TÉCNICO LISBOA

RoCKIn @ Home

Home Automated Devices - User Manual

João Mendes

Grant Report
Fev 2016

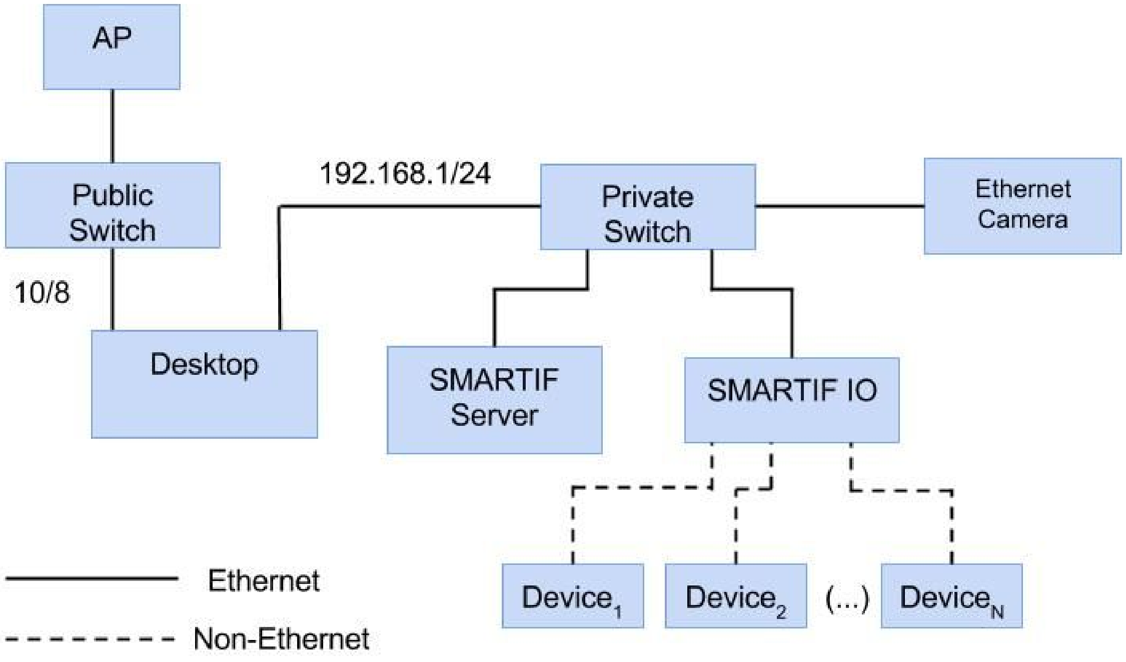
Introduction:

This document overviews the hardware and software of the network devices used during the RoCKIn project. The report contains an overview of the hardware used, its configuration, how to recover from a system failure and a guide through for a new and clean installation.

Everyone that is in charge or simply using the network devices should read this guide as it will introduce you to the system's characteristics and point out possible failures.

Hardware:

The @Home Testbed at IST is equipped with network devices capable of opening/closing the blind, turning on/off the lamps, and recognize the doorbell signal. The network is organized as shown in the figure below, followed by a description of each block. 10/8 is considered to be a network with netmask 255.0.0.0, while 192.168.1/24 a network with 255.255.255.0.



- Desktop: A computer used to operate the switches and the RefBox. This PC is equipped with two ethernet cards in order to communicate with both the networks with ease. The IP should be 192.168.1.1 and 10.0.0.1

- Private Switch: An ethernet switch used to connect all the devices (IPs: 192.168.1.56 for the SMARTIF server; 192.168.1.45 for IO; 192.168.1.100 for the Desktop; and 192.168.1.60 for the camera)
- Public switch: An ethernet switch that is open to the teams. Distributes the same network as the AP.
- AP: An Access Point where the robot is supposed to connect. This is the only connection between the robots and the network. Acts as a bridge between WLAN and LAN. The Access Points used work in Dual-band Standalone 802.11a/g/n. The models used are Cisco AIR - AP1042N-E-K9 16.
- Ethernet Camera: Perspective camera facing the Outside Hallway. The camera can have its parameters (frame rate, resolution, color gains) changed over ethernet and it is not motor controlled (no pan-tilt). The camera is an Axis P1344 and is independent of the network devices. Can be accessed as a normal IP camera. (IP 192.168.1.2)
- Devices: Different devices may exist in the house. In our test bed the devices are: a motor to control the window blinds (2 relays, 1 up and 1 down), 3 controlled power plugs (through a relay), 1 light dimmer, and 1 door bell button.
- SMARTIF IO: This module controls the different devices/sensors existing in the house. It is prepared to add more devices in case of need.
- SMARTIF Server: Device responsible for the communication between the SMARTIF IO mentioned above and the network. It is also where the system configurations (through the "SMARTIF Config Tool") are stored and changed. Technical details regarding SMARTIF products can be found at the official site.

As for the network configuration, we used the general ISR network. Considering the figure above, both the AP and the public switch are part of the ISR normal network. The private switch is to decouple the networks and to allow this network to work as stand alone without knowledge of the public network. This way the “Desktop” can even access a public network with DHCP and maintain all the configurations regarding the switches. This layout is even more useful since is a replica of the one designed for the competition RoCKin@Home2014. By having the exact same structure it is possible to also use the RefBox (designed for the competition) as a complete mock-up to test the robots in a competition-like scenario.

It is advisable to perform the common tests (ping, nmap, etc) to infer if all the IPs are set correctly.

Hardware Assembly:

The hardware deployment can be done with the aid of SMARTIF provided guide (in the attached folder). Here we can find the wiring that needs to be done for all the typical sensors/actuators. For the server and IO, independent power sources were provided. In the power layout used in the testbed, one of this power supplies also powers up the ethernet switch. Bear in mind that changing in the polarity of the IO and/or the devices/switches will most likely damage the equipment since it is not protected against this situations.

An important fact not mentioned in the provided documentation is that the inputs are, as seen in the images, current sinks. The problem is that they were not built to endure long time exposure. The need to use momentary switches (i.e., non-latching) is advised by SMARTIF. Although at the moment the only input the set-up has is the doorbell (non-latching), if one wants to add more switches please comply with the mentioned.

Another possibly dangerous fact is the heat dissipation of the dimmers. The dimmers tend to get extremely hot. If one needs to change or manually operate them, power it off some time before the operation.

After the electrical wiring one should proceed to the configuration of the devices.

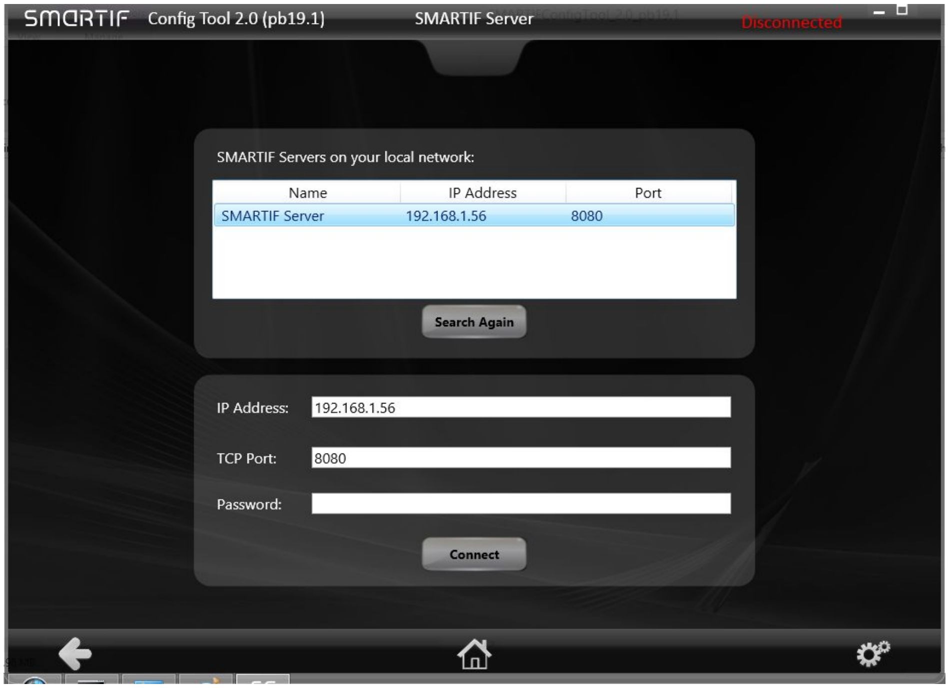
Hardware configuration:

Apart from hardware assembly, the system also needs to be configured. To do so, a configuration tool was provided by SMARTIF. This tool can only be run in windows (.exe) and allows a straightforward configuration of the devices, switches and actions. As an example: we can initialize a switch (input device); an output relay (that can be a light); and an action for this pair (on/off). This actions can change according to the input/output pair (eg: a blind is able to go up/down instead of on/off). As extra features it is possible to: define areas of the house, useful if the robots need to turn off all the lights in a pre-defined area of the house; save and upload the project, in order to use the same configuration in a different SMARTIF Server with ease; export of the configuration to a CSV file.

A guide through a simple configuration is now provided. We will be use as an example the addition of a light, a switch, and the action that will state what will happen when the switch is pressed.

These are the needed steps:

- Assure that the computer is on the same network as the SMARTIF network.
- Open the exe. If you are able to ping the SMARTIO, the tool should detect the server right away. Connect to it.



SMARTIF Servers on your local network:

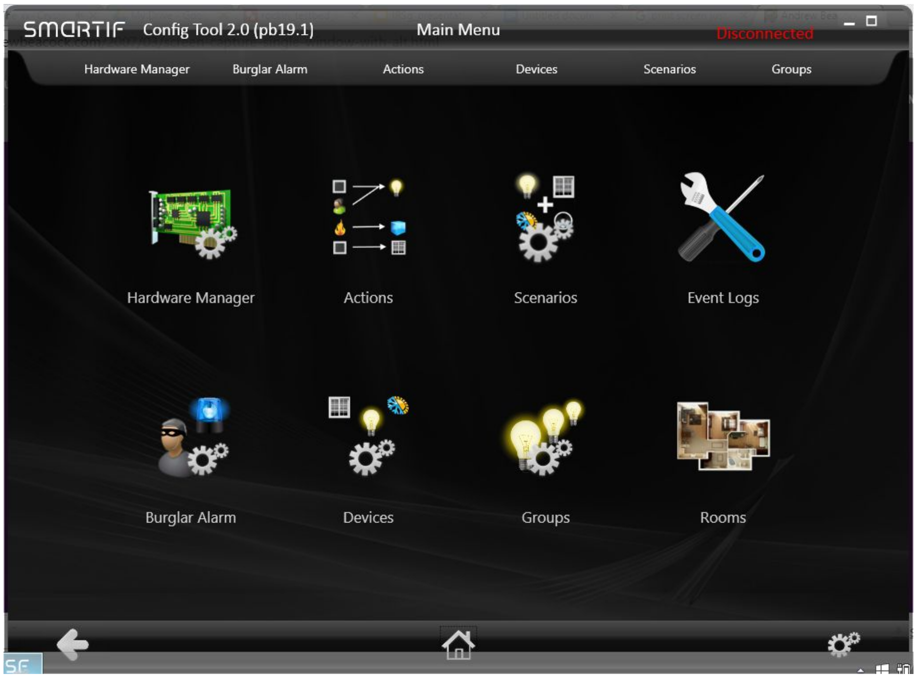
Name	IP Address	Port
SMARTIF Server	192.168.1.56	8080

IP Address: 192.168.1.56

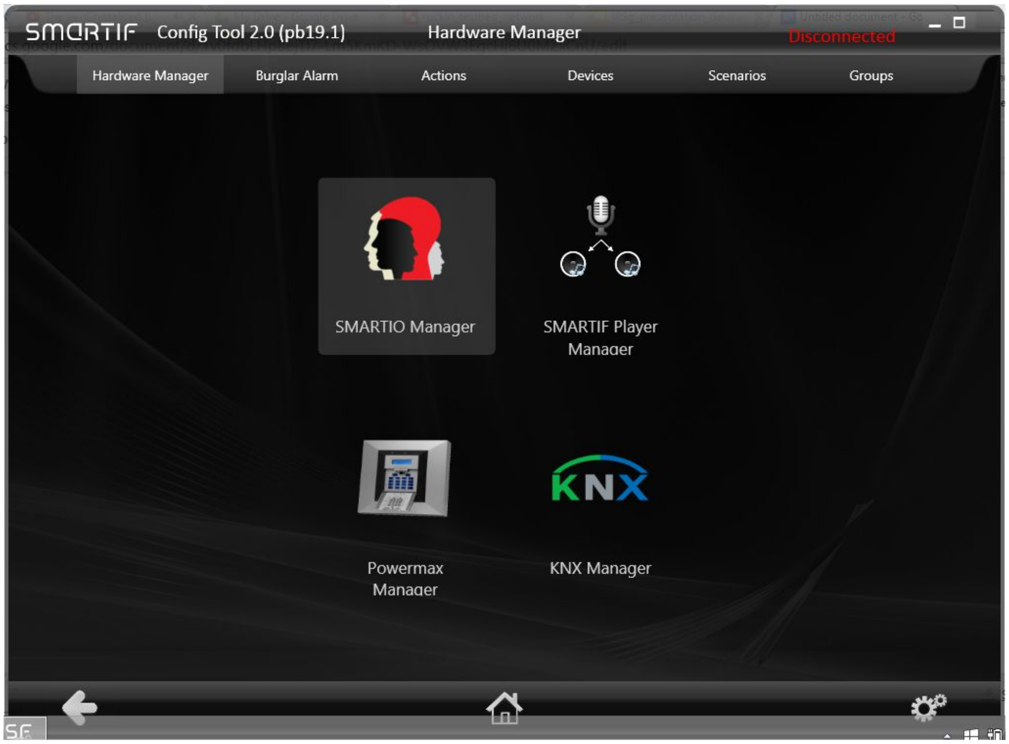
TCP Port: 8080

Password:

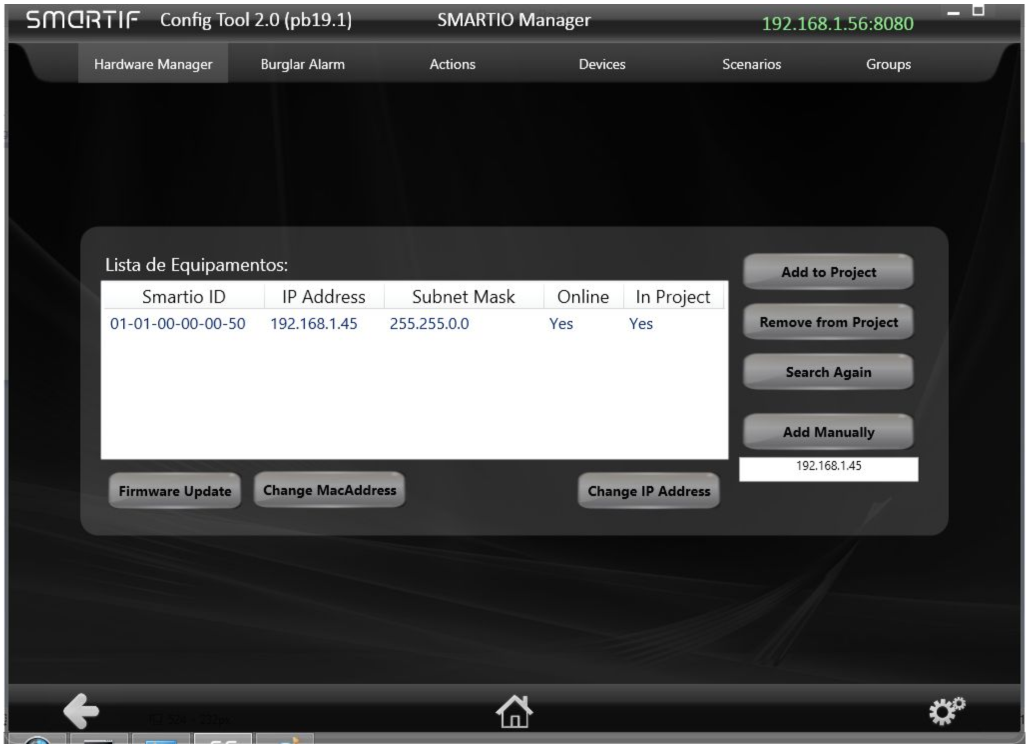
- After this step, the following menu should appear:



- Click: Hardware Manager;



- Click SMARTIO manager; Select the IO; add to project. This step is only necessary the first time. If the IO is already in the project the step is not needed.

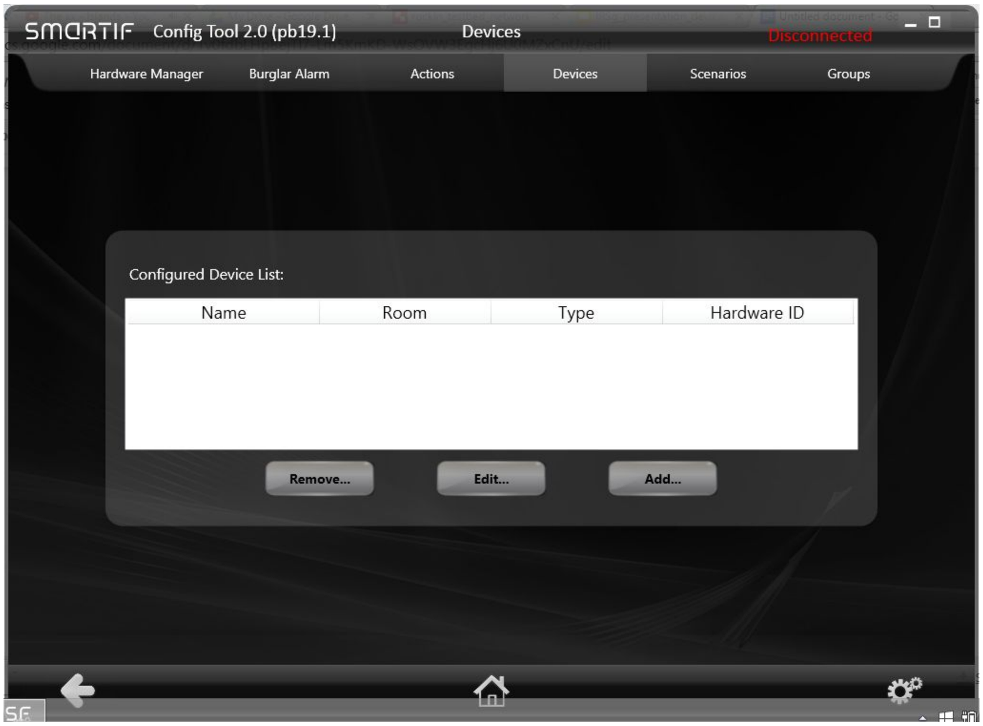


Lista de Equipamentos:

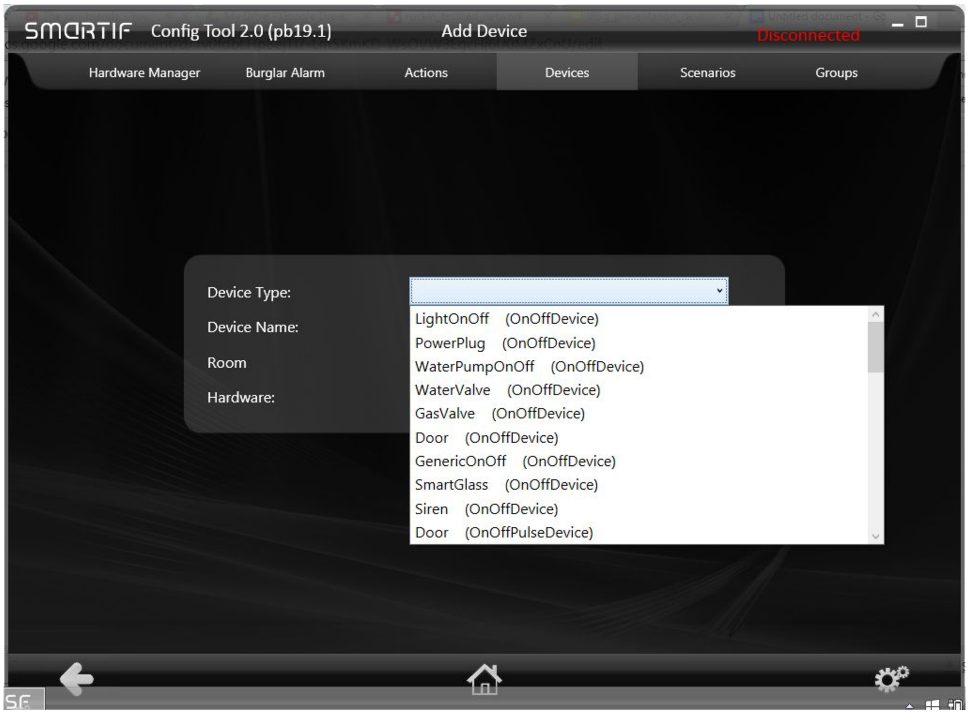
Smartio ID	IP Address	Subnet Mask	Online	In Project
01-01-00-00-00-50	192.168.1.45	255.255.0.0	Yes	Yes

192.168.1.45

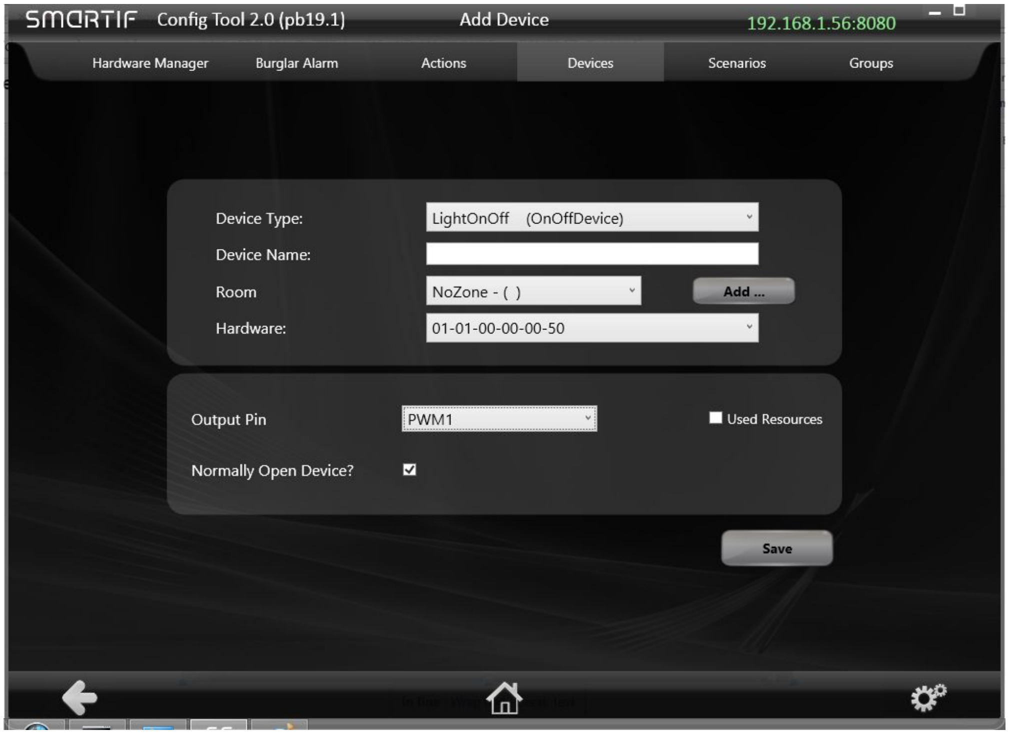
- Click: Devices;



- Select add. In all the drop down menus you be able to see the available options.



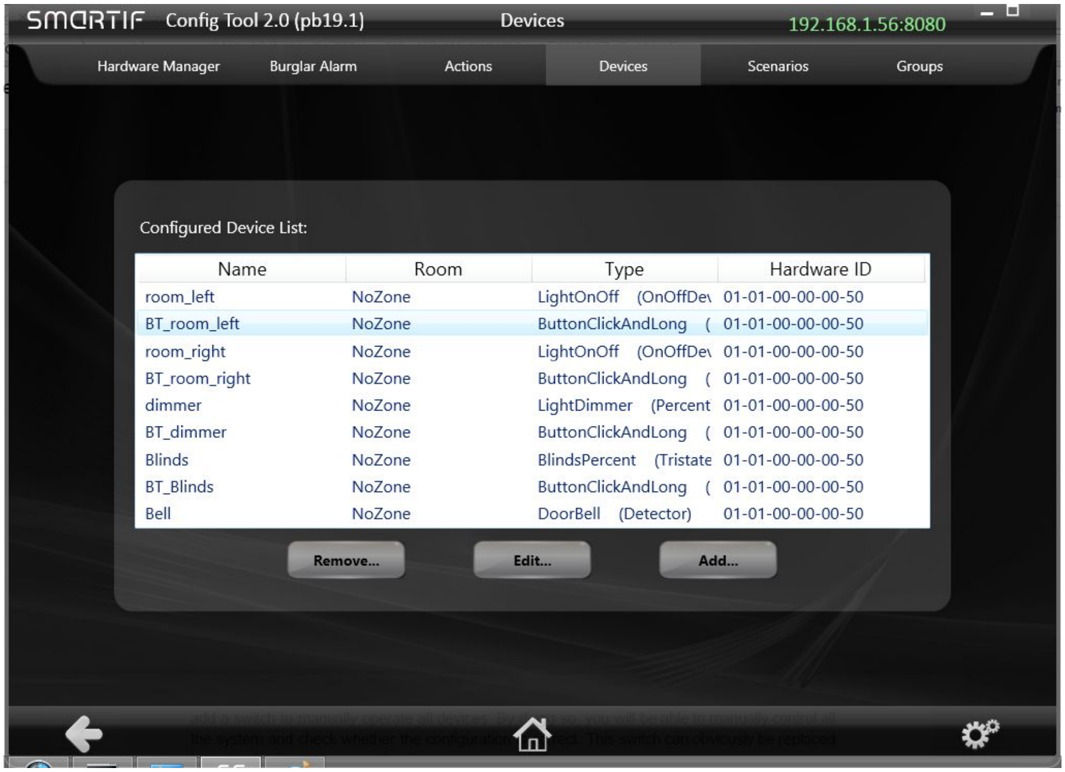
- Select LightOnOff (on “device type” drop menu); name it; choose the IO (on “Hardware” drop menu); select the port where the light was physically wired. Press save.



SMARTIF Config Tool 2.0 (pb19.1) Add Device 192.168.1.56:8080

Device Type: LightOnOff (OnOffDevice)
Device Name:
Room NoZone - ( )
Hardware: 01-01-00-00-00-50
Output Pin PWM1 □ Used Resources
Normally Open Device? ☑

- After this last step, the light will have been created. Simultaneously (and automatically) a switch to operate this light will have been created. By checking the “Action” tab the user will notice that an action was also created. This details (on the action or on the switch) can be edited now if needed.



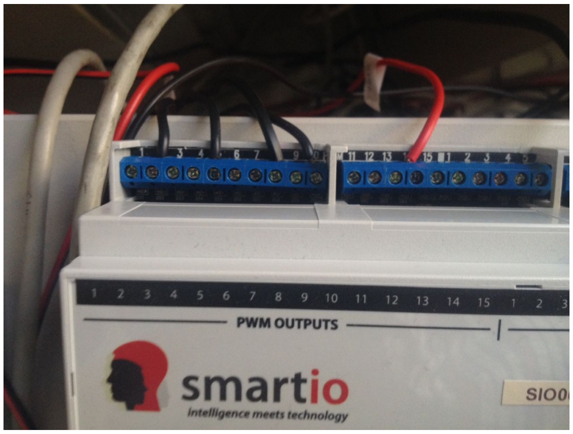
Configured Device List:

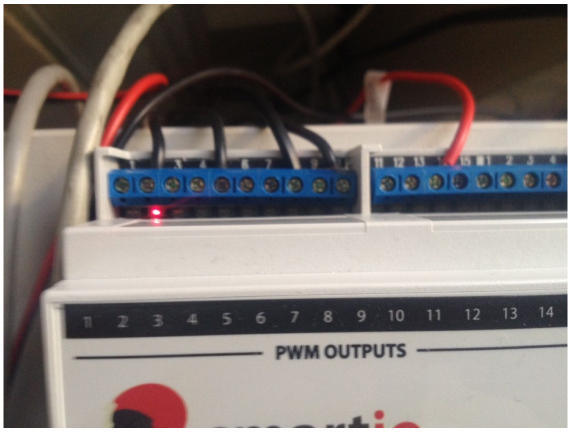
Name	Room	Type	Hardware ID
room_left	NoZone	LightOnOff (OnOffDe	01-01-00-00-00-50
BT_room_left	NoZone	ButtonClickAndLong (	01-01-00-00-00-50
room_right	NoZone	LightOnOff (OnOffDe	01-01-00-00-00-50
BT_room_right	NoZone	ButtonClickAndLong (	01-01-00-00-00-50
dimmer	NoZone	LightDimmer (Percent	01-01-00-00-00-50
BT_dimmer	NoZone	ButtonClickAndLong (	01-01-00-00-00-50
Blinds	NoZone	BlindsPercent (Tristate	01-01-00-00-00-50
BT_Blinds	NoZone	ButtonClickAndLong (	01-01-00-00-00-50
Bell	NoZone	DoorBell (Detector)	01-01-00-00-00-50

- Click Synchronize (bottom right corner) to program the SMARTIF Server. Changes only take effect in this moment.

The configuration is intuitive and the task is eased by the drop menus that are updated/greyed according to the changes that are being made. Small errors may occur when wiring or configuring. If you are not familiar with the system configuration, it is advisable to always add a switch to manually operate all devices. By doing so, you will be able to manually control all the system and check whether the configuration is correct. This switch can obviously be replaced by a simple wire acting as a shunt.

As a debug, one should manually operate the switch and look at the IO to see if the LEDs are being activated as intended. The LEDs are lit when the port is active and OFF is inactive as seen in the following pictures.





After the manual operation of all the devices, you can consider the SMARTIO and SMARTIF server to be installed and configured.

To finish this step, and as it will be needed in the next steps, save a XML file with the instructions (can be found in the options). The next step is to connect the refbox (or alternatively the devices_toolbox) with the SMARTIF server.

SOFTWARE CONFIGURATION:

In a previous grant of the RoCKIn project, a ROS interface to interact with the devices was built - Roah_devices. This interface has several utilities: services to perform action of the devices; a GUI; an offline mode (to use without the devices); etc. This software was built by João Reis and has a specific document where it is thoroughly explained. Although it is part of the RoCKIn refbox, it can be used in stand-alone mode.

This document will focus on the installation and configuration of the software in our refbox/smartif server. All the following instructions may be easier to follow if you have a keyboard and a monitor attached to the smatif server.

In this report, two different situations will be approached. In the first one (clean installation) we are considering that you do NOT have any prior installation/backup of the system. This instructions are considered to be advanced and not needed nor advisable if you already have a backup SD card image. The only exception is the case of adding/removing devices. Considering the hardware already assembled, jump to **Roah_devices ROS side setup.**

<u>Clean installation</u>

Roah_devices SMARTIF Server setup:

The following steps, were provided by João Reis in order to perform the software deployment. They were tested and performed in the past. They are proven to work but be extremely careful regarding paths and filenames.

In this tutorial we assume the Smartif server has the IP address 192.168.1.56 and password raspberry. If you already have roah_devices.jar, skip to 5.

1. Import the java project inside the *.jar into eclipse by clicking
   File | Import... | Existing Projects into Workspace.
2. Run the project to create a launch configuration.
   Open Main.java and click Run. Make sure you stop it
   with the red button in the console.
3. Export to a JAR file by clicking
   File | Export... | Java | Runnable JAR file. Select
   your launch configuration and destination
   roah_devices_server.jar somewhere. Finnish.
4. Close Eclipse.
5. scp roah_devices_server.jar root@192.168.1.56:/tmp/roah_devices_server.jar
6. ssh root@192.168.1.56
7. mkdir /var/lib/roah_devices_server
8. mv /tmp/roah_devices_server.jar /var/lib/roah_devices_server/
9. cp /etc/init.d/smartif /etc/init.d/roah_devices_server
10. sed -i s/smartifserver/roah_devices_server/g /etc/init.d/roah_devices_server
11. sed -i s/smartif/roah_devices_server/g /etc/init.d/roah_devices_server
12. sed -i s/SMARTIF/RoAH_Devices_Server/g /etc/init.d/roah_devices_server
13. update-rc.d roah_devices_server defaults
14. shutdown -r now && exit

When the smartif server restarts, it is possible to check on screen if the process has started. It is also possible to check this via shell ($ ps aux). If the process started, you will be able to launch the toolbox and communicate with the server. At this stage you are able to communicate with the server. Although, one last step is needed in order to actuate the devices.

Roah_devices ROS side setup:

At this stage we need to “guide” our software to interact with the correct hardware IO ports. To do so, we will be using the XML generated by the smartif software. The main idea is to use the information inside this file and use it in the toolbox code - Roah_devices. Once again, if the configuration of the hardware changes, for example by adding a new switch, you are obliged to do this step. Be also aware that if the system was working before that change, you only need to perform this step.

To do so, follow this instructions:

1 - $ java -jar smartif_xml_192_168_1_56.jar > smartif_conf.xml

2 - $ cat smartif_conf.xml | xmllint --format - > smartif_conf_f.xml   This will create an easier file to read.

2.1 - An error may occur due to bad syntax of the xml file. Edit it and remove the first line (easy to see since it is the only with a paragraph. “Keep alive....”. The file MUST start with “<?xml ....”)

3 - You should now look for the name of the devices, i.e., the names you gave during the configuration, on the smartif_conf_f.xml file. All the devices in the file should have a similar structure, for example: (taken from the files in attachment)

```
<device name="room_left" id="10" interface="OnOffDevice" devtype="LightOnOff" protocol="smartio"
zone="" visible="true">
        <parameters>
                <parameter name="outputPin" value="3.3" type="Hardware" valueType="HardwarePin"/>
                <parameter name="isNO" value="True" type="Constant" valueType="BOOLEAN"/>
         </parameters>
         <resources>
                <resource id="11" name="state" type="SmartioObjectResource" mode="writable"
datatype="OnOff"/>
         </resources>
</device>
```

In this case, the ID of this device is the 11. Pay attention to the fact that we do NOT need the number of the id of the device. We need the id of the resource of the device.

Gather the IDs of all the needed devices.

5 - This values should be placed inside the src/device_id.h file.

An example of the *.h and *.xml is provided in the attached files. In case of any change, study this two files as an example. Please keep in mind that all the changes made in the configuration (for example moving the bell from port XX to YY, In the hardware and in the configuration) must be also changed here. Typically if you are able to manually interact with the devices but not via software, something is wrong with this file (*.h not coherent with *.xml).

At this stage you should be able to interact with the physical devices via the toolbox. Since this toolbox is also used by the refbox, if it works while in standalone it will also work while under the refbox.

Copying the image to the SD card:

If a stable version of the SD card of the smartif server is available and no hardware changes (installation or configuration) were performed, one can easily re-upload the image to the card. The SD card system has proven to be very faulty in the past. A simple power disconnection may cause the distro installation to become unusable. It is strongly recommended to always have a stable version of the SD card backed-up.

Considering the SD image to be "image_working.img" we can use the linux built-in dd program to copy the image to the card.

1- Insert the card. There are laptops in ISR that have SD card connections.
2- Check if the card is mounted. If it is mounted, unmount it (all the partitions!)
3- run the following command. Check the name of the root of the SD card. In this case it is mmcblk0 (You can check the name of the device using the dmesg command) $sudo dd if=image_working.img of=/dev/mmcblk0
4- This operation will take some time. Remove the SD card and insert it on the smartif server. The smartif server must be powered off.
5- It should work right away. If possible, check if João Reis' code has been launched to confirm. Using the $ps mentioned above.

Possible known issues:

If you have restored the SD card from the image you might face a problem next time you run the configuration software (only needed if changes were performed on the hardware! Simple usage should not face this problem). An error mentioning that an activation is required may appear. In that case, you should provide internet in the 192.168.1./24 network and upload the licence in the software (you are allowed to browse in your files).

To provide internet to the server I would advise the usage of a router connected to the ISR network and configure a 192.168.1./24 on the LAN ports and DHCP on the WAN. Connect a monitor to the server and confirm that the gateway and dns are correctly assigned (should point to the router you are using).