dos respetivos detentores.
Nenhuma parte das especificações pode ser reproduzida, sob qualquer forma ou por quaisquer meios, nem utilizada para criar qualquer obra derivada, como tradução, transformação ou adaptação, sem a autorização da Open Grow, LDA.

Open Grow, LDA
Edifício Expobeiras
Prq. Industrial de Coimbrões
3500-618 Viseu
Portugal

(+351) 232 458 475

info@opengrow.pt
support@opengrow.pt

www.opengrow.pt
www.opengrow.pt/support