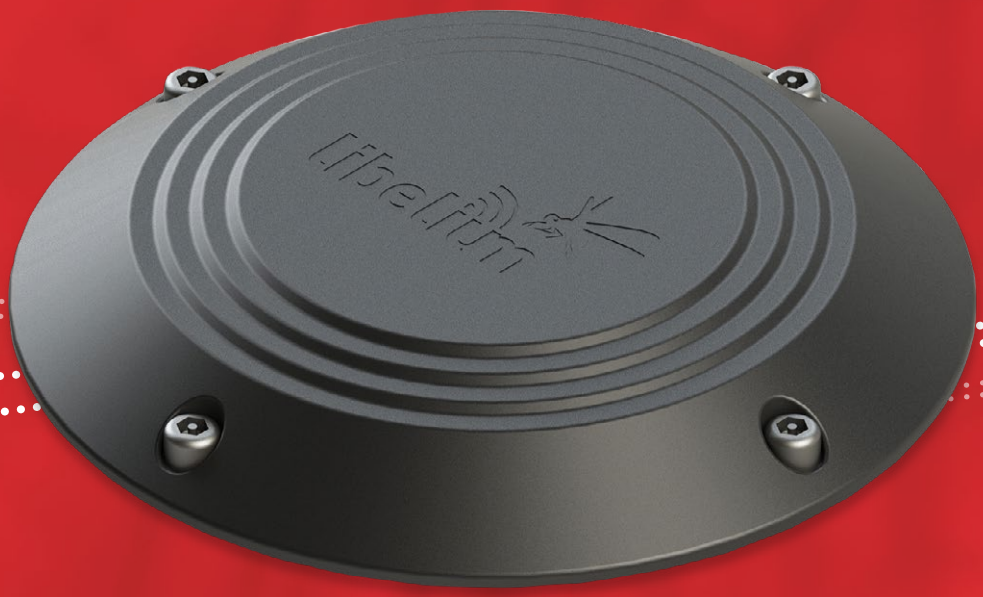
Plug & Sense! Smart Parking

Technical Guide



libelium

waspmote

Document version: v7.0 - 10/2016

INDEX

1. Introduction

The new version of **Waspmote Plug & Sense! Smart Parking**, the solution for Smart Cities that allows citizens **to detect available parking spots.**

The new device is **easier and cheaper to deploy** as it is mounted on the road surface. Unlike most market versions, it does not need to dig a hole in the ground for installation, reducing installation time from 30 to 5 minutes and allowing to be replaced by another unit in case of maintenance in just 10 minutes.

The smaller size -reduced over 50%-, its higher accuracy and reliability, and the faster time of detection, besides the independence from temperature are also important features of new Smart Parking device.

New sensor system is **fully compatible with LPWAN** radio technologies **-LoRaWAN** and **Sigfox-** to enable long range and low power consumption. It can be connected with both radios for the European 868 MHz band and for the 900-930 MHz band (US / Canada). One unique feature of the system is that it allows to use both radio technologies at the same time or changing from one to the other using the manager system from the Cloud.

With the new sensor system, one base station can give service to thousands of devices around a range of several kilometers in urban environment. This fact provides lower costs of installation since the number of **base stations can be dramatically reduced**. Besides, the new sensor model has been optimized for really low-power operation, so the battery lifetime is extended **up to 10 years** easily.

The new Smart Parking node has been granted with the CE / FCC / IC marks and provides a **robust software which works out-of-the-box**. Developers do not have to cope with programming the nodes, they just have to specify the values of key parameters in the firmware such as working cycle or night mode to be ready to work. **Remote management and bidirectional communicatio**n allows to change several parameters of the nodes from the Cloud. This means we can reprogram thousands of nodes by just setting the right values from our web browser in the management platform.

2. General

2.1. General and safety information

- In this section, the term "Waspmote" encompasses both the Waspmote device itself and it enclosure.
- Read through the document "General Conditions of Libelium Sale and Use".
- Do not allow contact of metallic objects with the electronic part to avoid injuries and burns.
- NEVER submerge the device in any liquid with the enclosure open.
- Keep the device in a dry place and away from any liquid which may spill.
- Waspmote consists of highly sensitive electronics which is accessible to the exterior, handle with great care and avoid bangs or hard brushing against surfaces.
- Check the product specifications section for the maximum allowed power voltage and amperage range and consequently always use a current transformer and a battery which works within that range. Libelium is only responsible for the correct operation of the device with the batteries, power supplies and chargers which it supplies.
- Keep the device within the specified range of temperatures in the specifications section.
- Do not connect or power the device with damaged cables or batteries.
- Place the device in a place only accessible to maintenance personnel (a restricted area).
- Keep children away from the device in all circumstances.
- If there is an electrical failure, disconnect the main switch immediately and disconnect that battery or any other power supply that is being used.
- If a hardware failure occurs, consult the Libelium Web Development section.
- Check that the frequency and power of the communication radio modules together with the integrated antennas are allowed in the area where you want to use the device.

2.2. Conditions of use

- Read the "General and Safety Information" section carefully and keep the manual for future consultation.
- Use Waspmote in accordance with the electrical specifications and the environment described in the "Hardware" section of this manual.
- Do not place Waspmote in contact with metallic surfaces; they could cause short-circuits which will permanently damage it.
- IMPORTANT It is the responsibility of the installer to find out about restrictions of use for frequency bands in each country and act in accordance with the given regulations. Libelium Comunicaciones Distribuidas S.L does not list the entire set of standards that must be met for each country.
- For further information go to:
  - CEPT ERC 70-03E - Technical Requirements, European restrictions and general requirements: http://www.erodocdb.dk/
  - R&TTE Directive - Equipment requirements, placement on market: http://www.erodocdb.dk/
- Further information you may need can be found at: http://www.libelium.com/development/waspmote
- The "General Conditions of Libelium Sale and Use" document can be found at: http://www.libelium.com/development/waspmote/technical_service

3. Hardware

3.1. Hardware description

Figure : Plug & Sense! Smart Parking

Operating frequency	Sigfox radio for Europe: 868.0 to 869.7 MHz LoRaWAN radio for Europe: 863.0 to 870.0 MHz Sigfox radio for USA: 902.0 to 928.0 MHz LoRaWAN radio for USA: 902.0 to 928.0 MHz
Power supply	Built-in Lithium batteries, expected lifetime of 4-6 years *
Antenna	Included
Detection	Magnetic
Mounting	Over the floor
Dimensions	230 mm diameter, 28 mm height
Protection	IP67, completely sealed housing
Operating temperature	-20 to +65 ºC

** Under normal circumstances and dependent on settings*

Figure : Plug & Sense! Smart Parking main characteristics

3.2. Power consumption

	Consumption
Measuring sensor	*TBD*
Transmission Sigfox	*TBD*
Transmission LoRaWAN	*TBD*
Sleep state	25 µA
Battery self discharge	< 1% month at +20 ºC

Figure : Plug & Sense! Smart Parking power consumption

4. How the node works

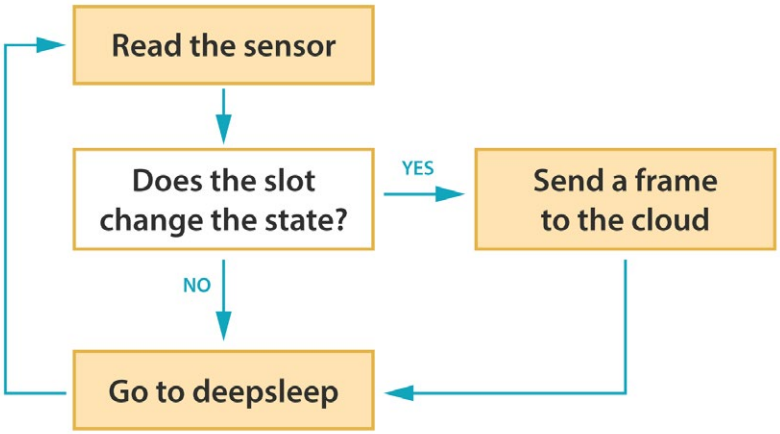


Figure : Basic working loop diagram

As the diagram indicate, the basic loop of the node consists in read the sensor and send a frame when the parking slot change it status. Then, it sleeps a desired time and starts the loop again.

Some events can forced the node to send a frame to the cloud. If a desired time elapsed since the last radio transmission, the node will send a Keep-Alive frame. This frame only contains basic data from the node (parking slot status and battery status). It is useful to know that there is no changes in the slot, but the node still working. The node also will send a frame each 24 hours with the working data of the day (number of measurements, number of transmissions,…).

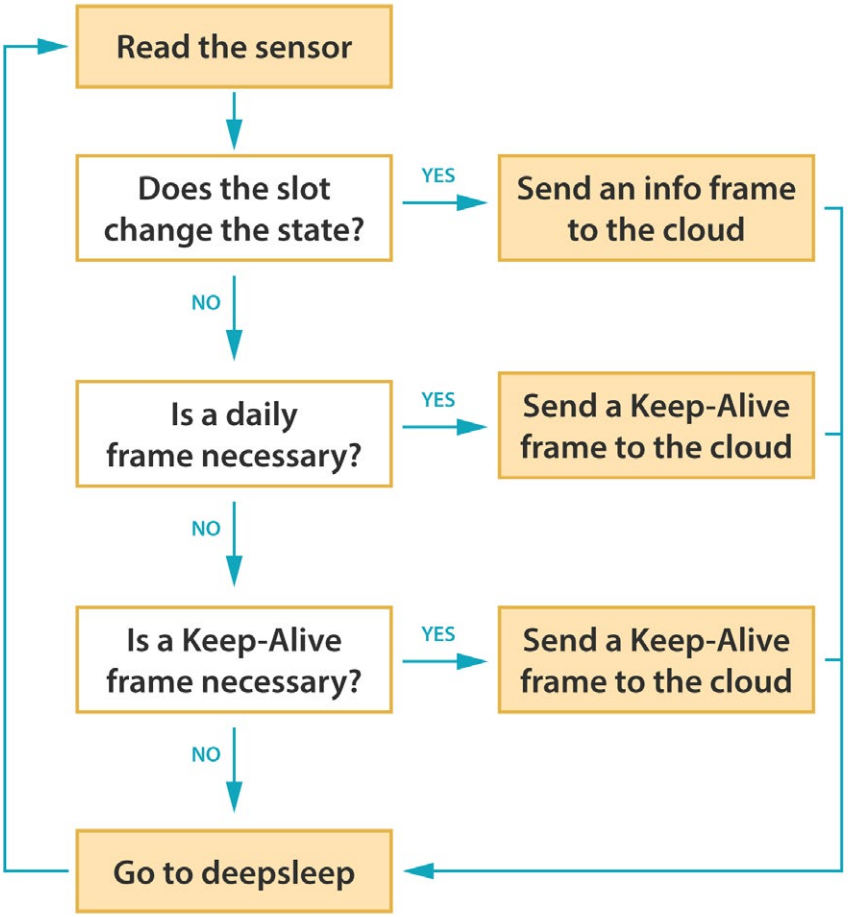


Figure : Extended loop diagram

5. Sleep modes

Plug & Sense! Smart Parking has 2 sleep modes: day mode and night mode. The second one has been developed to use when the parking slot is expected to have fewer changes (i.e. at night). Each mode has its own configuration parameters. The figure below shows an example for the node transmissions in a day. The time zone between 6 AM and 12 AM (in light gray) indicates that the node are working in day mode. In this mode, the sampling of the parking slot is made more regular (1 minute) and the Keep-Alive is only 2 hours. In the dark gray zone, from 12 AM to 6 AM, the node is working in night mode. As is shown in the example, the sampling time is greater (10 minutes) and the Keep-Alive increases too (3 hours).

Example configuration:

Parameter	Configuration
Sleep time	1 minute
Keep-Alive	2 hours
Night Mode start hour	00 hours
Night Mode duration	6 hours
Night Mode Sleep Time	10 minutes
Night Mode Keep-Alive	3 hours

Figure : Example configuration

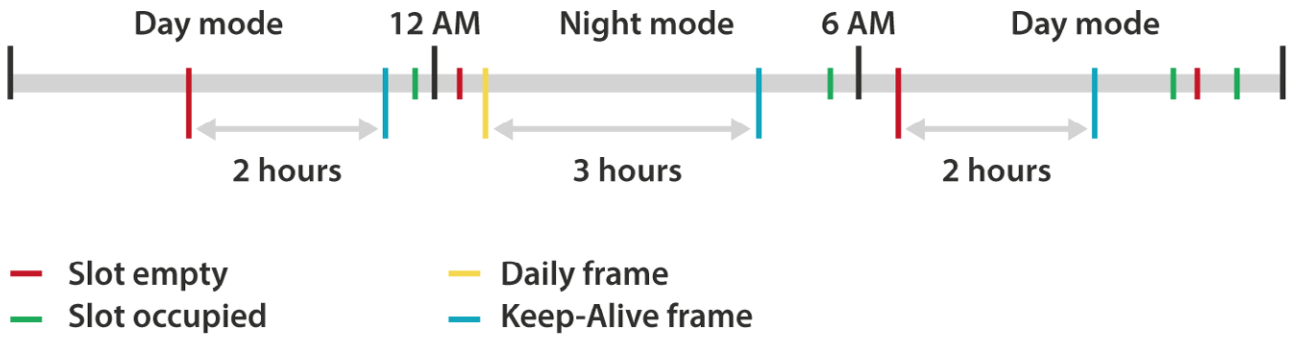


Figure : Day and night modes example

5.1. Day mode

It is the basic working mode and it has 2 configurable parameters:

- **Sleep time:** Sleep time between consecutive sensor measurements. 2 minutes option is configured by default.
- **Keep-Alive:** Elapsed time since last transmission to trigger a Keep-Alive frame. 10 hours option is configured by default. This frame only contains basic data from the node (parking slot status and battery status). It is useful to know that there are no changes in the slot, but the node still working. This mode can be disabled using both the USB Programmer or the Remote Manager.

5.2. Night mode

This mode has been developed to use when the parking slot is expected to have fewer changes (i.e. at night) and it has 4 configurable parameters:

- **Night Mode start hour:** Beginning hour of the night mode. 22 hours option is configured by default.
- **Night Mode duration:** Night mode duration time. 8 hour option is configured by default.
- **Night Mode Sleep Time:** Sleep time between consecutive sensor measurements (during night mode). 10 minutes option is configured by default.
- **Night Mode Keep-Alive:** Elapsed time since last transmission to trigger a Keep-Alive frame (during night mode). 4 hours option is configured by default. This frame only contains basic data from the node (parking slot status and battery status). It is useful to know that there is no change in the slot, but the node is still working.

This mode can be disabled using both the USB Programmer or the Remote Manager.

6. Transmission modes

Plug & Sense! Smart Parking has 5 transmission modes allowing to the user to choose between Sigfox, LoRaWAN y their combinations:

- **Sigfox.** This mode only use the Sigfox radio to send the data collected by the node. This mode is selected by default.
- **LoRaWAN.** This mode only use the LoRaWAN radio to send the data collected by the node
- **Sigfox + LoRaWAN.** In this mode the data collected is sent using the 2 radios. It is recommended to test the node with the two transmission technologies.
- **Sigfox → LoRaWAN.** This mode uses Sigfox radio as primary radio. If there is an issue with the Sigfox radio, it will use LoRaWAN to send the frames.
- **LoRaWAN → Sigfox.** This mode uses LoRaWAN radio as primary radio. If there is an issue with the LoRaWAN radio, it will use Sigfox to send the frames.

7. Frames

Plug & Sense! Smart Parking node can send 6 defined frames. All frames are 12 bytes length and they are the same for Sigfox and LoRaWAN. Bytes 0 and 1 are common for all frames. Byte 0 has the basic data from the node and frame and byte 1 its a frame counter. It can be used to detect lost frames.

Bit	Name	Description
7	Parking slot status	'0' indicates that the parking slot is empty
		'1' indicates that the slot is in occupied
6	Battery state	'0' indicates that the battery has a good level of charge
		'1' indicates that the battery has little charge and it will be necessary to change it. When the battery has little charge it is possible that the node does not work properly and the radios fail sending the frames.
5-4	Reserved	Reserved bits. Do not consider.
3	Frame type	0 – Info frame 1 – Keep-Alive frame 2 – Daily update frame 3 – Error frame 4 – Start frame 1 5 – Start frame 2 Values from 6 to 15 are reserved
2		
1		
0		

Figure : Byte 0 description

7.1. Info frame

It is the most common frame sent by the node. The node will send this kind of frame each time it detects that the parking slot changed from empty to occupied or vice-versa. The other bytes are used to give additional data to the user.

Byte	Name	Description
0	Basic data	Detailed description in the section "Frame"
1	Frame counter	Detailed description in the section "Frame"
2	Temperature MSB	Raw temperature from the node's internal sensor. The value stored in these two bytes is a 16-bit value in 2's complement form. To convert to Celsius degrees use the next formula: $Temperature(^{\circ}C)=\frac{(MSB\cdot 2^8+LSB)}{8}+25$
3	Temperature LSB	
4	X axis measurement MSB	Raw value from the sensor associated to the X axis. The value stored in these two bytes is a 16-bit value in 2's complement form.
5	X axis measurement LSB	
6	Y axis measurement MSB	Raw value from the sensor associated to the Y axis. The value stored in these two bytes is a 16-bit value in 2's complement form.
7	Y axis measurement LSB	
8	Z axis measurement MSB	Raw value from the sensor associated to the Z axis. The value stored in these two bytes is a 16-bit value in 2's complement form.
9	Z axis measurement LSB	
10-11	Reserved	Reserved bytes. Do not consider.

Figure : Info frame structure

7.2. Keep-Alive frame

This frame is used to indicate that the parking slot has not changed, but the node is still working.

Byte	Name	Description
0	Basic data	Detailed description in section "Frame"
1	Frame counter	Detailed description in section "Frame"
2	Timestamp (hh)	Current hours
3	Timestamp (mm)	Current minutes
4	Temperature MSB	Raw temperature from the node's internal sensor. The value stored in these two bytes is a 16-bit value in 2's complement form. To convert to Celsius degrees use the next formula: $Temperature(^{\circ}C)=\frac{(MSB\cdot 2^8+LSB)}{8}+25$
5	Temperature LSB	
6	X axis measurement MSB	Raw value from the sensor associated to the X axis. The value stored in these two bytes is a 16-bit value in 2's complement form.
7	X axis measurement LSB	
8	Y axis measurement MSB	Raw value from the sensor associated to the Y axis. The value stored in these two bytes is a 16-bit value in 2's complement form.
9	Y axis measurement LSB	
10	Z axis measurement MSB	Raw value from the sensor associated to the Z axis. The value stored in these two bytes is a 16-bit value in 2's complement form.
11	Z axis measurement LSB	

Figure : Keep-Alive frame structure

7.3. Daily update frame

This frame is sent daily at 1 AM. It contains a little summary. This frame can be deactivated using the Plug & Sense! Smart Parking USB Programmer or via radio, with the response given by the Remote Manager, hosted in the cloud.

Byte	Name	Description
0	Basic data	Detailed description in section "Frame"
1	Frame counter	Detailed description in section "Frame"
2	Sensor measurements MSB	Unsigned 16 bit counter. It stores the times that the sensor is used in the last 24 hours.
3	Sensor measurements LSB	
4	Sigfox transmissions MSB	Unsigned 16 bit counter. It stores the times that Sigfox radio is used in the last 24 hours.
5	Sigfox transmissions LSB	
6	LoRaWAN transmissions MSB	Unsigned 16 bit counter. It stores the times that LoRaWAN radio is used in the last 24 hours.
7	LoRaWAN transmissions LSB	
8	Resets	Number of resets generated in the last 24 hours
9	Config_id	Value of the configuration version loaded into the node
10-11	Reserved	Reserved bytes. Do not consider.

Figure : Daily update frame structure

This frame is sent daily at 1 AM. It contains a little summary. This frame can be deactivated using the Plug & Sense! Smart Parking USB Programmer or via radio, with the Remote Manager, setting to 0 the *enable/disable daily frame* bit.

The daily update frame is very special because the node waits for a response after it is sent. This response is useful for reconfiguring the node "over the air", without physical access. Also, a second use of this response frame is to synchronize the node's internal clock, thanks to a timestamp. This response can be configurated using the remote PHP.

7.4. Error frame

In some cases the node could send a frame if some internal components or processes fail.

Byte	Name	Description
0	Basic data	Detailed description in section "Frame"
1	Frame counter	Detailed description in section "Frame"
2	Error data	Detailed description below
3	Temperature MSB	Raw temperature from the node's internal sensor. The value stored in these two bytes is a 16-bit value in 2's complement form. To convert to Celsius degrees use the next formula: $Temperature(^{\circ}C)=\frac{(MSB\cdot 2^{8}+LSB)}{8}+25$
4	Temperature LSB	
5	X axis measurement MSB	Raw value from the sensor associated to the X axis. The value stored in these two bytes is a 16-bit value in 2's complement form.
6	X axis measurement LSB	
7	Y axis measurement MSB	Raw value from the sensor associated to the Y axis. The value stored in these two bytes is a 16-bit value in 2's complement form.
8	Y axis measurement LSB	
9	Z axis measurement MSB	Raw value from the sensor associated to the Z axis. The value stored in these two bytes is a 16-bit value in 2's complement form.
10	Z axis measurement LSB	
11	Battery level	Battery voltage in millivolts. To convert to millivolts use the next formula: $Battery\ voltage(mV)=((Battery\ level)\cdot 4)+2800$

Figure : Error frame structure

Bit	Name	Description
7-6	Reserved	Reserved bits. Do not consider.
5	Error Sigfox	Set to '1' when an error related with the Sigfox radio is detected. Clear when no issues detected.
4	Error LoRaWAN	Set to '1' when an error related with the LoRaWAN radio is detected. Clear when no issues detected.
3	Error RTC	Set to '1' when an error related with the RTC (internal clock) is detected. Clear when no issues detected.
2	Error X axis	Set to '1' when an error appears in the X axis of the sensor. Clear when no issues detected.
1	Error Y axis	Set to '1' when an error appears in the Y axis of the sensor. Clear when no issues detected.
0	Error Z axis	Set to '1' when an error appears in the Z axis of the sensor. Clear when no issues detected.

Figure : "Error data" byte structure

7.5. Start frames

When the node starts to work in the parking slot, it will send 2 frames. The first one is dedicated to the sensor and the battery. The second one is used to send some parameters about the chosen configuration.

7.5.1. Start frame number 1

Byte	Name	Description
0	Basic data	Detailed description in section "Frame"
1	Frame counter	Detailed description in section "Frame"
2	Temperature MSB	Raw temperature from the parking internal sensor. The value stored in these two bytes is a 16-bit value in 2's complement form. To convert to Celsius degrees use the next formula: $Temperature(ºC) = \frac{(MSB \cdot 2^8 + LSB)}{8} + 25$
3	Temperature LSB	
4	X axis reference MSB	Reference value from the sensor associated to the X axis. The value stored in these two bytes is a 16-bit value in 2's complement form.
5	X axis reference LSB	
6	Y axis reference MSB	Reference value from the sensor associated to the Y axis. The value stored in these two bytes is a 16-bit value in 2's complement form.
7	Y axis reference LSB	
8	Z axis reference MSB	Reference value from the sensor associated to the Z axis. The value stored in these two bytes is a 16-bit value in 2's complement form.
9	Z axis reference LSB	
10	Battery voltage MSB	Battery voltage in millivolts. The value stored in these two bytes is an unsigned 16-bit value.
11	Battery voltage LSB	

Figure : Start frame number 1 structure

7.5.2. Start frame number 2

Byte	Name	Description
0	Basic data	Detailed description in section "Frame"
1	Frame counter	Detailed description in section "Frame"
2	CODE_ID	Firmware version
3	NM_START	Beginning hour of the night mode
4	NM_PERIOD	Duration in hours of the night mode
5	NM_SLEEP_TIME	Sleep time between consecutive sensor measurements (during night mode)
6	NM_KEEP_ALIVE	Elapsed time since last transmission to trigger a Keep-Alive frame (during night mode)
7	RADIO_MODE	Selected transmission mode between Sigfox, LoRaWAN y their combinations
8	SLEEP_TIME	Sleep time between consecutive sensor measurements
9	KEEP_ALIVE	Elapsed time since last transmission to trigger a Keep-Alive frame
10	THRESHOLD	Threshold for detecting a vehicle over the parking slot
11	Reserved	Reserved byte. Do not consider.

Figure : Start frame number 2 structure

8. Smart Devices App

Libelium Smart Devices App is an important tool developed by Libelium that allows users install new firmware versions and program the configuration of the new Libelium devices in a few clicks. At the moment it is only available for SmartParking and MySignals products, but the list will be incremented shortly.

8.1. Installation

First of all and before installing anything, users have to take into account the platform where the application is going to be installed. To install the Libelium Smart Devices App, it is compulsory to have installed the JDK 1.8. If it is not installed in the computer, you can follow the steps and download it from this website:
https://docs.oracle.com/javase/8/docs/technotes/guides/install/install_overview.html

Once installed JDK, users can download the application using the appropriate link depending on the operative system:

- **Ubuntu:** http://downloads.libelium.com/smart_device_app/SmartDeviceApp_linux64.zip
- **Windows:** http://downloads.libelium.com/smart_device_app/SmartDeviceApp_windows32.zip
- **Mac:** http://downloads.libelium.com/smart_device_app/SmartDeviceApp_macosx64.zip

Then customers only have to extract the content of the SmartDeviceApp zip file downloaded in a place with the right permissions, and finally execute the file called "SmartDeviceApp" that will initialize the application. Please, note that the extension of this file will depend on the operative system the user is using at the moment (sh for Linux and OSX, and bat for Windows).

8.2. Smart Parking

This section provides several options to Smart Parking users in order to take full advantage of all possibilities the devices offers.

8.2.1. Programmer

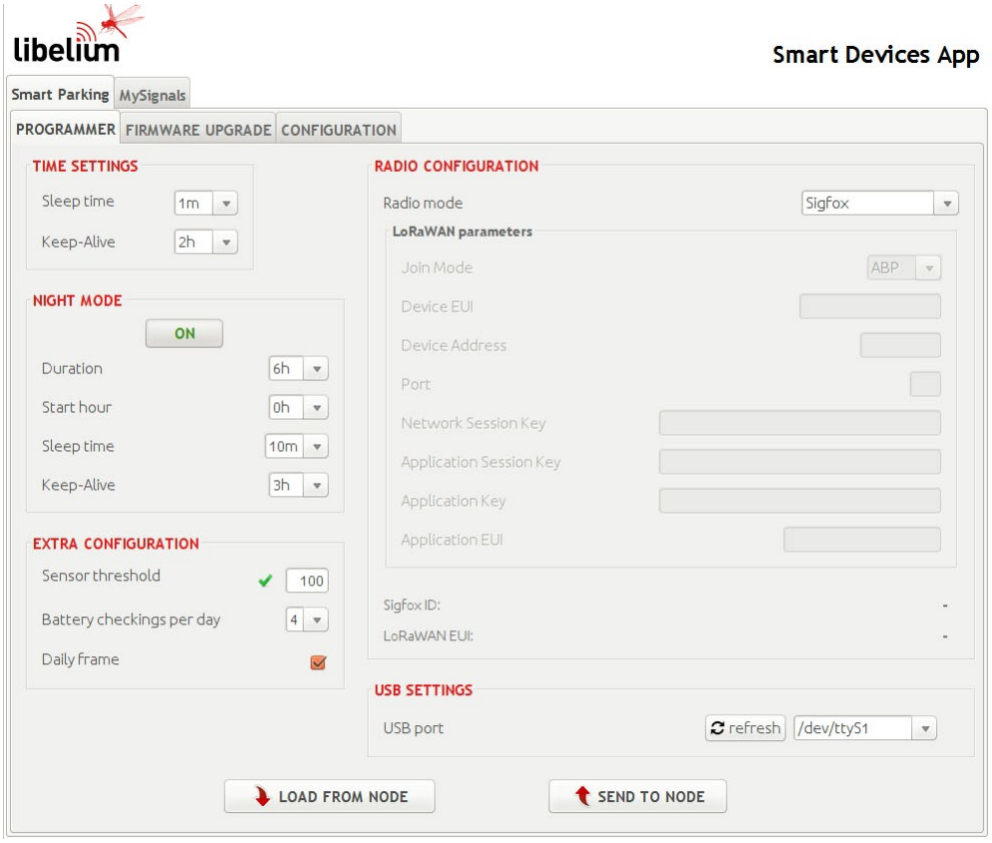


Figure : Smart Parking configuration form

Users can read and write all configuration parameters of their devices in this section. The process is quite simple. Just connect the device to the computer where the Smart Devices App is installed using the USB cable provided.

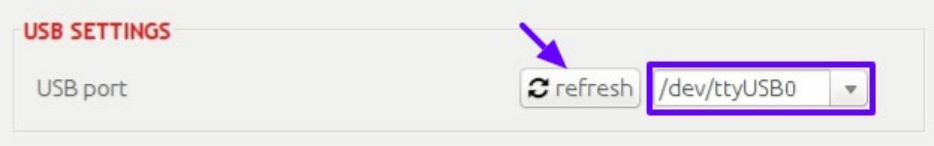


Figure : USB settings

Then, refresh the "USB settings" block which is in the bottom-right corner, clicking in "refresh" button. Once done it, the port where the device has been connected must be selected.

But before configuring the device, it is very important that users take in consideration the following list where all parameters are explained:

TIME SETTINGS

- **Sleep time:** Sleep time between consecutive sensor measurements.
- **Keep-Alive:** Elapsed time since last transmission to send a Keep-Alive frame.

NIGHT MODE

- **On/Off button:** Button to activate/deactivate this option. If it is not active, the following fields will not be effective.
- **Duration:** Night mode duration time.
- **Start hour:** Night Mode start hour.
- **Sleep time:** Sleep time between consecutive sensor measurements (during night mode).
- **Keep-Alive:** Elapsed time since last transmission to send a Keep-Alive frame (during night mode).

EXTRA CONFIGURATION

- **Sensor Threshold:** Threshold for detecting a vehicle over the parking slot.
- **Battery readings per day:** Battery readings per day.
- **Daily frame:** Enable/Disable daily frame sending.

RADIO MODE

- **Radio mode:** Radio transmission mode among Sigfox, LoRaWAN or their combinations.

LoRaWAN parameters

- **LoRaWAN join mode:** LoRaWAN join mode, ABP or OTAA.
- **Device EUI:** LoRaWAN device EUI.
- **Device Address:** LoRaWAN device address.
- **Port:** LoRaWAN port.
- **Network Session Key:** LoRaWAN network session key.
- **Application Session Key:** LoRaWAN application session key.
- **Application Key:** LoRaWAN application key.
- **Application EUI:** LoRaWAN application EUI.

Sigfox ID: Sigfox ID that will be loaded from the device.

- **LoRaWAN EUI:** LoRaWAN EUI that will be loaded from the device.

USB SETTINGS

- **USB Port:** In this list will be displayed all available USB ports to work out with the device. If you plug your device and the port is not listed, you have click on "Refresh" button in order to update the list.

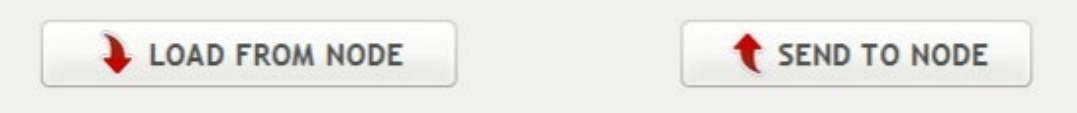


Figure : Load configuration from node & Send configuration to node buttons

"Load from node" button will read all parameters from the node and will displayed the information in the form. On the other hand "Send to node" button will overwrite the configuration in the node. All available fields have to be filled, with the proper format. If any parameter doesn't have an acceptable format, a red cross like this IMAGEN app_cross.png will be displayed near it and you can't write the information in the node. If the information introduced is valid, a green tick IMAGEN app_tick.png will be shown.

8.2.2. Firmware upgrade

In this tab, users can select the firmware version to install in their devices.

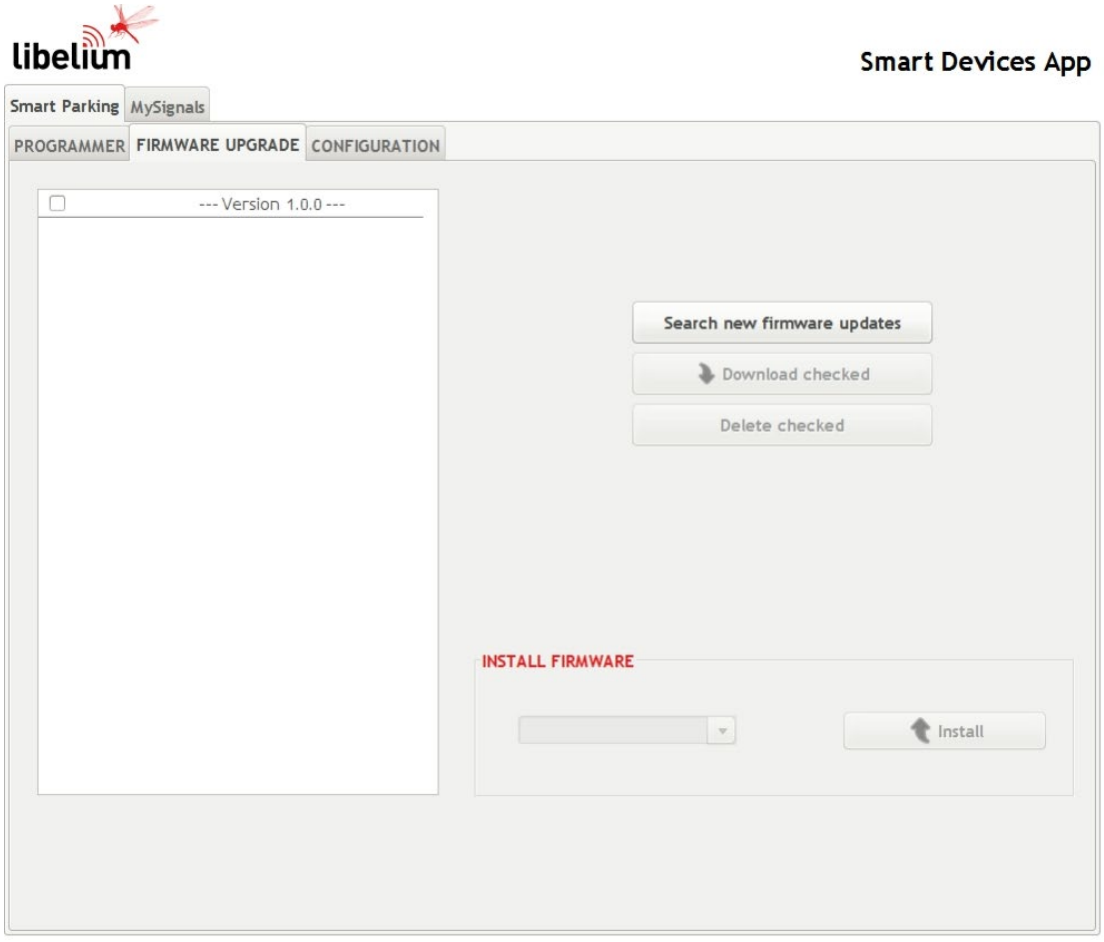


Figure : Smart Parking firmware upgrade form

The list with all available firmware is loaded when the program starts, but users can update it on demand, clicking on "Search new firmware updates" button. Before installing the firmware, it is necessary to download it. This process is very simple, just mark the check of the version you want to install from the list IMAGEN app_check.png and click on "Download checked" button.

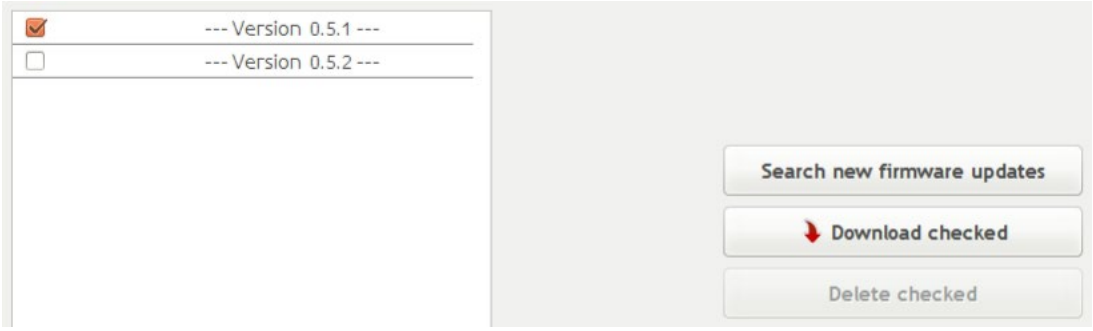


Figure : Download the firmware selected

When this item is downloaded, a disk will be displayed near it IMAGEN app_disk.png, indicating it is downloaded. Once the firmware is downloaded, it is ready to install using the "Install Firmware" section at the bottom. In the drop-down will appear all downloaded versions. Select one and then hit on "Install" button.

Figure : Install the firmware selected

You can also delete the downloaded firmware marking the check from the list IMAGEN app_check.png and then clicking on "Delete checked" button.

Figure : Delete downloaded firmware

Remember that the USB port must be selected in the programmer tab.

8.2.3. Configuration

In the last tab "Configuration", it will be displayed all external parameters that the software uses to work. Users can modify this values in order to get the wished application behavior.

In this case, there is only one parameter available, the API key to connect to Libelium Cloud. This value is provided by Libelium and it is very important to control the access and get some results needed in the programmer tab. If users don't fill this field, the software don't work.

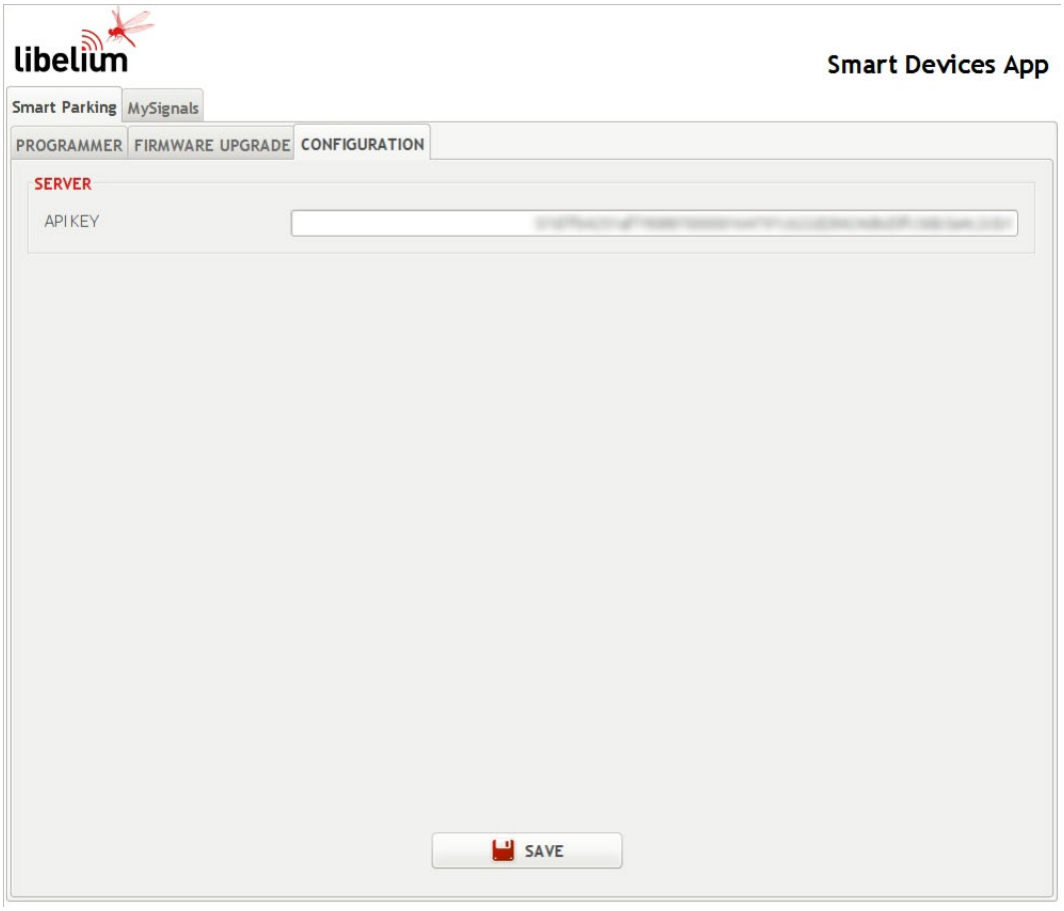


Figure : Smart Parking configuration form

9. Callback Server

For using the SigFox and LoraWAN callback functionality is needed to configure a back-end web server application that will receive SigFox and LoraWAN requests.

Libelium provides the source files of a standard back-end web server application, including a simple web graphic interface to configure SmartParking nodes. Users can deploy those source files as their customer server app for receiving the SigFox and LoraWAN backend requests in order to provide the proper answer to the Smart Parking devices through the service chosen.

Deploying this simple back-end web server application has standard requirements for any web application: Apache server, PHP libraries, static and public IP. After this process is finished, SigFox and LoraWAN backends should be configured using the resulting URL:

https://my_server.com/path/zip/extracted

No need of DB is needed for this simple implementation provided by Libelium but it is very easy to extend its functionality implementing connections to MySQL, PostgreSQL or any other DB storage if required.

NOTE: *Customers have to ask for this source code to Libelium Sales Department after buying the nodes.*

9.1. Installation

Before installing the web server, you have to bear in mind that this computer has to be accessible from the Internet, SigFox and LoraWAN backends will try to connect to this server. We recommend to implement existing security policies for web services: SSL, Firewall, user credentials and tools to avoid DoS attacks.

Libelium back-end web server application is developed using PHP language, so any web server supporting this technology may be used. Apache web server with PHP libraries is a suitable configuration to be used. DB Storage packages like MySQL, Postgresql or any other database engine is optional for custom requirements.

Visit the following URL's to find some tutorials about performing this installation steps in several Operating Systems.

- **Ubuntu**: https://help.ubuntu.com/community/ApacheMySQLPHP
- **Windows**: http://www.ampsoft.net/webdesign-l/how-to-install-apache-php-mysql.html
- **Mac**: http://jason.pureconcepts.net/2012/10/install-apache-php-mysql-mac-os-x/

Furthermore, some platforms use other technologies apart from the web server in order to get a fully functional callback server running.

Visit the following URL's to read how to install nodejs and npm in the web server.

- **Ubuntu**: http://www.hostingadvice.com/how-to/install-nodejs-ubuntu-14-04/
- **Windows**: http://blog.teamtreehouse.com/install-node-js-npm-windows
- **Mac**: http://shapeshed.com/setting-up-nodejs-and-npm-on-mac-osx/

9.2. Deploying

Extract the ZIP file provided by Libelium containing the standard back-end web server application source files in your server. Configure the web server with the path where the source files has been extracted. Following instructions describe how to perform the configuration using Ubuntu and Apache environment. Create a new configuration file my_server.conf, usually Apache web server context configuration files should be placed in /etc/apache2/sites-available directory.

The following example of an Apache configuration can be used to change the paths according to your own installation:

```
<VirtualHost *:80>
        ServerAdmin webmaster@localhost
        ServerName my_server.com
        ServerAlias my_server.com

        DirectoryIndex index.html index.php
        DocumentRoot /path/zip/extracted

        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined

        <Directory /path/zip/extracted>
                Options Indexes FollowSymLinks
                AllowOverride All
                Require all granted
        </Directory>
</VirtualHost>
```

Check the right owner/group and permissions of all the files extracted in order to be accessible.

Usually using *www-data* group is default in Ubuntu environment.

Check the permissions of all folders and files.

Usually using 0770 for directories and 0660 for files is default in Ubuntu environment.

Enable this new configuration site created and restart the Apache server.

9.3. Making the server accessible from anywhere

Compulsory for Sigfox and Actility platfoms.

Optimal deployment includes a server name pointing to a public and static IP, using dynamic DNS could be done with services like no-ip which has a free package http://www.noip.com/. No-ip has also a client application responsible for updating any IP address changes in the background, more information in: http://www.noip.com/download

Finally, the URL to configure the SigFox and LoraWAN platforms to send the callback request to will be conformed with the server name and the paths:

https://my_server.com/path/zip/extracted/

9.4. Remote node configuration web

The remote node configuration tool is available in the URL you established in your configuration file previously *(http://my_server.com/)*. This tool is used to configure the nodes in a remote way when received messages require an specific answer. Similar to the Java application, this web form can modify some parameters of the parking nodes remotely. The aim of this remote manager is to generate a binary file with the configuration selected:

1. Response to the Start Frame number 1, transmitted by the node in the activation process. This response frame allows the initial time synchronization of the internal clock. In normal conditions, the activation process happens only once.

2. Response to the Daily Update Frame with a timestamp, in order to perform a time synchronization of the internal clock. This will happen once per day and helps to keep the node's time drift to minimum.

3. Response to the Daily Update Frame with a new configuration for the node. In normal conditions, this will never happen in the lifetime of the node (or very few times). A change in the node's configuration will only happen when the administrator of the network decides to change the configuration which was set via the USB Programmer.

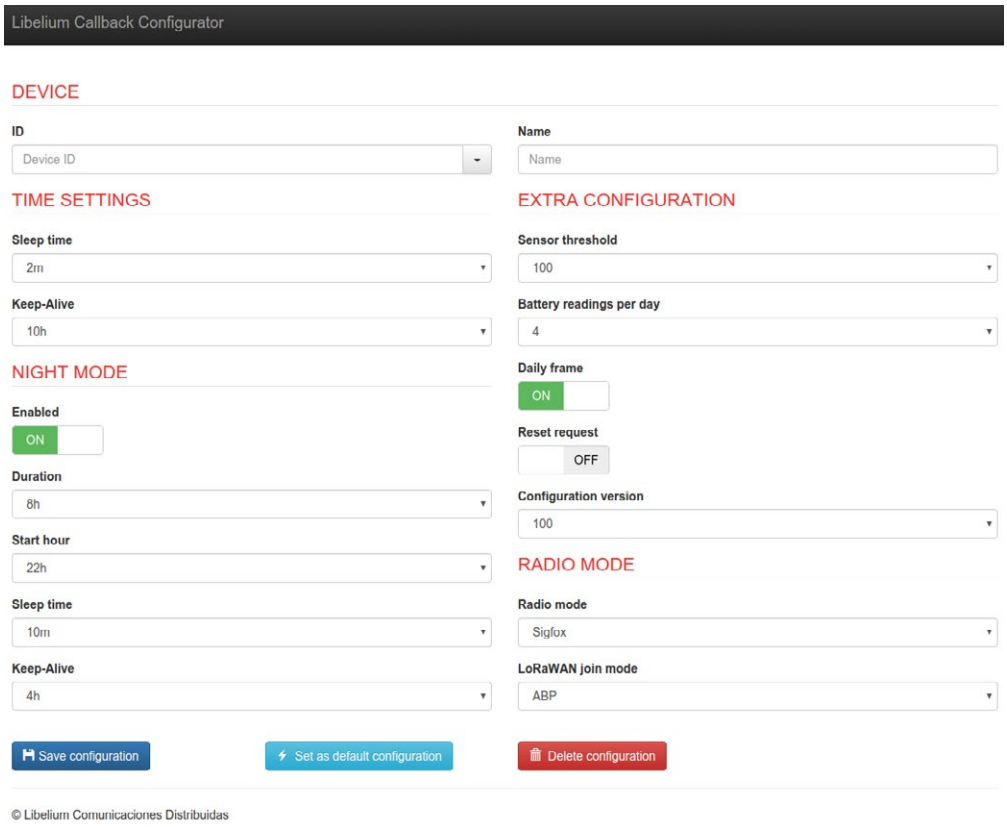


Figure : Libelium Callback Configurator screen-shot

Users can create, update and delete the same configuration for several devices. To do that, just type the ID separated by semicolons (;) like this: 00145F;001460;001461 and the rest configuration parameters will be saved with each device ID.

Here there are the description for all fields:

DEVICE

- **ID:** Device ID of the node. In each platform section we explain how to get this parameter.
- **Name**: A name associated to the device ID to make easier for you the identification of the node.

TIME SETTINGS

- **Sleep time**: Sleep time between consecutive sensor measurements.
- **Keep-Alive:** Elapsed time since last transmission to send a Keep-Alive frame.

NIGHT MODE

- **Enabled:** Button to activate/deactivate this option. If it is not active, the following fields will not be effective.
- **Duration:** Night mode duration time.
- **Start hour:** Night Mode start hour.
- **Sleep time:** Sleep time between consecutive sensor measurements (during night mode).
- **Keep-Alive:** Elapsed time since last transmission to send a Keep-Alive frame (during night mode).

EXTRA CONFIGURATION

- **Sensor Threshold:** Threshold for detecting a vehicle over the parking slot.
- **Battery readings per day:** Battery readings per day.
- **Daily frame:** Enable/Disable daily frame sending.
- **Reset request:** Enable/Disable the reset when a vehicle abandon the parking slot.
- **Configuration version:** Code version identifier.

RADIO MODE

- **Radio mode:** Radio transmission mode among Sigfox, LoRaWAN or their combinations.
- **LoRaWAN join mode:** LoRaWAN join mode, ABP or OTAA.

Before configuring the platform that we will explain in the following sections, we recommend to configure all the radio nodes in the web server in order to generate the proper answer to them when a message is received. If, for certain reasons, users can not configure it, don't worry. All nodes will synchronize date and time with the server but not the remote configuration. To get the whole functionality, each radio node has assigned an ID that has to be stored internally. Users can create or update the information of every device.

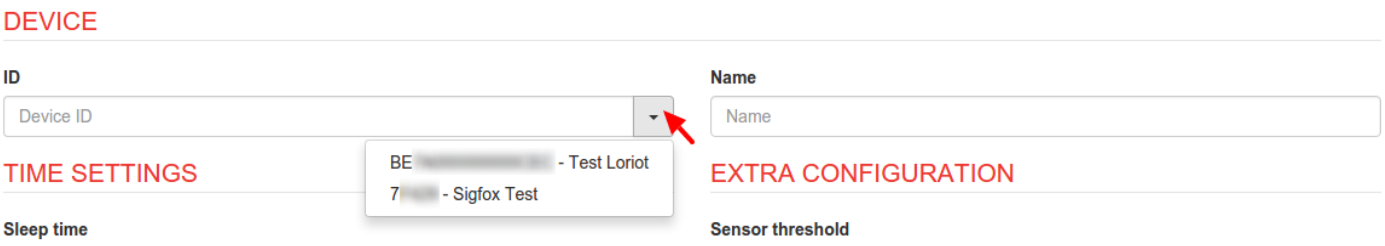


Figure : Device creation detail

Update: Search the node in the list, and click on it. All parameters saved will be loaded in the form.

Create: Overwriting ID and Name fields with ones which do not exist, when the information is saved, it will be created. If the device ID exists, it will be updated like if you select the device from the list.

Getting the IDs from the node is quite easy from the Libelium Smart Devices App. The IDs to be created in your web server are marked in the following picture:

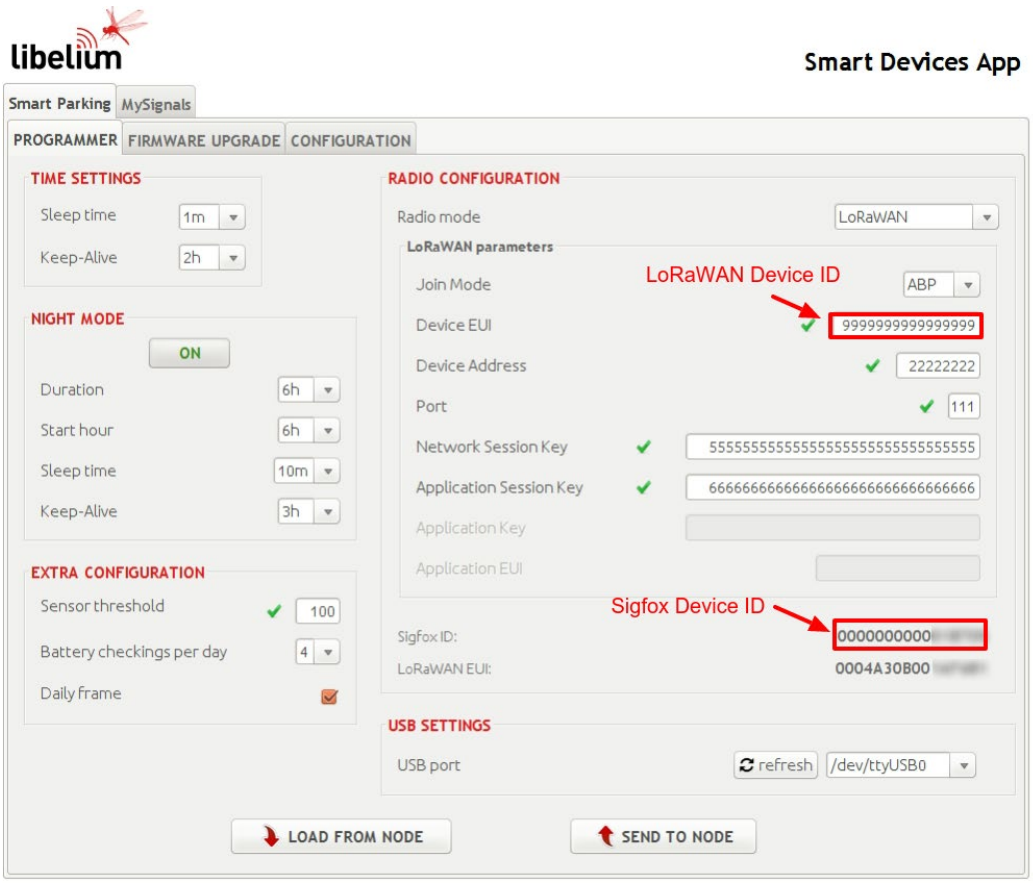


Figure : Device IDs readed from the Smart Devices App

Then, it is necessary to configure the rest parameters. Click on "Save configuration" and a background procedure will start to save all information needed to generate a valid answer to the node.

10. Developing the network

10.1. Application considerations

10.1.1. Deployment of the motes

The optimum deployment point will be the one where the probability of detection is maximum, which means minimizing the probabilities of false detection (caused by other vehicles or objects near the lot under control) and false rejection (owed to a not high enough variation in the magnetic field above the mote with a vehicle parked in the spot).

This optimum deployment spot will depend on the kind of parking lot that we are going to monitor. In the case of parallel parking lots the mote should be deployed below one of the car sides, as shown in figure below, while for perpendicular parking spots the most adequate place will be the one nearest to the center of the motor or the backside of the vehicle.

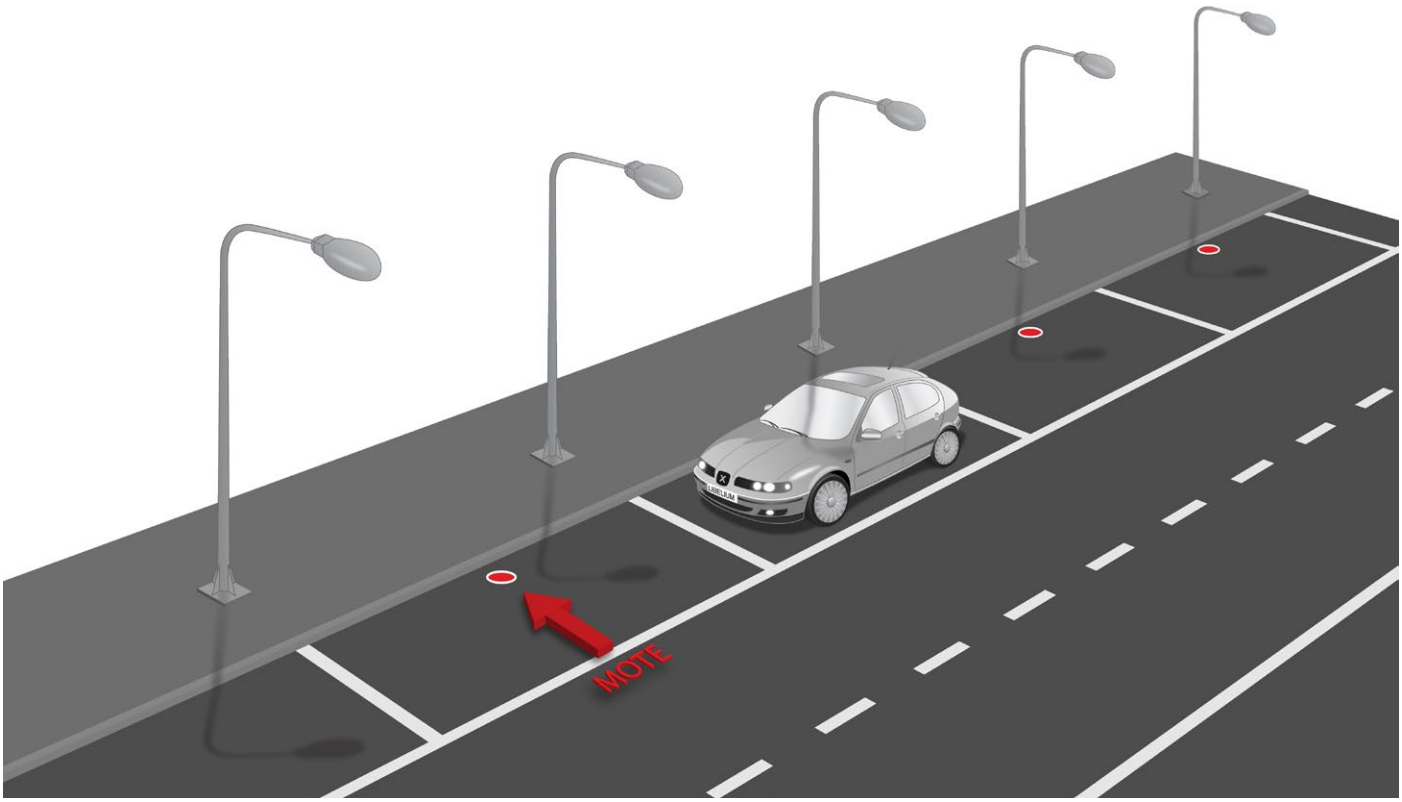


Figure : Diagram of the deployment points of the motes for parallel parking lots

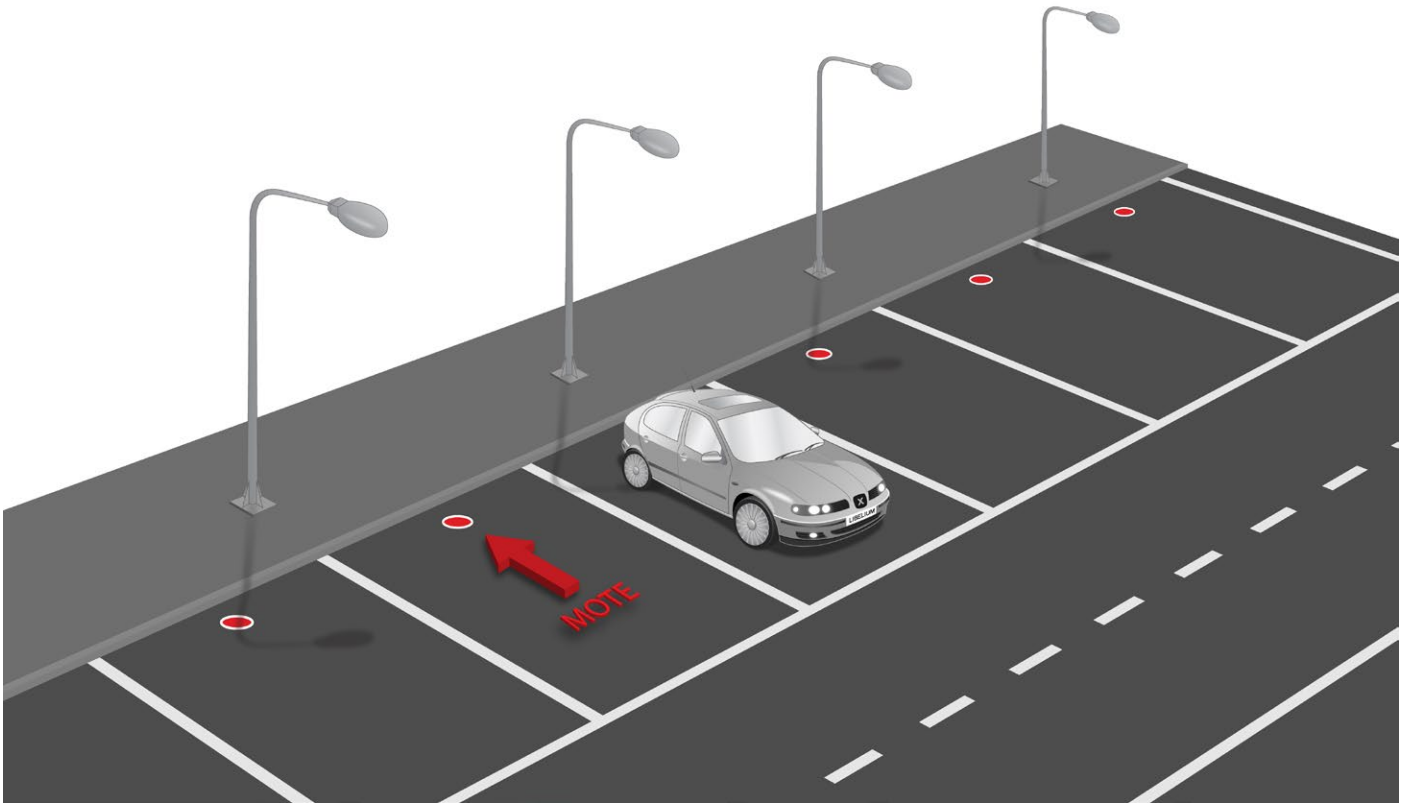


Figure : Diagram of the deployment points of the motes for perpendicular parking lots

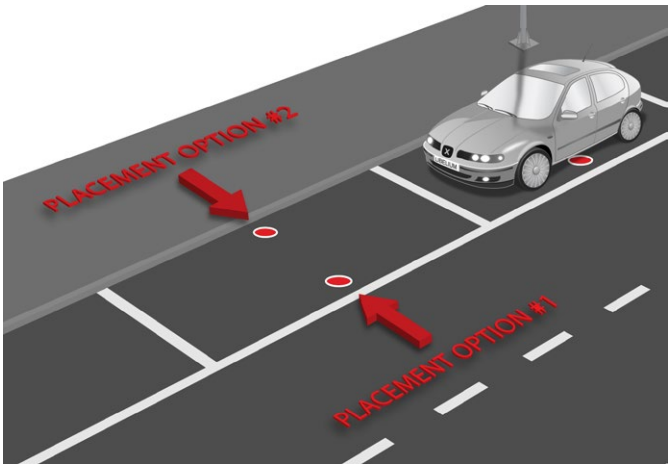


Figure : Placement options of the motes for parallel parking lots

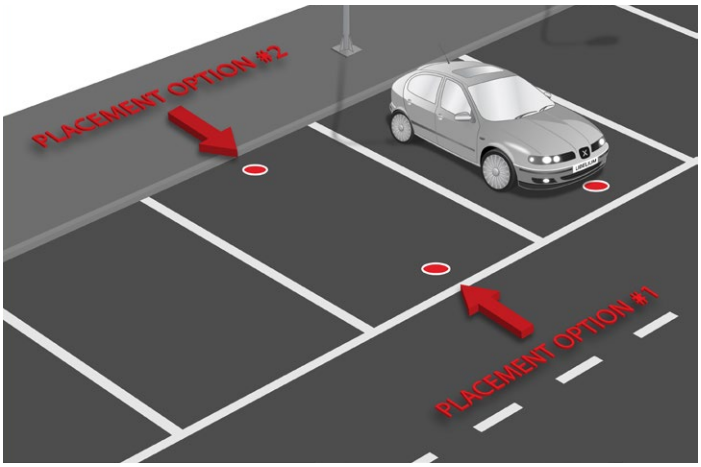


Figure : Placement options of the motes for perpendicular parking lots

Other consideration to be taken into account in mote deployment is the communication between this one and the gateway or router that will receive the data and process or redirect it. This is a very variable issue that will have to be analyzed independently for each scenario.

10.1.2. Interference of other vehicles

As pointed in section "Deployment of the motes", the presence of other vehicles in contiguous spots or near places may influence in the detection, modifying the detection threshold. This influence earns special importance in perpendicular lots, where the distance between the mote and the contiguous vehicles is shortest, and in the non-delimited parallel lots. The best way to avoid this disturbance is to take into account the state of the near spots in the detection decision when the variation in the magnetic field is very close to the detection threshold.

11. Device Installation

Important: Before deploying the nodes on the street, make sure that enough tests have been performed in order to achieve a 100% functional network and that all the necessary information related to the mote, such as identification numbers of the radios, has been compiled and stored, since once the mote is installed, the access will be very limited.

11.1. Assembly and set up

Step 1: Connect the USB Programmer to the node. Please note that an inadequate connection of the USB Programmer can damage the node.

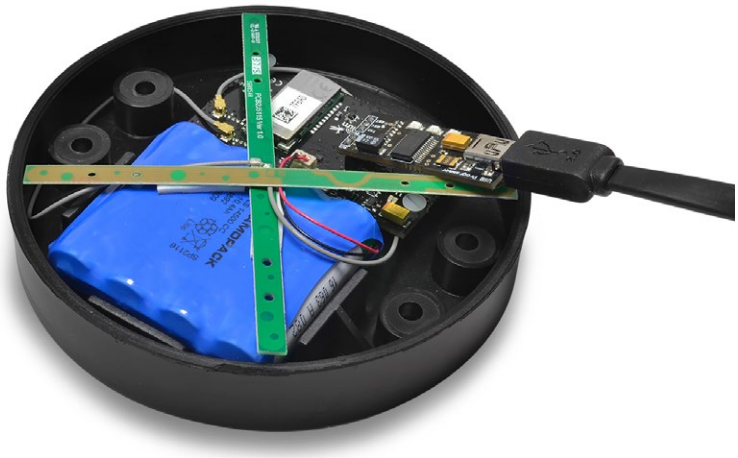


Figure : Plug & Sense! Smart Parking with the USB Programmer tool connected

Step 2: Once the parameters have been configured and the USB port selected in the Plug & Sense! Smart Parking Programmer, turn on the node and send the configuration with the button "Send to node". If the node has been programmed successfully, the next message will be shown.

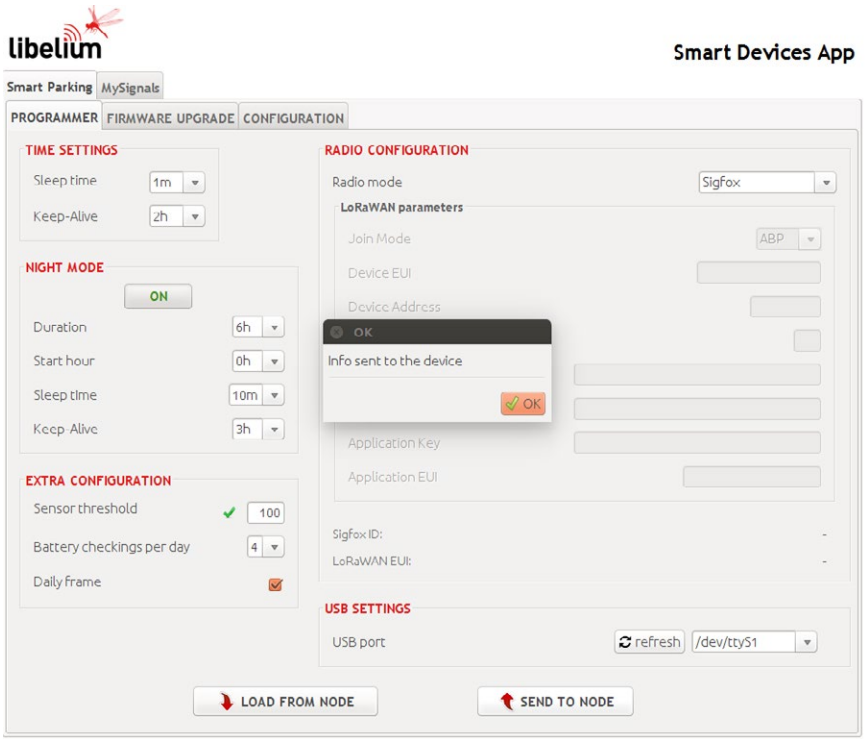


Figure : Configuration successfully programmed

Step 3: Now the node is configured. Turn off the node .

Figure : Battery connection

Step 4: Before closing the enclosure, the node must be powered on. It's mandatory that the node is powered off before this step. The node blinks the onboard LEDs (first the red LED and after the green LED) and it will go to deepsleep state. In this state the node **must not be reset with the magnetic switch before the node is deployed**. If it is reseted, the node start to work and will send frames generating an unwanted consumption. Besides, the node would perform the calibration process before it is in the real location.

Figure : LEDs blinking

Step 5: Now, the node can be closed. It is ready to deploy.

11.2. Installation and boot

Step 1: Indicate the 4 holes in the asphalt.

Figure : Indicating the holes for the node

Step 2: Drill the holes.

Figure : Drilling the holes

Step 3: Install the node in its final position and screw with the 4 special screws. Anti-vandalic screws are recommended to avoid problems: anyone could unscrew the node from the ground.

Figure : Screwing the node

Figure : The node finally screwed

Step 4: Use the magnet to reset the node. Once the node has been reset, it will start to configure the radios and get a calibration of the parking slot. In this stage the parking slot **must be** empty, so the node learns when the slot is not occupied.

Figure : Using the magnet to reset the node

Step 5: When the node gets the reference calibration, it will send two frames to the cloud: Start frame 1 and Start frame 2.

Time	Delay (s)	Header	Data / Decoding	Location	Base station	RSSI (dBm)	SNR (dB)	Freq (MHz)	Rep	Callbacks
2016-03-21 12:54:42	1.4	0000	0501011706050203020 1ff00		07B8	-133.00	14.05	868.1116	3	
					07BC	-98.00	19.95	868.1115	3	
					0B3A	-130.00	15.36	868.1125	2	
2016-03-21 12:54:34	1.3	0010	090a0de30cf9000f Temp: 24.9 °C VDD idle: 3.338 V VDD tx: 3.299 V RSSI: -85.0		0B3A	-128.00	22.61	868.1242	1	
					07B8	-134.00	11.80	868.1232	1	
					07BC	-98.00	16.55	868.1232	1	
2016-03-21 12:53:57	< 1	0000 ack required	040000190003fe71fc930000		0B3A	-132.00	12.81	868.1409	3	
					08C9	-138.00	7.30	868.1348	1	
					07B8	-136.00	9.30	868.1280	3	

Figure : Start frames received in the Sigfox backend

11.3. Configuring the parking nodes in the callback server

Customers have installed the callback server, as explained before. Now, all nodes have to be configured using the on-line form available.

Considering that a real deployment will have groups of nodes working with different configurations (for example, street #1, street #2), it is recommended that the final Remote Manager is able to generate the configuration depending on the group that each node belong to. This can be achieved if the cloud keeps record of the groups, and which nodes are inside each group.

Each backend can perform a callback. Please, see the Sigfox and LoRaWAN sections to get more information.

12. Sigfox

This section explains how to route the information received from the Sigfox platform to the back-end web server application installed,deployed and configured in section 3 generating a response when needed.

12.1. Device configuration

In first place, configuring a new Group in the Sigfox back-end is a must:

- 1. Hitting on group top menu section, the parent group it is displayed.
- 2. Then, before creating a new one, we have to be sure that it does not exist. We can't create two same groups in the same level, in this case an alert is shown while the form is saved noticing the error. If the group exists, we have to skip the rest of these steps and jump to the Device Type paragraph.
- 3. We are ready now to create the new group clicking in "New" button. In the next pop-up windows, you have to select the parent group clicking on it. Finally, fill in the form with the information required, checking the parent group written matches the one chosed in the previous step.

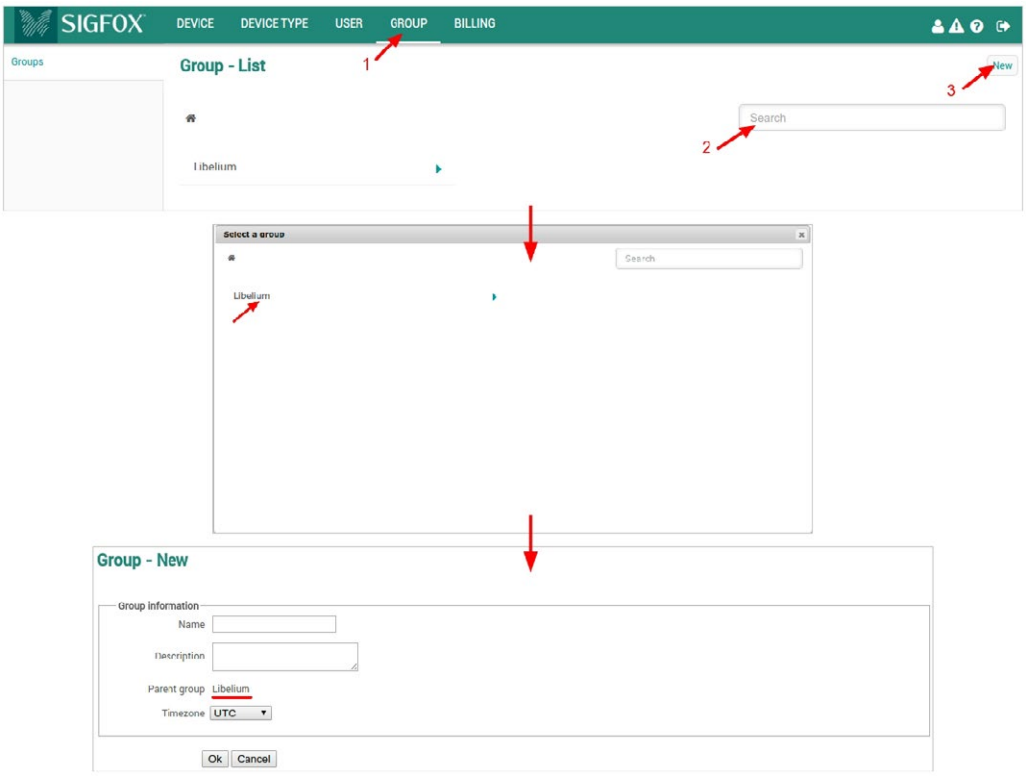


Figure : Group creation schema

Is time now to create the Device Type in the back-end, clicking on "Device Type" in the top menu and then on "New" button.

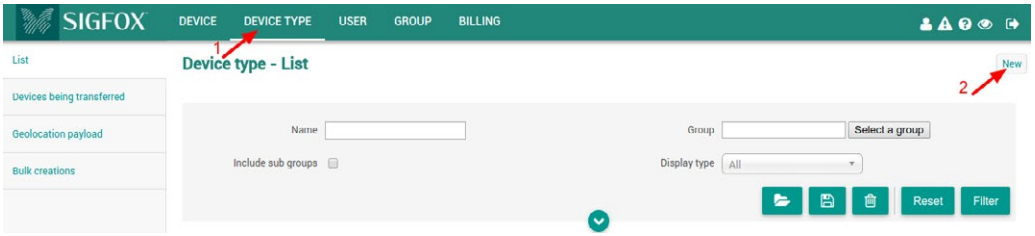


Figure : Device type creation schema

A new pop-up window will be shown to select the group where you want to create the Device Type, usually the group created previously.

After selecting the desired Group a new form will appear to fill all information required to create the Device Type. Device Types are shared for all nodes with the same functionality. Fulfil the form with the information you want, don't forget to select CALLBACK as Downlink mode.

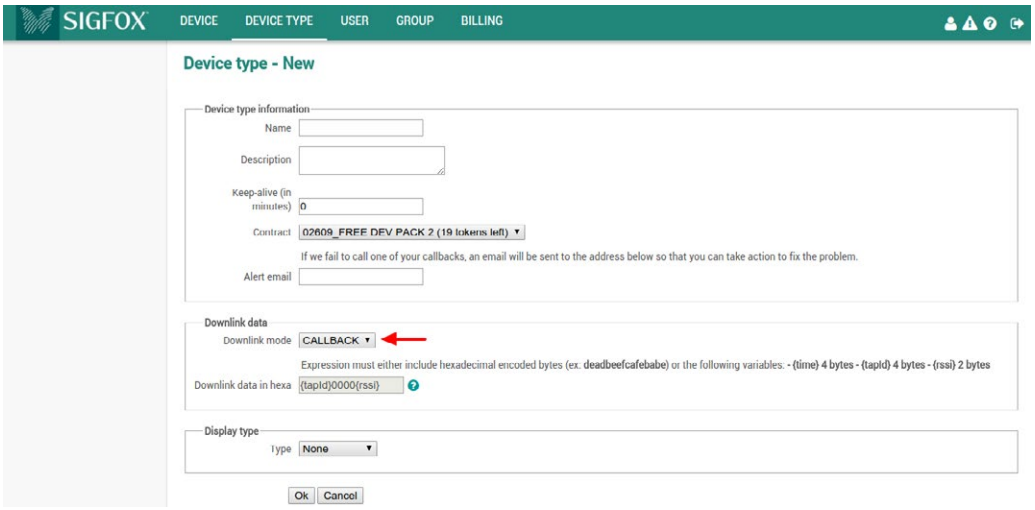


Figure : Device type form

Once created, configuring the callback is a must. Click on "Callbacks" option in the left menu. Create a new callback clicking on "New" button in the new window loaded.

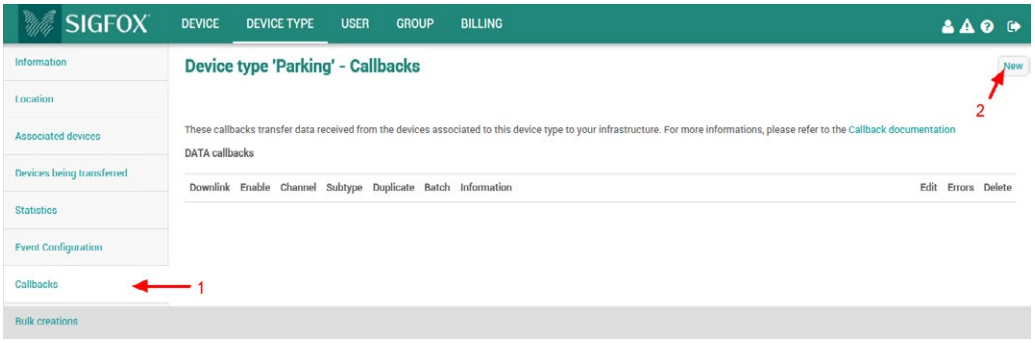


Figure : Callback creation

A new form that must be completed will be displayed as you see in the next image.

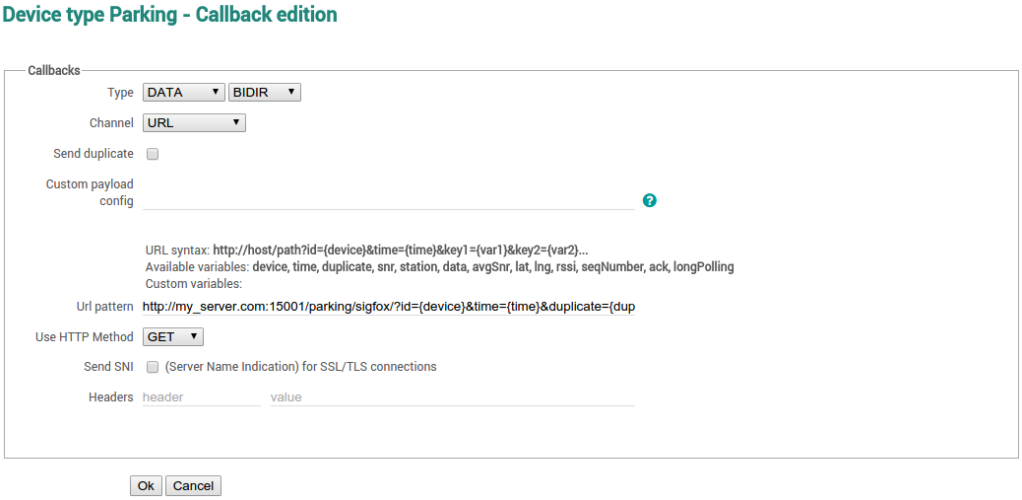


Figure : Callback form

"Url pattern" field should be filled with the URL to the back-end web server application deployed in your server following steps described in section 3, that provides access to the Sigfox service.Some extra variables has been added following the information displayed in the form, in order to send to the service as much information as possible.

```
http://my_server.com/services/sigfox/?id={device}&time={time}&duplicate={duplicate}&snr={snr}&station={station}&data={data}&avgSnr={avgSnr}&lat={lat}&lng={lng}&rssi={rssi}&seqNumber={seqNumber}&ack=true
```

The list of all callbacks available will be shown after the callback form is saved. In this list it is mandatory to enable this entry with a downlink for the service in your server receive information. Perform following 2 actions to enable SigFox downlink:

1. Activate "Downlink" column. There is a bullet in this column, and it must be activated. To activate, just click on it and leave the bullet coloured.

- ● Downlink active
- ○ Downlink inactive

2. "Enable" the callback. In the "Enable" column of the callback created, there is a check icon and you have to ensure that it is checked with a green colour.

- ☑ Callback enabled
- ☐ Callback disabled

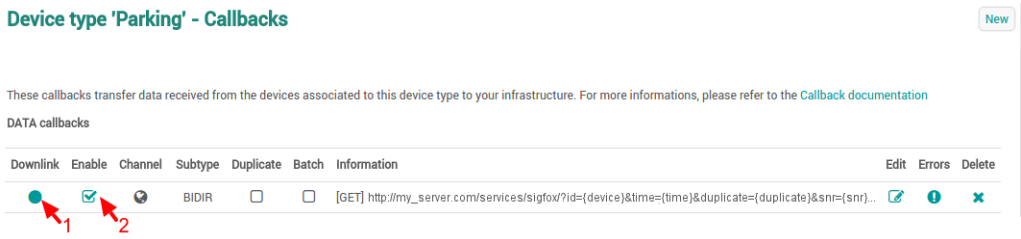


Figure : Callback list

Finally it is time for creating the device, clicking on "Device" option in the top menu and, once loaded the next window, on "New" button.

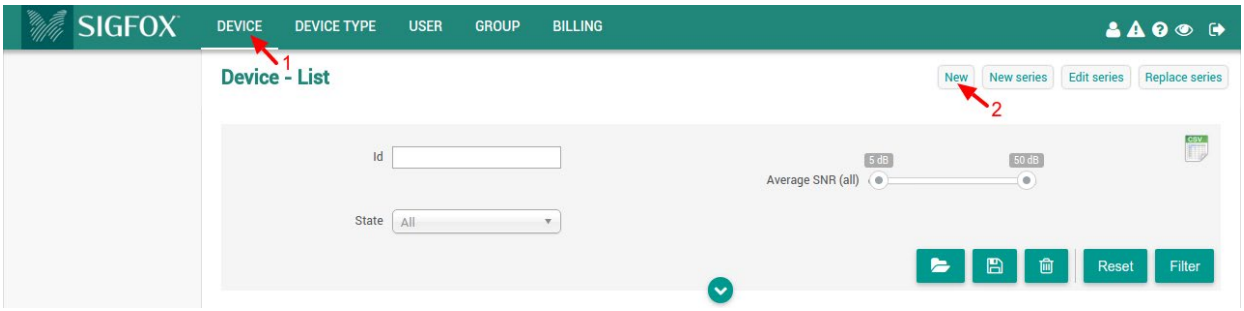


Figure : Device creation

As it happened with the Device Type, we have to select the group first after continue with the process. Then, in the new device form, we have to fill the following information:

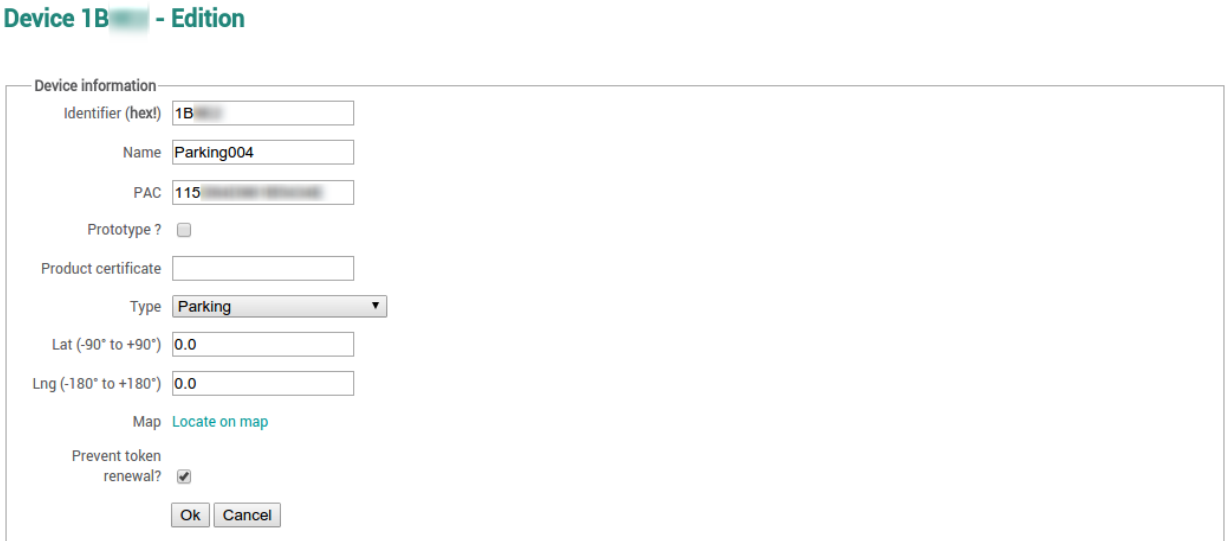


Figure : Device form

- **Identifier:** Device ID in hexadecimal format, given by the manufacturer.
- **Name:** Device name. Use a descriptive name in order to identify your device easily.
- **PAC:** Porting Authorization Code (PAC) is a unique hexadecimal number to identify the device regardless of the network. This code is given by the device manufacturer.
- **Product certificate:** Leave in blank.
- **Type:** We have to look for in this drop-down the device type created before and select it.
- **Lat (-90º to +90º) / Lng (-180º to +180º) / Map:** You can grab the coordinates of your node if you want using these three fields.
- **Prevent token renewal?:** Leave unchecked.

12.2. Server configuration

Configuration parameters to be modified in order to be able to communicate to the node and run the whole system.

The `services.ini` file, located in `data` folder, has to be updated with the following information in the `sigfox` section.

```
[sigfox]
log_level = “ALL”
log_file = “../../logs/sigfox.log”
```

- **log_level:** This level is the minimum level to save logs in the system. Select among these levels:
  - OFF: This option deactivate the log.
  - ERROR : It only reports ERROR messages.
  - INFO: It reports ERROR + INFO messages.
  - DEBUG: it reports ERROR + INFO + DEBUG messages.
  - ALL: It reports everything happened in the process.
- **log_file:** The relative path where the log messages will be saved.

13. Loriot

This section explains how to route the information received from the Loriot platform to the back-end web server application installed,deployed and configured in section 3, generating a response when needed.

13.1. Device configuration

Create a new application in the dashboard after log in the Loriot dashboard.

Figure : Create an application

In this application all devices have to be added clicking on "Devices" left menu option and the device list will be displayed. Click on "Generate new device" button for adding a new device.The new one will be created in the end of the list. Click on it to get more information.

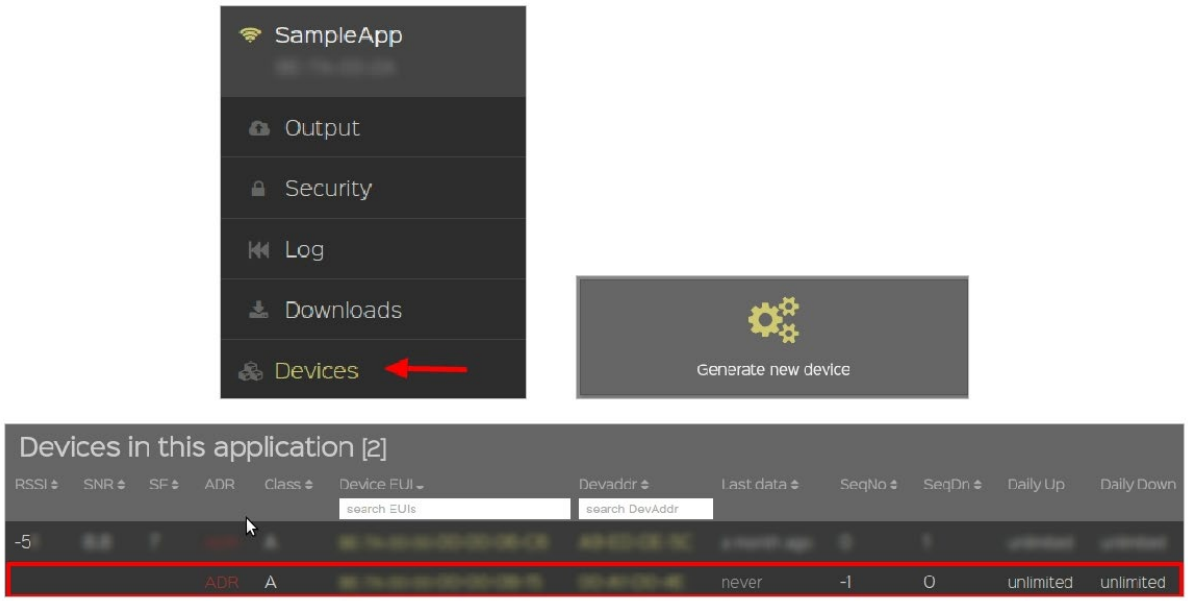


Figure : Device creation schema

Data output section is the responsible to stablish the communication between the platform and the customer server. The configuration is in the main window of the application created, clicking on our application name in the left menu, the first right block "Network Application" contains the "Data output" option.

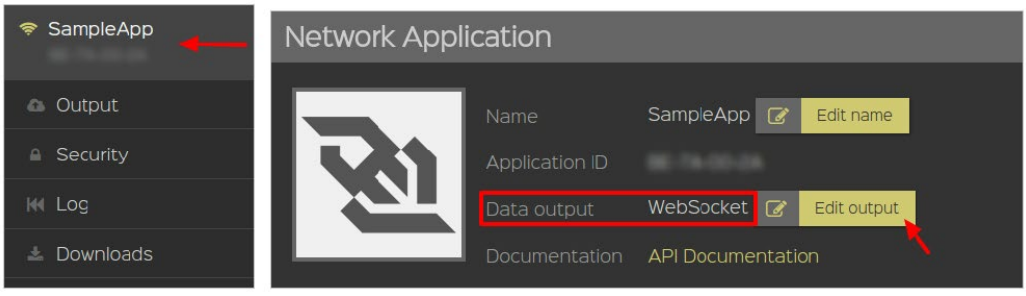


Figure : Output application selection

Clicking on "Edit output", all information available about the selected output option will be displayed. To change the data output click on "Change" button in the detailed window and select one of the multiple choices from the emergent list.

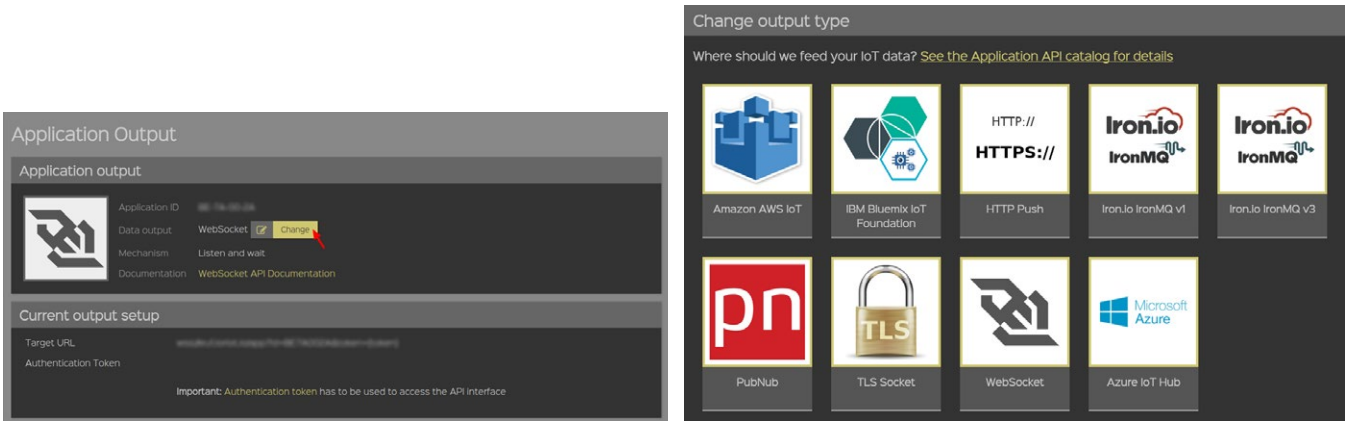


Figure : Selecting the output application

For this project, we have chosen "Websocket" because it is the preferred method for the company to deliver the end-device data.

- It's cloud-friendly (existing HTTP traffic optimization can be used)
- It's bi-directional, real-time interface
- It's easy to implement
- It's lower overhead compared to REST
- It's already supported in all major web browsers
- It's already supported in many programming languages

13.2. Server configuration

Configuration parameters to be modified in order to be able to communicate to the node and run the whole system.

The `services.ini` file, located in `data` folder, has to be updated with the following information in the `loriot` section.

```
[loriot]
log_level = "ALL"
log_file = "../../logs/loriot.log"
websocket_url = "wss://eu1.loriot.io/app?id=xxxxxxx&token=xxx-xxxx"
service_url = "http://my-server.com/services/loriot/"
```

- **log_level:** This level is the minimum level to save logs in the system. Select among these levels:
  - OFF: This option deactivate the log.
  - ERROR : It only reports ERROR messages.
  - INFO: It reports ERROR + INFO messages.
  - DEBUG: it reports ERROR + INFO + DEBUG messages.
  - ALL: It reports everything happened in the process.
- **log_file:** The relative path where the log messages will be saved.
- **websocket_url:** The URL of the websocket listed in the Loriot dashboard. In the last window shown "Application output", in the second section "Current output setup" there are the target URL and the Authentication Token to access to the API interface.

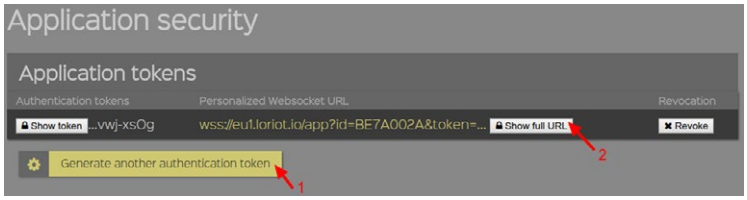


Figure : Output setup

Clicking in "Authentication token" link, a new window will be shown with the tokens generated in the Loriot application.

To create a new one, just click on "Generate another authentication token" button, and a new record will be created in the table. Finally, copy the full URL hitting on "Show full URL" and paste the URL in this parameter of the service configuration file.

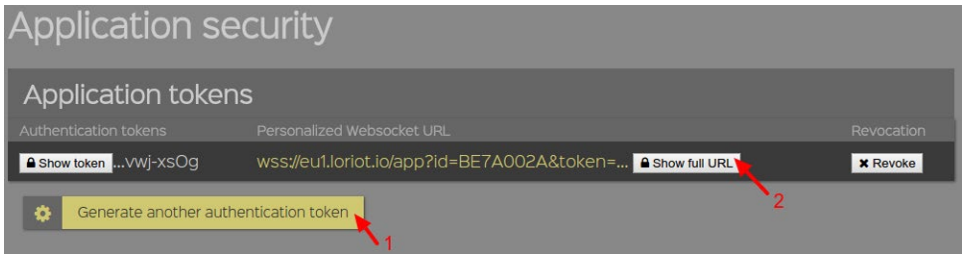


Figure : Generating an application token

- **service_url:** The URL to access to the Loriot service in your server previously configured. (http://my_server.com/services/loriot/)

13.3. Start the web-socket connection

This platform requires an additional step, starting the process that establish the connection to the Loriot server in order to receive and send information in all nodes.

Open a SSH connection to the web server in a terminal, navigate to the Loriot service folder, in your_`web_server_path/services/loriot`. Execute this command:

```
nohup nodejs websocket.js >>../../logs/loriot.log 2>&1 &
```

to create the web-socket communication. Kill this process to close the web-socket communication channel.

14. Actility

This section explains how to route the information received from the Actility platform to the customer server and generate a response if it is needed.

14.1. Device configuration

Firstly, it is necessary to configure a new AS routing profile in the Device Manager clicking on "AS routing profiles" on the left sidebar menu.

Figure : AS Routing profiles menu

A list will be displayed with all existing AS routing profiles. Below, in the second section, New AS routing profile gives the capability to add new AS routing profiles. Clicking on Add button to create a new AS Routing profile.

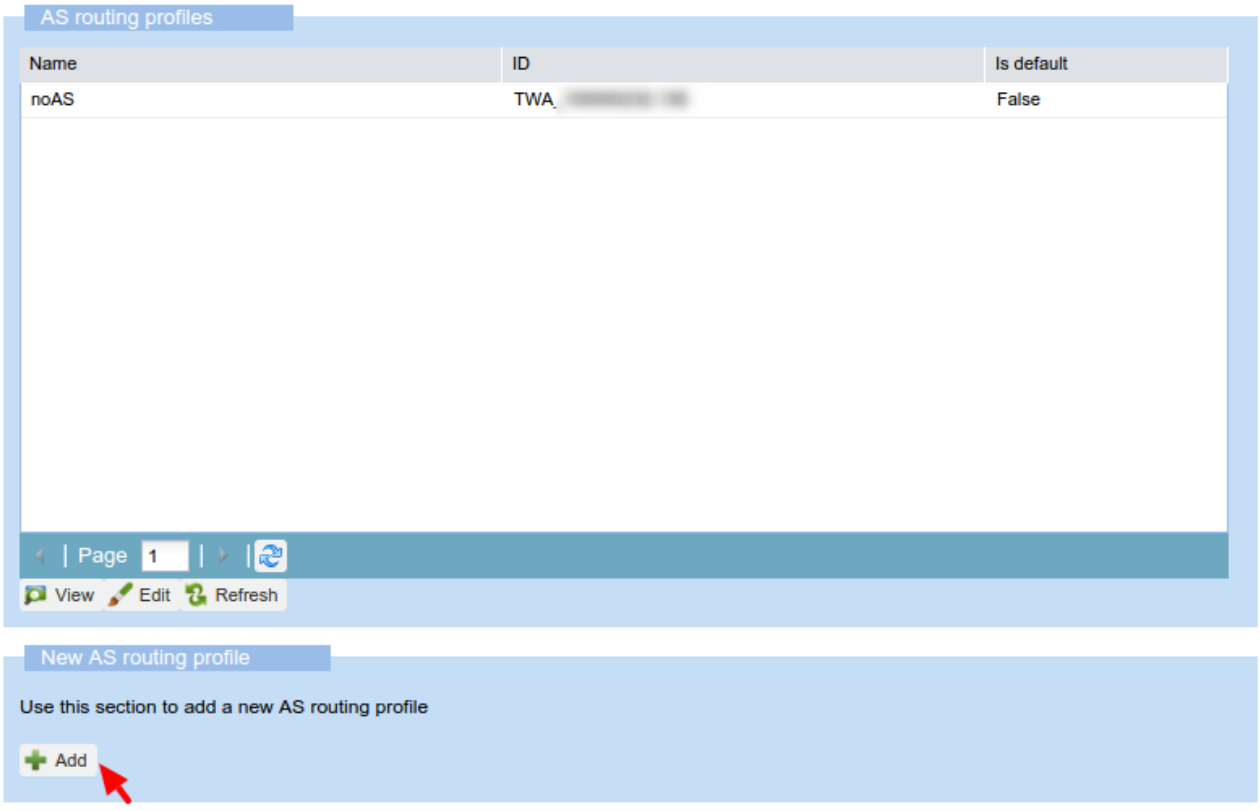


Figure : AS Routing profiles

A name must be typed in the new window displayed. Clicking on Create button to continue the process.

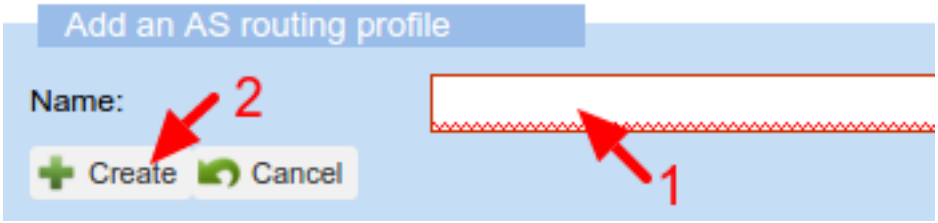


Figure : AS Routing profile name

In the new window, it is necessary to mark the check "Is default" and click the "Add button" in the Add a route section.

Figure : AS Routing profile route

A new section "Route" will appear with a default route created. The name will be always "bigONG1". Only our destination URL must be in this table. The first step will be delete the default route, clicking on the row and then on the "Delete button"

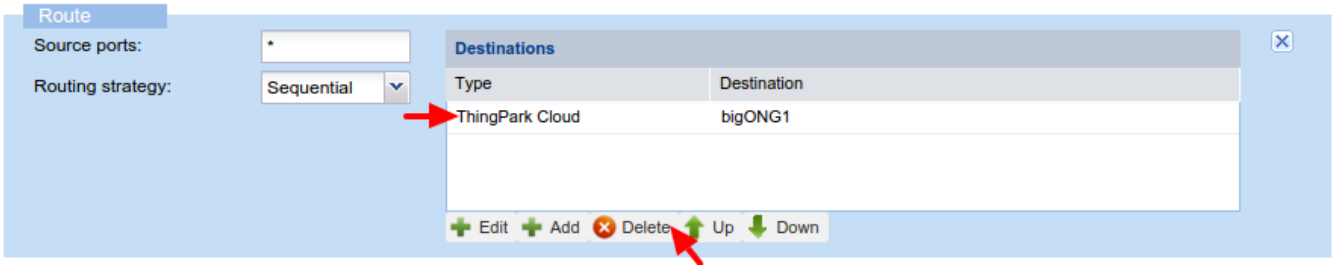


Figure : Add a route

Once deleted, it is time to create the destination URL.

- 1. Hitting on "Add" button a new pop up window will be displayed.
- 2. In "Type" field "Third party AS (HTTP)" must be selected.
- 3. In "Destination" field, the URL to access to the Actility service in your server previously configured has to be typed. (http://my_server.com/services/actility/)
- 4. Finally, hitting on "Add" button, the configuration will be saved.

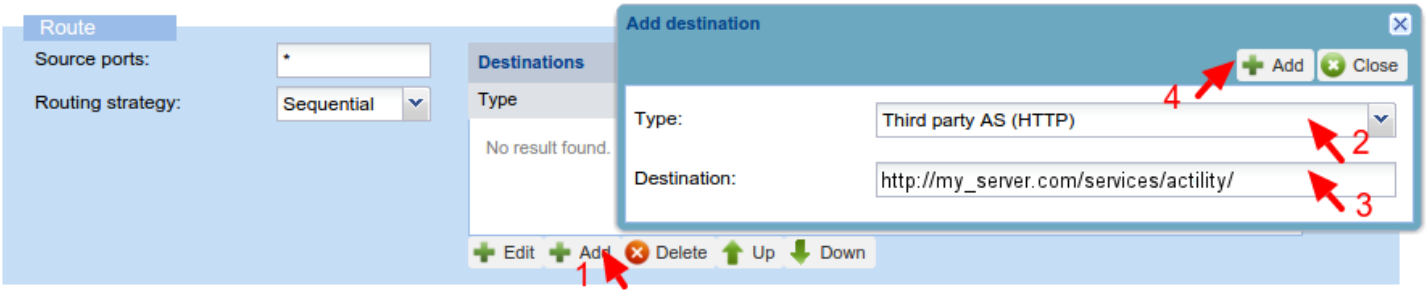


Figure : Route destination

Once saved, the configuration will appear in the list. The last step is creating your device and associate it to the AS routing created. You have to right-click on "Devices" option in the left menu, and then hit on "Create device".

It is necessary to fill in all compulsory fields in the form displayed in the new window, according to the values in the Libelium Smart Devices App. Remember to select your AS routing in "AS routing profile"

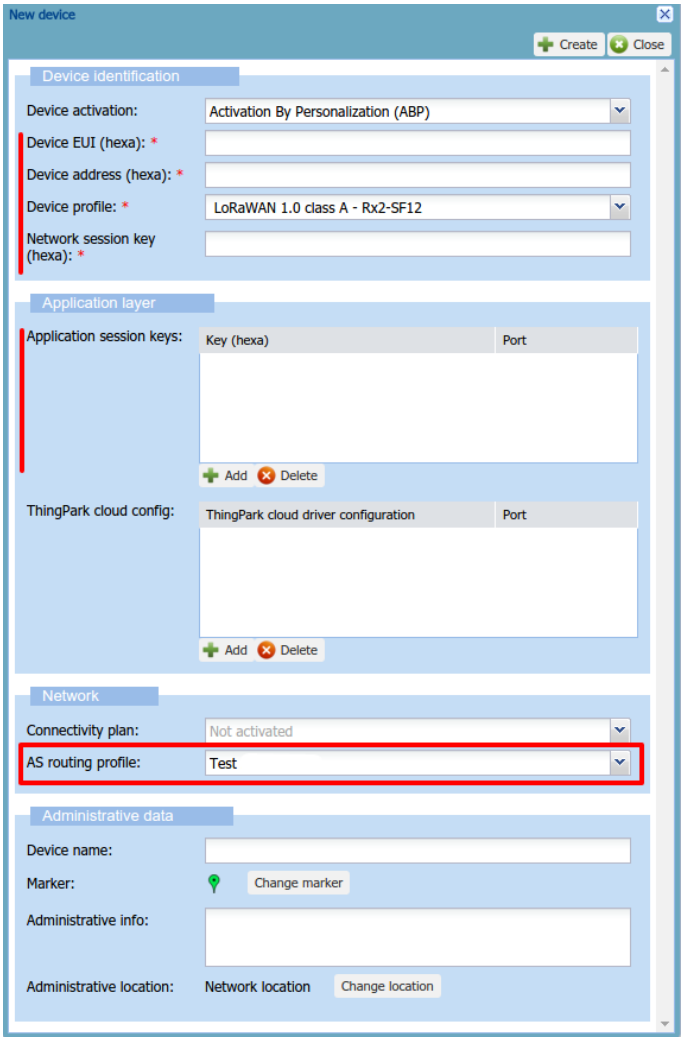


Figure : Create a new device

On the other hand, if the device is created, changing the network routing to the AS Routing profile is quite easy. First, select a device from the list and hit on "Edit" button. Then click on Network section in the left menu.

Figure : Steps to follow when the device is created

In the "Network/cloud routing" section, the new AS Routing profile created must be selected clicking on "Change" button and then selecting the profile created from the dropdown box displayed in the new pop-up window. Finally, click on "Save" button.

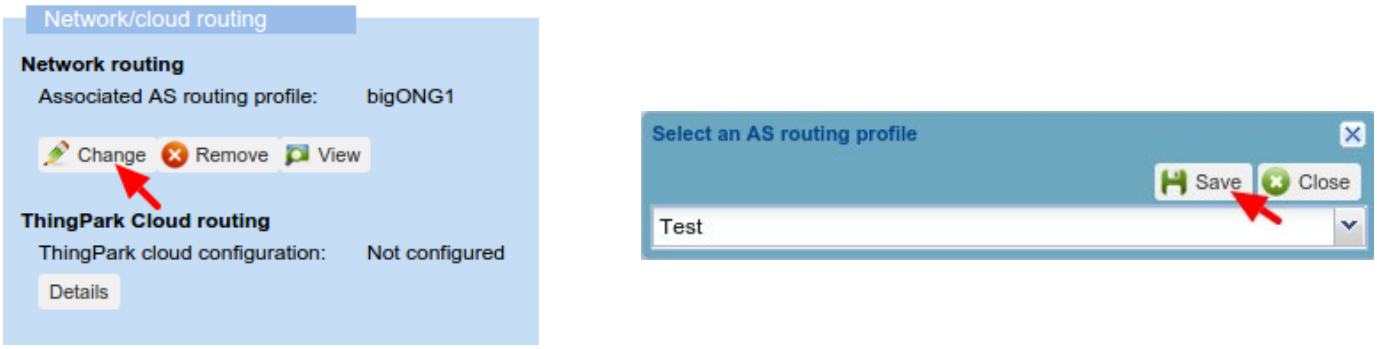


Figure : Select the AS Routing created

14.2. Server configuration

In the customer server installed, users have to configure some parameters in order to be able to communicate to the node and run the whole system.

The services.ini file, located in data folder, has to be updated with the following information in the actility section.

```
[actility]
log_level = "ALL"
log_file = "../../logs/actility.log"
server_url = "http://lrc99.thingpark.com:8807/sensor/"
```

- **log_level:** This level is the minimum level to save logs in the system. Users can select one among these levels:
  - OFF: This option deactivate the log.
  - ERROR : It only reports ERROR messages.
  - INFO: It reports ERROR + INFO messages.
  - DEBUG: it reports ERROR + INFO + DEBUG messages.
  - ALL: It reports everything happened in the process.
- **log_file:** The relative path where the log messages will be saved.
- **server_url:** The downlink URL that Actility gives to send information from the server to the node. This URL will be the address of the primary Actility LRC cluster.

15. Troubleshooting

15.1. Windows does not recognize USB ports

Sometimes it is possible that your computer does not recognize the USB port where the board is plugged. The reason could be that the driver for the device is missing in your Operating System. Proceed as described to fix this issue.

Open the Device Manager in order to see what device is not being detected. There are several ways to open this window depending on the Windows version installed, usually typing "Device Manager" in the Start > Search option works.

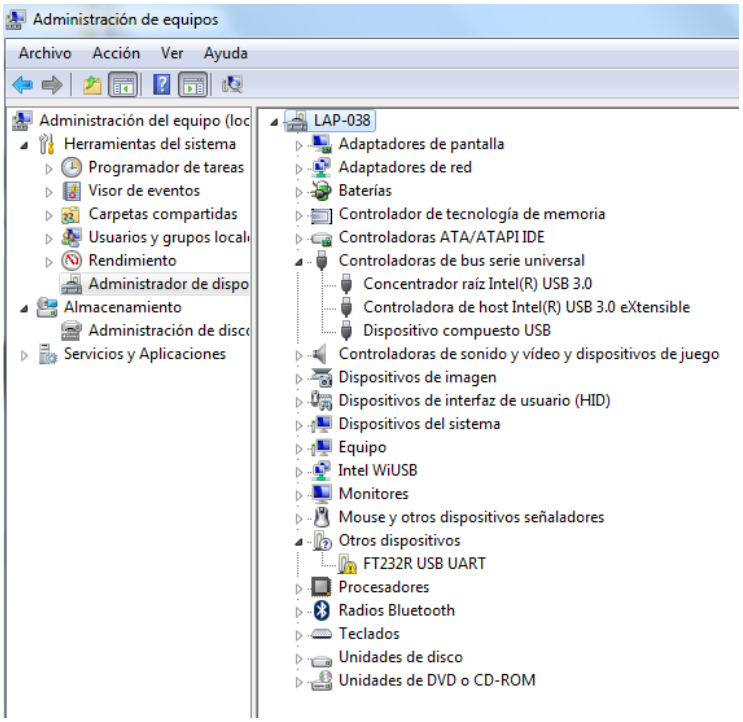


Figure : Windows Device Manager

The devices that are not detected are always marked with an alert icon as the next image shows. Right-click and select the first choice, "Update Driver Software".

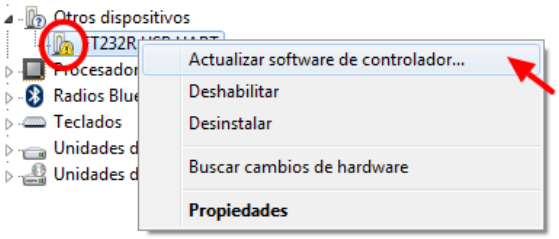


Figure : Update driver

Then, the driver must be searched in the computer, in particular the same path where the Smart Devices App has been installed.

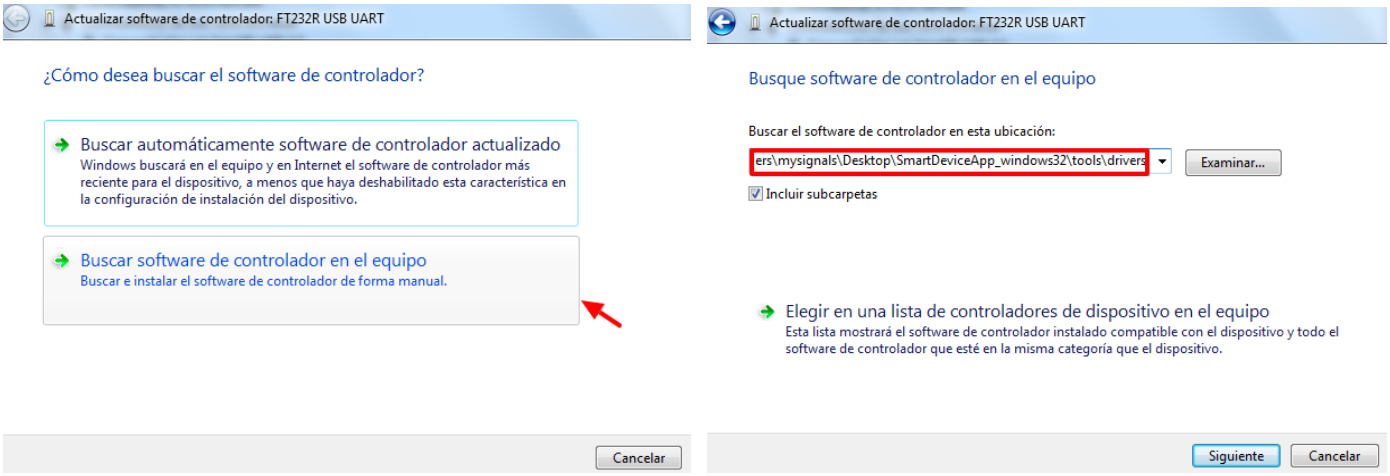


Figure : Search the driver in the computer

Once validated the path, the next pop-up will be displayed, noticing the driver is not verified. Users have to confirm clicking on "Install this driver software anyway".

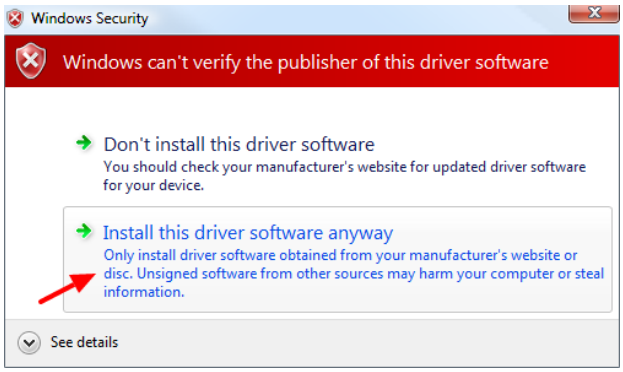


Figure : Security warning

After a while, the driver is installed.

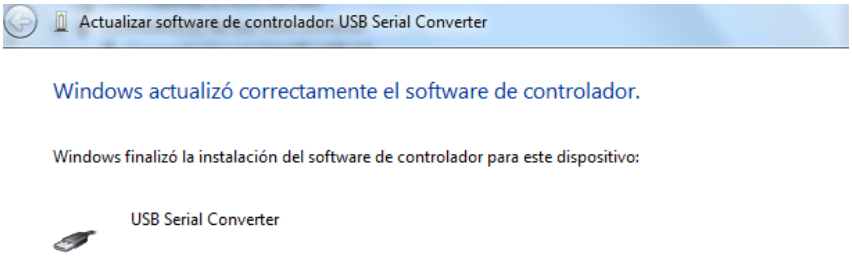


Figure : Driver installed

The process is not finished. The first driver installed is "USB Serial Converter",check the Device Manager list to verify the status. If the warning remains near the USB serial port, the "Update driver software"process must be repeated again.

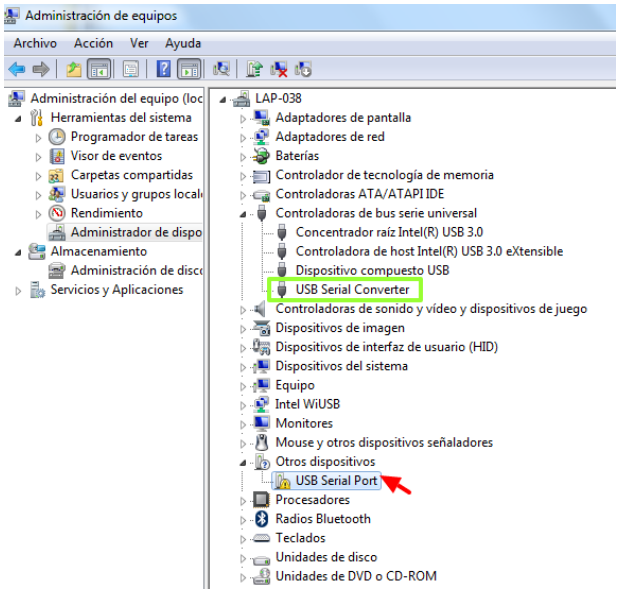


Figure : USB devices to verify

When the process is finished, check that both drivers have been installed.

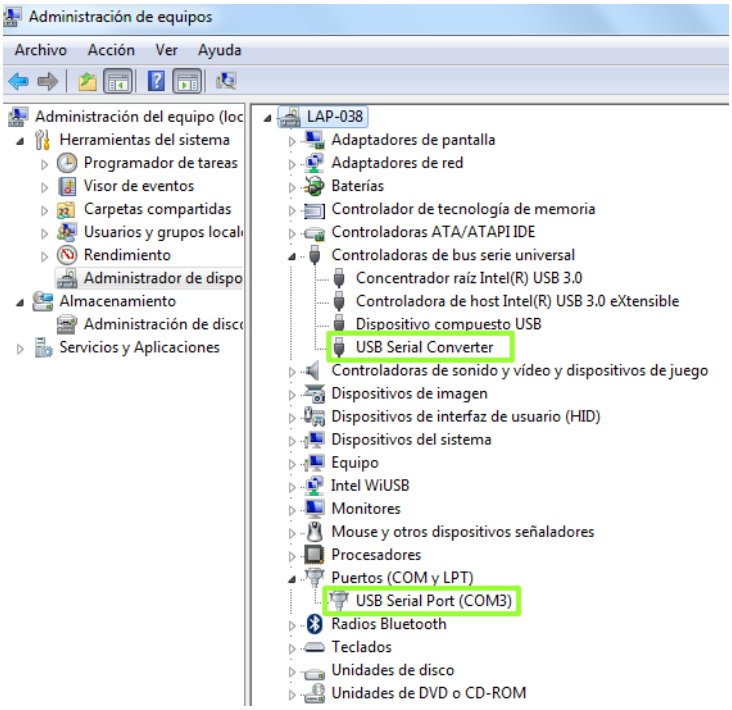


Figure : USB devices with all drivers installed

Now, the Smart Device App is ready to detect the ports and make the operation you want to do.

16. Certifications

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

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

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

www.libelium.com/products/meshlium

List of certifications for Plug & Sense! and Meshlium:

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

Figure : Certifications of the Plug & Sense! product line

You can find all the certification documents at:

www.libelium.com/certifications

17. Disposal and recycling

In this section, the term "Waspmote" encompasses both the Waspmote device itself as well as it enclosure.

When Waspmote reaches the end of its useful life, it must be taken to an electronic equipment recycling point.

The equipment must be disposed of in a selective waste collection system, and not that for urban solid residue. Please manage its disposal properly.

Your distributor will inform you about the most appropriate and environmentally friendly disposal process for the used product and its packaging.