CARLO GAVAZZI
Automation Components

CARLO GAVAZZI

SBP Tool Car Park Manual

Rev. 7.3.3/0.1, 24/10/2017

Summary

1 INTRODUCTION 5

1.1 Requirements 5
1.1.1 Minimum hardware requirements 5
1.1.2 Software requirements 5
1.1.3 How to read the software version number 5

2 INSTALLATION 6

2.1 How to connect with a mini-USB cable 7
2.1.1 How to install the mini-USB driver for Windows 7 / Vista / XP 7
2.1.2 How to install the mini-USB driver for Windows 10 / 8.1 / 8 10
2.1.3 How to connect to the controller by means of a modem 18

3 USER INTERFACE 19

3.1 File Menu 19
3.2 View Menu 20
3.3 Reports menu 21
3.4 Add menu 21
3.5 Program setup menu 22

4 PROJECT STRUCTURE 23

4.1 Wizard 23
4.1.1 Area 1 23
4.1.2 Area 2 24
4.1.3 Area 3 24
4.2 Locations 25

5 LANE, LINE, POSITION 26

6 HOW TO CREATE A NEW PROJECT 28

6.1 How to add a new configuration to the current project 32
6.2 How to automatically find and address the sensors – The controller is connected to the modules 35
6.3 How to manually place the discovered modules 36
6.4 How to automatically place the modules 37
6.4.1 Single line addressing 37
6.4.2 Multi line addressing 39
6.5 How to manually add modules – The controller is not connected 41

7 HOW TO CALIBRATE A SENSOR 43

7.1 Step 1: Select the modules 44
7.2 Step 2: Write the distance from the floor and the type of mounting 46
7.2.1 How to write different settings to each single sensor 46
7.2.2 How to write the same distances and mounting ways into the selected sensors 47
7.3 Step 3: Calibrate the sensors 50
7.3.1 Remote calibration 50
7.3.2 Local calibration 51
7.3.3 Diagnostic signals 52

8 HOW TO DEFINE THE COLOURS OF THE LED IN THE SENSOR SBPUSLXX 53

9 HOW TO PROGRAM THE CONTROLLER SBP2WEB24 AND THE SBP2CPY24 54

9.1 SBP2WEB24 54
9.2 SBP2CPY24 56
9.2.1 How to reset the configuration into the CPY server 57

10 HOW TO READ THE CONFIGURATION FROM A CONTROLLER 58

11 LIVE SIGNALS 59

12 ZONE COUNTER FUNCTION 60

12.1 What is a zone? 60
12.2 Detection Points (DPOs) 60
12.3 Initialization and adjustment 60
12.4 Detection points (DPO) function 61
12.4.1 Entrance/exit with no direction detection 61
12.4.2 Entrance/exit with direction detection 62
12.4.3 How to set the Options field of the counter in the DPO function 64
12.4.4 Live signals for the DPO function 65
12.5 MZC function 66
12.5.1 How to set a predefined value of the counter using signals 68
12.5.2 How to manually increase/decrease the counter 70
12.5.3 How to set the number of available bays with the calendar 70
12.5.4 How to remotely access the output status of the MZC function 72
12.5.5 Live signals in the MZC function 73

13 INDICATOR FUNCTION 74

13.1 How to add the Indicator function with the fast procedure 75

14 MODULES 77

14.1 How to manage the filters on the Modules window 77
14.2 How to manage the filters in the Signals window 79
14.3 Car Park modules 82
14.3.1 Ultrasonic sensors 82
14.3.2 Lane indicator 85
14.3.3 Counter sensor 86

15 TIME SERVER 88

1 Introduction

The SBP tool has been developed for the configuration of the SBP2WEB24, a programmable Linux embedded PC specially designed for car park and building automation applications.
The SBP tool is also used to write the configuration into the SBP2CPY24 servers.
All functions are represented by graphic symbols, and all function related parameters are set up locally in the PC, and then transferred to the SBP2WEB24 via Ethernet. Some of the function parameters can be changed later via remote connection to the controller (web server, email, SMS, Modbus...) as described later in this manual.
Likewise, data from the SBP2WEB24 can be uploaded and modified.

The PC does not need to be connected to a SBP2WEB24 controller in order to make a configuration.

The figures in this manual may differ from the figures on your screen. This is not necessarily an error, but may be caused by revision differences.

The contents of this manual may be altered without notice.

1.1 Requirements

1.1.1 Minimum hardware requirements

- A Microsoft® Windows®-based PC
- Display with a resolution capability of minimum 1024x768 pixels
- 1 GB of disk-space
- An Ethernet-port and cable or SD-card reader or USB port 2.0 or higher

1.1.2 Software requirements

- Microsoft® Windows® 10/8.1/8/7/Vista/ (32 or 64-bit)
- Microsoft Dot-Net Framework 4.5

1.1.3 How to read the software version number

The SBP tool revision number has the following structure:

Major	Minor	Revision	Build number

- **Major:** this identifies the main features of the software. It is incremented when new features are added or there are big changes in the existing ones.

- **Minor:** this identifies the version of the relevant *major* release and it is incremented when there are bugs fixing or small new functions.

- **Revision:** this identifies the status of the release.
  1 *beta*: for internal use only
  2 *controlled beta:* to be shared with selected customers for field testing
  3 *final*: available for everybody on the Carlo Gavazzi website

- **Build number:** this is an internal number that includes the day on which the release has been compiled

2 Installation

The SBP tool can be downloaded from Carlo Gavazzi's Product Selection website. To install the software you just have to double click on the *Setup.exe* file and follow the instructions on the screen.

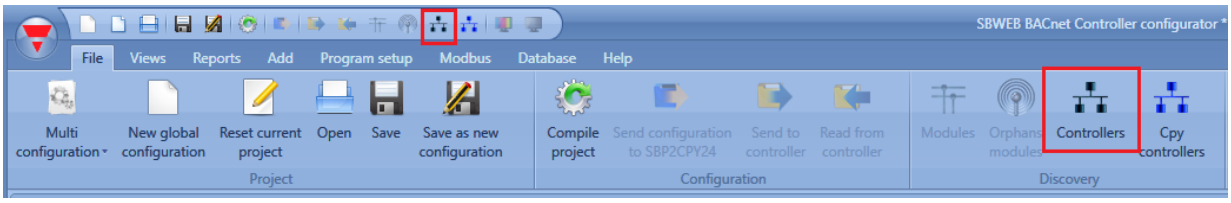
In order to get in contact with the SBP2WEB24 controller, the user has several connection modes available:

- via Ethernet connection
- via Mini USB cable
- via Modem connection

After a power on, the SBP2WEB24 master unit is ready to work after about 1 minute. Only when the yellow BUS Led starts flashing is the master unit ready.

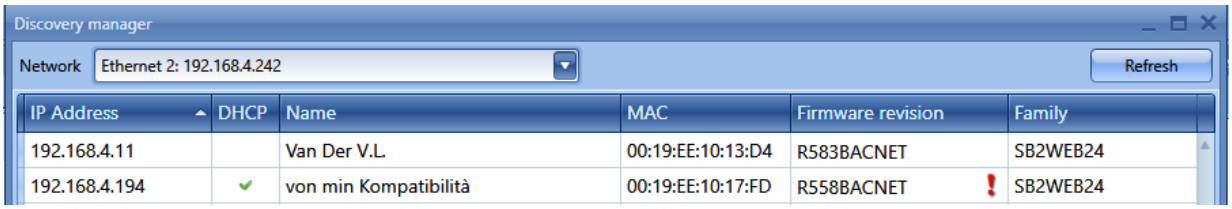
How to connect the master unit SBP2WEB24 with Ethernet connection

To connect to a master unit SBP2WEB24, the user has to click on the icon highlighted in red in the picture below: the SBP tool will start the discovery of the SBP2WEB24 connected to the Ethernet network.



Important note
If the PC is running the Windows Firewall or a Third party Firewall / Antivirus, make sure that the ports 52325, 10000, 10001, 10002, 80 and 443 are not blocked (input/output packets). These ports are used by the SBP Tool to search for the master unit in the network and for communication. If a firewall blocks these ports, the SBP Tool will not be able to find the controllers in the network or to use the Live Signals. Be sure that the master unit's IP settings match the IP settings of the PC used: it must have the same IP class and the same net mask address.

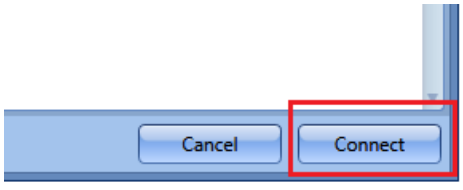
When the PC has more than one network card or has many IP addresses, it is possible to select the right network from the list (see picture below): it must be the same one as the SBP2WEB24 is connected to.



When the icon marked in red is clicked, the SBP Tool starts looking for the controller(s). If one or more units are found, a window will pop-up, allowing you to make a selection of which controller to connect to.

Select the master unit on the list, or *Cancel* if you do not want to get connected at this time.

After the user has pressed *Connect*, the red Led on the selected SBP2WEB24 starts flashing and the connection will be established.



2.1 How to connect with a mini-USB cable

The SBP2WEB24 controller can be connected to the PC by means of the *mini-B* port available on the front of the housing, protected by the front cover. Please refer to the *Hardware manual* for technical information.

The following items are required to obtain the connection:

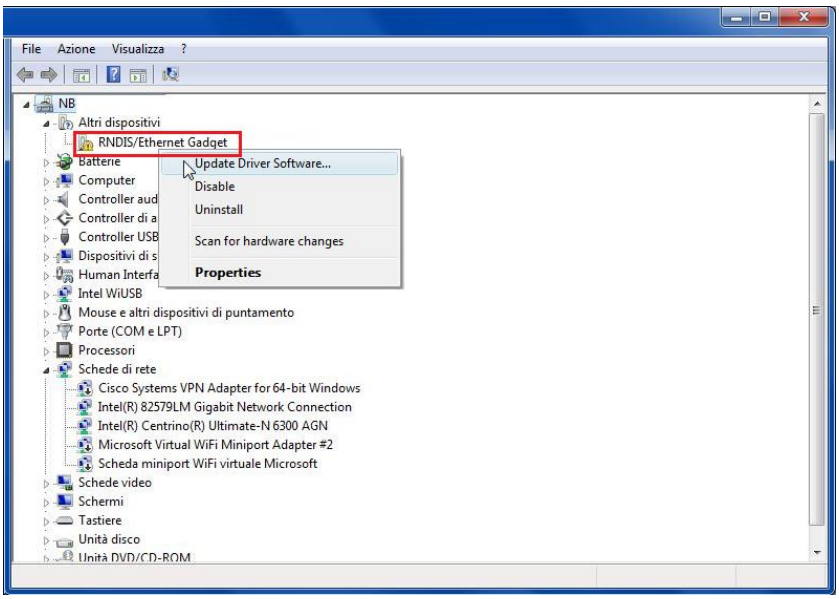
- A **USB 2.0 cable-Type-A / Mini-B** (this is not provided with the SBP2WEB24 controller).
- *mini-USB driver.zip* package available on www.productselection.net website in the SBP2WEB24 pages.

The driver installation procedure can change slightly depending on the operating system in the user's PC: please follow the instructions below.

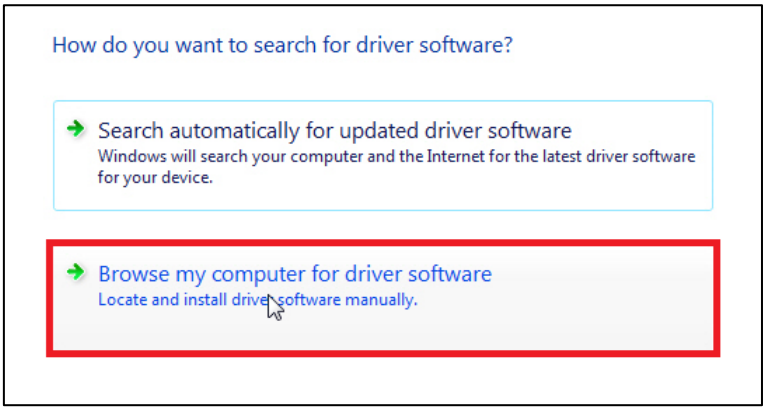
2.1.1 How to install the mini-USB driver for Windows 7 / Vista / XP

Open the archive "*mini-USB driver.zip*" and save the file *linux.inf* in your PC, then install the driver by following the instructions below:

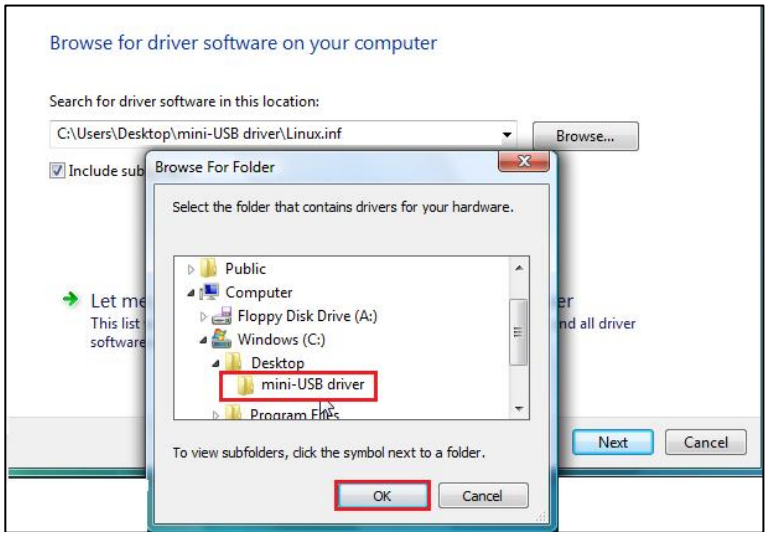
1. Plug the USB connector into a free USB port of the PC and the mini-USB connector into the mini-B port of the SBP2WEB24
2. Go to *Control Panel* and open *Device Manager*.
3. Find the device *RNDIS/Ethernet Gadget*, right-click on it and select *Update Driver Software…,* as shown in the picture below.



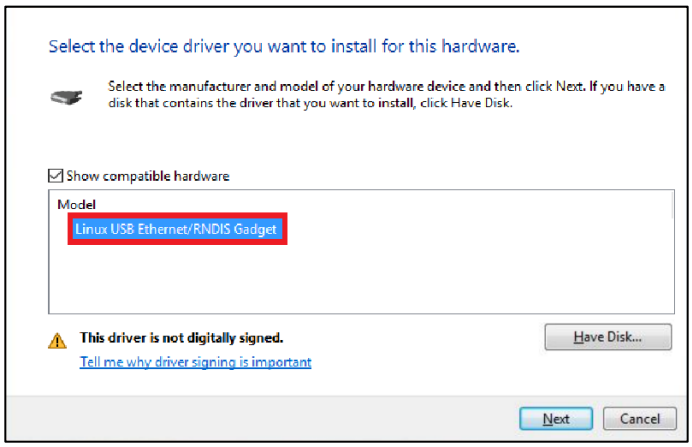
4. Select the *Browse my computer for driver software* option.



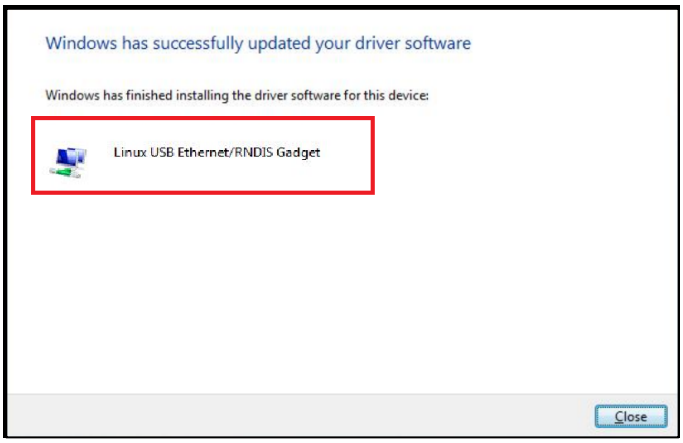
5. Browse for the *linux.inf* driver file and press *OK.*



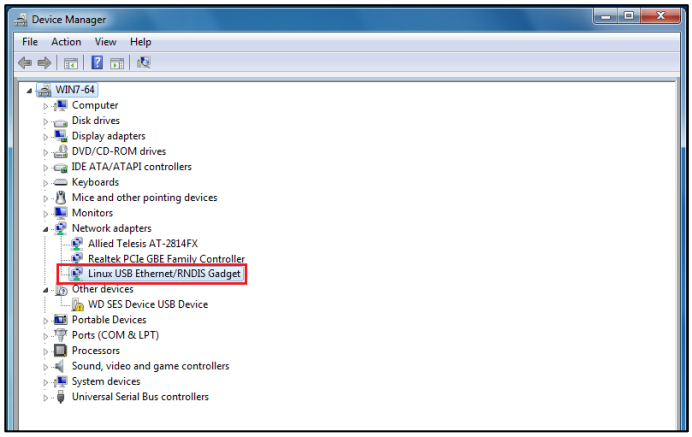
6. The driver *Linux USB Ethernet/RNDIS Gadget* will be detected: press *Next* to proceed.



7. The driver will be installed, as shown in the picture below.

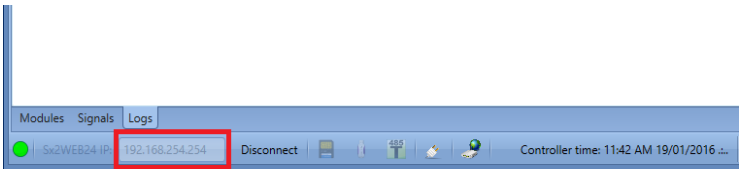


8. When the driver is installed, in the *Network adapter* category a virtual network board named *Linux USB Ethernet/RNDIS Gadget* will be added, as shown in the picture below. The driver automatically gives a dynamic IP address to the Controller/PC according to the actual IP of the PC.



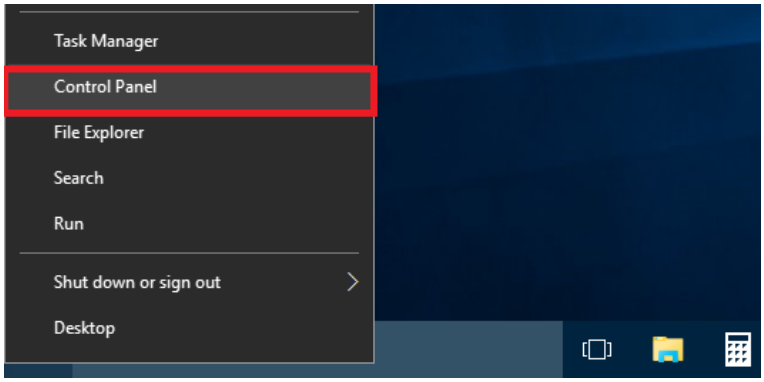
For example, if the PC has the IP address 192.168.0.10, the virtual board will be created with a new address 192.168.254.xxx and the IP address 192.168.254.254 will be assigned to the SBP2WEB24 controller.

9. Insert the IP address 192.168.254.254 in the SBP Tool and press *Connect,* as shown in the picture below:

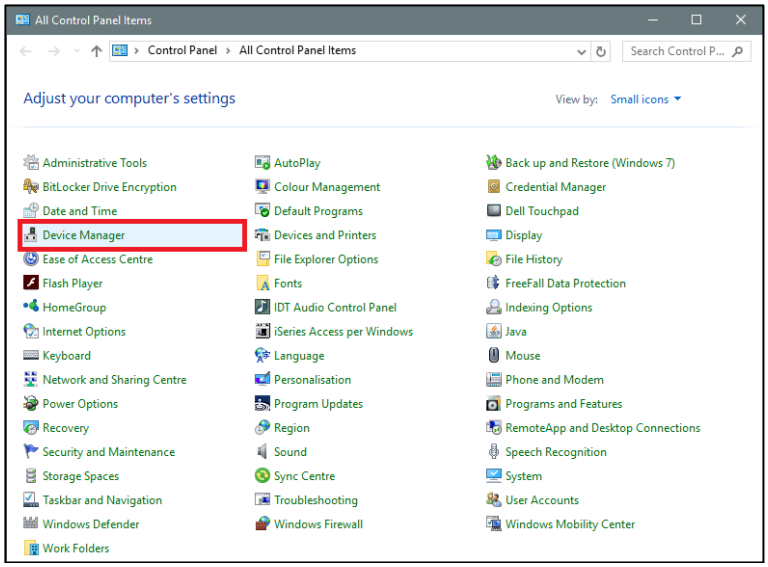


2.1.2 How to install the mini-USB driver for Windows 10 / 8.1 / 8

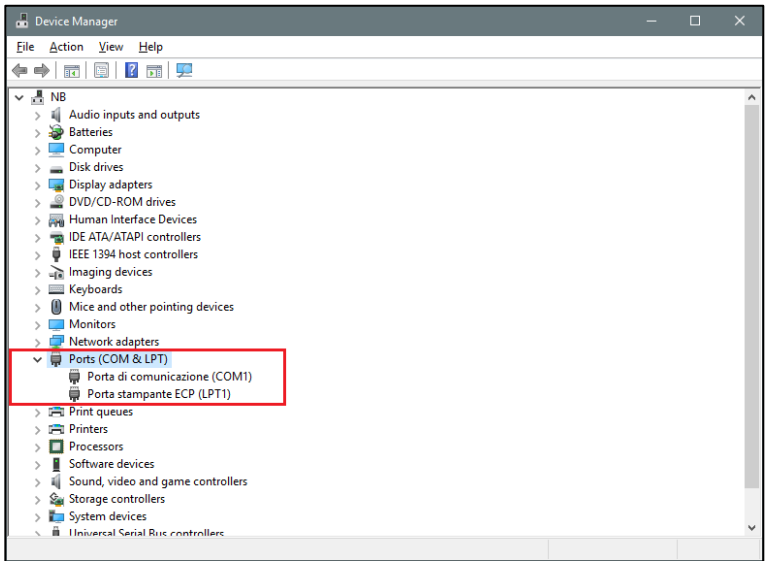
1. Open *Windows 10 Control Panel* by right-clicking on the *Start* button and click on *Control Panel*.



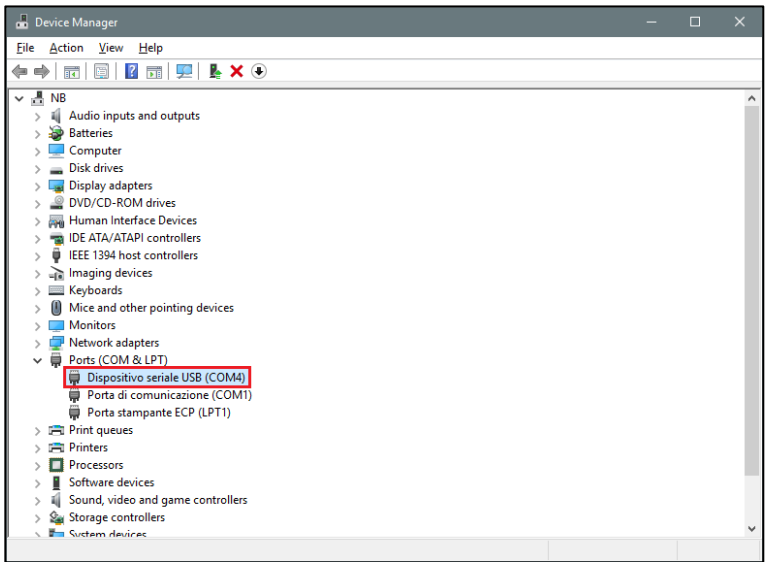
2. In *View by: Small icons*, click on *Device Manager.*



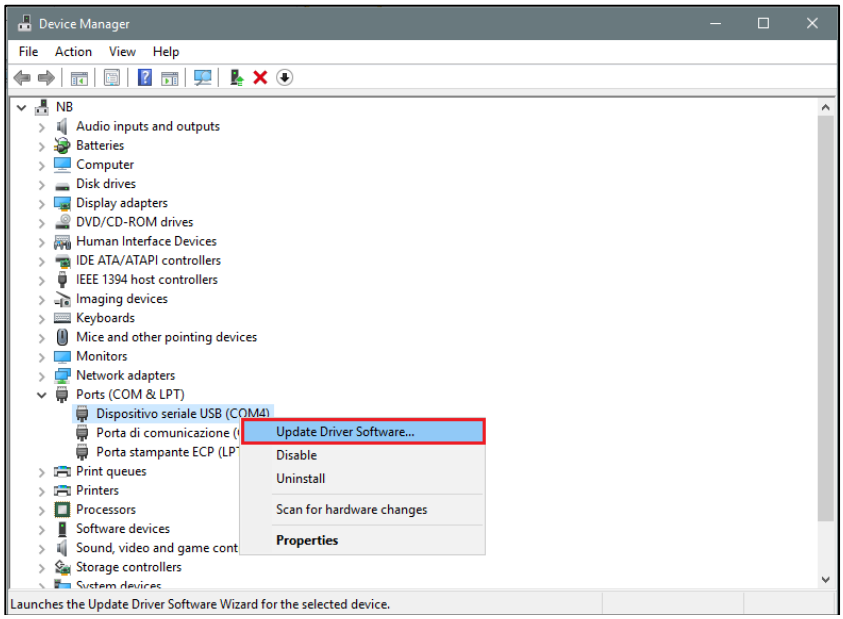
3. Before connecting the mini-USB cable to the PC and to the SBP2WEB24 controller, in the list of hardware categories double-click on the category Ports (COM & LPT) and take note of the serial communication (COM) ports in use. In the example shown below, the only serial communication port is ***COM1***.



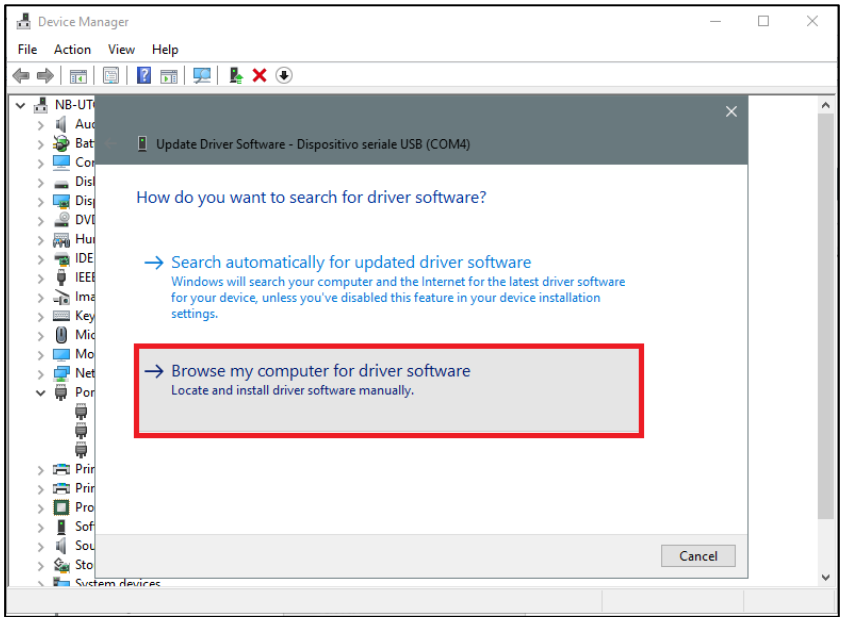
4. Plug the mini-USB cable into the PC and into the SBP2WEB24 controller. A new serial communication port (COM*x*) will automatically be added after a few seconds. In the example shown below the new port added is ***COM4.***



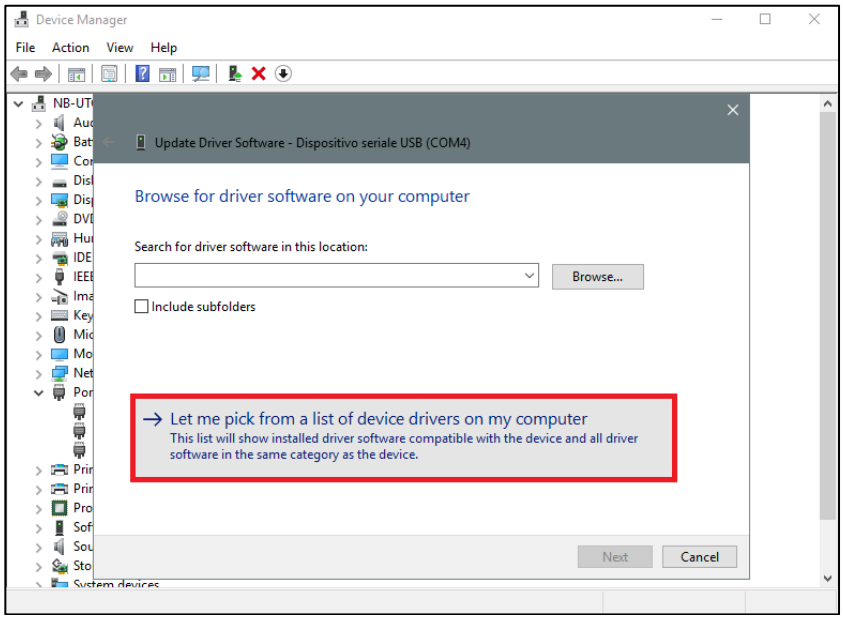
5. Right-click on the device and select *Update Driver Software*...



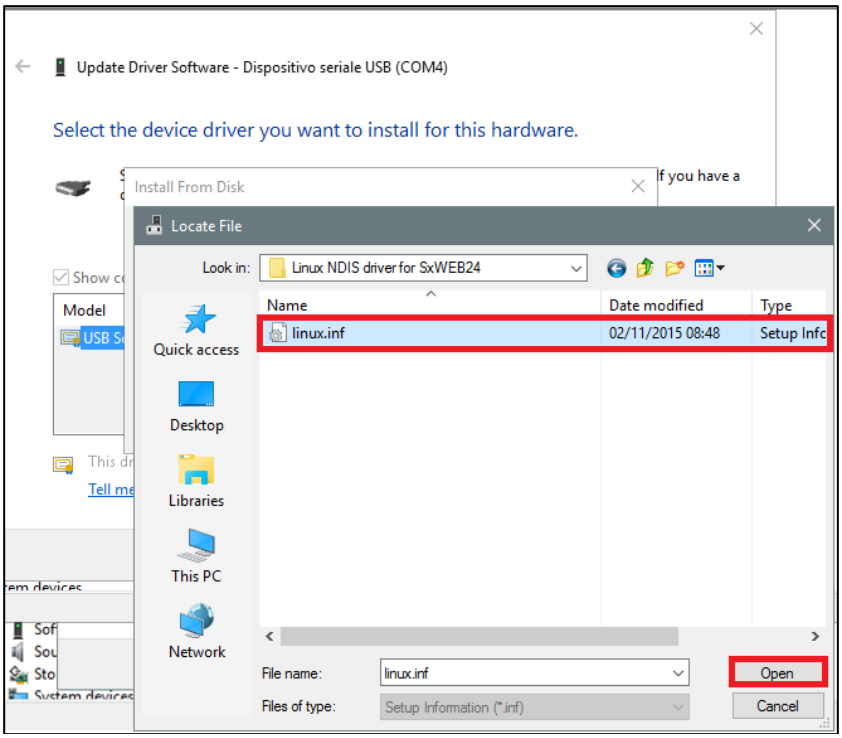
6. In the pop-up window, select the second option, *Browse my computer for driver software*



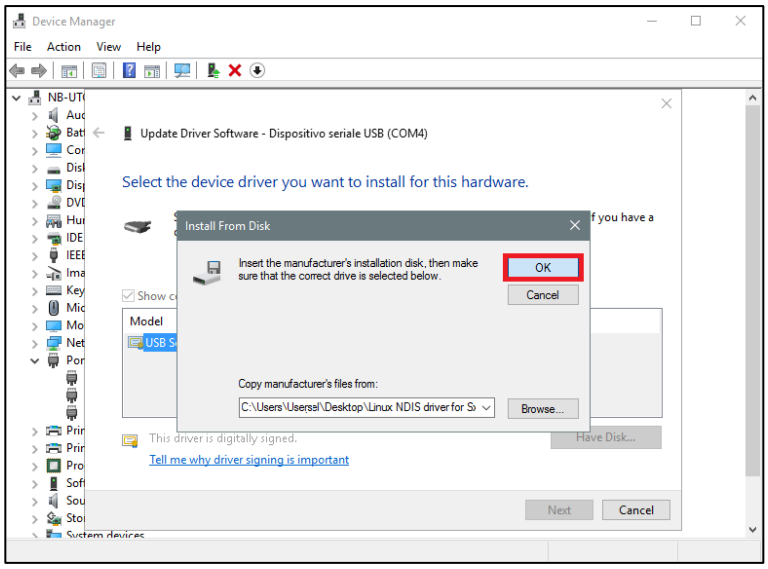
7. In the next page, select the option *Let me pick from a list of device drivers on my computer*



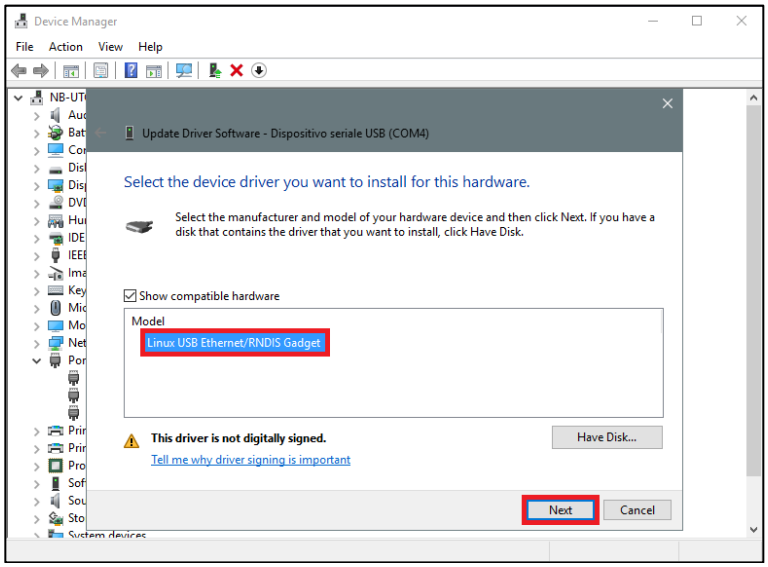
8. Browse for the *linux.inf* file, save it in the PC and press *Open*



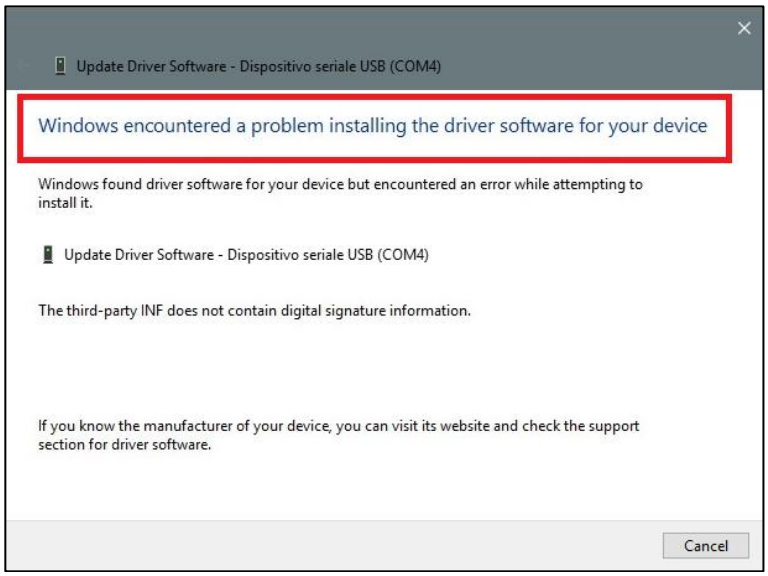
9. Press *OK* in the next window, as shown in the picture below



10. The *Linux USB Ethernet/RNDIS Gadget* is ready to be installed. Click on *Next* to continue the installation. See the



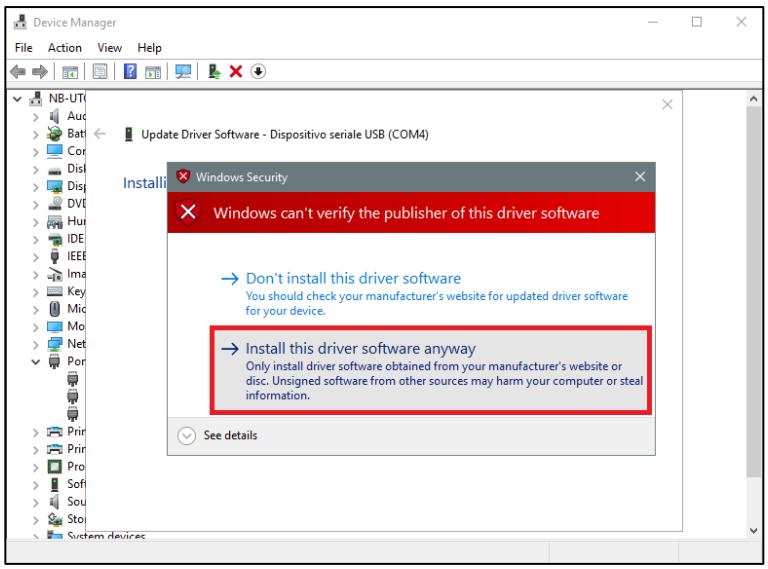
Follow the instructions below to disable driver signature enforcement **ONLY** if the user gets the error shown in the picture below, otherwise go to step 11.



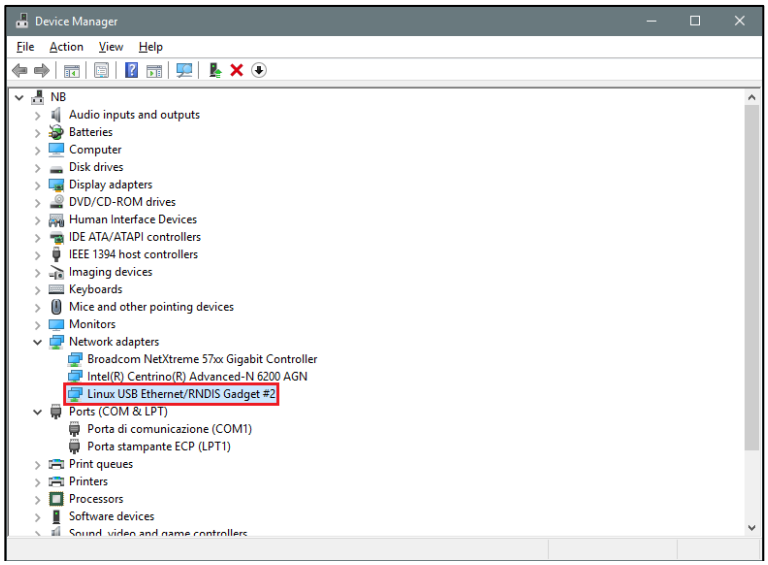
Windows 10 enforces driver signatures by default. This can be disabled to install drivers that are not digitally signed. Please refer to the following steps to disable driver signature enforcement.

1. Click the **Start** **menu** and select **Settings**.
2. Click **Update and Security**.
3. Click on **Recovery**.
4. Click **Restart now** under **Advanced Startup**.
5. Click **Troubleshoot**.
6. Click **Advanced options**.
7. Click **Startup Settings**.
8. Click on **Restart**.
9. On the Startup Settings screen press 7 or F7 to disable driver signature enforcement.
10. The computer will restart and the user will be able to install non-digitally signed drivers.

11. **N.B. Windows 10 does not detect the Digital signature, so to install the driver correctly, click on *Install this driver software anyway.***



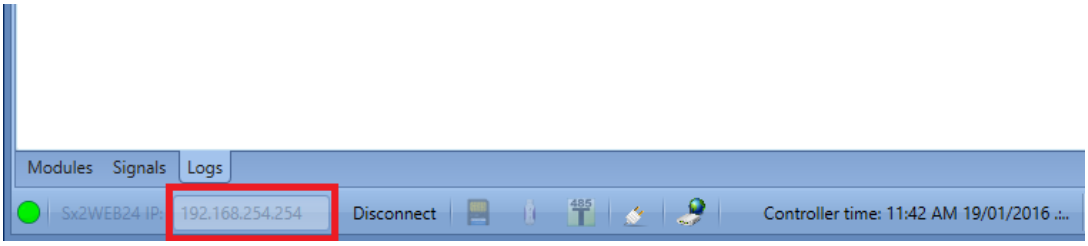
12. When the driver has been installed, in the *Network adapter* category a virtual network board named *Linux USB Ethernet/RNDIS Gadget #2 will* be added, as shown in the picture below.



13. The driver automatically gives a dynamic IP to the Controller/PC according to the actual IP of the PC.

For example, if the PC has the IP 192.168.0.10, the virtual board will be created with a new address 192.168.254.xxx and the address 192.168.254.254 will be assigned to the SBP2WEB24 controller.

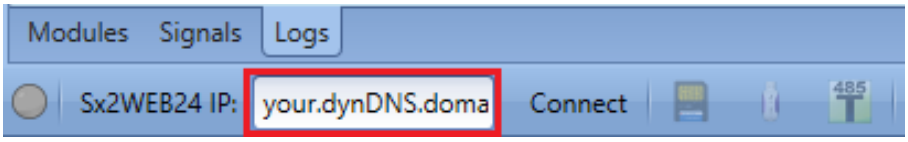
The user can type the IP address **192.168.254.254** in the address bar and click *Connect*. The green circle icon appears in the left part of the address bar when the connection is established, as shown in the picture below.



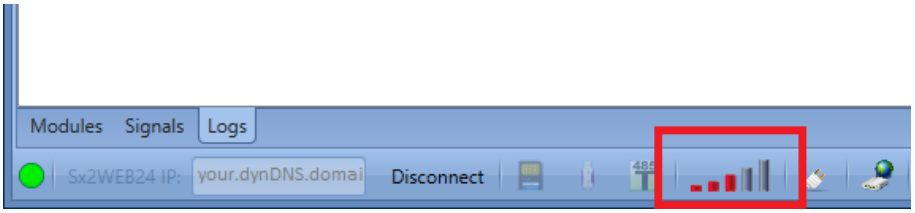
2.1.3 How to connect to the controller by means of a modem

When the user has to connect to the controller for the first time, the modem connection is not available, **so first it has to be configured.** Please refer to the section *How to configure the modem* in the system manual for more information (**http://www.productselection.net/MANUALS/UK/sx_tool_manual.pdf**).

In the IP address bar, the user can type in the dynamic IP address delivered by the ISP, or they can directly fill in the *DynDNS domain* registered (see *How to set the DynDNS parameters when using the modem*): the second option is preferable since the SBP Tool is able to auto-update the dynamic IP address delivered for the DynDNS account each time the modem is reset.

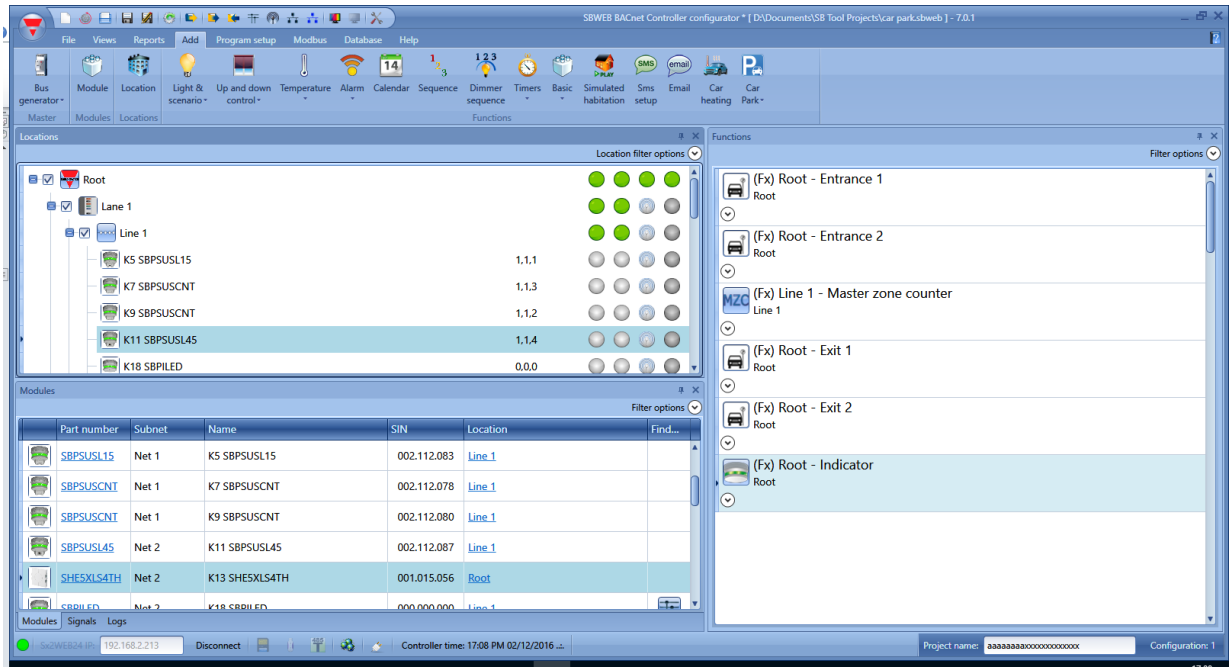


When a connection is established with the modem, the network signal strength will be shown by a set of bars, as shown in the picture below. The higher the number of red bars, the stronger the network signal.



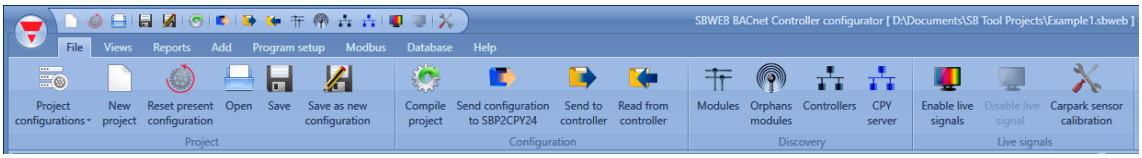
3 User interface

When the SBP tool starts, the following window appears:



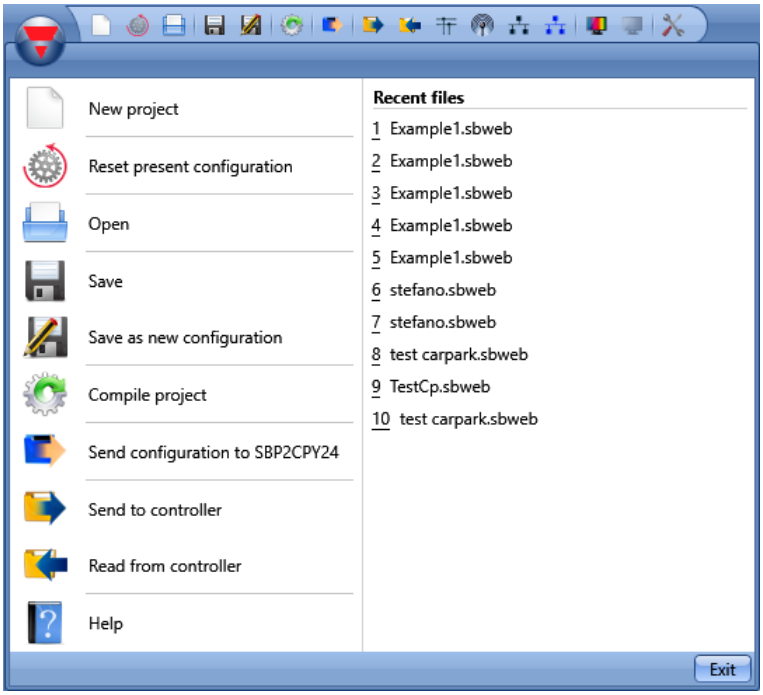
The user interface uses the standard ribbon tool often used by a lot of software nowadays. To access a ribbon, just click on the relevant menu.

3.1 File Menu



In the File menu the user can create a new project (single and multi-configuration), open an already existing one or save it as in a standard menu File.
In addition some functions strictly related to the connection with SBP2WEB24/SBP2CPY24 are available and listed here:

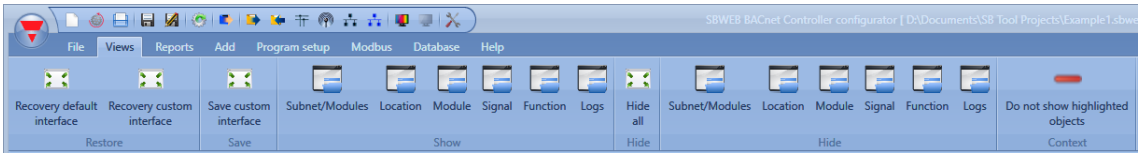
- Compiling a project
- Uploading/downloading of a project
- Discovery functions of the SBP2WEB24, SBP2CPY24 and modules connected in the network
- Addressing and calibration of car park sensors
- Enabling/Disabling of Live signal monitoring



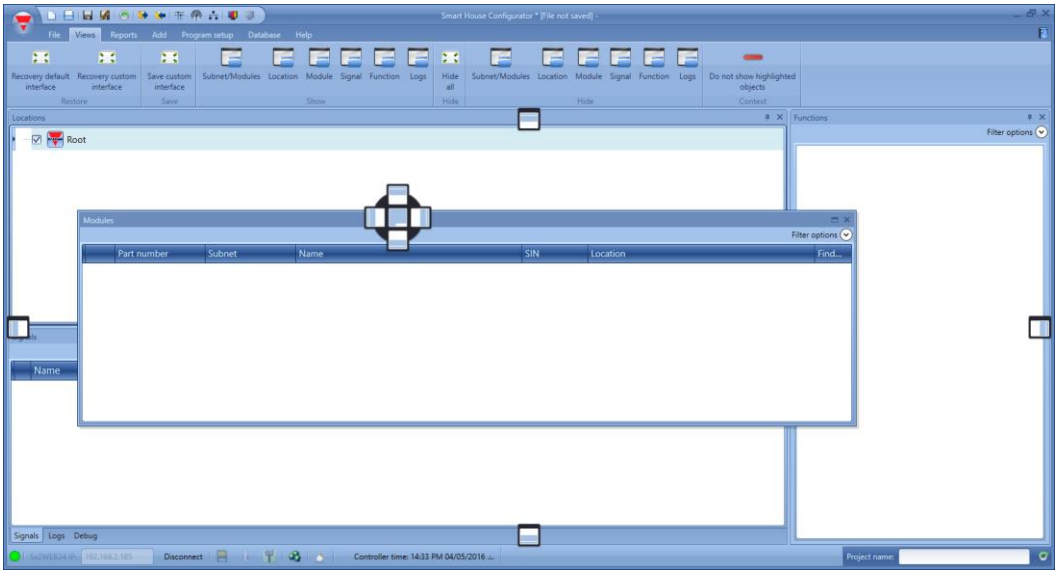
The user can access the menu file either with the quick menu in the upper part of the window on the right of the red Carlo Gavazzi triangle or by clicking on the triangle.

3.2 View Menu

In the View menu, the user can configure the preferred position of the windows relevant to Locations, Modules, Signals, Functions and Subnet. It can also remove the contents view.



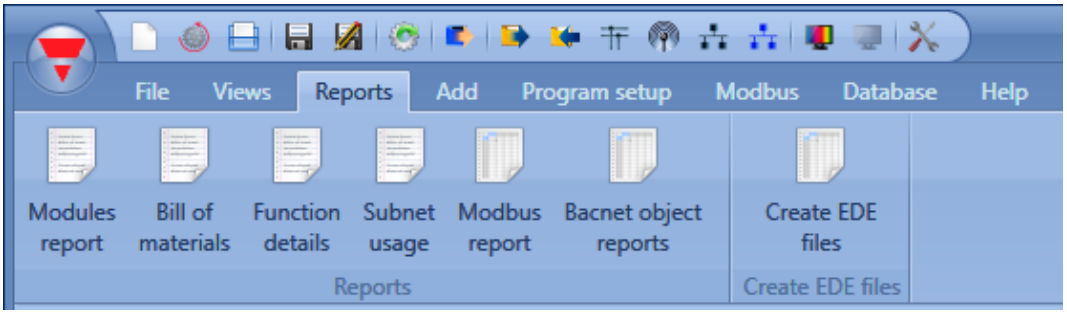
The windows are floating and can be positioned by using the five docking areas shown in the figure below. The position of the windows can be saved.



3.3 Reports menu

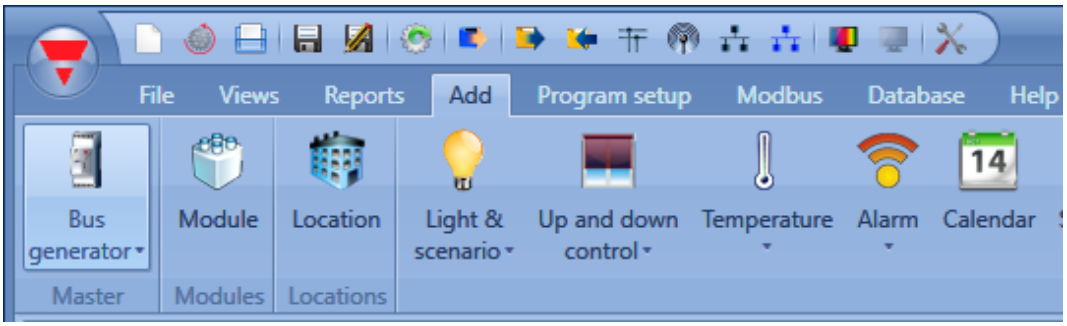
In the reports menu, five different kinds of reports can be created, saved and /or, printed.
The user can select from:

- *Modules list*: the complete list of modules is shown.
- *Bill of materials*: the list of modules is organized by type of modules.
- *Function details*: each function is described with the details of used signals.
- *Subnet usage*: this report indicates the number of used signals and the total current consumption.
- *Modbus TCP/IP report:* in this report the Modbus map of the project is reported.



3.4 Add menu

In the Add menu the user can select what to add: bus extension, modules, locations and functions.



Bus generators: if a new Dupline network is needed, a new bus extension module has to be added.
Module: a new module is added (light switch, pir sensor, output module…)
Location: The user has to define floors, rooms or any other type of location to have a clear structure of the installation.
Functions: some predefined functions can be defined and configured by means of the wizard tool. The predefined functions are:

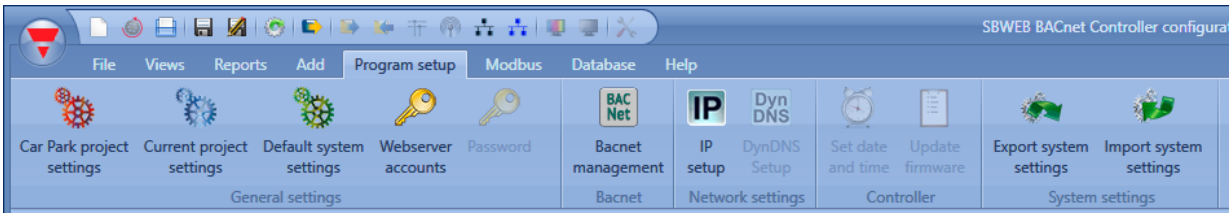
- Lights & scenario
- Up and down control: for controlling blinds, curtains, windows
- Temperature System Functions
- Alarm: intruder, smoke, water, siren
- Calendar: can execute activities during the year
- Sequence: executes a list of chosen functions
- Dimmer sequence
- Timers
- Basic functions: this section contains Counter, Logic Gates, Analogue comparator, Mathematical function, Analogue output, Hour counter
- Simulated habitation: records and then plays back the light activations

- Sms setup: manages the SMS for the remote control of functions
- Email
- Car heating

For the building automation functions, database management, communication protocols and any other function that is not covered here, please refer to the system manual at this link: http://www.productselection.net/MANUALS/UK/sx_tool_manual.pdf

3.5 Program setup menu

In the *Program setup* menu, the user can configure the settings relevant to a specific project, the general settings of the installer, the IP address and time and date of the SBP2WEB24. The user can also update the firmware and configure the webserver and the password to access the controller.

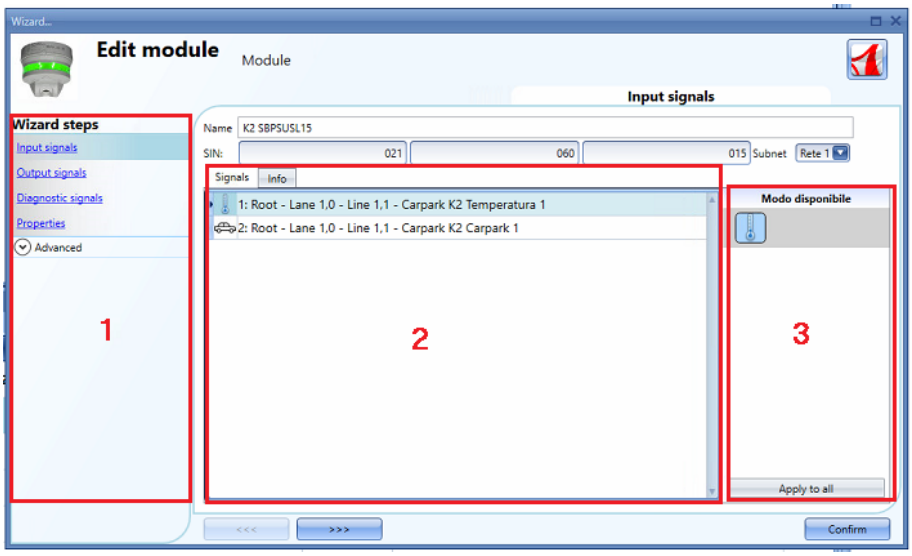


4 Project structure

In a project the user has to define the locations of the installation, add the required modules (sensors, display…) and place them in the relevant location. then create the functions.
Location, *modules* and *functions* are the pillars of the project structures: they are described in detail in the following paragraphs.

4.1 Wizard

Each object, whichever type it is, is created and configured by means of the *Wizard* tool.



The wizard is a tool that drives the user in the configuration of an object, guiding them in small steps to the complete setup of a module, location or function. The aim of the wizard is to reduce the effort of understanding the complete process of a configuration, making it easy and fast. The different steps can be filled in one by one just by clicking on the *">>>"* button and going through all of them, or by clicking on those required visualized in blue in the area on the right (Area 1).

In the picture above an example of the wizard tool divided into three areas is shown.

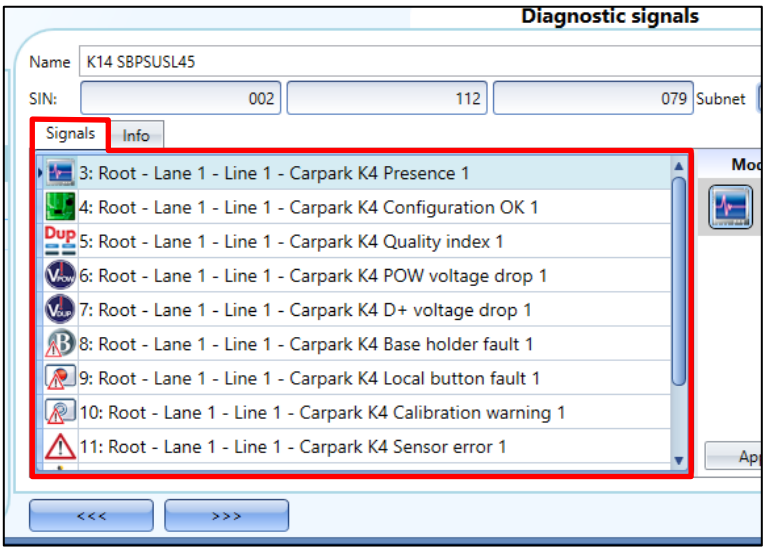
4.1.1 Area 1

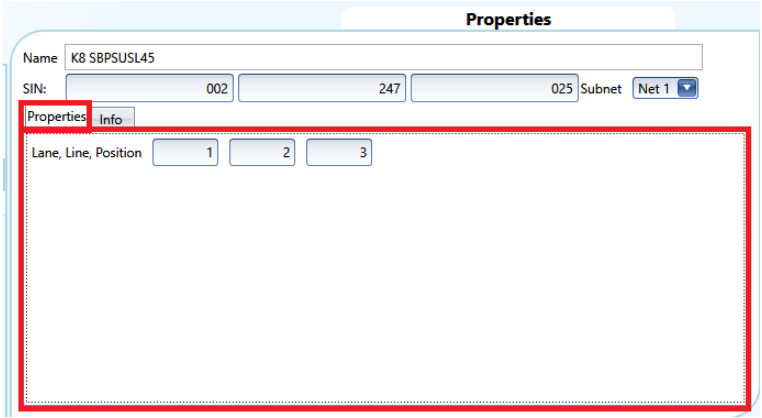
Area1 is divided into two sections. The one on top contains the basic signals/settings the user has to insert/define to create the object ('object' is a general word to indicate location, modules or functions). For example, in a light function the basic signals are the input and the output signals, which are the minimum settings required to create this kind of function. In general, most functions show the input and output signals in the basic settings.

In the lower section the "Advanced" functionalities are editable: if they are not enabled they are hidden to the user in order to give an easy user interface to the not-so-skilled installer.
The list of Advanced functionalities will appear: tick the relevant box to enable the one required. For example, if the Lux sensor is to be used, click on this. Once the complete list disappears, by clicking again on *Advanced*, the enabled one/ones will appear for quick access.

4.1.2 Area 2

This is the area where the available signals are shown or where properties can be changed.



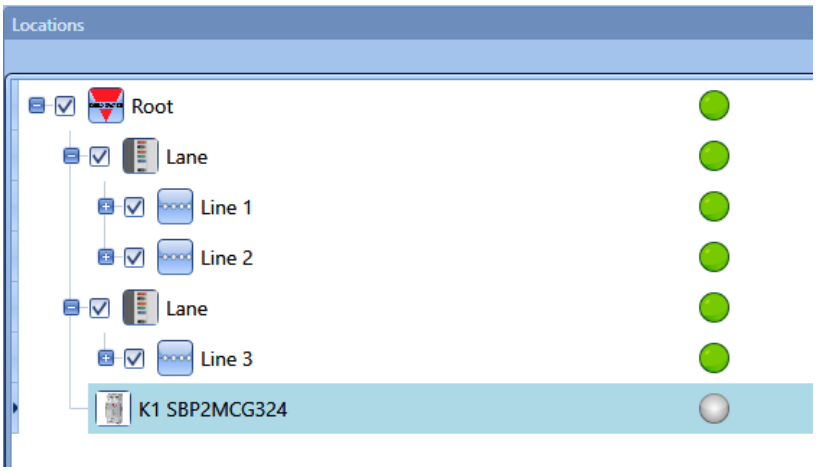


4.1.3 Area 3

This is the area where properties of signals are shown and in some modules also changed.

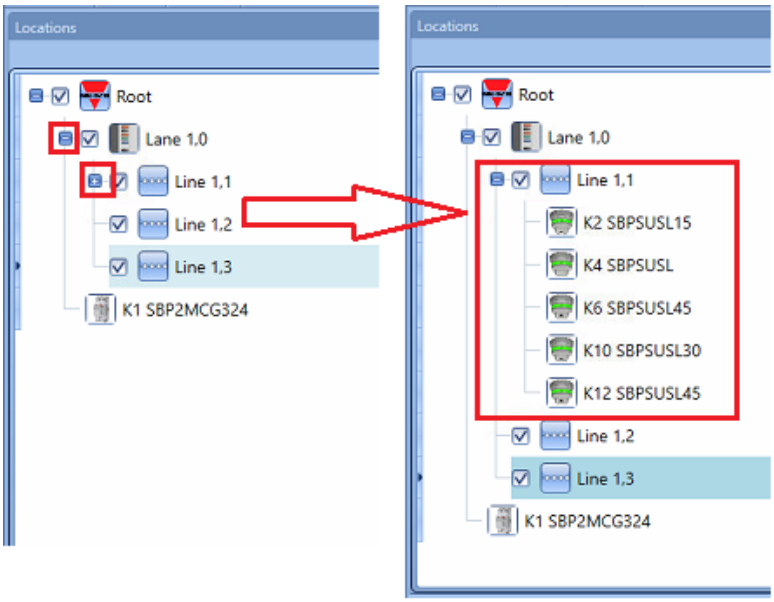
4.2 Locations

In the *Locations* window the user can define the structure of the project, starting from the locations where modules have to be placed: the user has to define lane, line or any other type of location to have a clear structure of the installation.



From the Location window, the user can select which locations have to be shown or not: all the modules and functions related to the locations will be hidden/shown depending on the locations selected in the project tree.

Click on the *plus sign* (+) to expand a Location and to see its sub-locations and modules:



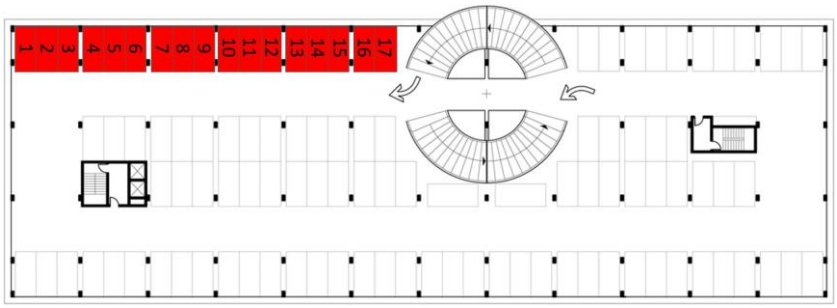
The checked box next to each Location indicates that it will be shown by default. The user can choose which Locations to show or to hide by clicking on the relevant symbol.

- To show the modules associated to a location, the check box must remain checked
- To hide the modules associated to a location, the user has to uncheck the location. The objects associated to its sub-locations will also be hidden.

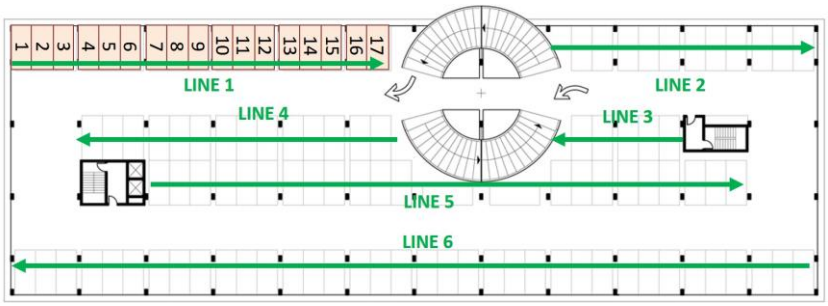
5 Lane, line, position

In a car park project commissioned with the SBP tool and the CPY server, it is important to understand the concept of lane-line-position to program the sensors and the displays.

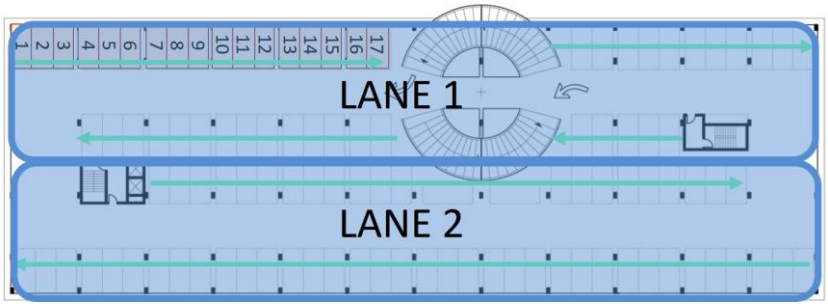
Position: sensors are grouped and each sensor in a group is represented by a number which corresponds to its address. In the picture below, there are 17 sensors, and each of them is identified by a number from 1 to 17.



Line: any group of sensors is a line: any line is also indicated by a number

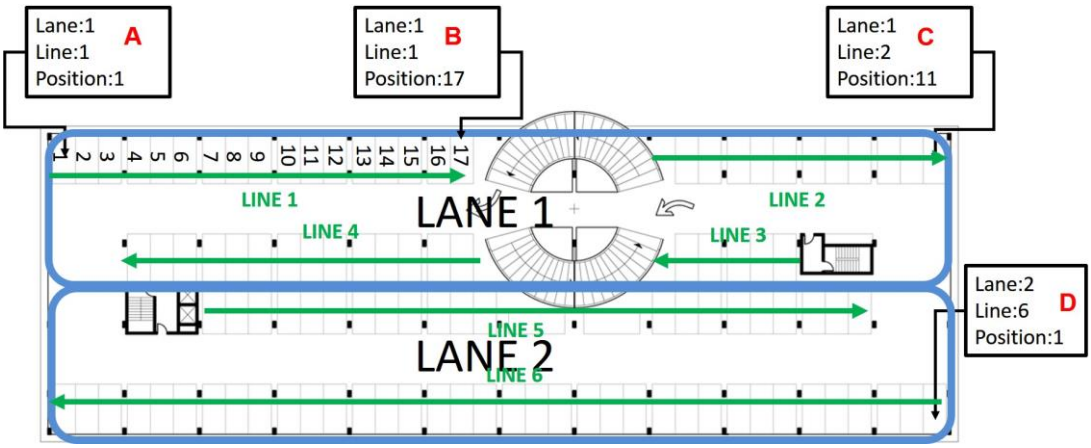


Lane: any group of lines is a lane: each lane is indicated with a specific identification number in the project

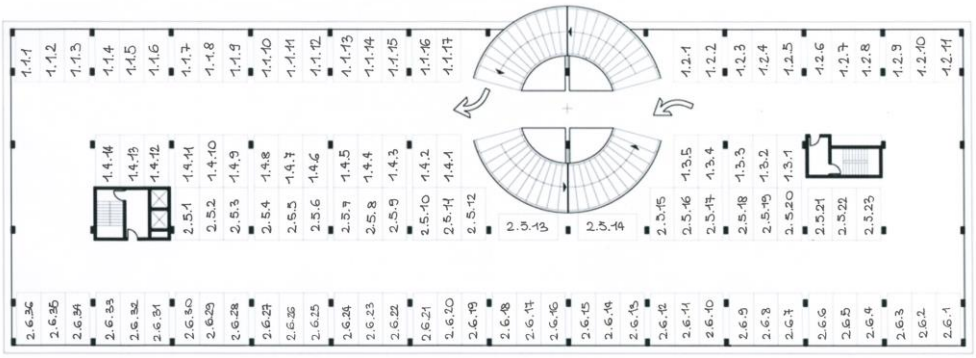


According to the grouping explained above, each sensor is uniquely identified by its position in a line: the car park address is made up of three numbers, each separated by a dot, very similar to the SIN number, and it represents the lane, line and position of the sensor.

Address sensor A: 1.1.1 (lane.line.position)
Address sensor B: 1.1.17
Address sensor C: 1.2.11
Address sensor D: 2.6.1

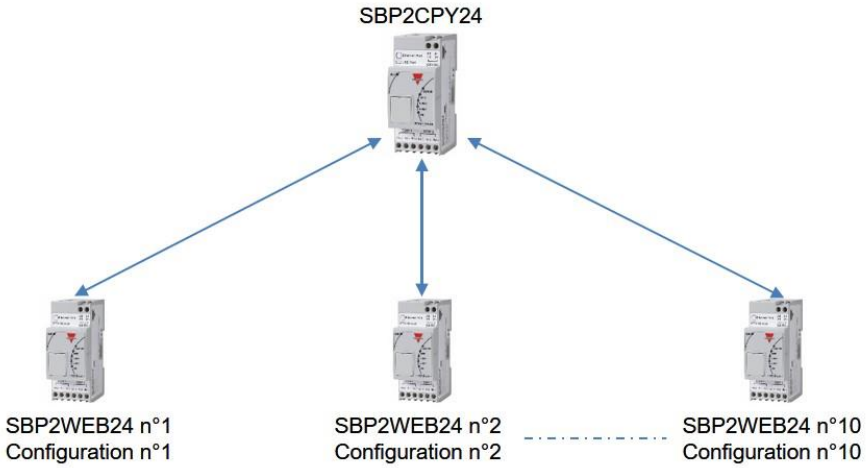


In the following image, the whole car park has been mapped with the address of each sensor.

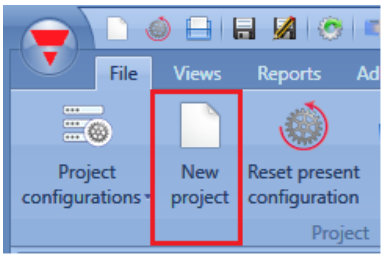


6 How to create a new project

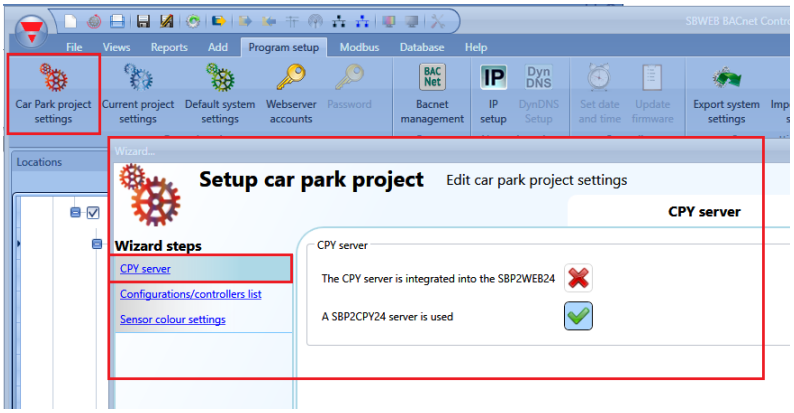
A car park project can include up to ten configurations for up to ten SBP2WEB24s each, communicating with one SBP2CPY24, as shown below.
The SBP2CPY24 includes the CAR Park server, which can be integrated into the SBP2WEB24 in small installations.



When the tool is opened or when a new project is created by clicking on the icon *New project,* one configuration is added by default.



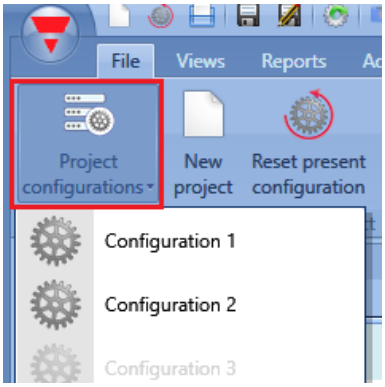
In the new project, select where the car park server is located: if there is only one SBP2WEB24 the car park server can be integrated into it: otherwise, in the case of a multi-configuration project a SBP2CPY24 is needed. In the *Program setup* menu, click on *Car Park project settings* and then in the new window select *CPY server:*



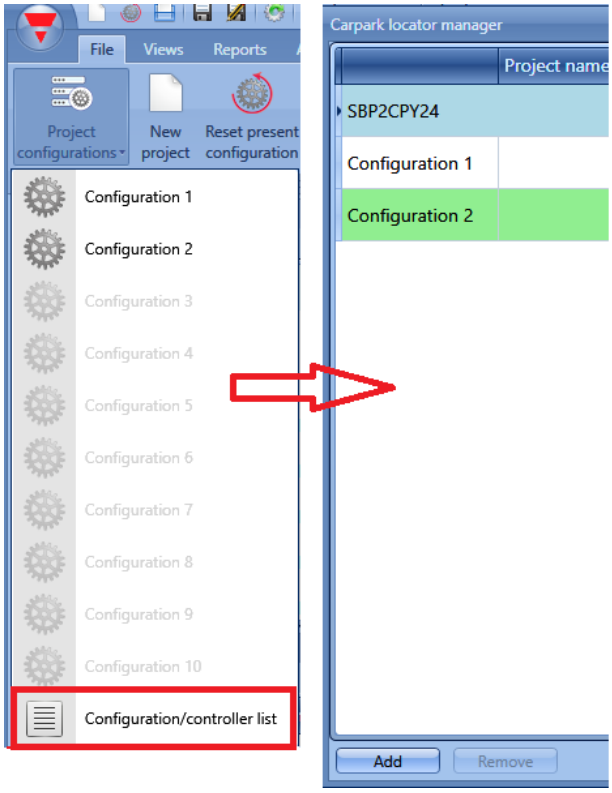
Should the CPY server be included into the SBP2WEB24, the address to access it is:

***controller_ipaddress*/CP3App**

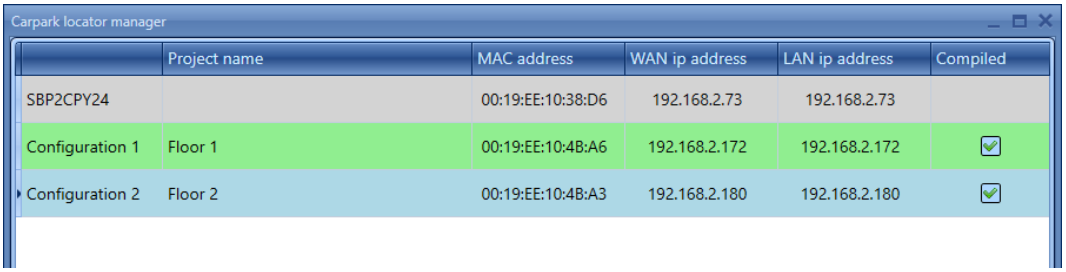
The list of configurations is displayed by clicking on the icon *Project configurations:*



To add a new configuration to the project with the relevant SBP2WEB24, click on *Configuration/Controller list,* as shown in the picture below:



The *Car park locator manager* window will appear with the list of the configurations and the SBP2WEB24 controllers associated to them:

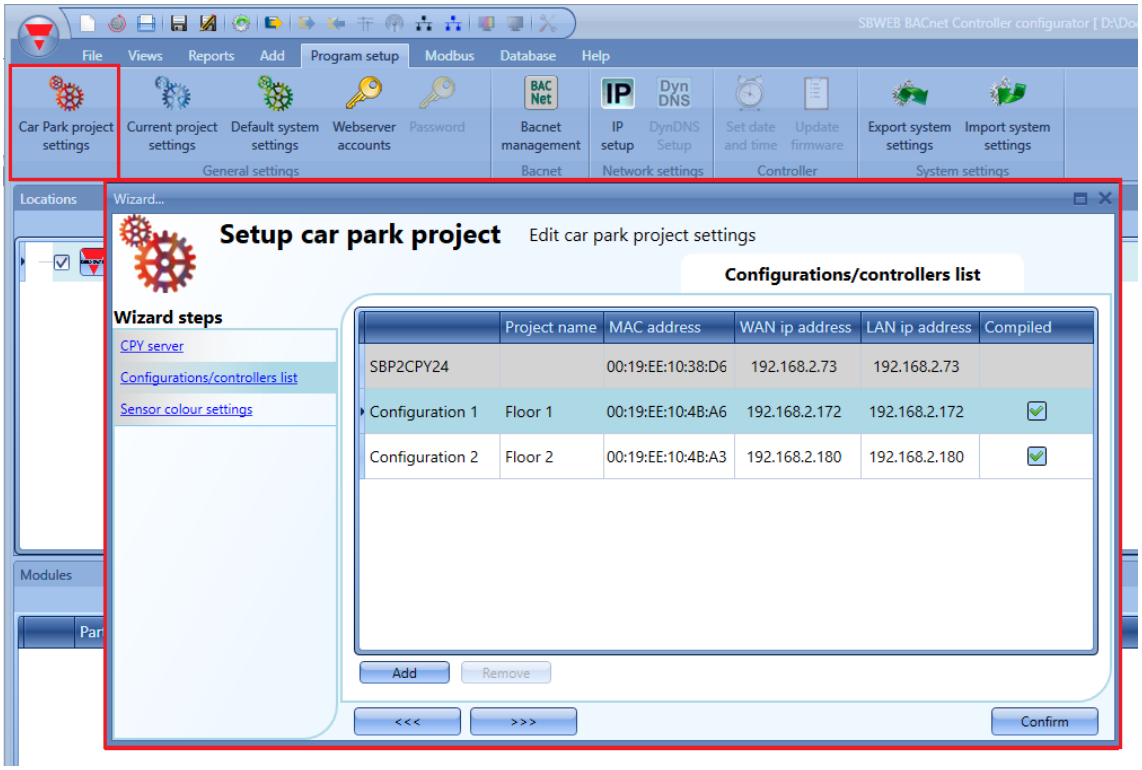


The first item on the list is the CPY server that can be integrated into the SBP2WEB24 controller or the SBP2CPY24 dedicated item in a multiconfiguration project. Later, the different SBP2WEB24s configurations (up to 10) are shown.

The following table shows the available parameters:

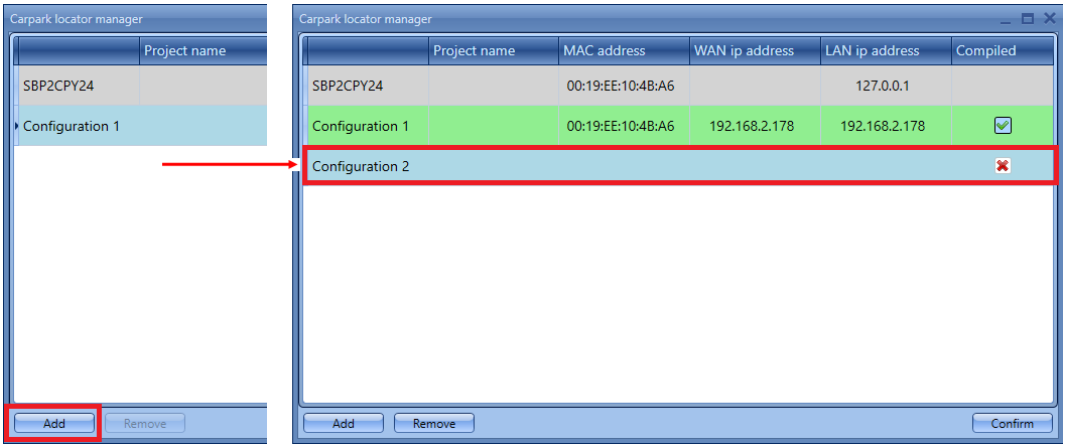
Field	Description
Project name	This field shows the project name of the configuration, the user can change it
MAC address	This field shows the physical address of the SBP2WEB24/SBP2CPY24 modules
WAN ip address	This field shows the current public IP address of the SBP2WEB24/SB2CPY24: this identifies the IP address (or DynDNS name) the user can insert to get the access from the Internet (when the user needs the connection of the LAN where the Car Park modules are installed)
LAN ip address	This field shows the IP address of the SBP2WEB24/SB2CPY24 devices in the LAN where the Car Park module is installed
Compiled	When a small red cross is shown, the user has to compile the project before sending the configuration to the SBP2WEB24 controller

Alternatively, the user can access the same list from the *Car Park project settings* window:



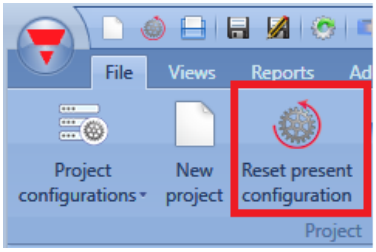
6.1 How to add a new configuration to the current project

From the *Carpark locator manager* window, by clicking on the *Add* button, a new line is added for a new configuration, as shown in the picture below:

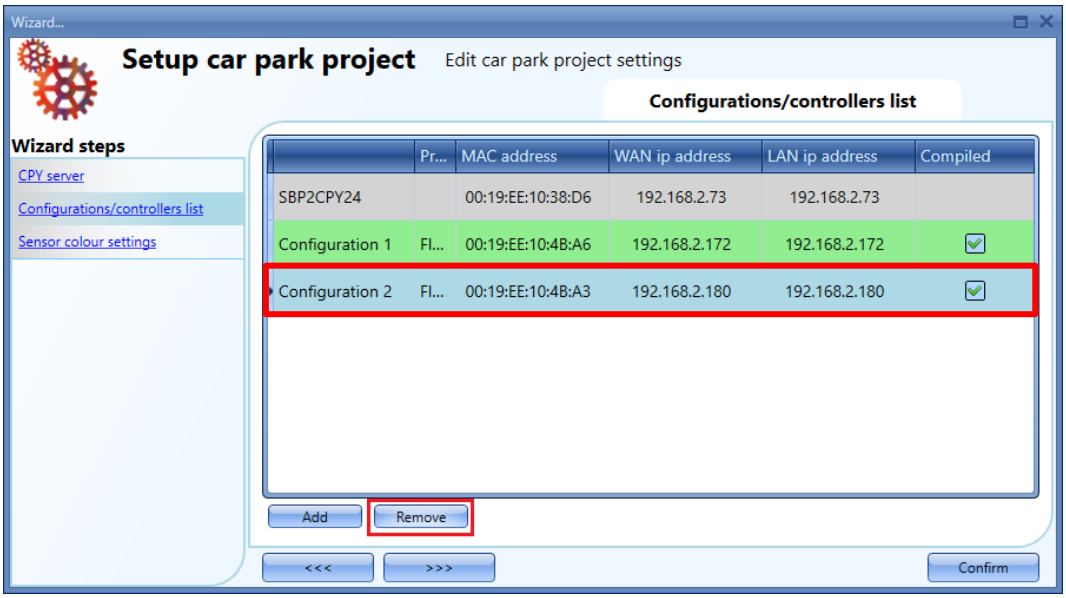


A new configuration can be added to the project at any time.
The green line indicates the configuration that is now open in the project.

To reset a configuration in the project, select the configuration from the list and click on *Reset present configuration*: the reset will delete all the locations, modules and functions.



To remove a configuration, select it from the list and click on *Remove,* as shown below:

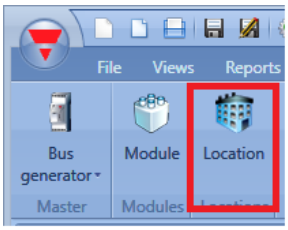


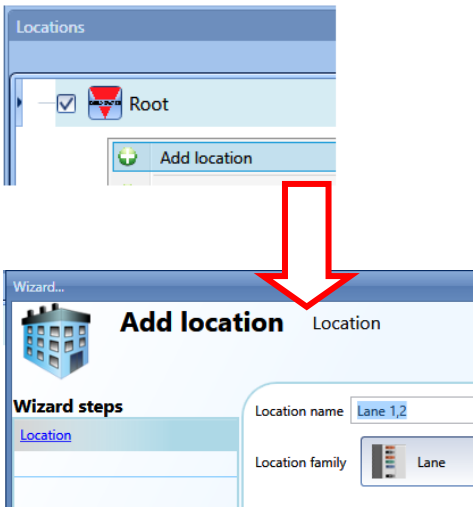
Please note that the present open configuration cannot be deleted.

Once the structure of the project has been defined with the relevant configurations, SBP2WEB24s and CPY server, the location tree has to be generated.

There are different ways to create a location tree:

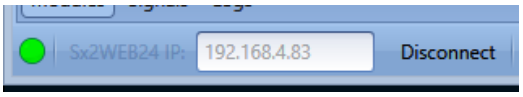
1) Starting from the *Add* menu from the top menu or by right clicking in the *Location* window



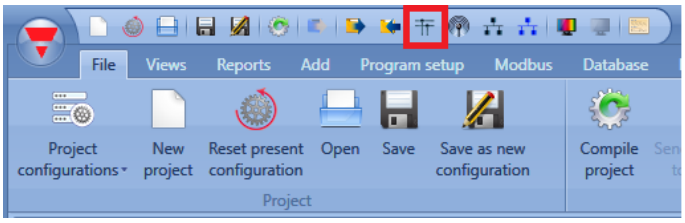


2) In a faster way, the location tree can also be generated starting from the *Discovery manager window:* to open this, follow these steps.

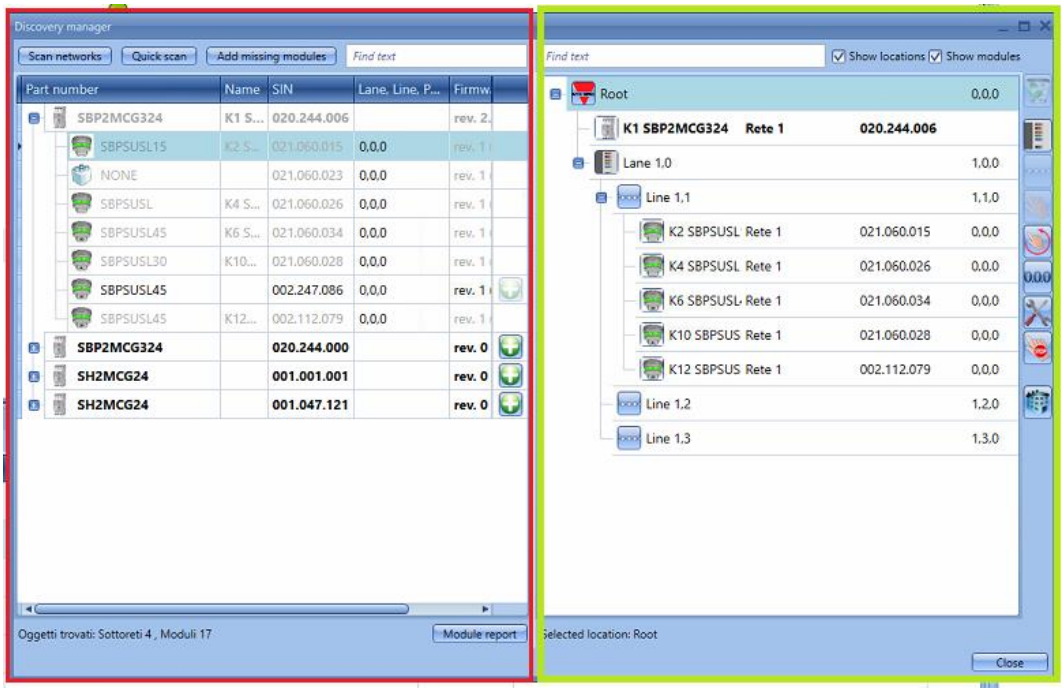
a) Connect to the controller



b) Launch a discovery of the Dupline module



c) The *Discovery manager* window will appear:



On the left of the window (marked in red), all the modules connected to the bus are prompted once the *Scan networks* push button is pressed.

The *Quick scan* button will list only the modules of the configuration written into the SBP2WEB, while the *Scan networks* button will list all the modules connected to the controller.

On the right, the location tree is shown. To generate this, only three push buttons have to be used:

To delete a location and all its objects, the push button with the basket has to be used

This adds a location *Lane*. It is enabled if locations such as Root, Floors, etc are selected. It is not possible to add a Lane from a Lane location or from a Line location.

This adds a location Line. It is enabled only if a location Lane is selected since a line is a group of bays (parking spaces) in the lane

This starts/stops the single line addressing

This starts/stops the multi-line addressing

This resets the car park addresses of all the sensors

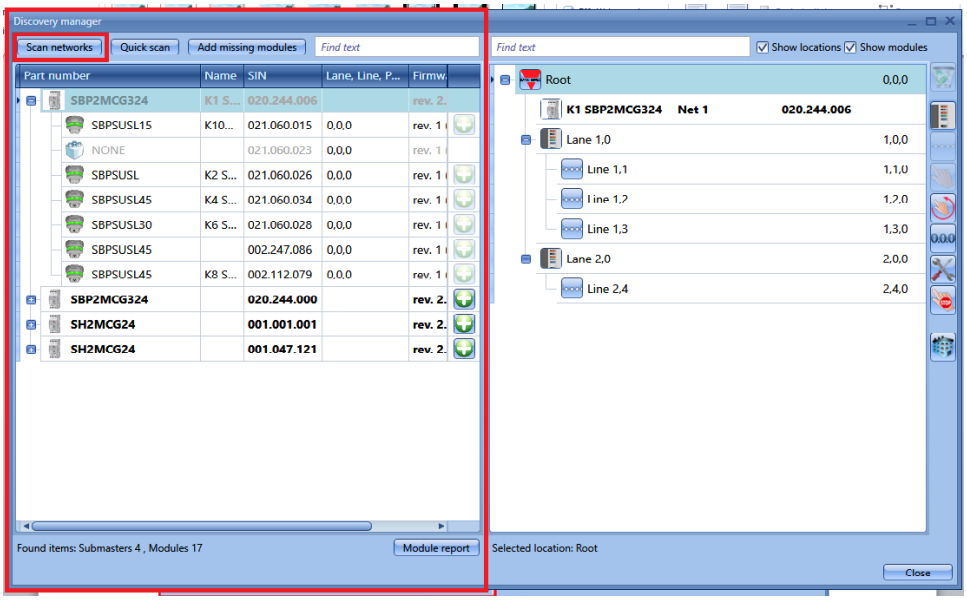
This stops any running activity such as a network discovery or an automatic addressing

This adds any type of locations. It is not enabled if a location line is selected.

Once the location tree is ready, the user can proceed to add the modules: there are two ways of doing this, an automatic scan of the network launched by the *Discovery manager* window, and a manual mode.

6.2 How to automatically find and address the sensors – The controller is connected to the modules

In the *Discovery manager* window, click on *Scan networks*: all the modules present on the bus will be prompted.



Once the modules are discovered, they have to be placed in the locations and car park addresses have to be assigned.

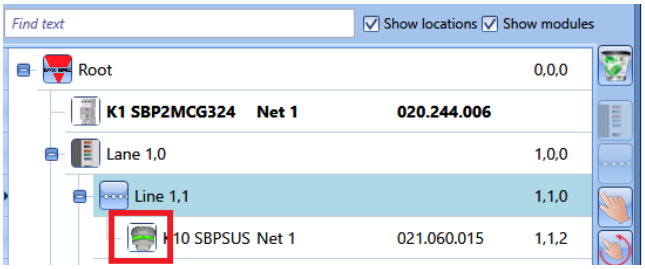
6.3 How to manually place the discovered modules

There is one important rule to understand when placing modules: the sensor SPBSUxxx can be placed only in the location *Line,* and in the location *Line* only SBPSUxxx sensors can be placed. There are no restrictions for the other modules/locations.

Modules can be manually placed in the locations in the following ways:

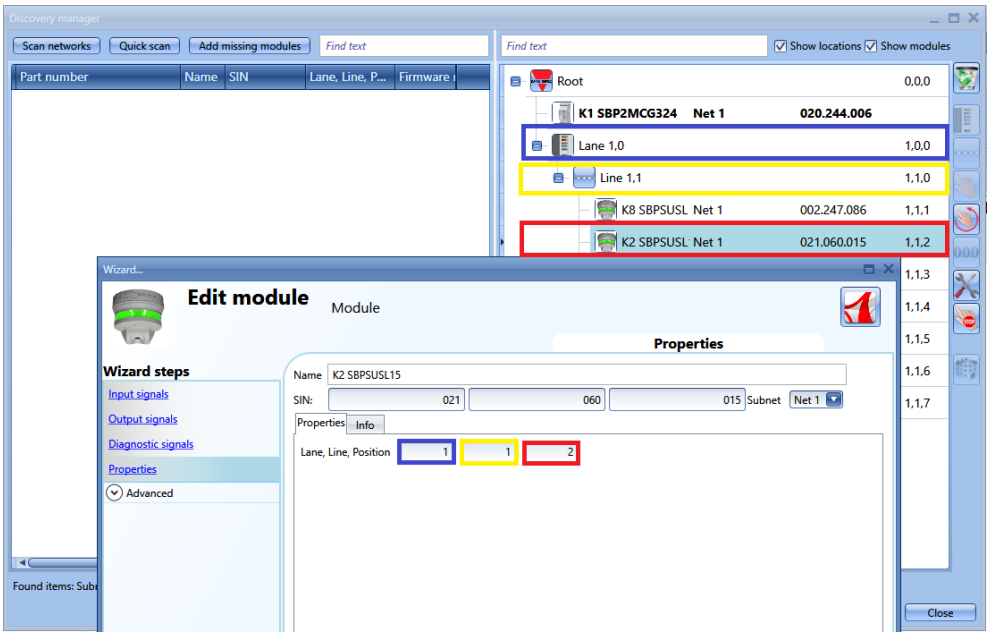
1) By selecting the locations and then clicking on the icon.
This icon will be enabled only if the selected location is allowed to contain that type of module.
2) Modules can be dragged and dropped onto the selected location: the drag&drop functionality also follows the rule mentioned above.

When modules are placed manually, the car park addresses also have to be entered manually, taking into account the lane and line addresses and the position of the sensor in the line.
Click on the icon relevant to the module in the location tree:



The wizard of the module will be opened.

In the *Properties* field enter the number of the lane and line the module belongs to and its position in the line.



6.4 How to automatically place the modules

There are two ways to place modules in the lines:

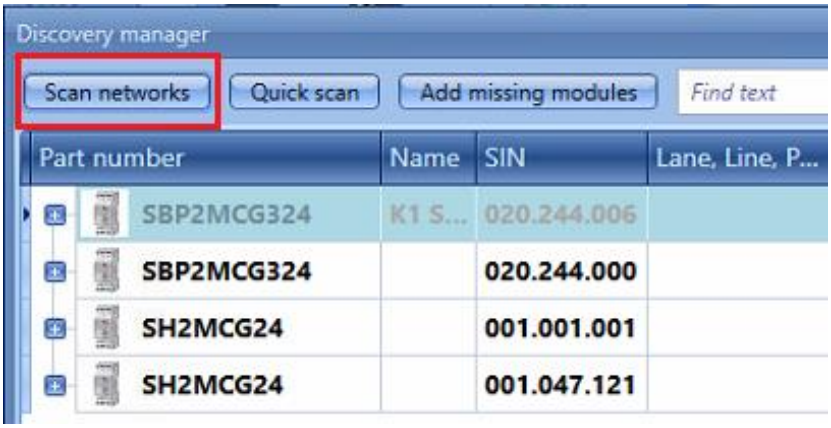
1) line by line, i.e. *Single line addressing*
2) more lines at the same time, i.e. *Multi line addressing*

6.4.1 Single line addressing

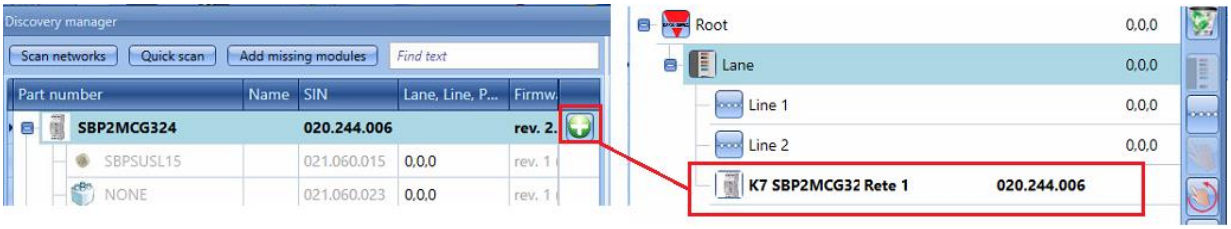
This type of addressing is available only if a location *Line* is selected.
The auto addressing lets the user place the sensor in the selected line simply by clicking on the push button on the sensor: the car park address will be set automatically. This procedure will address only the sensor present in the selected line.

The following steps have to be executed:

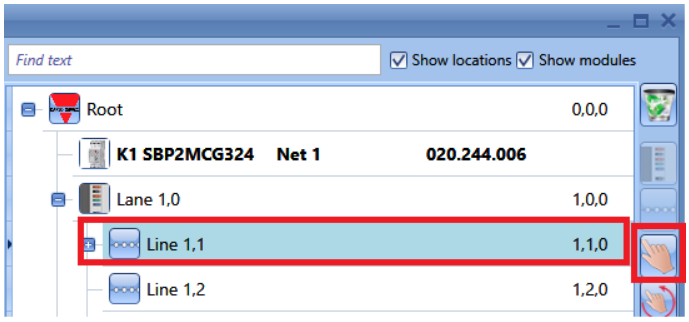
1) Launch the modules discovery



2) Add the three-wire Dupline generator to any location, except to location *Line*

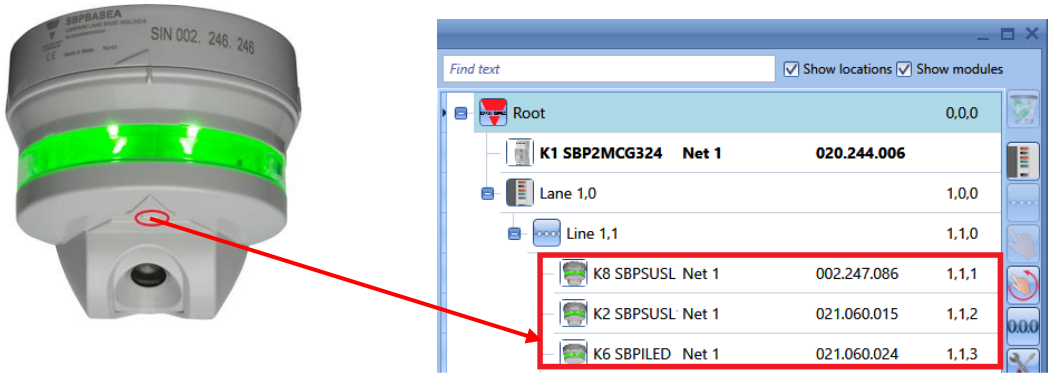


3) Select the line where the sensor is placed and click on the icon marked in red to start the automatic addressing



All the sensors with no address will start blinking a yellow LED.

4) Walk to the line where the sensors to be addressed are and press the button on the sensors one by one: the car park addresses will be assigned automatically and the sensors will be shown in the correct position in the location tree



As highlighted in red, as soon as the push button on the sensor is pressed, the sensor is moved from the left side of the window to the right side and the car park address is shown next to the SIN number.

5) Once all the sensors belonging to the line have been addressed, click again on the icon to close the procedure.

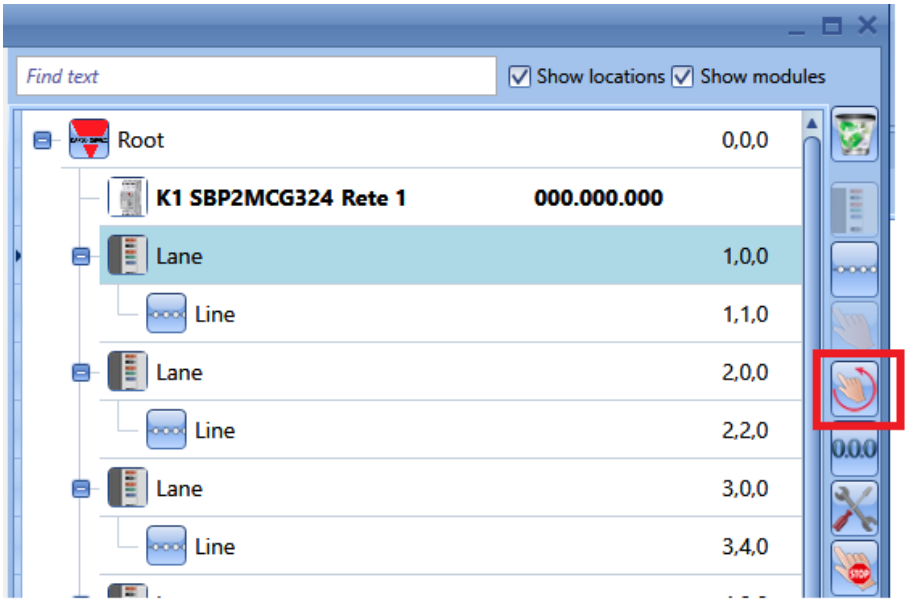
6.4.2 Multi line addressing

This procedure lets the user consecutively address the sensors belonging to many lines, without launching the procedure described above many times.
The user just needs to launch it once and walk around the parking area, pressing the buttons on the sensors according to the line order defined in the software tool.

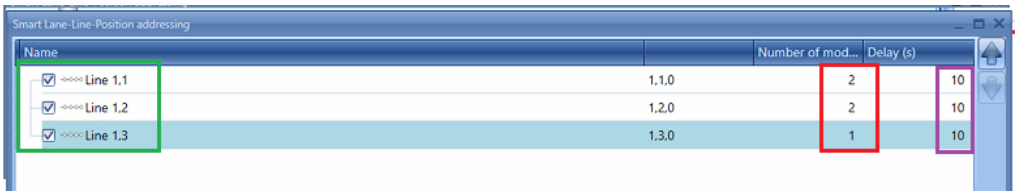
The following steps have to be executed:

1) and 2) as in the *Multi line addressing* mode

3) Select any location and click on the icon marked in red:



The following window will appear, in which the user has to select the lines to program by clicking on the boxes highlighted in green.



In the field marked in red, the number of sensors for each line has to be edited: in this way the system knows when one line is completed and it is time to switch to the following one/s.
In the field highlighted in purple, the delay between the addressing of two lines has to be set: this delay is needed to allow the installer to walk from one line to the following one: it can also be set to 0.

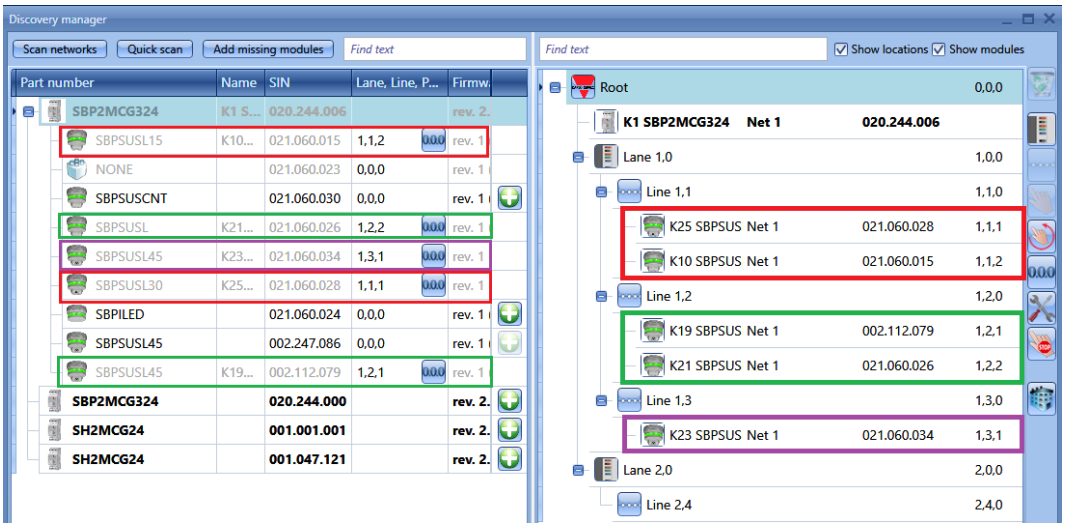
Once all the settings have been completed, click on *Confirm* to start the addressing: all the sensors will start blinking.

4) Walk to the lines to be addressed and press the button on the sensors: they will be added to the lines according to the order in which they are activated and the *Number of modules* set for each line. ***Please wait until the sensor that has just been programmed does not stop blinking fast before pressing the button on the following one.***

Example

1 According to the settings shown in the picture above, the first two sensors whose buttons are pressed are placed in line 1.1
2 the sensors stop blinking for 10 seconds
3 automatic addressing is enabled again (the sensors will start blinking again) and the following two activated sensors will be placed in line 1.2
4 then there is another delay of 10 seconds during which the sensors stop blinking
5 automatic addressing is enabled again (the sensors will start blinking again) and the following activated sensor will be placed in line 1.3
6 after another delay of 10 seconds, the intelligent addressing procedure is closed

5) The tree of the installation will be completed as shown in the picture below:



The first sensor whose button has been pressed is the one with SIN number 021.060.028 and it is placed in line 1 of lane 1, so its car park address will be 1.1.1.

The second sensor is the one with SIN number 021.060.015 and it is placed in line 1 of lane 1, so its car park address will be 1.1.2.

At this point the system automatically changes to line 2, since we have told it that two sensors have to be placed in line 1.

The third sensor whose button has been pressed is the one with SIN 002.112.079 and it is placed in line 2 of lane 1, so its car park address will be 1.2.1

The fourth sensor is the one with SIN 021.060.026 and it is placed in line 2 of lane 1, so its car park address will be 1.2.2

At this point the system automatically changes to line 3, since we have told it that two sensors have to be placed in line 2.

The fifth sensor whose button has been pressed is the one with SIN 021.060.034 and it is placed in line 3 of lane 1, so its car park address will be 1.3.1

At this point the system automatically closes the procedure.

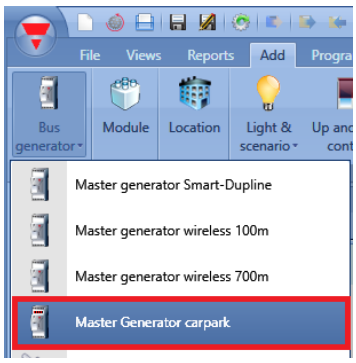
In the window on the left, as soon as the button is pressed, the relevant module is put in light grey and the only action that can be done is to reset the car park address by clicking on

6.5 How to manually add modules – The controller is not connected

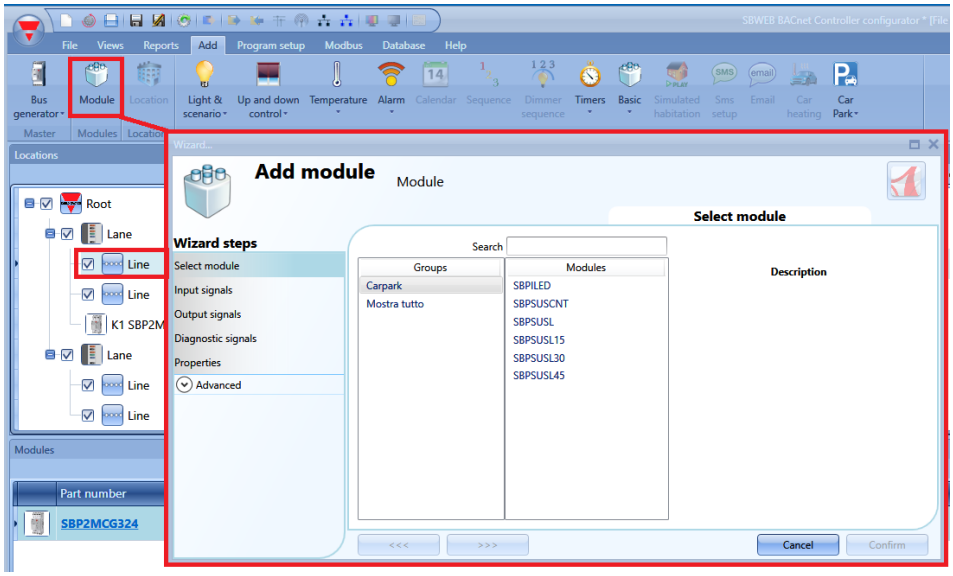
If the configuration has to be built off line (without the connection to an SBP2WEB24 or to any Dupline network), the modules can be added and addressed manually.
The following steps have to be followed to create the project.

1) Generate the location tree by clicking on *Add Location*

2) Add the three-wire master channel generator SBP2WEB24

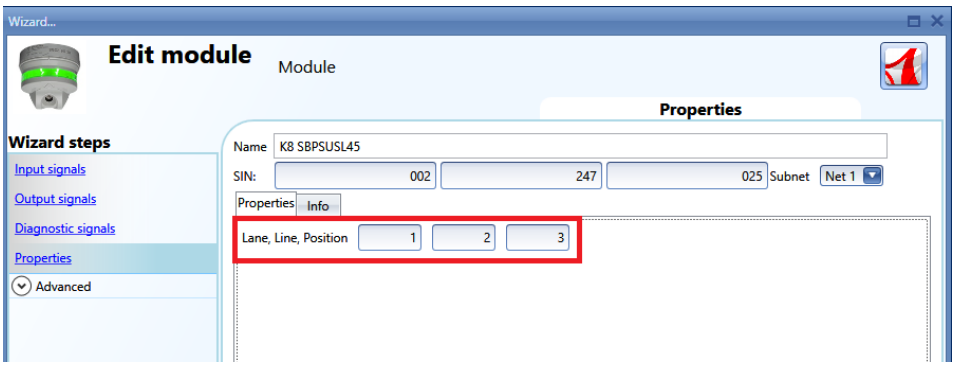


3) Add the sensors in the relevant *Line:* select the Line and click on *Module*. Select the sensor and click on *Confirm*.



In the location *line* only the sensors SBPSUSLxx can be placed.
The sensors SBPSUSLxx can only be placed in the location *line.*

4) Once the modules are added to the project, the addressing can be done manually or it can be done automatically at a later moment when it will be possible to connect to a Dupline network. **We strongly suggest doing it automatically to speed up the installation time and to reduce the possibility of making errors.**
Should this not be possible, the lane.line.position address can be edited manually from the wizard of the module.

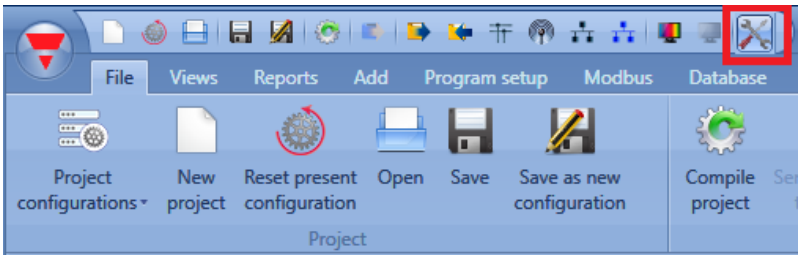


7 How to calibrate a sensor

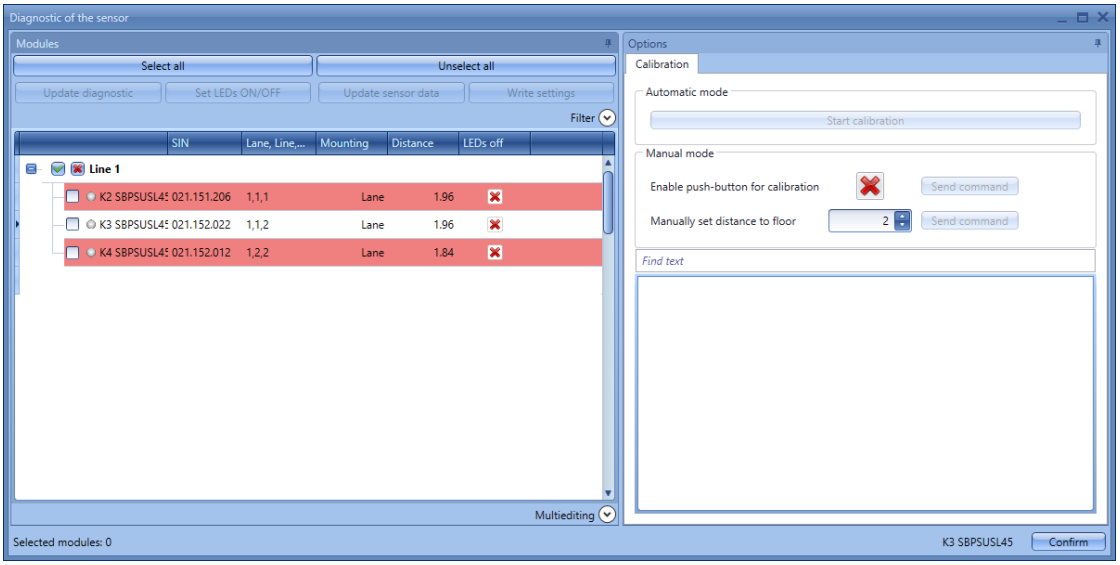
Once a sensor is mounted over the parking bay, it **needs** to be calibrated to distinguish between a vacant parking space (bay) and an occupied one.
The calibration has to be carried out with no car so that the distance in free air between the sensor and the floor is programmed into the sensor: according to this programmed distance, the sensor will then be able to recognise if the bay is occupied or not (see *Carlo Gavazzi CP3 - Guide and installation manual*).

The calibration has to be done after writing the configuration into the SBP2WEB24 controller and when the parking space is empty.

To start the calibration process, click on the icon marked in red once the controller is connected to the SBP tool:



The following window will appear:



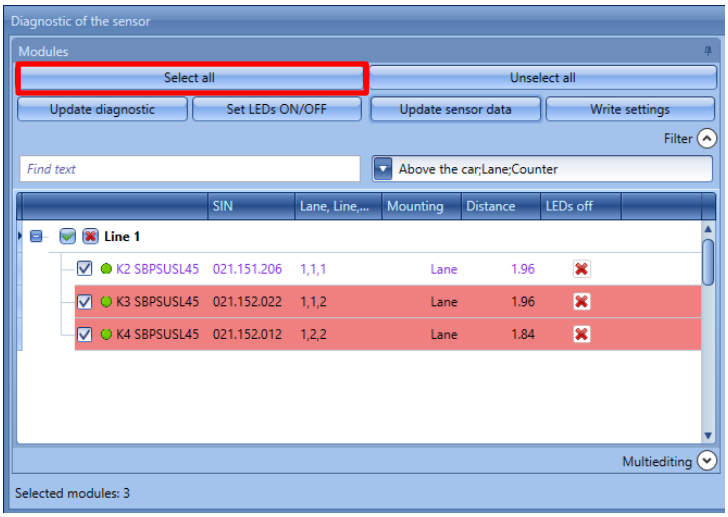
- The occupied sensors are marked in red
- The unoccupied sensors are marked in white

For each sensor, the type and distance are shown.

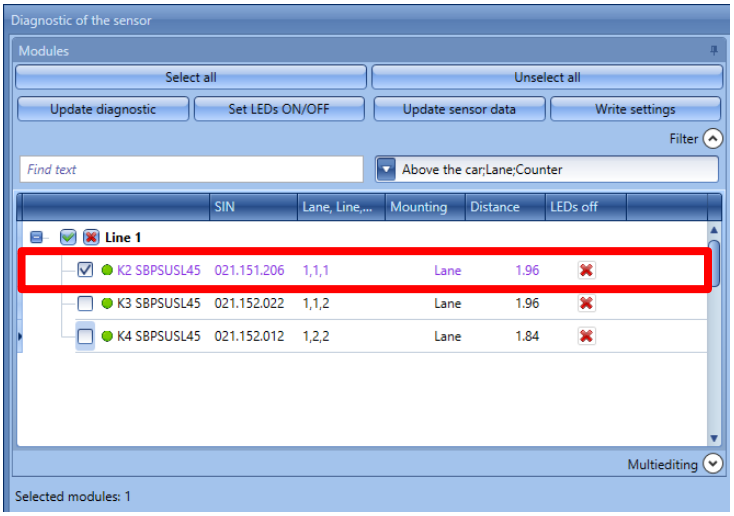
7.1 Step 1: Select the modules

To select the modules:

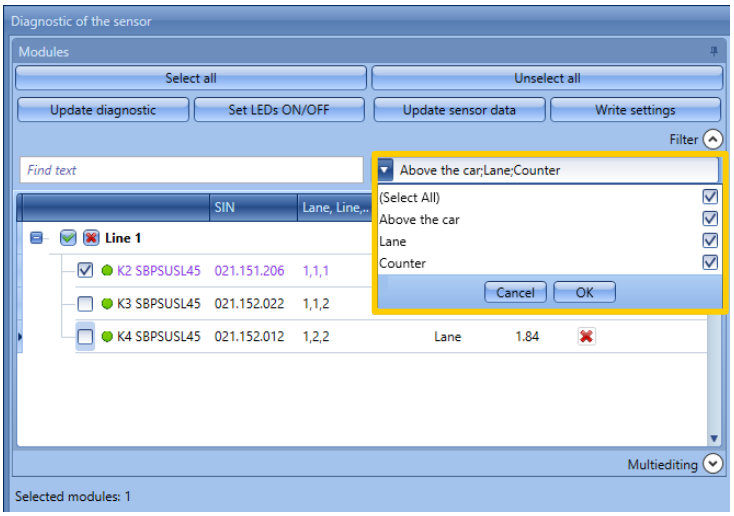
a. Click on *Select all* to calibrate all the sensors



b. Check the sensors one by one



c. Select the sensors according to the type

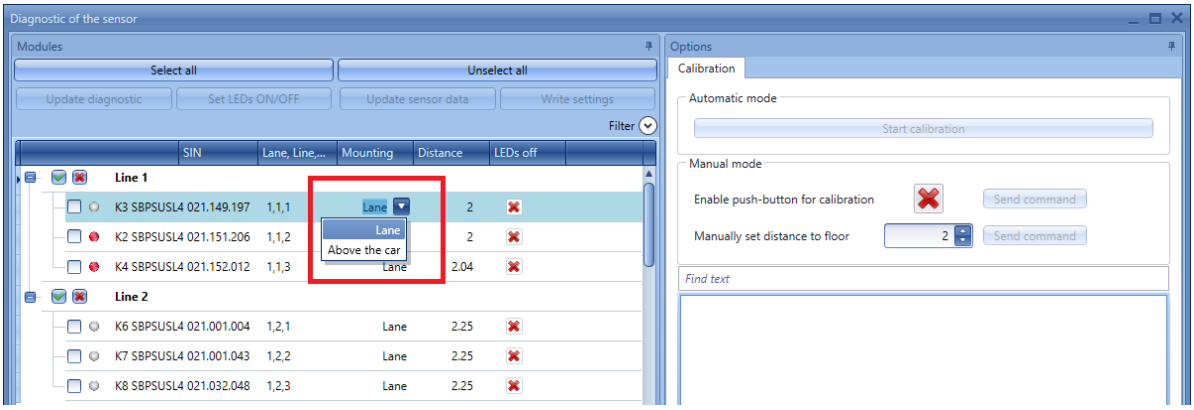


It is strongly suggested to calibrate not more than 20 sensors at the same time.

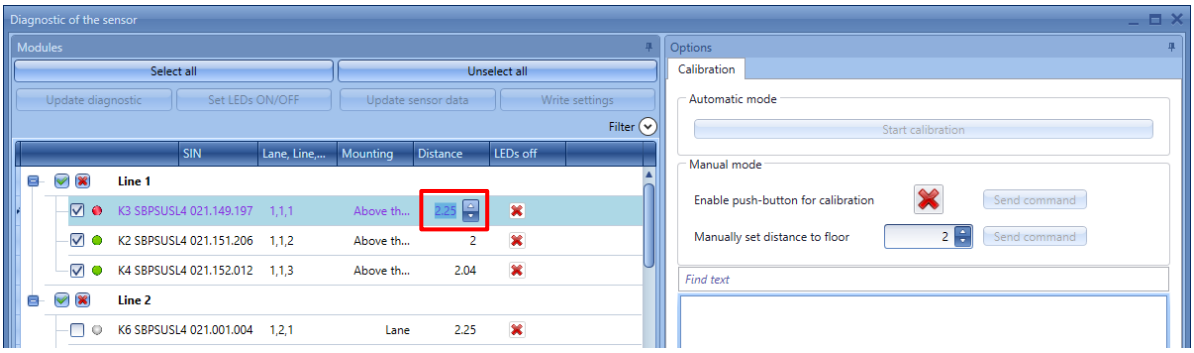
7.2 Step 2: Write the distance from the floor and the type of mounting

7.2.1 How to write different settings to each single sensor

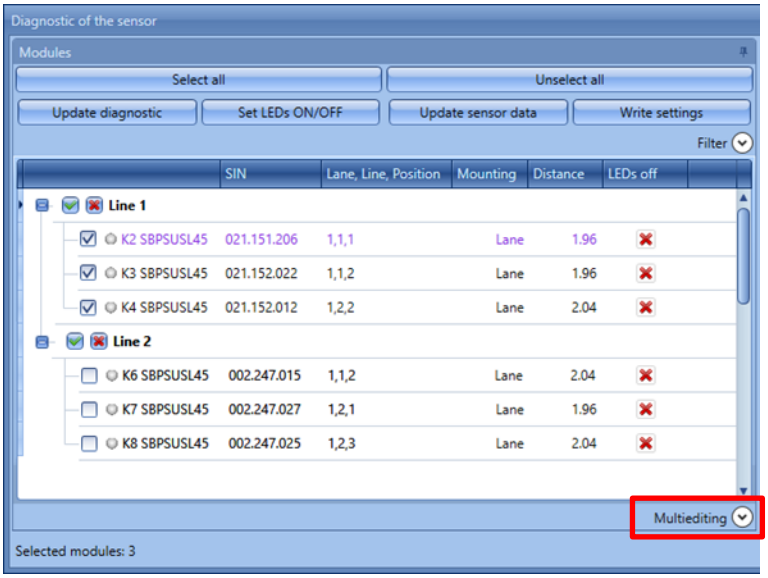
1) Select the type of mounting for each sensor: *Lane* if it is mounted in the lane or *Above the car* if it is mounted above the parking space.



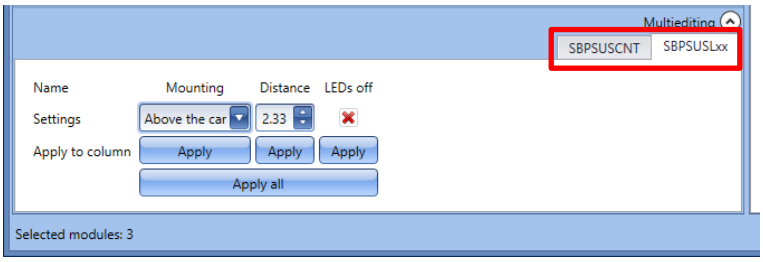
Set the distance from the floor in the column *Distance* for each selected sensor and then click on *Write settings*



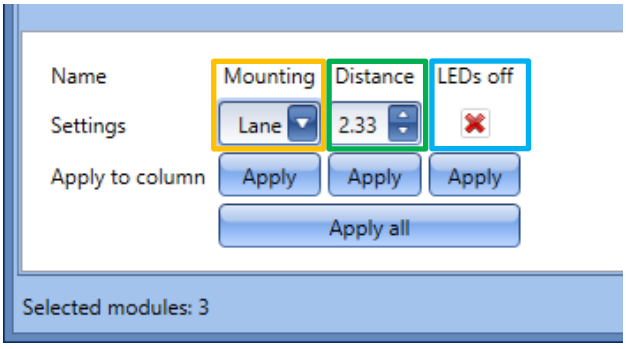
7.2.2 How to write the same distances and mounting ways into the selected sensors



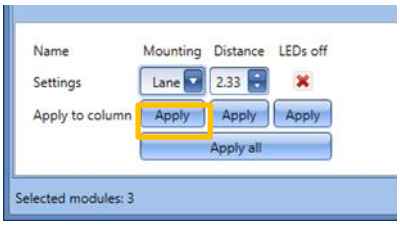
a) Open the *Multiediting* session in the bottom of the window.



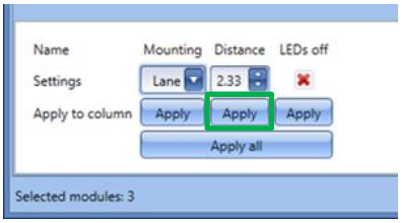
b) Select the part number of the sensors:



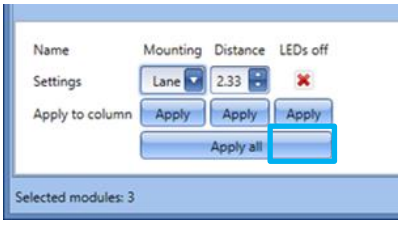
c) Select the type of mounting, distance and the LED status



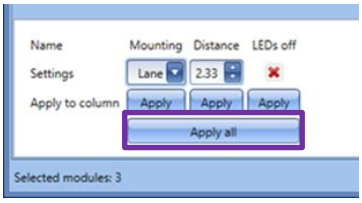
To write only the mounting way to the selected sensors, click on the *Apply* button below the *Mounting* field:



d) To write only the distances to the selected sensors, click on the *Apply* button below the *Distance* field.

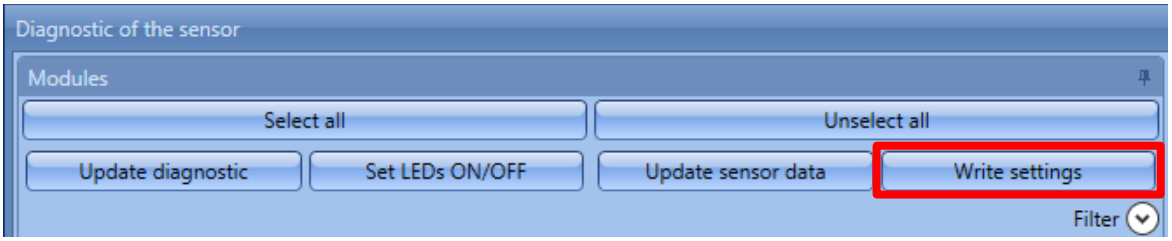


e) To manage the state of the LEDs of the sensor: when the small red cross is selected the LEDs of the sensors are enabled, click the small red cross to disable the LEDs lit;

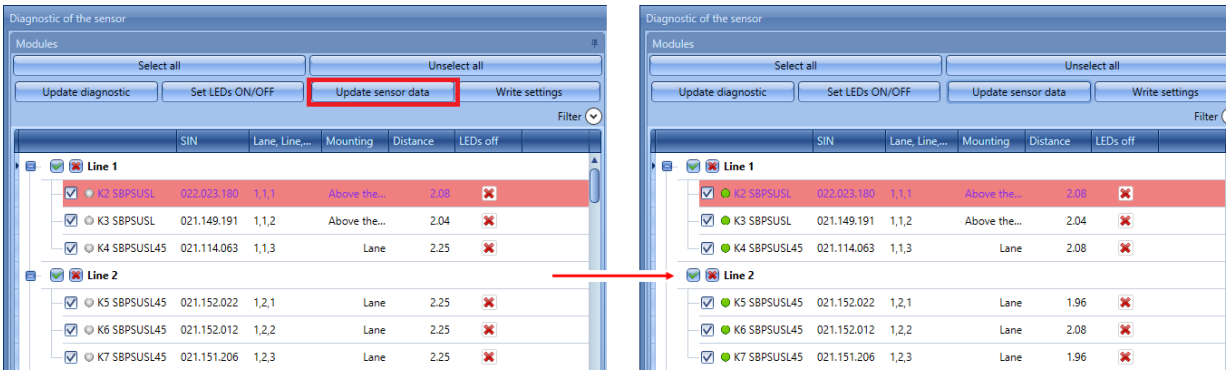


f) To write the type of mounting, the distance and the status of the LEDs of the sensor, click on *Apply to all*

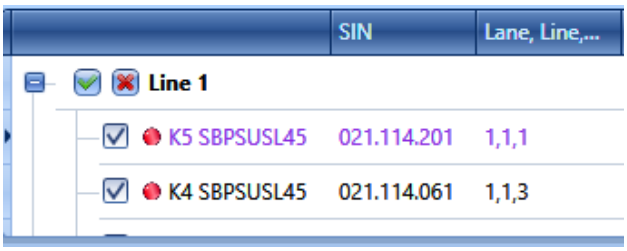
To send the new data, click on *Write settings* button, as shown in the picture below:



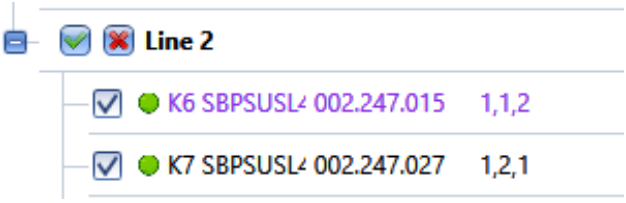
The user can read the configuration of the sensors by clicking on the *Update sensor data* button**:** the system will read the configuration of all the selected sensors, the dot next to the sensor name will turn to green colour, as shown in the example below:



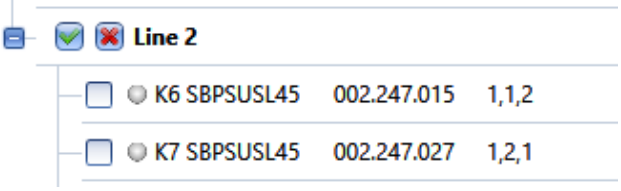
If the settings in the SBP Tool and the one in the sensor are consistent, the dot next to the sensor part number is green, otherwise it is red.



Data need to be synchronized.



Data are synchronized



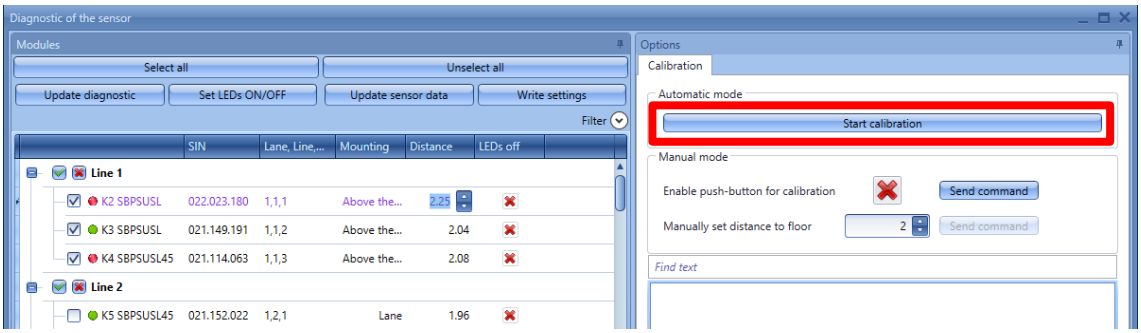
Data are not synchronized

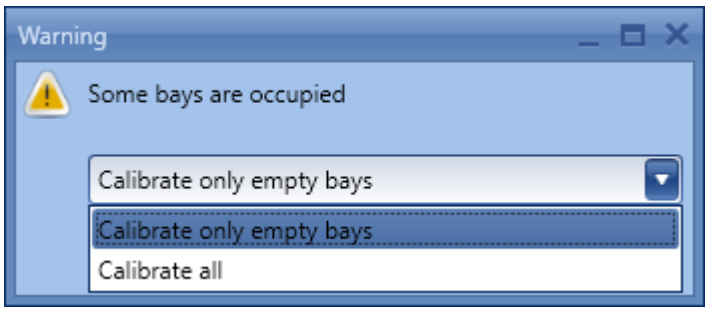
7.3 Step 3: Calibrate the sensors

There are two ways to calibrate the sensors: launching the calibration commands remotely from the tool and locally by means of the push–button on the sensor.

7.3.1 Remote calibration

Once the distance has been written, click on *Start calibration:* a window will appear asking which sensors have to be calibrated.





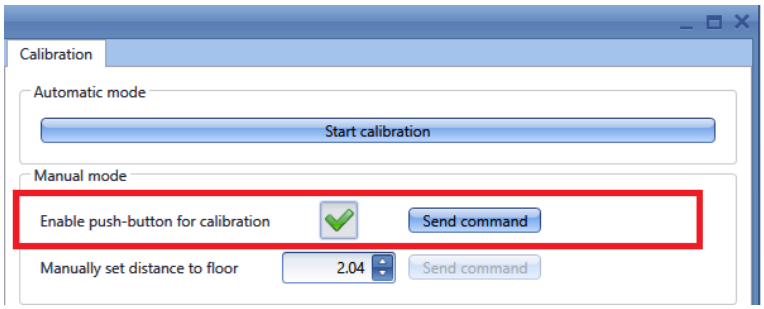
Calibrate only empty bays: if this option is selected, the system will calibrate only the unoccupied spaces

Calibrate all: if this option is selected, the system will calibrate all the sensors

Once the choice has been made, click on *Confirm*: an automatic process will start for the selected sensors, without needing to do anything.

7.3.2 Local calibration

a) Enable the push button on the sensor and click on *Send command.*



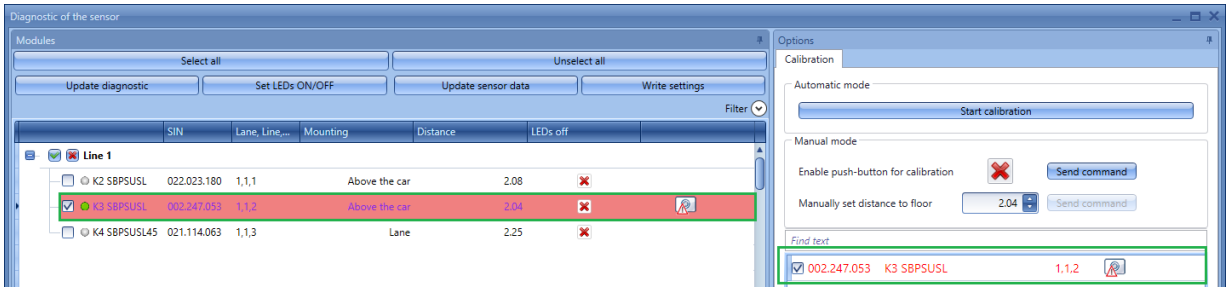
b) Start the calibration by walking from sensor to sensor and pressing the buttons: the yellow LED flashes slowly for 15 seconds to let the installer arrive and then it flashes fast when ready to be calibrated.

c) Once the calibration process is completed, we suggest disabling the push-buttons so that non-authorised people will not use them improperly.

If the settings *Above the car* is selected, the distance should not be set because it is over-written by the calibration process.

7.3.3 Diagnostic signals

To update the status of the sensors, click on *Update diagnostic:* the software will read the diagnostic status of the selected modules.

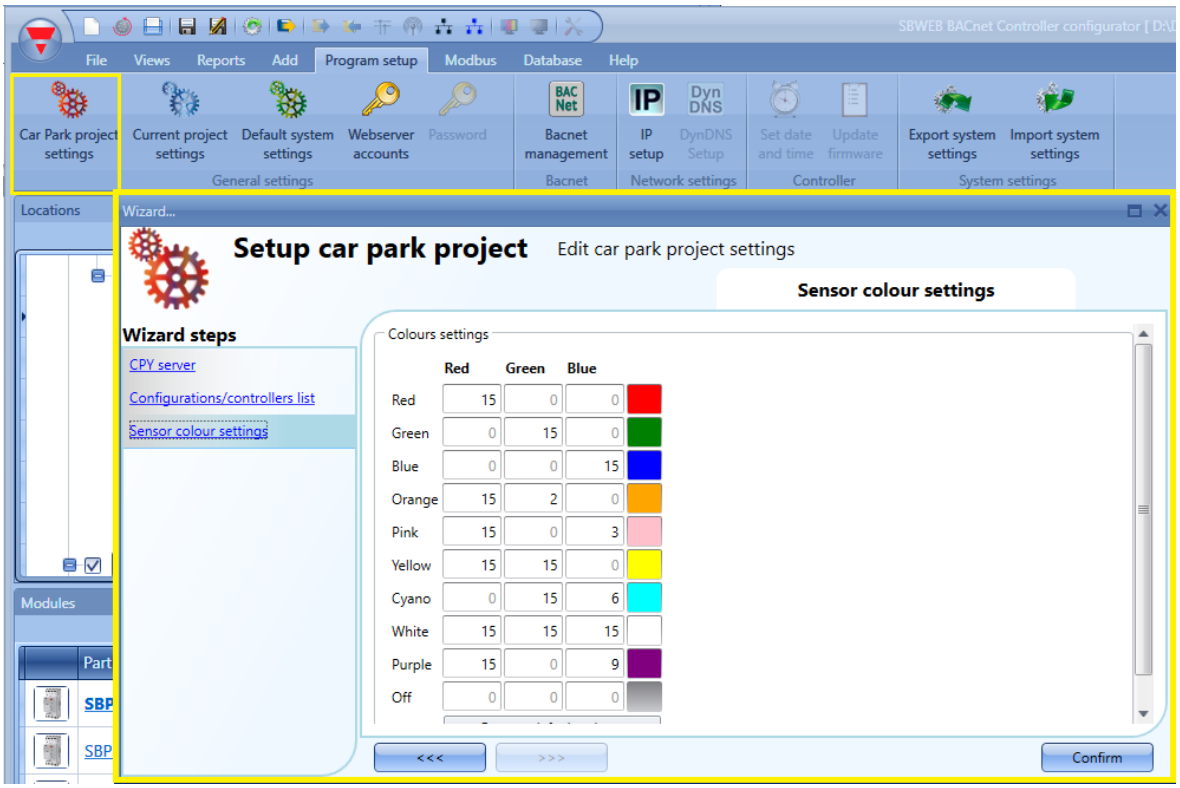


The diagnostic signals that can be detected by the system are the following:

Icon	Tool tip	Description
	Cross talk error	Signal received from other sensors
	Faulty sensor element	Ultrasonic sensor is faulty or covered
	Faulty pushbutton	Pushbutton on the sensor is always active
	Faulty base holder	The base holder of the sensor is damaged
	Dupline voltage drop	The Dupline cable is too long or the current consumption is too high, so there is a voltage drop
	Power voltage drop error	Voltage drop on the Dupline third wire

8 How to define the colours of the LED in the sensor SBPUSLxx

The sensors SBPUSLxx have an RGB LED for which the available colours have to be defined in the menu C*ar park project settings.*
These available colours will then be associated to the different types of categories the sensor indicates, such as vacant, engaged, for VIPs, for expecting mothers, etc. The association is carried out by means of the CPY server (see CPY server manual).



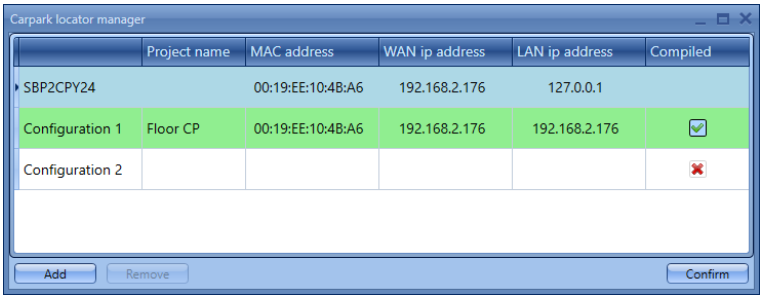
9 How to program the controller SBP2WEB24 and the SBP2CPY24

Once the project is completed, the configuration has to be downloaded into the SBP2WEB24 and into the car park server SBP2CPY24.
Please note that the SBP2CPY24 has to be programmed before the SBP2WEB24.

9.1 SBP2WEB24

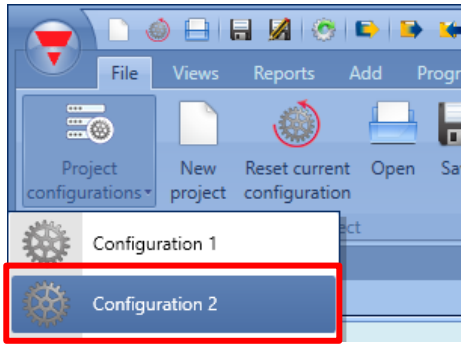
1) In a multi-configuration project, select the configuration to send and connect to the relevant controller.

   a. From the *Carpark locator manager* window the user has to add a new configuration by clicking on the *Add* button: a new configuration line will be added



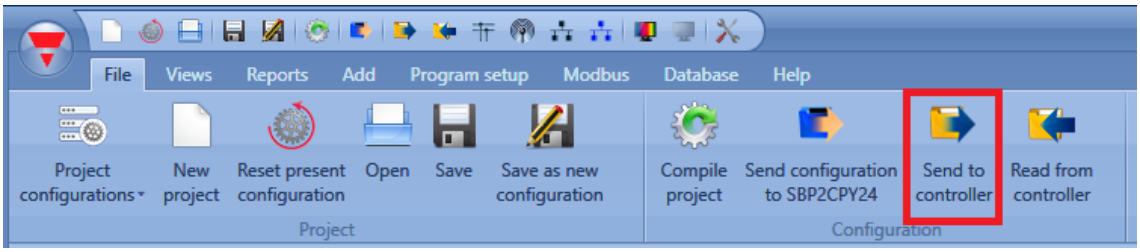
   b. Click on *Confirm* button to save the change.

   c. From the Project configurations select the new configuration that has been added, as shown in the example below the user selects the *Configuration 2*:



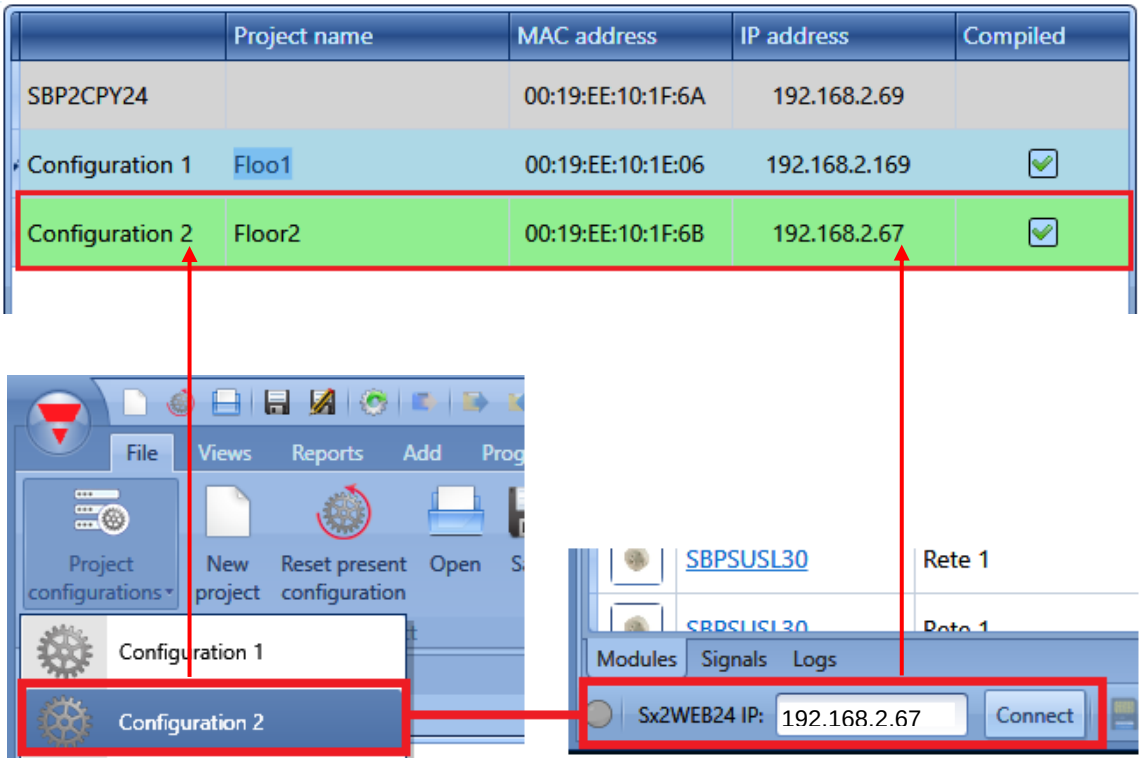
   d. The user has to select the IP address of the controller related to the new configuration.

   e. When the configuration has been done, the user can send the configuration to the controller by clicking on the icon marked in red in the picture below:



2) Repeat the operation from *1.a* to *1.e* for all the controllers/configurations in the project. Should the installation have only one controller, this operation has to be carried out only once.

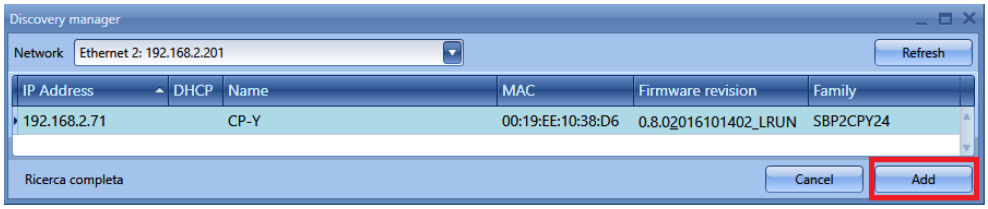
Should the CPY server be into the SBP2WEB24, the address to access it is:
***controller_ipaddress*/CP3App.**



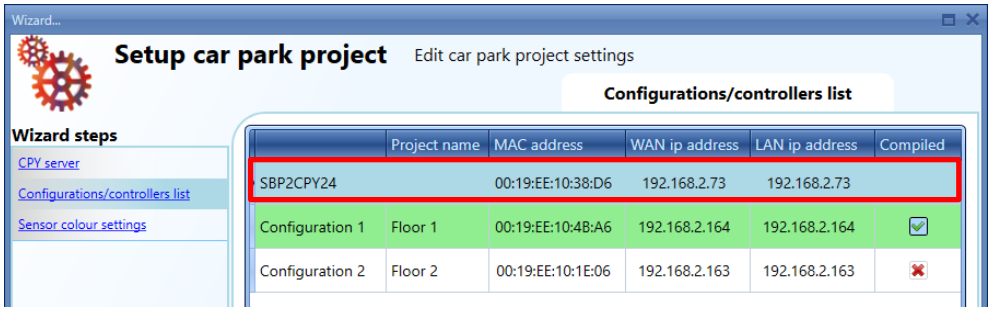
9.2 SBP2CPY24

1) To find the connected SBP2CPY24, click on the icon marked in red to start the discovery:

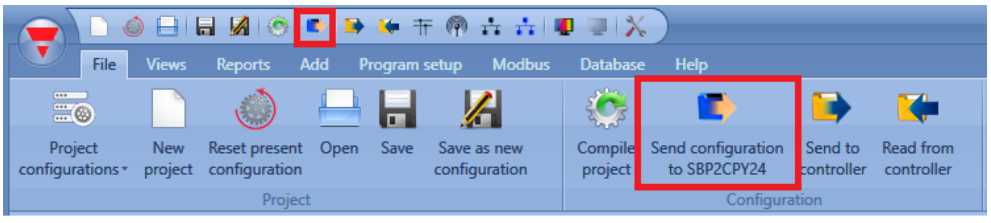
The following window will appear with all the SBP2CPY24s in the network:



Select the one to be added to the Car Park project and click on *Add*: the SBP2CPY24 will be added to the project and it will be listed in the window showing all the devices included in the project.



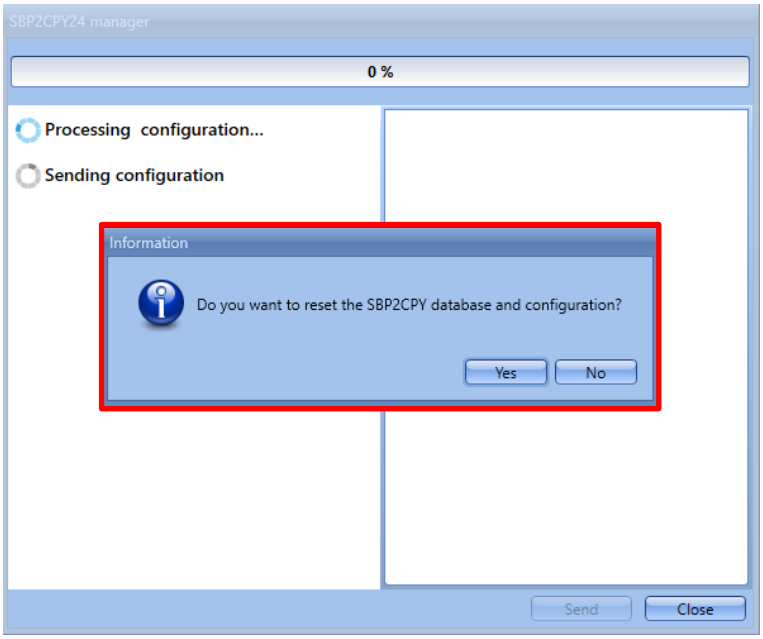
2) Click on the icon marked in red to download the project into the SBP2CPY24



Each time a change is made in the configuration, the user has to click the icon marked in red to send the configuration of the project to the SBP2CPY24.

9.2.1 How to reset the configuration into the CPY server

If a new configuration has to be done, clicking on *Send configuration to SBP2CPY24* button the following windows is shown:



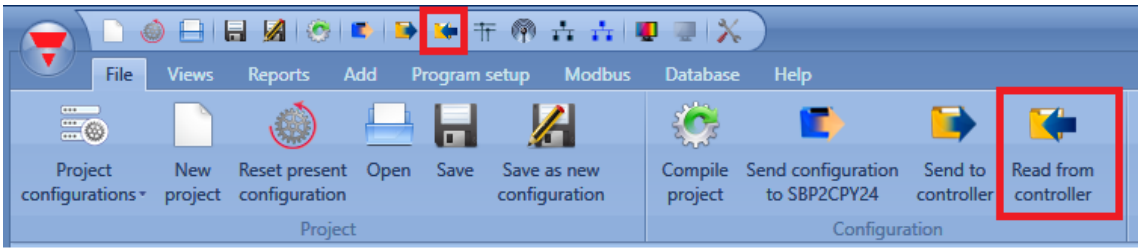
The user has to click on *Yes* to reset the database of the SBP2CPY24 in order to align the CPY project with the Car Park project.

If the user clicks on *No* button, the last valid CPY project is the previous database status will be kept on the SBP2CPY24 but this should be unusable

Note: It is strictly suggested in order to keep aligned the CPY project and Car Park project, to reset the database on the CPY server every time a new configuration is made.

10 How to read the configuration from a controller

To read the configuration from a controller, connect to it and click on the icon marked in red as shown below:

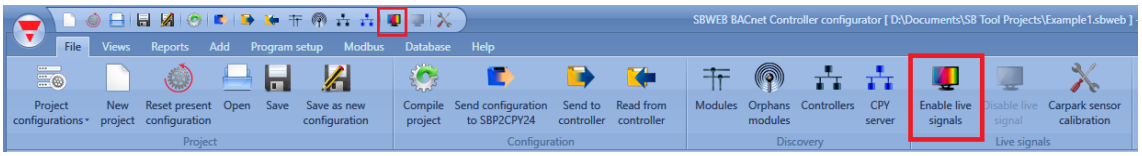


In the case of multi-configuration projects, this operation has to be done only once, connecting the SBP tool to any of the controllers belonging to the installation.

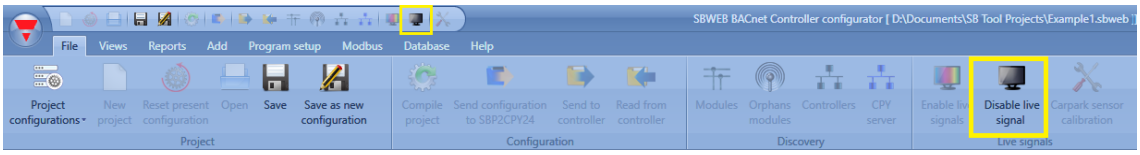
11 Live signals

If *Live signals* is enabled, it is possible to check the status of each sensor (vacant/engaged), to see the status and voltage of the bus and to read the value of each signal.

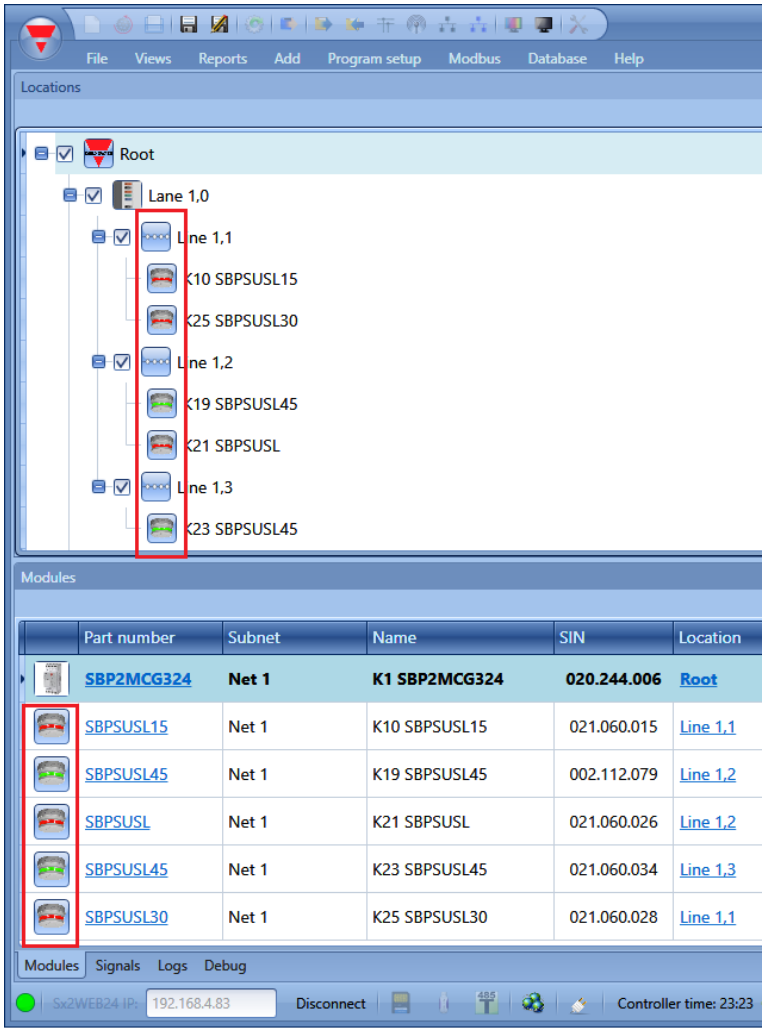
To enable it, click on the icon marked in red:



To disable it, click on the icon marked in yellow:



Once Live signals is enabled, it is possible to see the status (vacant/engaged) of each sensor, as shown below:



If the sensor is shown with the red LED, the parking bay is engaged, while if the LED is green, the parking bay is vacant.

For more information about the *Live signals* feature, please refer to the system manual at this link: http://www.productselection.net/MANUALS/UK/sx_tool_manual.pdf.

12 Zone counter function

The Master Zone Counter (MZC) is a zone count system which has the ability to detect and count cars when they enter and exit zones in the Carpark facility and to send the information to the displays and to the SB2CPY24 server.
The count system consists of a number of count zones, each of which has a certain number of entry and exit points for the cars. These are called detection points (DPOs) and this is where the sensors must be mounted to detect passing cars.

12.1 What is a zone?

A zone is typically a level of the parking facility, but can also be a part of a level or even the entire Carpark. A zone has a certain number of parking spaces available, and the aim of the zone count system is to detect and count the cars entering and leaving the zone, thereby keeping track of the number of available spaces. This means that, once the maximum number of available spaces is set, the Master Zone Counter (MZC) function will deduct from it every time a car enters the zone, and will add to it whenever a car exits from the zone.

12.2 Detection Points (DPOs)

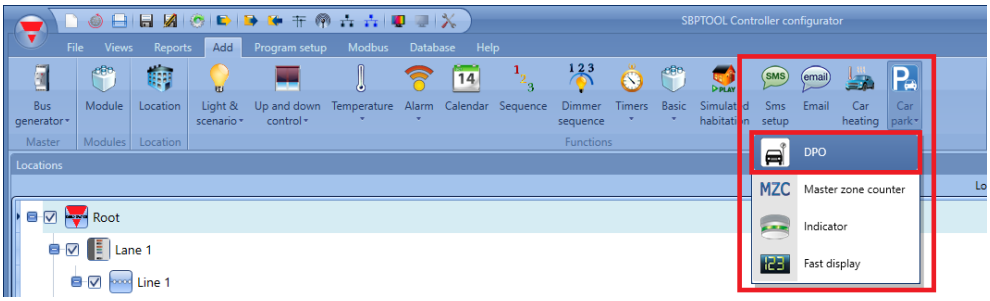
A detection point is a lane or driveway where cars enter or leave a zone. A typical example of a DPO is a ramp between two levels, but it could also be the entry point from the street into the Carpark, or the exit point. In many cases, a detection point is involved in two zones. For example, a DPO which is an exit point for level 2 could at the same time be an entry point for level 3.
Each detection point needs sensors to detect the passing cars. Dupline® SBPSUSCNT sensors are usually used, but other types of sensor, such as standard photo-electric or loop detectors, can also be used. This is carried out by connecting the sensor output to a Dupline® input module.
The zone counting gives the option of using either one or two sensors in each DPO. Two sensors with a distance of 2-3 m between them are recommended because this offers the possibility of detecting the direction of the car and allows more efficient filtering to avoid any false detection. Sometimes cars drive in the wrong direction in a one-way lane, and in a two-sensor solution the MZC is able to manage this so that the count will still be correct. In two-directional lanes it is mandatory to use two sensors. When configuring detection points, a timeout value has to be defined: it allows valid car detection as long as the delay period is less than the timeout value from the point where sensor 1 becomes inactive until the point where sensor 2 becomes active. With a typical distance of 2-3 m between the sensors, 1 sec is the recommended value. Too high a value increases the risk of detection faults. Single sensor DPOs are mainly used when it is not possible or difficult to use two sensors, for example in an outdoor installation with loop detectors.

12.3 Initialization and adjustment

In the initial configuration, the installer has to define the number of spaces in each zone. The actual number of available spaces in each zone at the initial stage must also be defined. From that point, the Master Zone Counter function will increase or decrease the zone count values as the cars enter or leave the zones through the relevant detection points. Since any count system runs the risk of accumulating detection faults, it is important to have a manual count adjustment facility that can be used from time to time whenever required. In the Dupline® Carpark count system, this manual adjustment is carried out via the SBP2CPY230 webserver which can be accessed from a pc or a laptop or via the SBP Tool by means of the *Live signals*.

12.4 Detection points (DPO) function

For each entry/exit point, a DPO function has to be created. Select the DPO function from the *Add* menu.

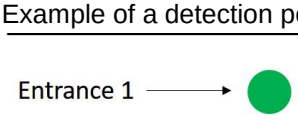


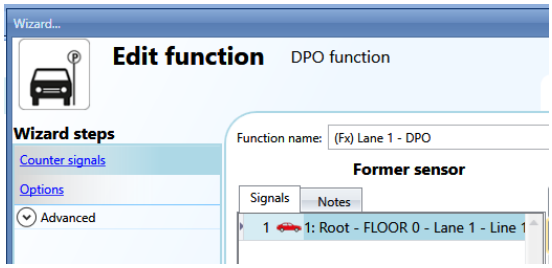
The following window will appear:

12.4.1 Entrance/exit with no direction detection

In this situation, one or more sensors have to be added in the field *Counter signals/ Former sensor*. The DPO function counts without sign and can be used to monitor one direction only (Entrance or Exit). The counter value is increased every time the sensor is activated.
Up to 10 sensors can be added.

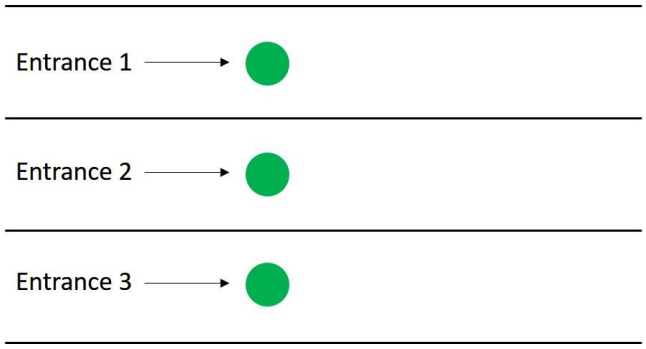
Example of a detection point with one sensor:



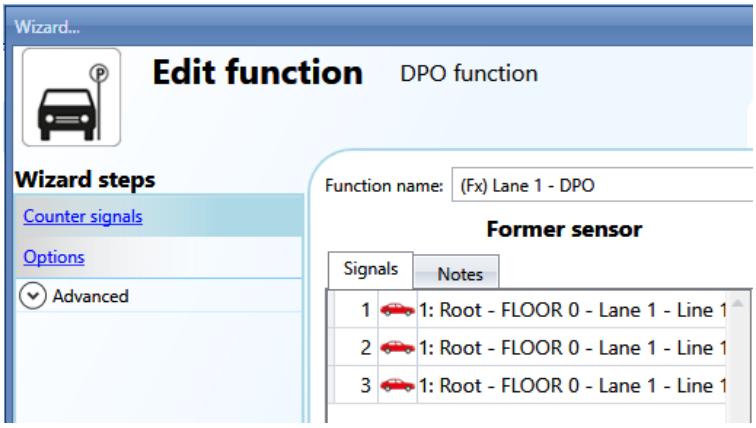


Only one sensor has to be added in the *Former sensor* field:

Example of a detection point with many sensors (one large entrance):



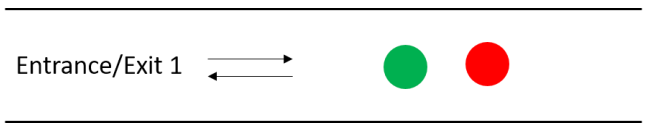
Three sensors have to be added in the *Former sensor* field.



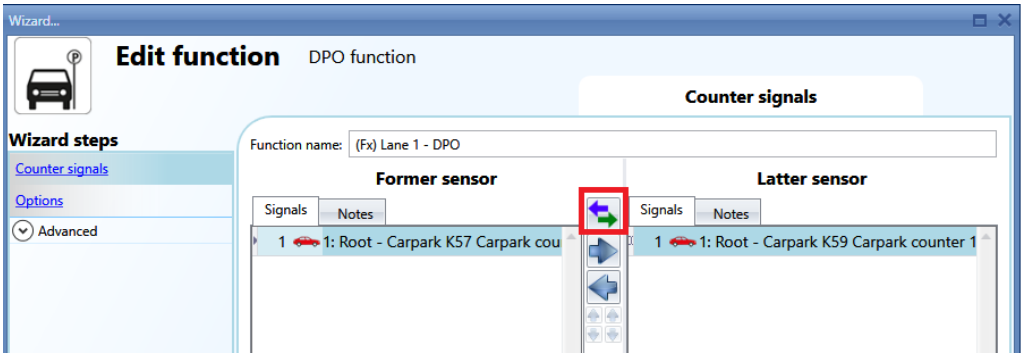
12.4.2 Entrance/exit with direction detection

In this situation two sensors are used to detect the direction of the car: the first sensor to be activated has to be added in the *Former sensor* field, while the second sensor has to be added in the *Latter sensor* field. When the first sensor is activated first, the counter of the function is increased, while when the second sensor is activated first, the counter is decreased.

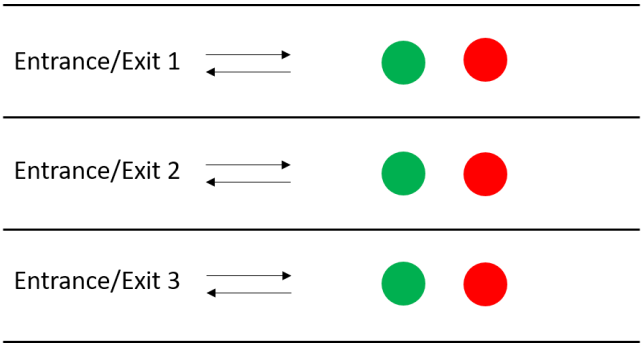
Example of a detection point with one Entrance/Exit:



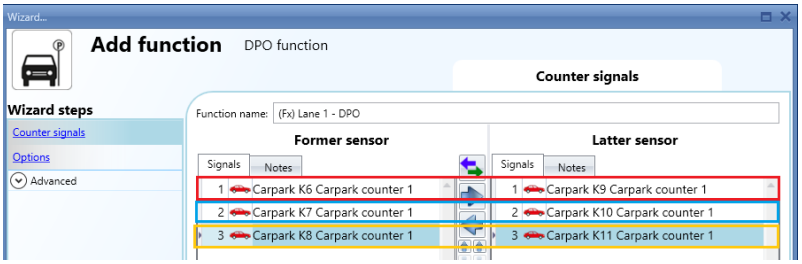
Sensor K57 will be the first to be detected if a car is entering the zone, while K59 will be the second. They can be swapped at any time by clicking on the icon marked in red.



Example of a detection point with many Entrances/Exits (one large entrance):



In this example, six sensors have to be added in the function:



The sensors are coupled according to their position in the list: K6-K9, K7-K10, K8-K11.

The position of the sensors can be changed at any time:

Move the sensor to the right column

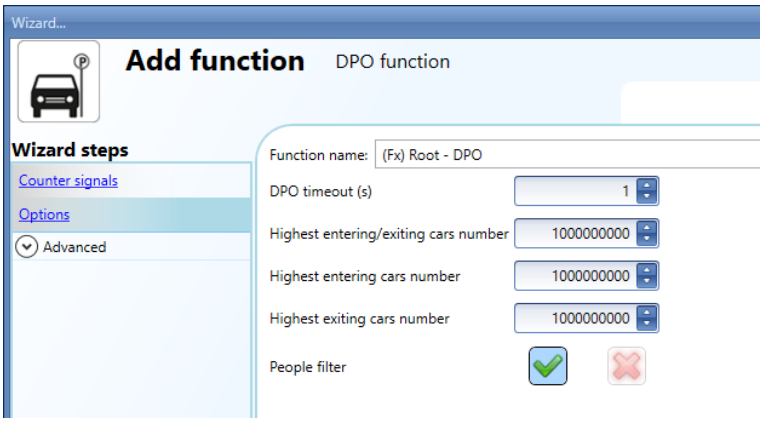
Move the sensor to the left column

Move the sensor up in the list

Move the sensor down in the list

12.4.3 How to set the Options field of the counter in the DPO function

By clicking on the Options field of the DPO function, the following window will appear:



DPO timeout (sec): This is the time in which the second sensor has to be activated after the first sensor. It can be set from 1 to 10 seconds.

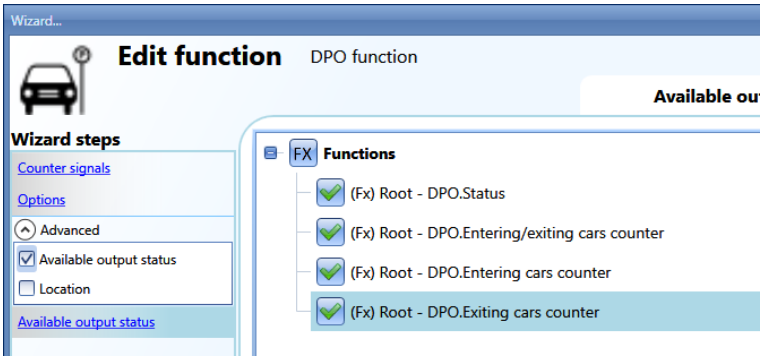
Highest entering/exiting cars number: If the detection point has two sensors to recognize the car's direction, this is the maximum value the counter can reach. This is a number with sign: it is increased if the car is entering the zone (the sensor in the field *Former sensor* is detected first), and decreased if the car is exiting (the sensor in the field *Latter sensor* is detected first). It can be set from -2.000.000.000 to +2.000.000.000.

Highest entering cars number: If the detection point has only one sensor, this is the maximum value the counter can reach. It can be set from 0 to +2.000.000.000. This is increased if the DPO function is linked to an *entry* in the MZC function.

Highest exiting cars number: If the detection point has only one sensor, this is the maximum value the counter can reach. It can be set from 0 to +2.000.000.000. This is increased if the DPO function is linked to an *exit* in the MZC function.

People filter: if the green V icon is selected, the people filter will be enabled in order to avoid the people detection

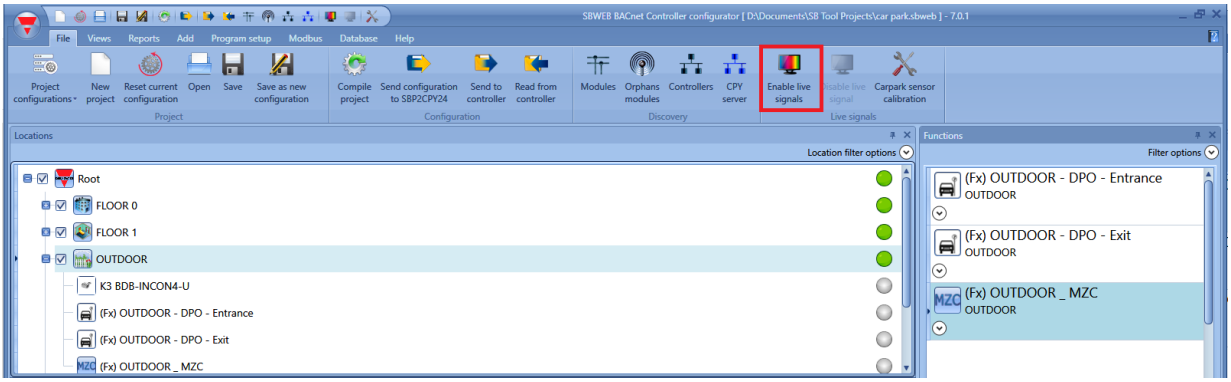
These parameters can also be read and written via Live signals, Modbus and BACnet, by enabling them in the *Advanced* fields of the DPO function.
They cannot be logged in the SBP2WEB database, since this is done by the MZC function.

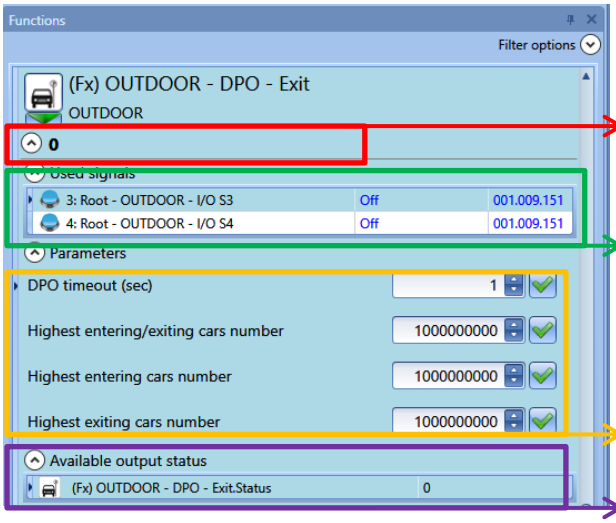


The DPO.status is set to 1 (or -1 if two sensors are used) only for the short time a sensor is activated (the car is passing under it).

12.4.4 Live signals for the DPO function

The counters of the DPO function can be set/reset using live signals. To enable this feature, click on the icon marked in red, after having downloaded the configuration with a DPO function into the SBP2WEB24.



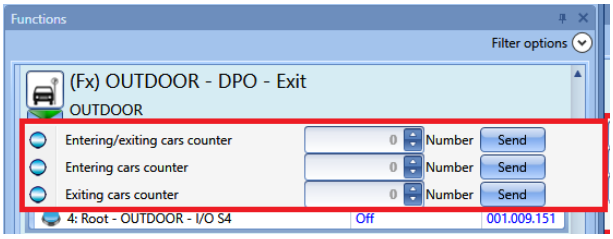


This is the status of the function: it turns to "1" only when the sensors linked to the function are activated by a car (i.e. for a very short time)

The status of the sensors linked to the MZC function are shown: if a car is activating a DPO function, a "1" is shown for the time the car is under the sensor (very short time, less than 1 second).

The parameters of the function can also be changed in live signals without needing to write the configuration again.

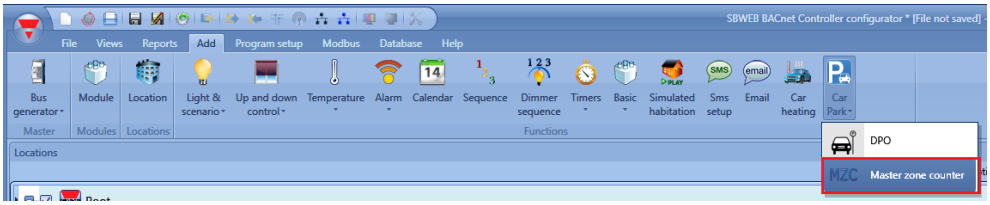
The three counters and the status are shown here for debug purposes.

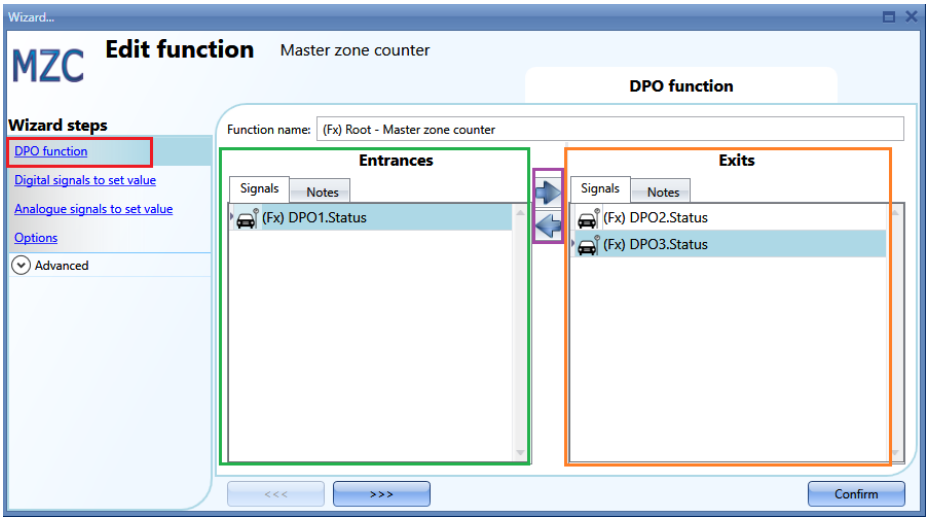


The counters of the functions can be set or reset in live signals by means of the menu marked in red that appears by clicking on the green arrow.

12.5 MZC function

The Master Zone Counter function counts down the available spaces in a zone.
The counter decreases in number when the sensors of a DPO set as an entry detects a car, while it increases when the sensors of a DPO set as an exit are activated by a car.



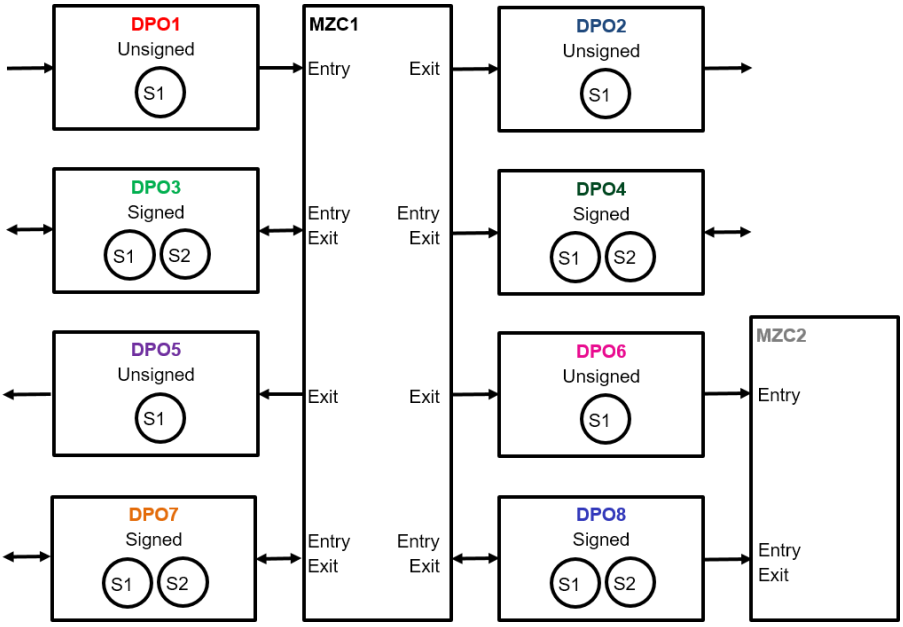


The first things to configure are the *Entrances* and the *Exits* of the zone the function is counting.
In the *Entrances* field (marked in green), the DPO functions that represent the entry for the zone have to be added by double clicking the *Signals* area.
In the Exits field (marked in orange), the DPO functions that represent the exit for the zone have to be added by double clicking the *Signals* area.

The following rules have to be followed:

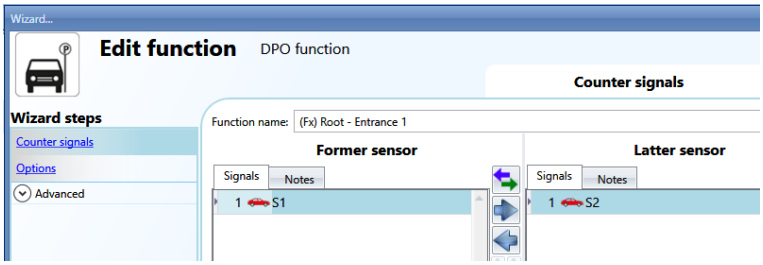
1) Each MZC function can manage up to 50 DPO functions in each direction
2) Any DPO added into the *Entrance* field will decrement the counter (the number of available bays decreases if cars enter the zone)
3) Any DPO added into the *Exit* field will increment the counter (the number of available bays increases if cars exit the zone)
4) Two MZC functions can share the same DPO function, but under the condition that the DPO function must be used as an entrance in the first MZC function and as an exit in the second MZC function.
5) When the position of a DPO function is changed, for example from entrance to exit (using the arrows marked in purple), if it is used also in a second MZC function, its position is automatically changed from exit to entrance in the second MZC function.

The following diagram shows the different possible combinations:

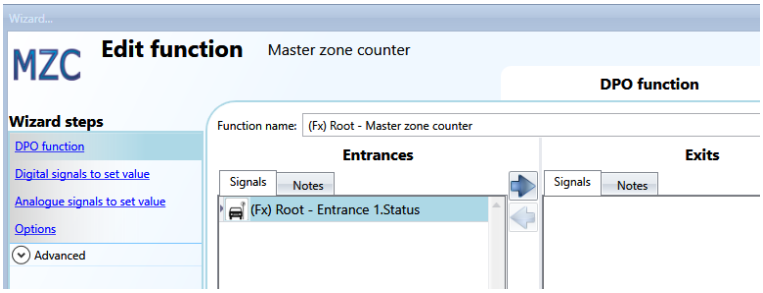


If a DPO function with two sensors is used (S1 as former sensor and S2 as latter sensor), it has to be added only once into a MZC function: see example below.

DPO function with two sensors to recognise the direction of the car:

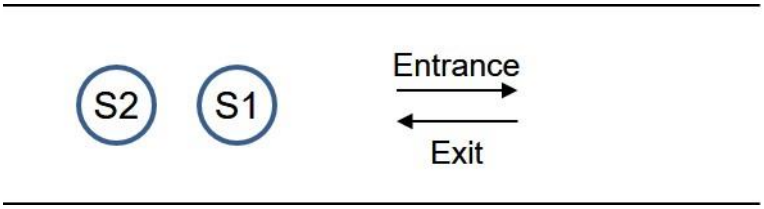


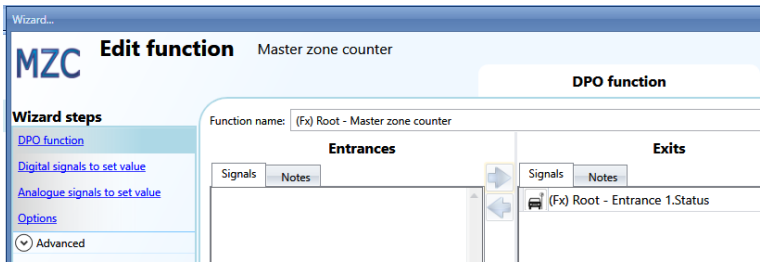
Example 1
When the car is moving from S1 to S2, it is entering the zone:



If the DPO function is added as an *Entrance,* the counter will be affected in this way: when the car is moving from S1 to S2, it is entering the zone and the counter is decreased; if the car is moving from S2 to S1, it is exiting the zone and the counter is increased.

Example 2
When the car is moving from S1 to S2, it is exiting the zone:

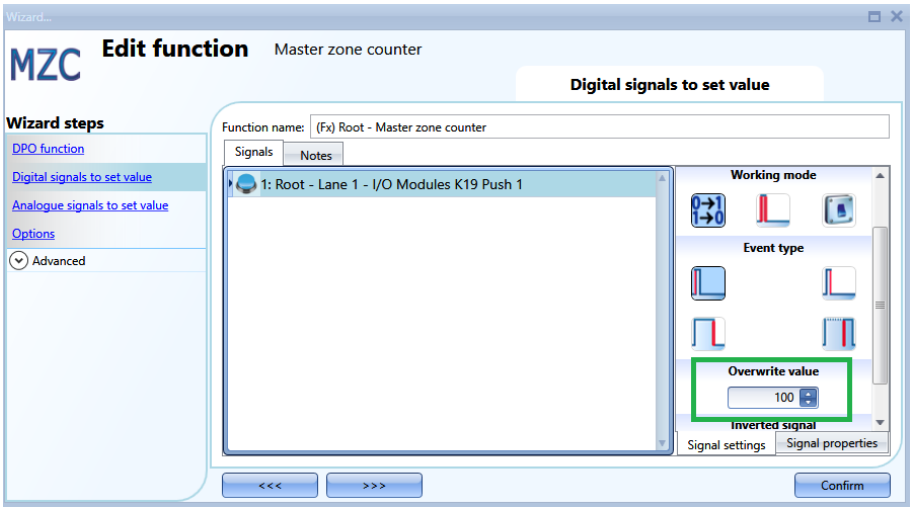




If the function is added as an *Exit,* the counter will be affected in this way: when the car is moving from S1 to S2, it is exiting the zone and the counter is increased; if the car is moving from S2 to S1, it is entering the zone and the counter is decreased.

12.5.1 How to set a predefined value of the counter using signals

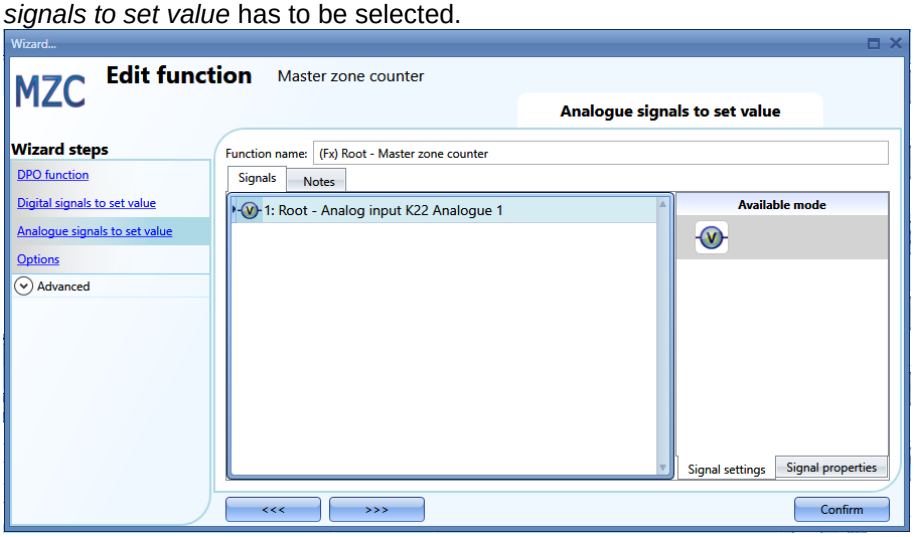
The counter of the available bays can be set to a predefined value using any digital or analogue signals. If the counter has to be set with a digital signal such as a push button, the field *Digital signals to set value* has to be used:



The event type then has to be selected according to the standard rules of the SBP Tool (short/long press, falling/rising edge), and the value (the empty bays) the MZC is to have once the signal is activated must be filled in as shown in the green rectangle.
Every time the number of empty bays is changed, the *Car in transit* and *Surplus of available bays* values

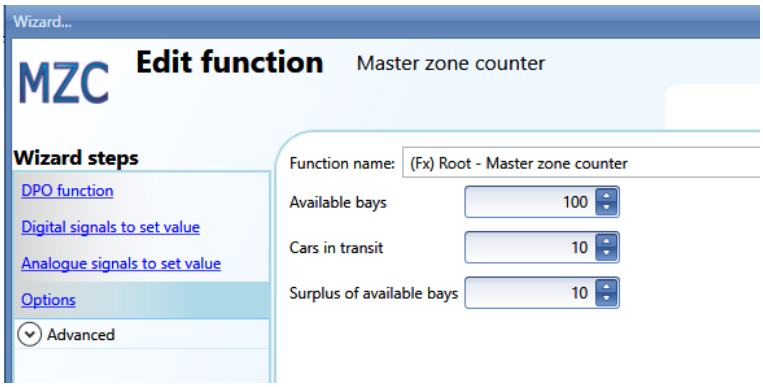
are also reset to the settings in the *Option* field.
The *Overwrite* value can be defined for each signal, and up to 10 signals can be added.

If the user needs to set the MZC function equal to another function or to an analogue signal, *Analogue signals to set value* has to be selected.



When the signal changes, the new value is written in the MZC function (i.e. the available empty bays). Also the *Car in transit* and *Surplus of available bays* values are reset to the settings in the *Option* field. Up to 20 signals can be added.

In the Options field the following parameters have to be set:



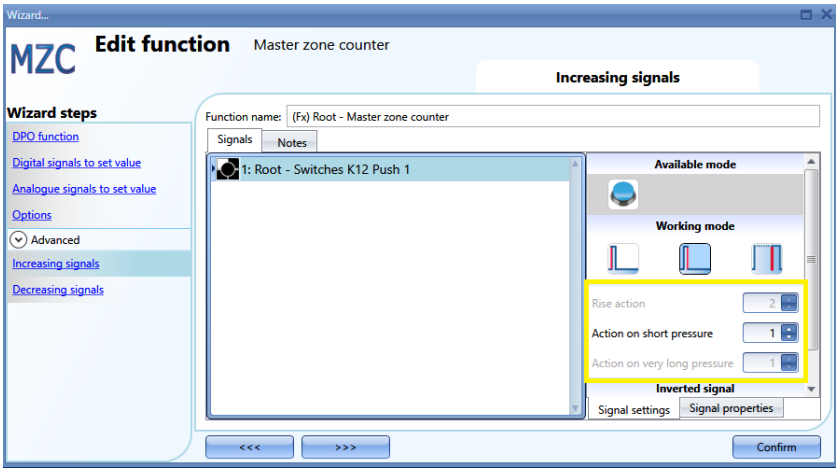
Available bays: this is the value from which the function starts to count down. It is the number of available bays in the zone. The counter is decremented when a car enters the zone until it reaches zero and then it is incremented each time a car exits the zone.

Cars in transit: when all the bays of the zone are occupied, there are cars going around to look for a space. This parameter takes this number into account: it is not mandatory and the user can decide whether or not to use it. The counter relevant to this parameter will be incremented once the counter for the available bays reaches zero. It will be decremented each time a car exits a zone.
Example: Available bays=100, Cars in transit=10
Once the *Available_bays* counter arrives at zero and other cars are entering the zone, the *Cars_in_transit* counter is decremented. It is then incremented when the cars exit the zone. Only when this counter reaches the predefined value (10 in this example), will the *Available_bays* counter be incremented again.

Surplus of available bays: This can be used for debug purposes, to check the number of cars exiting the zone.

12.5.2 How to manually increase/decrease the counter

To increase/decrease the counter of the function, digital signals can be used.
In the *Advanced* section, enable the field *Increasing signals* and/or *Decreasing signals*, and select the signals.

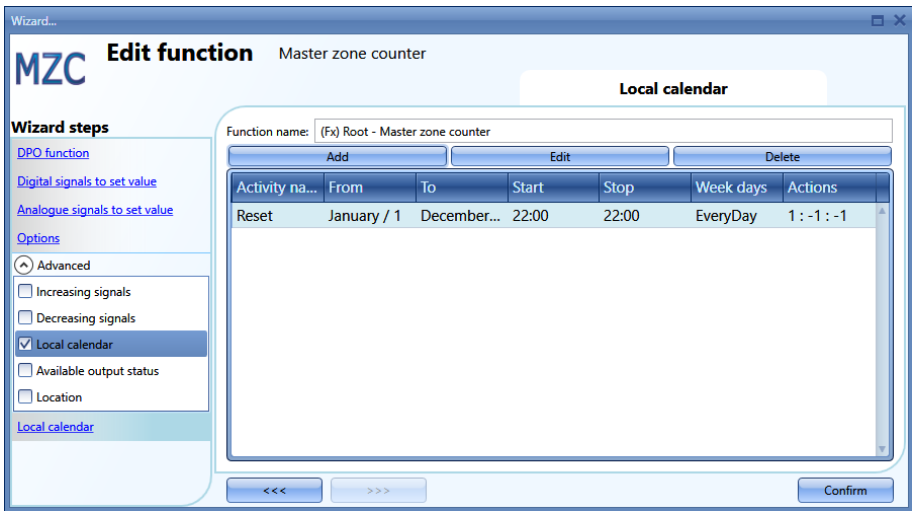


If a pushbutton is chosen, different types of actions are available that will make the counter increase: rising edge (i.e. as soon as the pushbutton is pressed), falling edge (i.e. as soon as the pushbutton is released), very long pressure (i.e. when the pushbutton is kept pressed for the set seconds). For each of these actions, the increasing value can be set (yellow rectangle).

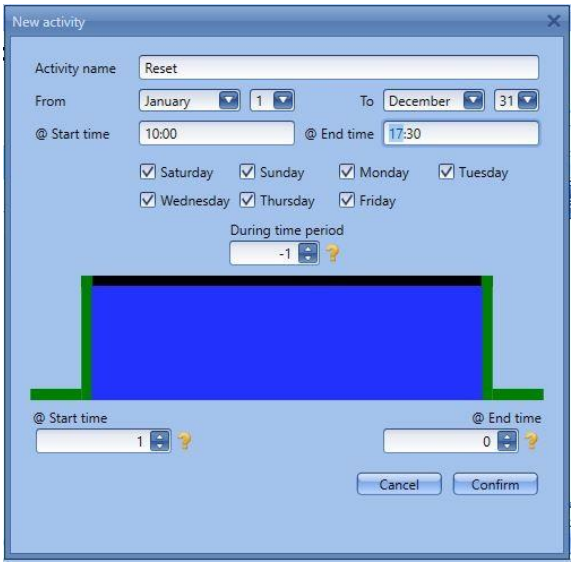
If a level signal is selected, such as a switch or a function, the available actions are rising edge (i.e. when the switch is closed or the function is activated), falling edge (i.e. when the switch is open or the function is deactivated), or both.

12.5.3 How to set the number of available bays with the calendar

The counter value can be changed/reset at a certain time of the day using the calendar. In the *Advanced* field, enable *Local calendar.*



Click on *Add* to open a new activity window:



Activity name: In this field the user defines the name of the event that will appear on the calendar. This is a mandatory field.

From*:* the start date for the calendar activity.

To*:* the end date of the calendar activity.

@ ***Start time****:* The time of the start of the activity.

@ ***Stop time***: The time of the end of the activity.

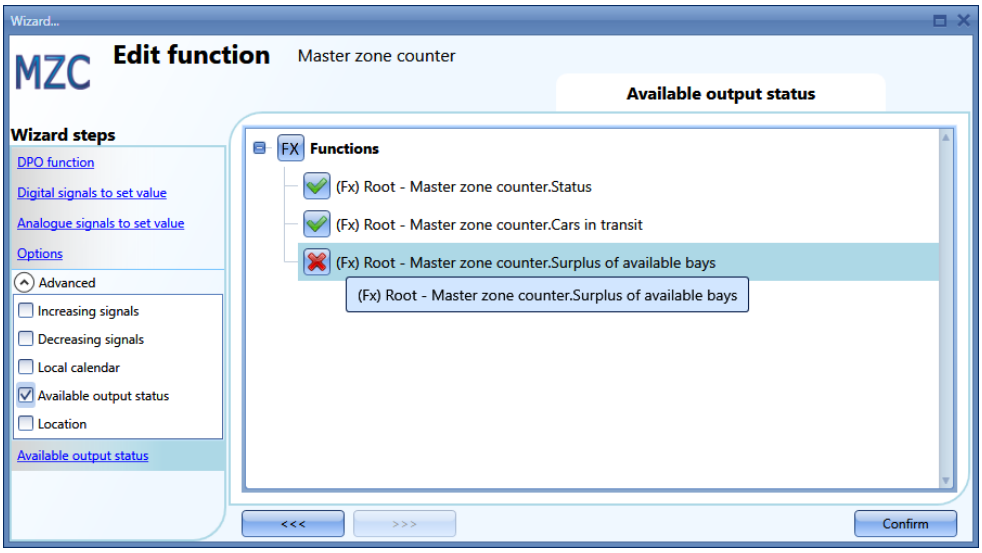
Days: The user should select the days when the calendar activities must be applied.

@ ***Start time***: in this field the user can select the value that will be written in the counter of the function
- No action (-1)
- MZC=0 (0)
- MZC=parameters *Available bays* (1)
- MZC= 2 (2)
- MZC= 3 (3)
- ……
- MZC=10000 (10000)

@ ***End time***: in this field the user can select the value that will be written in the counter of the function
- No action (-1)
- MZC=0 (0)
- MZC=parameters *Available bays* (1)
- MZC= 2 (2)
- MZC= 3 (3)
- ……
- MZC=10000 (10000)

12.5.4 How to remotely access the output status of the MZC function

In the *Advanced* section, enable *Available output status*:



The green V must be present if the different counters have to be used in live signal, Modbus or webserver.

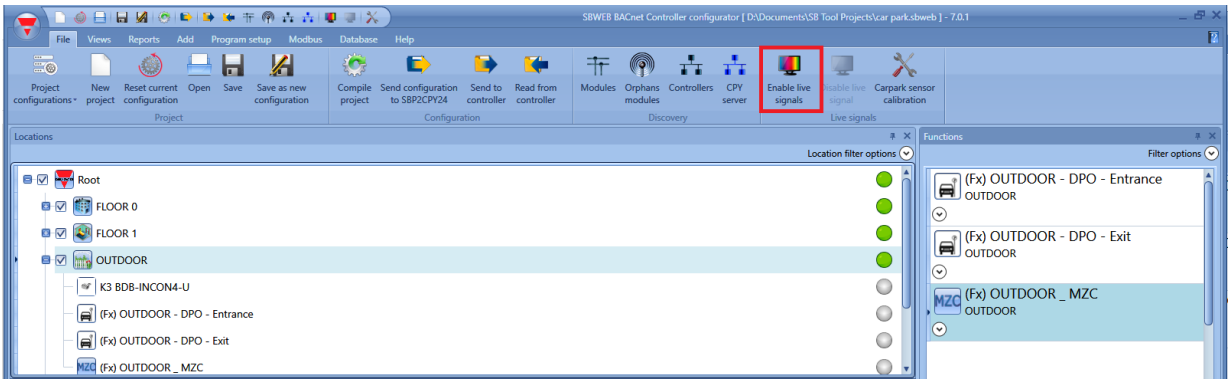
Master zone counter.status: This is the counter of the available bays and it indicates how many bays are empty now

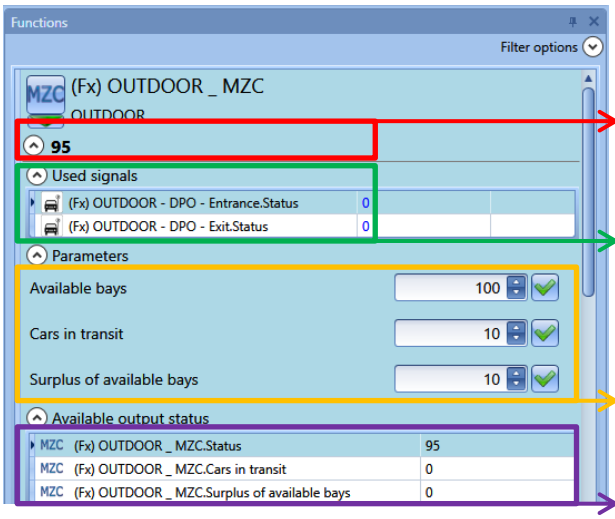
Master zone counter.Cars in transit: This is the counter of the cars in transit and it indicates how many cars are driving through the zone when all the bays are occupied

Master zone counter.Surplus of available bays: This can be used for debug purposes, to check the numbers of cars exiting the zone.

12.5.5 Live signals in the MZC function

The counter of the MZC function can be set/reset using live signals. To enable this feature, click on the icon marked in red, after having downloaded the configuration with a MZC function into the SBP2WEB24.



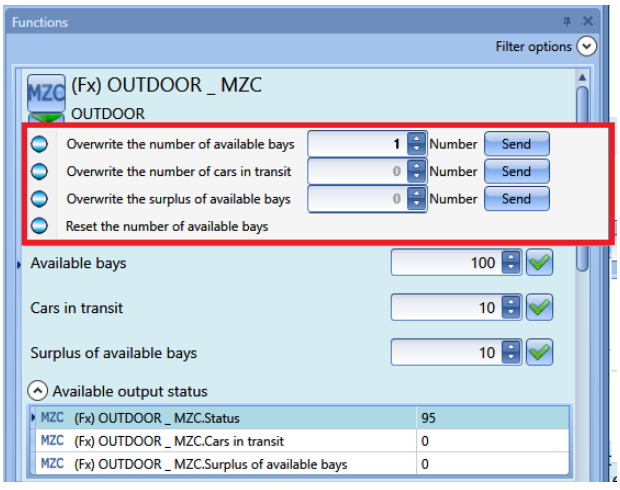


This is the counter that indicates the number of available bays: it is decreased when a car enters the zone and increased when a car leaves the zone.

The status of the DPO linked to the MZC function are shown: if a car is activating a DPO function, a "1" is shown for the time the car is under the sensor (very short time, less than 1 second.

The parameters of the function can also be changed in live signals without needing to write the configuration again.

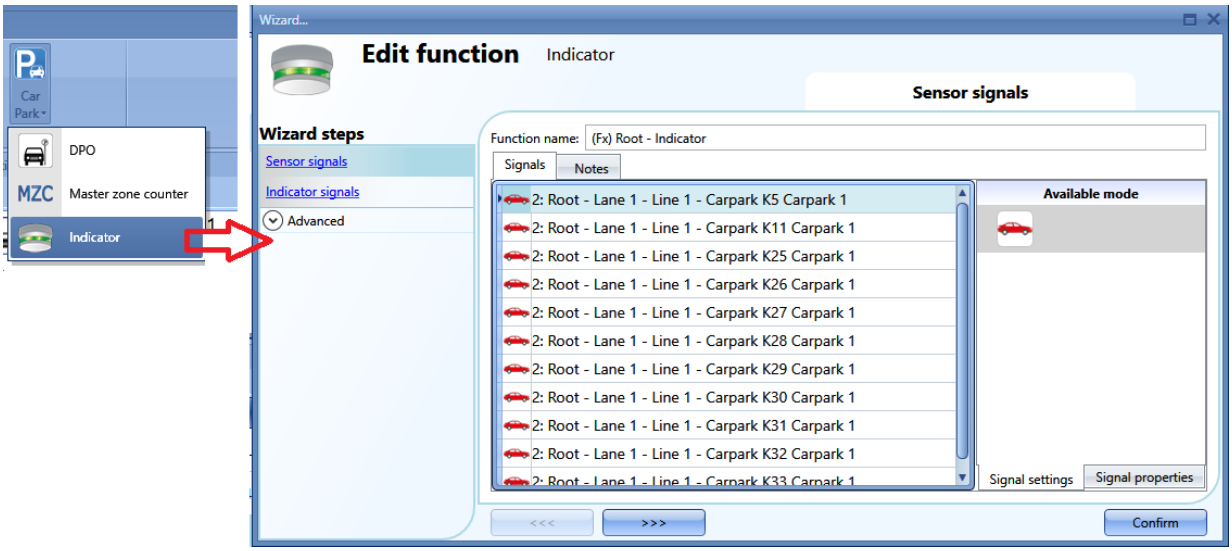
The three counters are shown here for debug purposes.



The counters of the functions can be set or reset in live signals by means of the menu marked in red which appears by clicking on the green arrow.

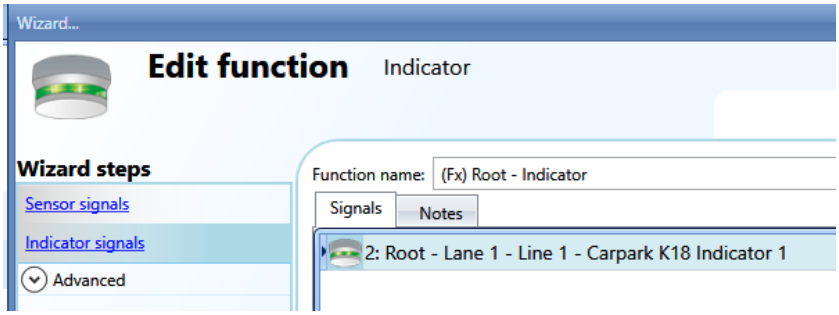
13 Indicator function

Should the user need an indicator to show if there are empty bays in the lane, the indicator function can be used to make the LED of the SBPILED change colour according to the status of the sensors in the lane.

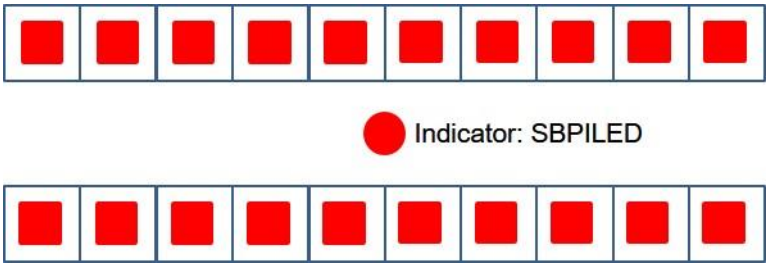


In the *Sensor signals* field, the user has to add the sensors which will give the indication about the lane or line: the system will do an OR of all the sensors added in this field.

In the *Indicator signals* field, the user has to add the indicator modules SBPILED which are mounted in the lane, to give a fast indication about the availability of the bays.

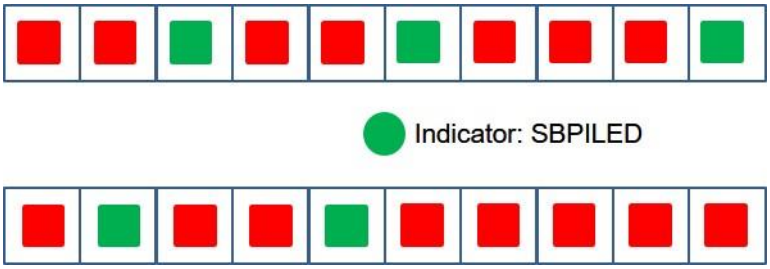


Example 1: lane with all the bays occupied



The red squares are the sensors that have to be linked into the *Sensor signals* field and which detect cars under them. The red circle is the indicator that has to be added into the *Indicator signals* field. Since there are no available spaces, the indicator is showing a red light to advise the driver not to enter the lane, since all the bays are occupied.

Example 2: lane with some available bays

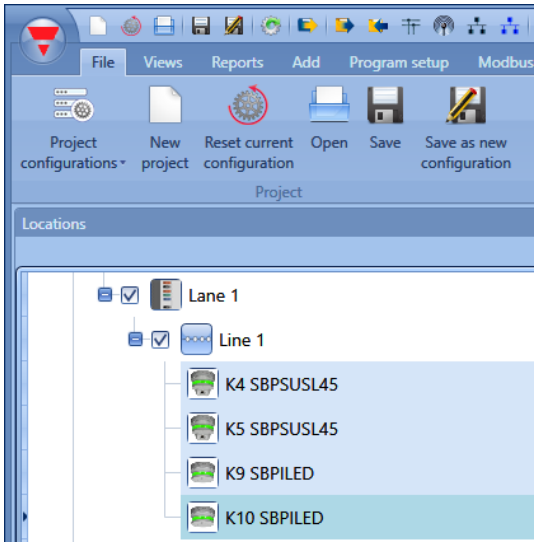


The red and green squares are the sensors that have to be linked into the *Sensor signals* field (red are the occupied ones, green the empty ones). The green circle is the indicator that has to be added into the *Indicator signals* field. Since there are some available spaces, the indicator is showing a green light to advise the driver that there are some empty bays in the lane.

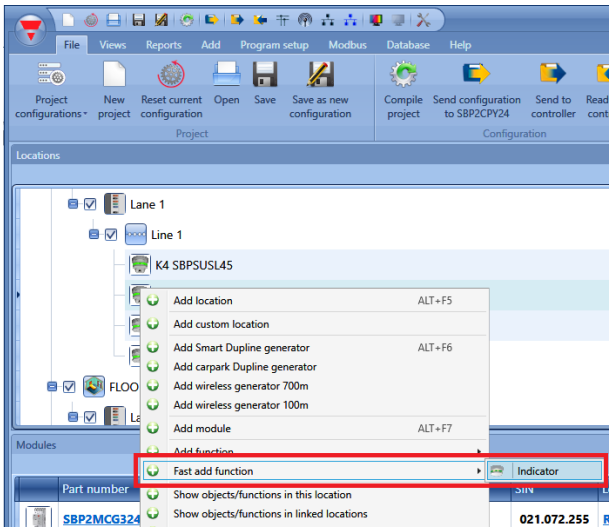
1) The function is automatically generated by placing the indicator modules SBPILED in the *Indicator signals* field, and the SBPUSLxx in the *Sensor signals* field.

13.1 How to add the Indicator function with the fast procedure

This function can also be added in a fast way in the *Location* window by following these steps:

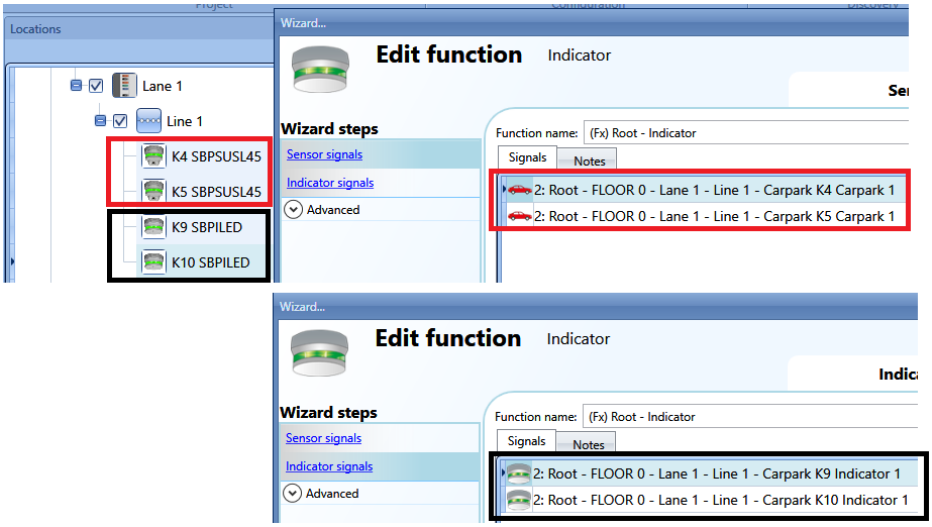


1) Select the indicator modules and the sensors in the location where the indicator function has to be placed: a multiple selection can be carried out by clicking on the module while pressing the *Shift or CTRL key.*
If also the location is selected, the function will be placed here, otherwise it will be placed in the location *Root*.



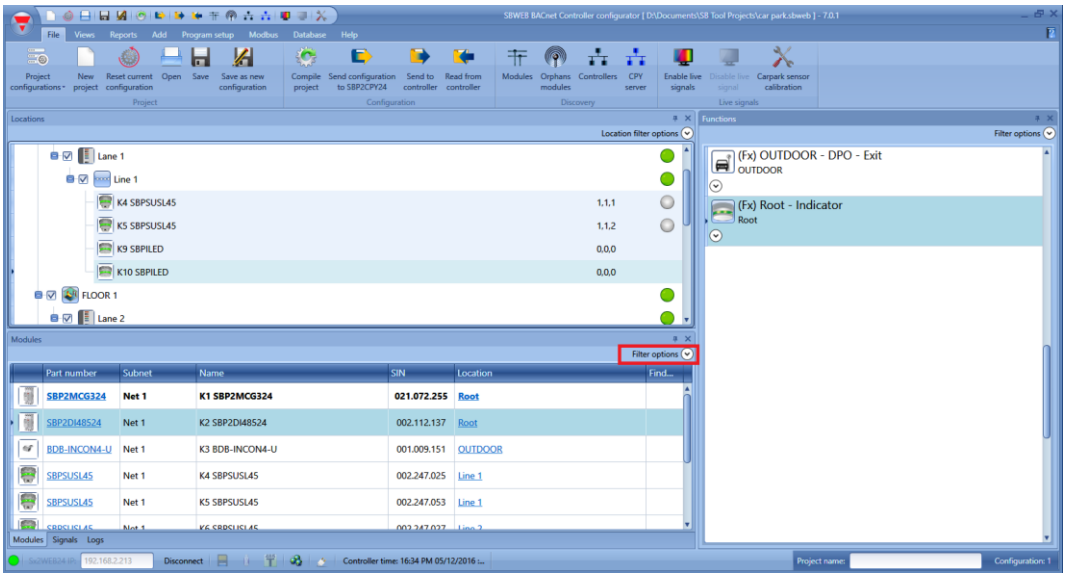
2) Right click on the selected modules and select *Indicator* from the *Fast add function* menu.

3) The function will be automatically generated by placing the indicator modules SBPILED in the *Indicator signals* field, and the SBPUSLxx in the *Sensor signals* field.



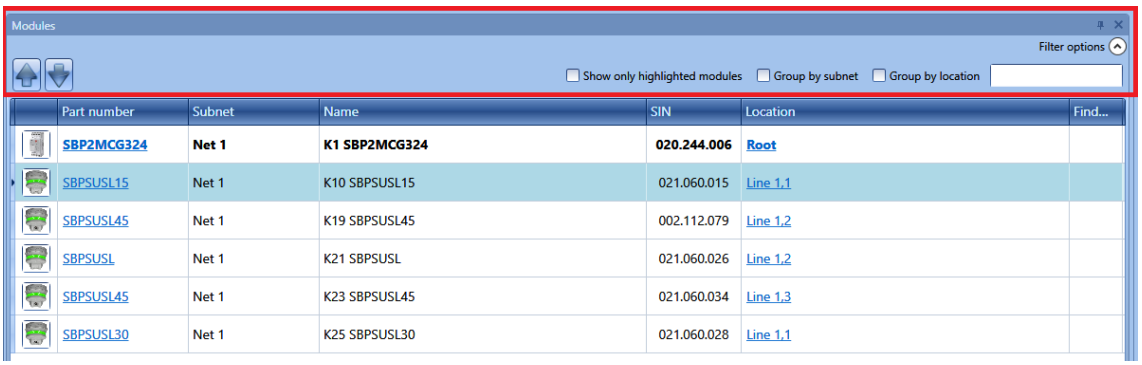
14 Modules

To configure a module, the user has to click on the picture of the module in the *Modules* window of the SBP Tool (see below):



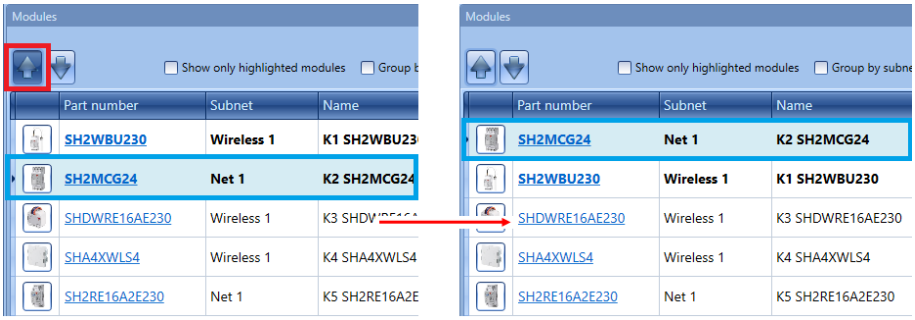
14.1 How to manage the filters on the Modules window

The modules can be sorted or filtered by the *Filter options*: by clicking on the icon, the *Filter options* panel will be opened. The available filters are shown in the top of the *Modules* window, as shown in the red rectangle:

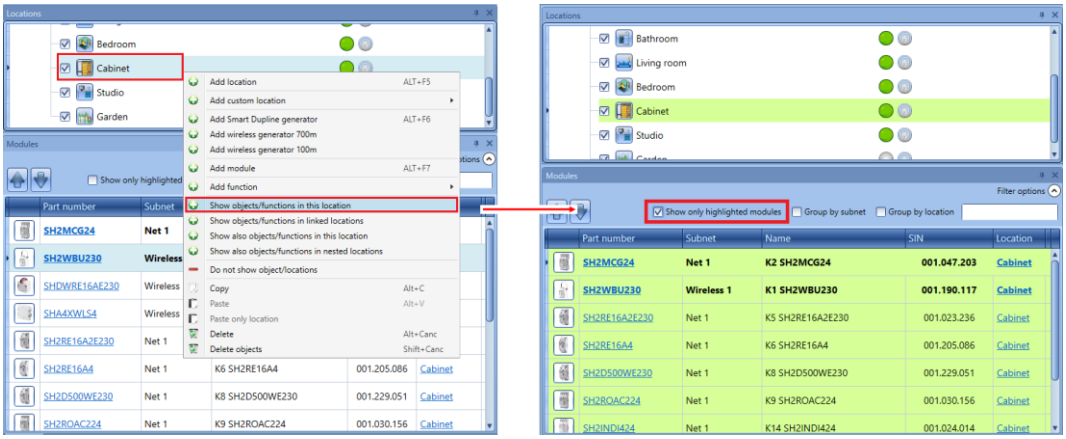


Only the modules specified by the filter will be presented: the filters can be used individually or they can be mixed.

Up/Down arrows:
Once a module is selected, by clicking on the arrows it can be moved up or down.

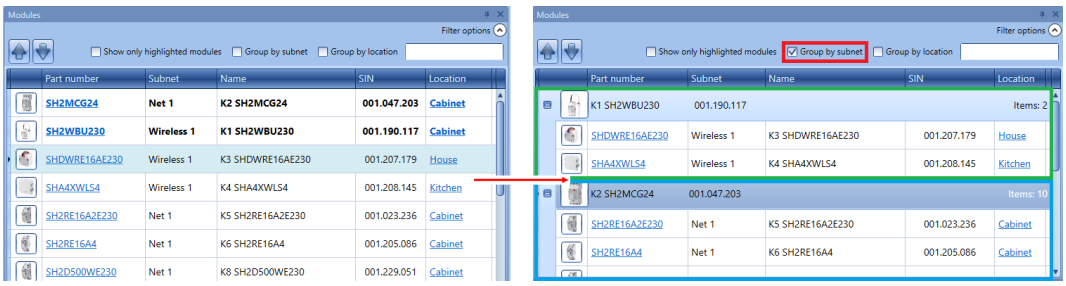


Show only highlighted modules:
When this filter option is checked, only the modules that are highlighted (according to the *highlight* functionality available on the *Locations/Functions* window) are presented in the *Modules* window.
In the picture below an example is shown:

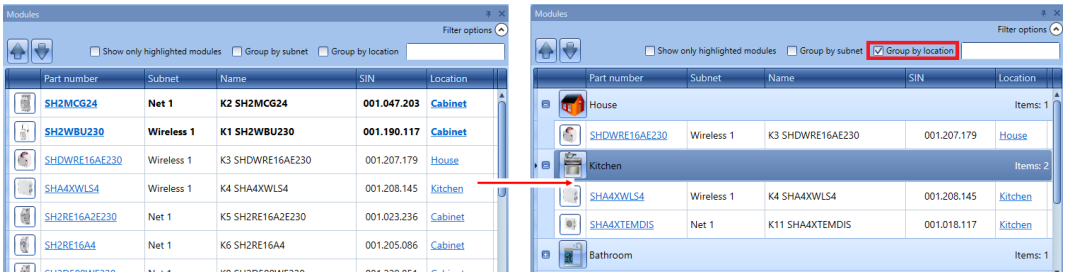


Group by subnet:
The modules are grouped by:

- Master generator (each network generated by SH2MCG24, SH2DUG24 and SH2WBU230N);
- Modbus serial communication port (COM1MASTER or COM2MASTER);
- TCP/IP Modbus connection



Group by location:
The modules are grouped by the locations defined in the project.

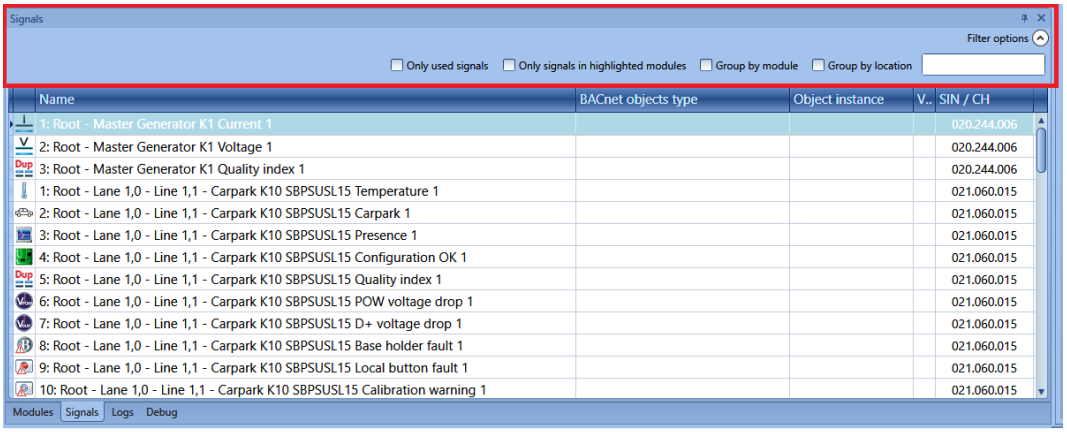


Search box:
The results will include all the words that contain the input string and the search will be carried out on the modules available in the project. The user can insert the full name or part of it.

14.2 How to manage the filters in the Signals window

The signals can be sorted or filtered by *Filter options*: by clicking on the icon, the filter options panel will be opened: only the signals specified by the filters will be displayed.

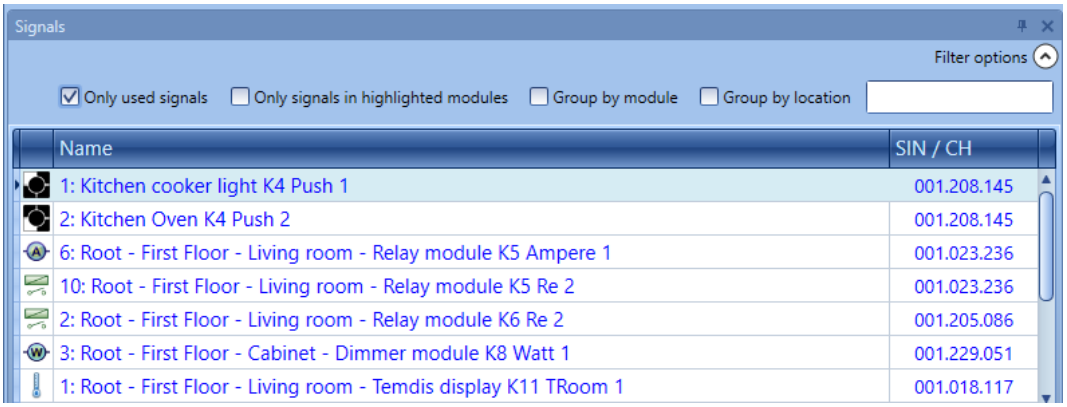
The available filters are shown in the top of the *Signals* window: see the red rectangle below:



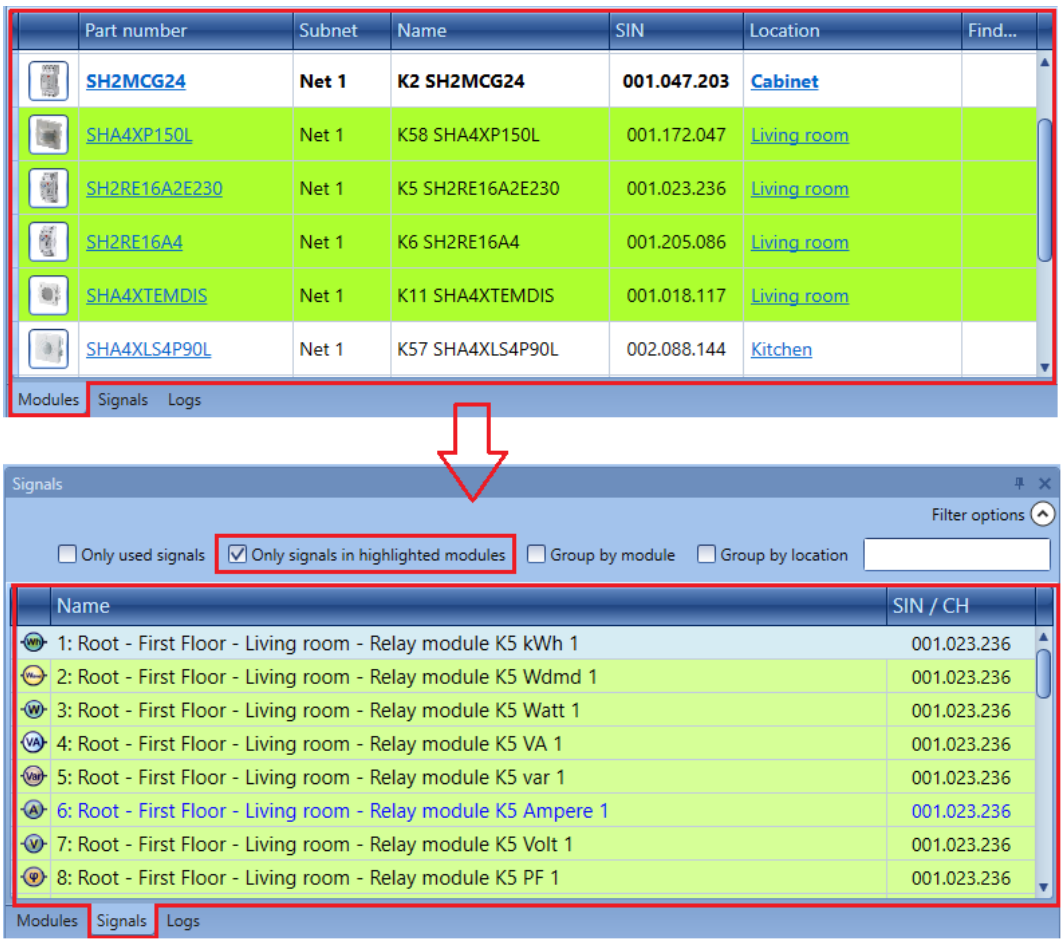
When the option panel is shown, the following filters are available. The filters can be used individually or can be mixed:

For more details about the SIN number, please refer to the Sx tool software manual:
http://www.productselection.net/MANUALS/UK/sx_tool_manual.pdf

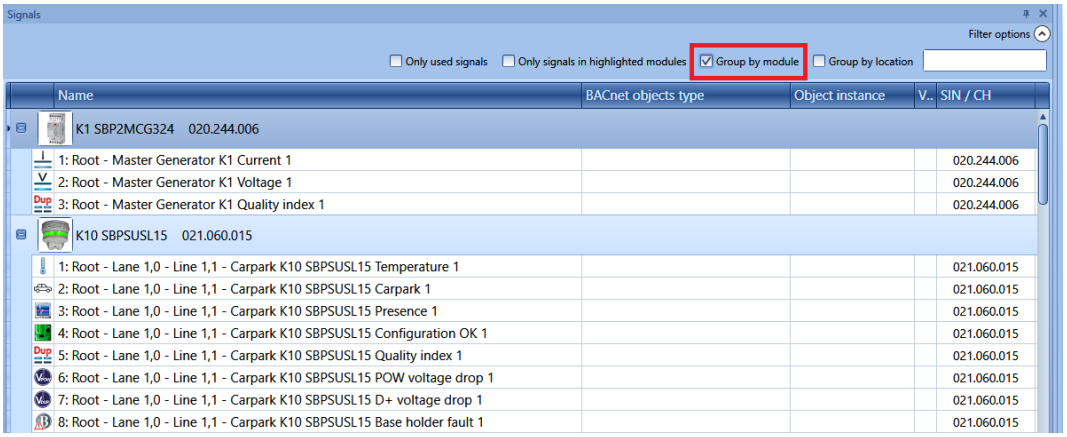
Only used signals:
Only the signals already used in at least one function are presented and they are highlighted in blue:



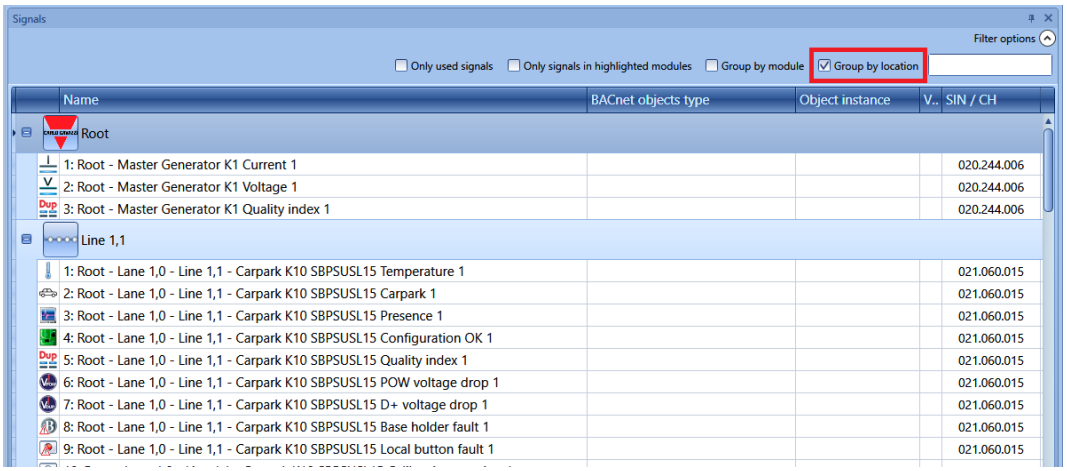
Only signals in highlighted modules:
When this filter option is selected, only the signals belonging to the highlighted modules are shown in the *Signals window*.
See the example below:



Group by module:
The signals are grouped by their own module, as shown in the example below:



Group by location:
The signals are grouped by location.



Search box:
The results will include all the words that contain the input string and the search will be carried out on the modules available in the project. The user can insert the full name or part of it.

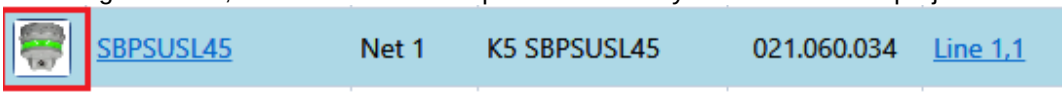
14.3 Car Park modules

14.3.1 Ultrasonic sensors

SBPUSL: indoor sensor 0° degrees (although the little 'o' is enough by itself)
SBPUSL45: lane mount sensor 45° degrees

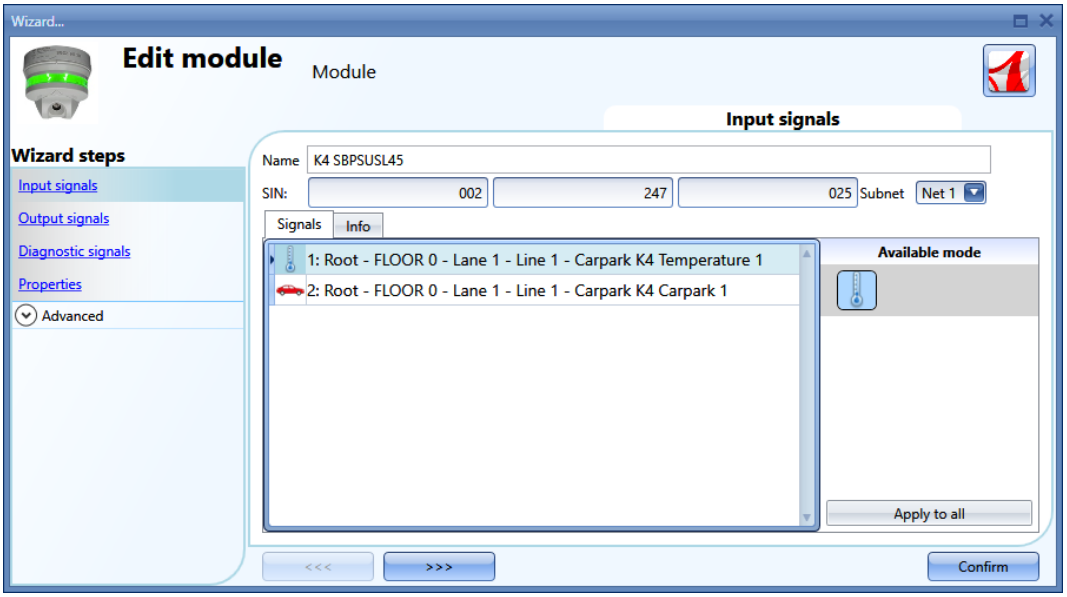
These modules can be added only in the location *Line*.

To configure these, click on the relevant picture once they are added to the project:



The configuration wizard will appear

Input signals



In the *Input signals* field, two signals are available to be used in any functions or in the database or read via Modbus or BACnet:

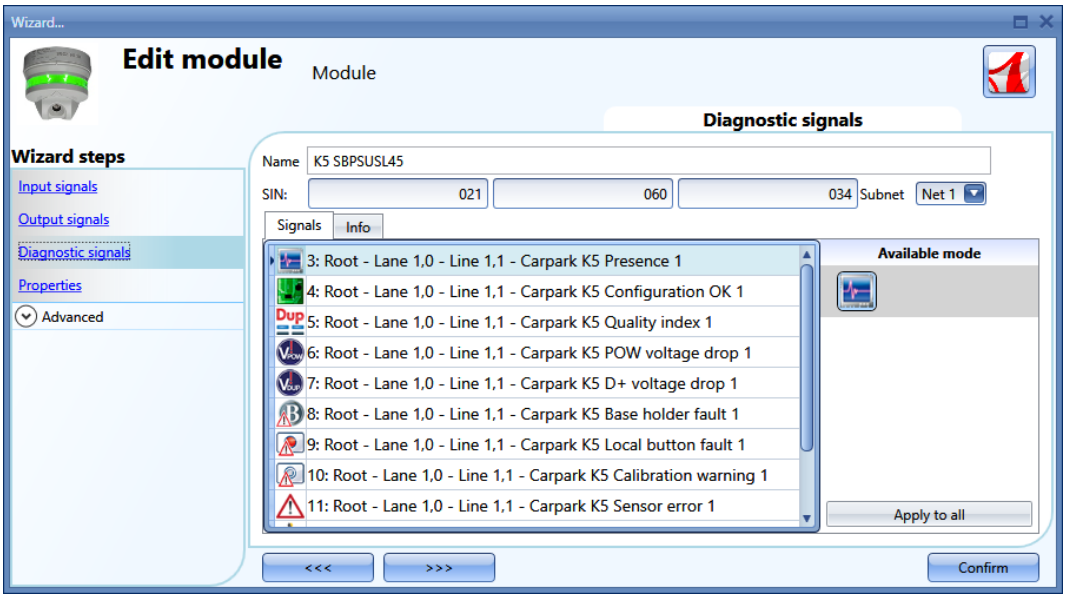
1) The temperature read by the sensor
2) The presence of a car in the parking bay monitored by the sensor

For more information about how to use these signals, please refer to the software manual: **http://www.productselection.net/MANUALS/UK/sx_tool_manual.pdf**

Output signals

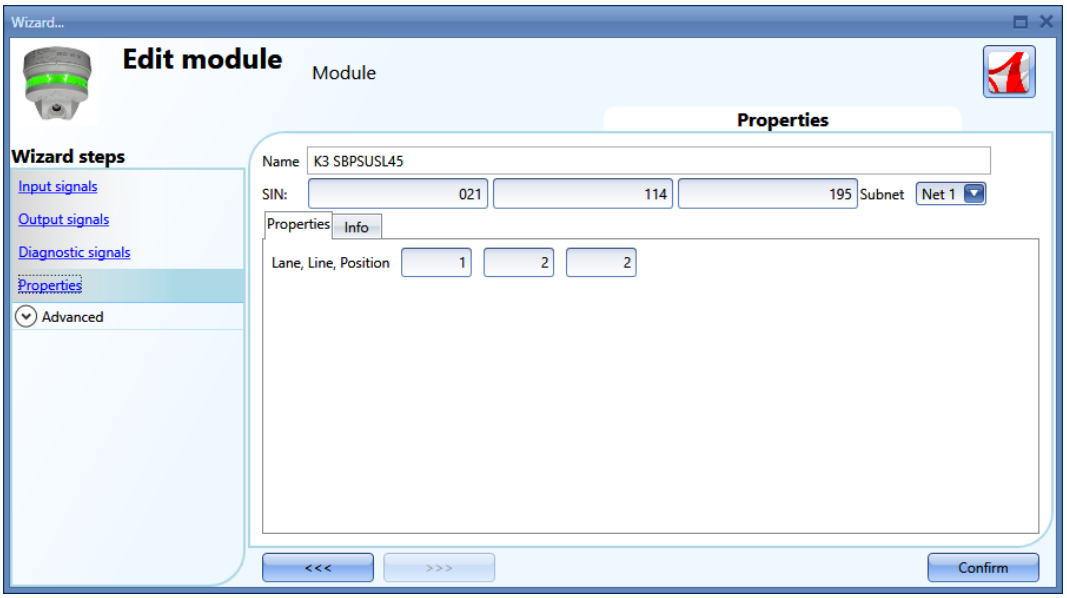
There are no output signals that can be managed by the controller.

Diagnostic signals



Icon	Description
	The module is alive
	The module is programmed
	Quality index of the Dupline bus (=100 ok, <100 noise is present on the bus)
	Voltage drop on the Dupline third wire
	Voltage drop on the Dupline bus
	Base holder is faulty
	The pushbutton of the sensor is always on
	Calibration error: no echo received
	Ultrasonic sensor is faulty
	Cross talk: signals received from another sensor
	Dupline power voltage
	Dupline bus voltge

Properties



In the properties window, the following parameters can be edited:

SIN
This field can be compiled manually or automatically by means of the Dupline network discovery

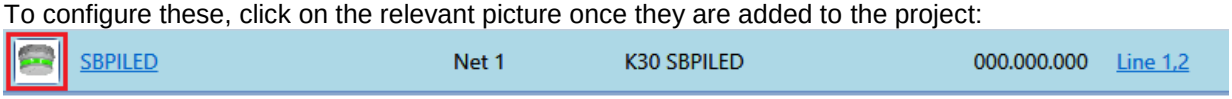
Subnet
The network can be selected manually or automatically by means of the Dupline network discovery

Lane, line, position
Car park address

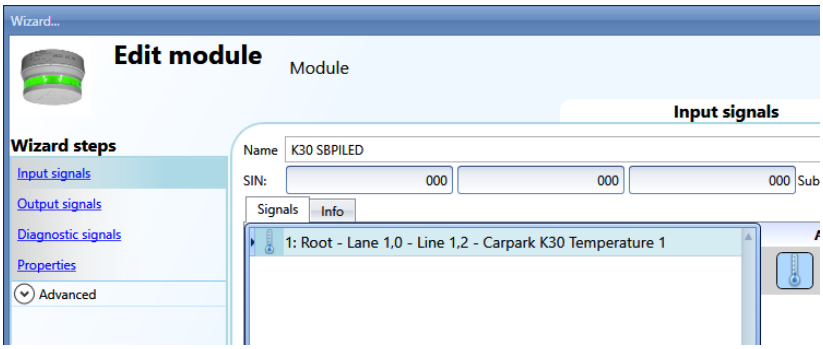
14.3.2 Lane indicator

SBPILED: indoor lane indicator

To configure these, click on the relevant picture once they are added to the project:



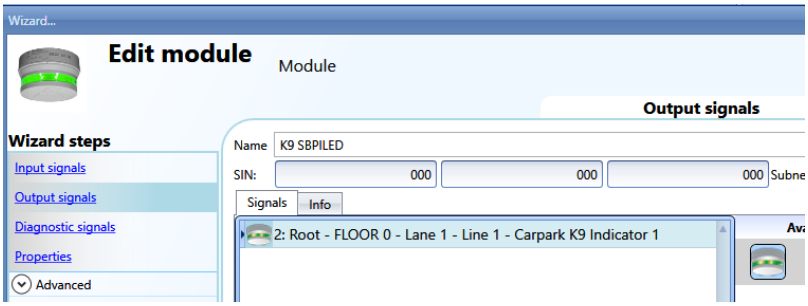
Input signals



In the *Input signals* field, one signal is available to be used in any function or in the database or read via Modbus or BACnet: for example the temperature read by the sensor

For more information about how to use these signals, please refer to the software manual: **http://www.productselection.net/MANUALS/UK/sx_tool_manual.pdf**

Output signals



One output signal, the LED, can be managed by any logic in the system.

Diagnostic signals
They are those as described for the sensors.

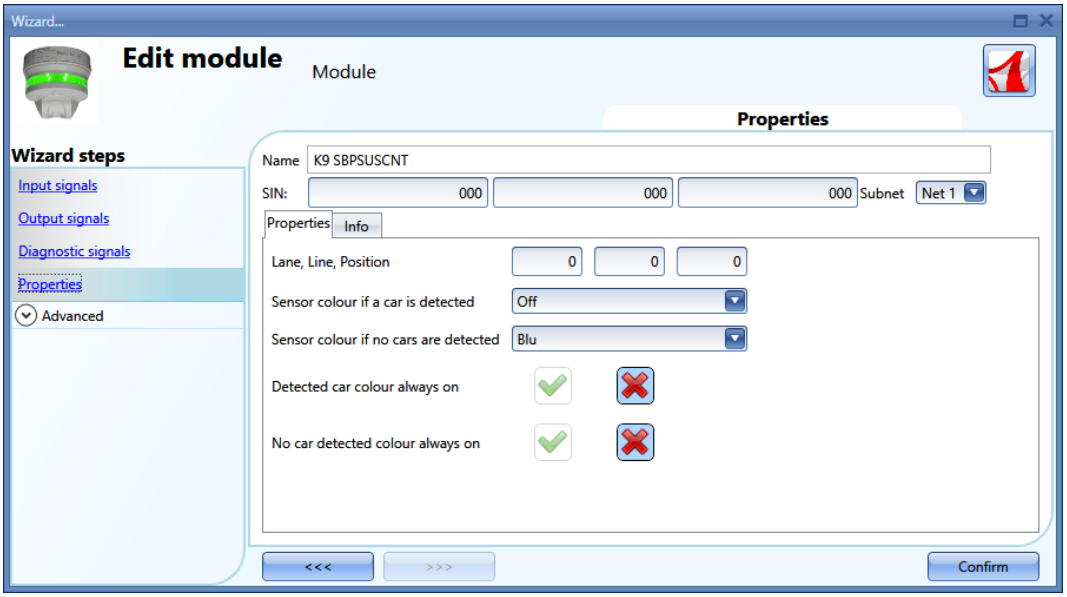
14.3.3 Counter sensor

SBPUSCNT: indoor counter sensor

These modules can be added only in the location *Line*.

The fields *Input signals*, *Output signals*, *Diagnostic signals* are the same as already described for the SBPSUSL45.

Properties



Sensor colour if a car is detected: Colour of the sensor when a moving car is detected

Sensor colour if no cars are detected: Colour of the sensor if no cars are moving under it

Detected car colour always on: The LED of the sensor will always be the one selected in the field *Sensor colour if a car is detected.*

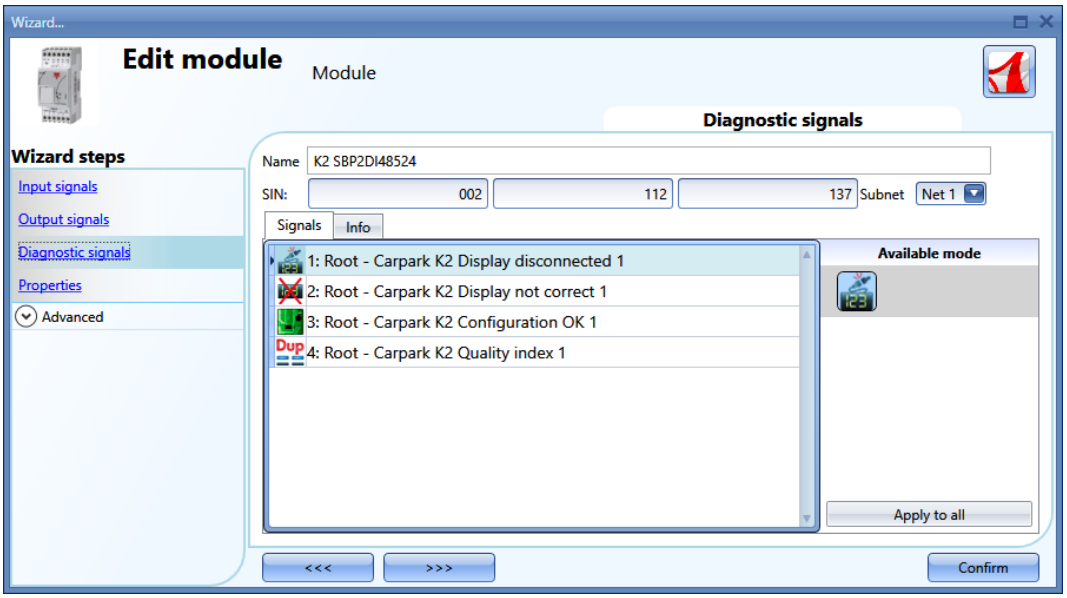
No car detected colour always on: The LED of the sensor will always be the one selected in the field *Sensor colour if a no cars are detected.*

RS485 to smart-dupline interface

SBP2DI48524

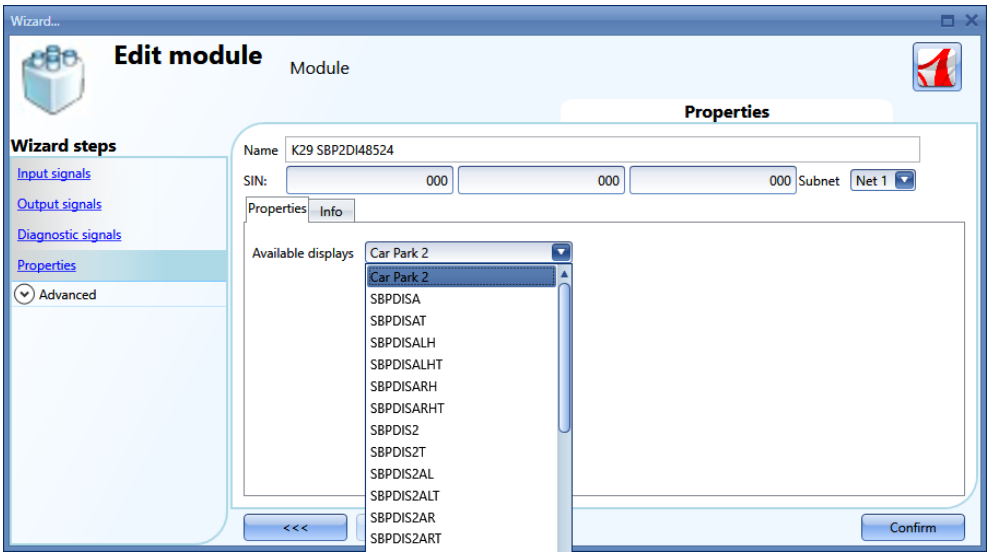
In the RS485 to smart-dupline interface there are no *Input/Output signals*

Diagnostic signals

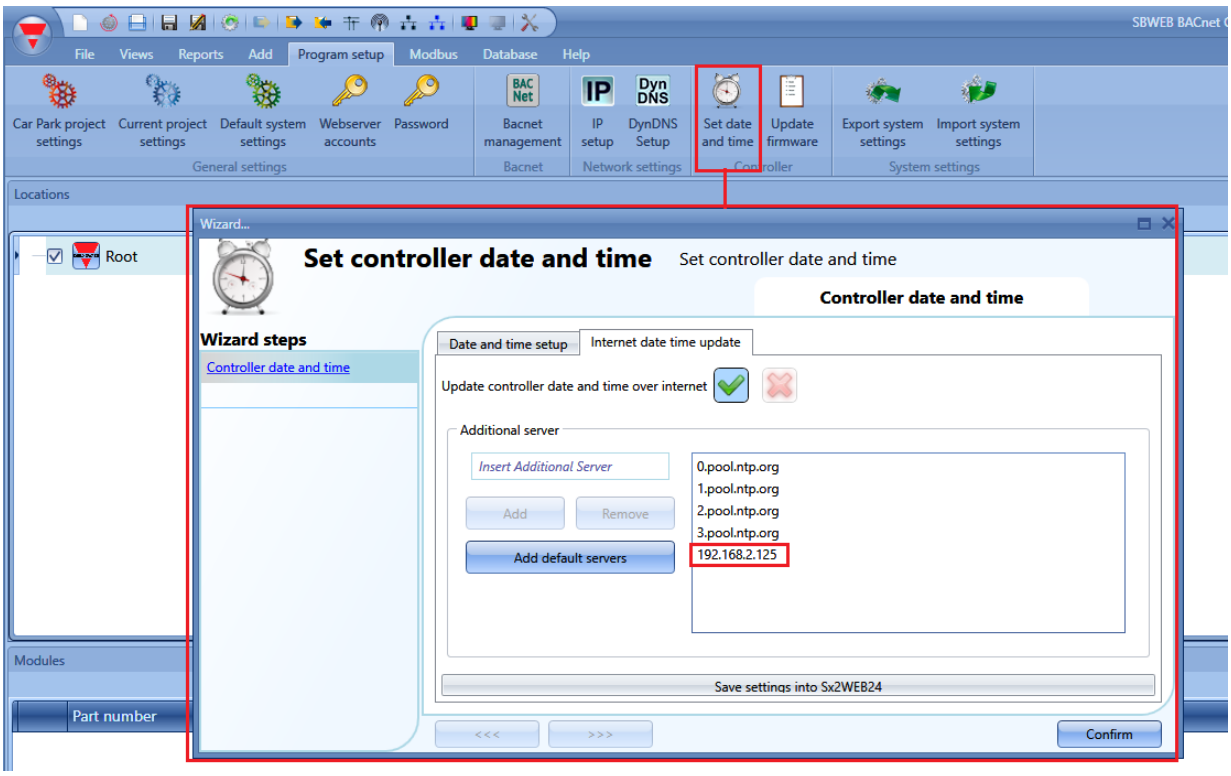


	This signal indicates if the display is properly connected
	This signal indicates if the type of display connected is the one selected in the field *Properties*
	This signal indicates if the module has been programmed
	Quality index of the Dupline bus (=100 ok, <100 noise is present on the bus)

Properties



In this field, the display connected to the SBO2DI48524 has to be selected.

15 Time server

If the car park controller SBP2WEB24 needs to synchronise the time via Internet, it has to go through the SBP2CPY24, so in the *Set controller date and time* window, the address of the SBP2CPY24 has to be added. It can also be a DYNDNS address.