Manual

Modbus Studer binding on openHAB

Date : 26.03.2021

Version : V1.0.0

Table of Contents

Preface

Welcome to the official guide for using the binding of Modbus Studer into openHAB. The purpose of this tutorial is to assist you on the first steps for make your own monitoring solution: we are sure you will explore openHAB's features for improve your Studer's installation in the best way possible. More information about openHAB here.

openHAB was developed for home appliances devices, Studer is now part of this big family with many others devices. We will install openHAB in a Rasperry Pi 3 B+ for reading values from your installation, the guide is mainly divided into three steps: **setup** materials needed, **add** a Studer's device and **visualize** values from devices.

This manual is intended for software and IT developers for example:

- Final-users with IT and/or software engineering background willing to extend the monitoring functionalities on their Studer system, by running openHAB in a RaspberryPi.
- People wanting to combine bindings, for example (Fronius, Tesla, Samsung..etc).
- Companies making hardware, using Studer equipment in their product.

The implementaion of openHAB binding is not supported nor warranted by Studer.

This manual has been developed by Euclide Innovation. You could find more information in the website: Euclide-Innovation.com

1 What I need?

The components needed are splitted into two groups: Studer's devices and a compatible platform for running openHAB. The procedure for configuring a Studer's device is the same for each devices on the family (Xtender, VarioTrack…).

In the case of this guide, we will use the follows:

OPENHAB'S PLATFORM

Raspberry Pi 3 B+

- (Micro) SD Card 16 GB
- USB Cable compatible for Modbus
- Ethernet Connection (WiFi possible)
- A good power supply

STUDER'S DEVICES

Xcom-485i*

- BSP/Xcom-CAN
- Xtender
- VarioTrack
- VarioString

***Mandatory**

You can find all the components and documentation to Euclide-Innovation.com

2 Installing openHABian

As mentioned on first point, we will use as compatible openHAB platform the Raspberry: because of its low price, small form factor and low energy consumption, the Raspberry Pi is a quite famous platform for openHAB and will allow to create a nice monitoring solution.

For doing that, the community of openHAB has developed a self-configuring Linux system setup, that it's called **openHABian**, a special OS suitable for many different Raspberry Pi version.

2.1 Raspberry Pi (Prepackaged SD Card Image)

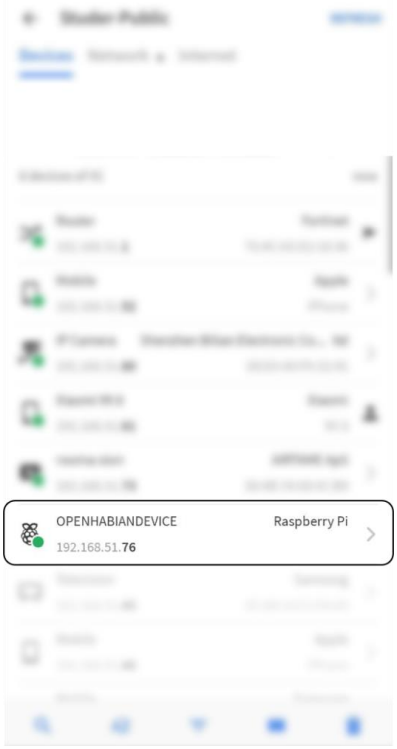
Flash, plug, wait, enjoy: the provided image is based on the Raspberry Pi OS Lite (previously called Raspbian) standard system. On first boot, the system will set up openHAB and the mentioned settings and tools. All packages will be downloaded in their newest version and configured to work without further modifications. The whole process will take a few minutes (around ~30 minutes), then openHAB and all other needed tools to get started will be ready to use without further configuration steps. openHABian is designed as a headless system, you will not need a display or a keyboard, just a computer/smartphone connected on the same network of your Raspberry Pi.

SETUP:

- Download the latest "openHABian" SD card image file
- Write the image to your SD card (with Raspberry Pi Imager or Etcher)
- Insert the SD card into your Raspberry Pi, connect Ethernet (WiFi also supported) and power on.
- Wait approximately **15-45 minutes** for openHABian to do its magic. (Check the progress in your browser into Raspberry Pi's IP address)
- The system will be accessible by its IP or via the local DNS name openhabian (or whatever you changed 'hostname' in openhabian.conf to)
- Be patient: in the meanwhile installation is running proceed to follow steps.

2.2 FINDING IP ADDRESS OF RASPBERRY PI

If you need to find the IP address of your Raspberry Pi, use Fing: there is a desktop version and mobile version. Use your mobile phone connected on the same network of your Raspberry Pi for finding the IP address.
Tips: for Linux you could use Angry Ip Scanner.



Screenshot of Fing's app.

3 Configure & Connect

The Xcom-485i will made the bridge between openHAB (Raspberry Pi) and Studer's devices: it allows the integration of Studer's protocol in advanced control applications using Modbus RTU 485, the universal communication protocol in the industrial environment.

3.1 SETUP THE XCOM-485I

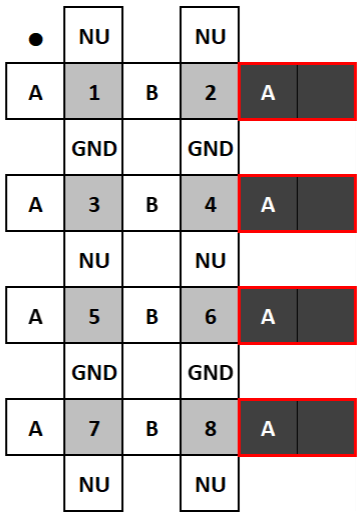
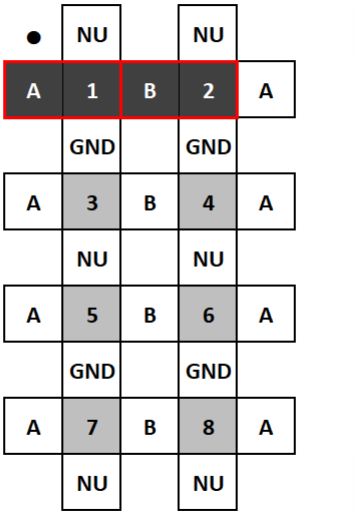
No special configuration are needed on the Xcom-485i. The table above it's just for reference: if you need more information about the Modbus protocol please check the documentation provided by Studer Innotec.

Modbus	**Value**
Addressing*	1 to 63
Baud Rate	9'600 bps
Parity	Even
Protocol	Modbus RTU only
Byte transmission	1 start bit
Electrical Interface	RS-485 on 2 wires
Connector Type	RJ-45

**The addressing must be kept from 1 to 63.*

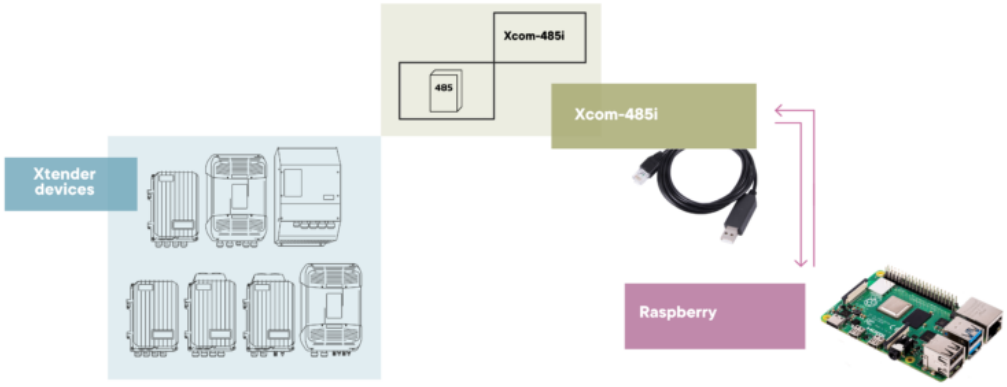
3.2 PINOUT CONFIGURATION XCOM-485I

Open the Xcom-485i, there are two screws in the back case of the device. You need to configure the jumpers according to your output cable. In case you're using the cable mentioned in the guide (Cable Modbus RJ45-USB) the pinout will be as the image on the right:

DEFAULT	CONFIGURATION
	
	

3.3 CONNECT THE XCOM-485I TO RASPBERRY PI

Once you configured the jumpers according to your cable, close the case of the Xcom-485i and connect it from the RJ45 port side "External" to our Raspberry Pi USB. Follow Studer's Innotec rules for Xcom-485i position on Studer's bus. If the Xcom-485i is the last device on your chain, must be put in "T" position on the switch for termination.



4 Install & Configure Modbus “Studer” binding

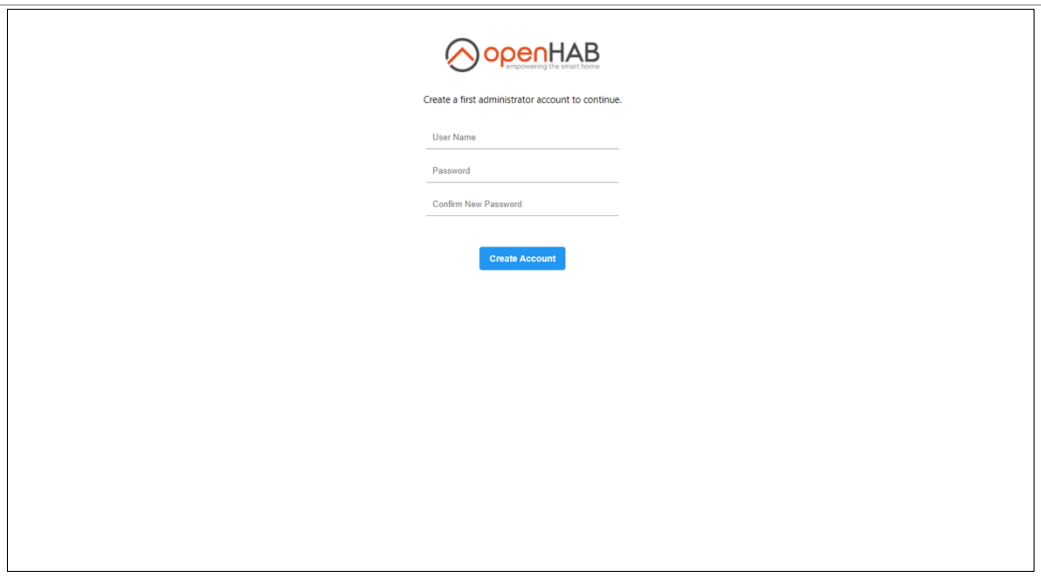
Nowadays Studer is integrated on openHAB (under Modbus binding): this allow you to visualize some of the most important values for monitoring your installation **without writing a line of code!** The complete list of info, that openHAB call “items” are available on the binding’s page here.

Once the installation of openHABian is finished (we left the installation in progress chapter 2.1 while we configured the Xcom-485i) you will able, with your computer connected on the same network of your Raspberry Pi, to join into the first window of openHAB.

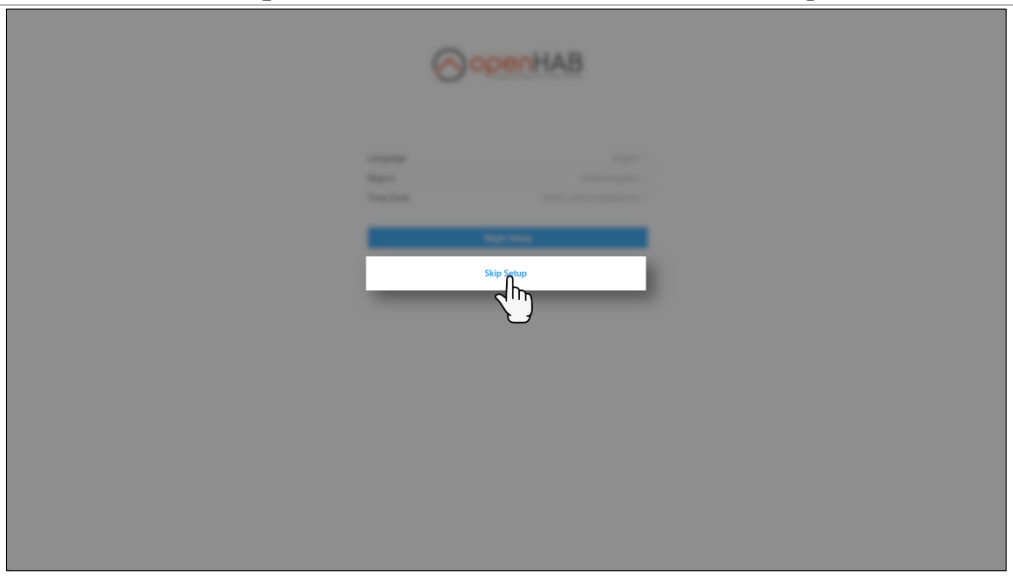
The web address is the IP you discovered on chapter 2.2, remember to add the port 8080 at the end of the IP found.

IPaddressofyourRaspberryPi:8080

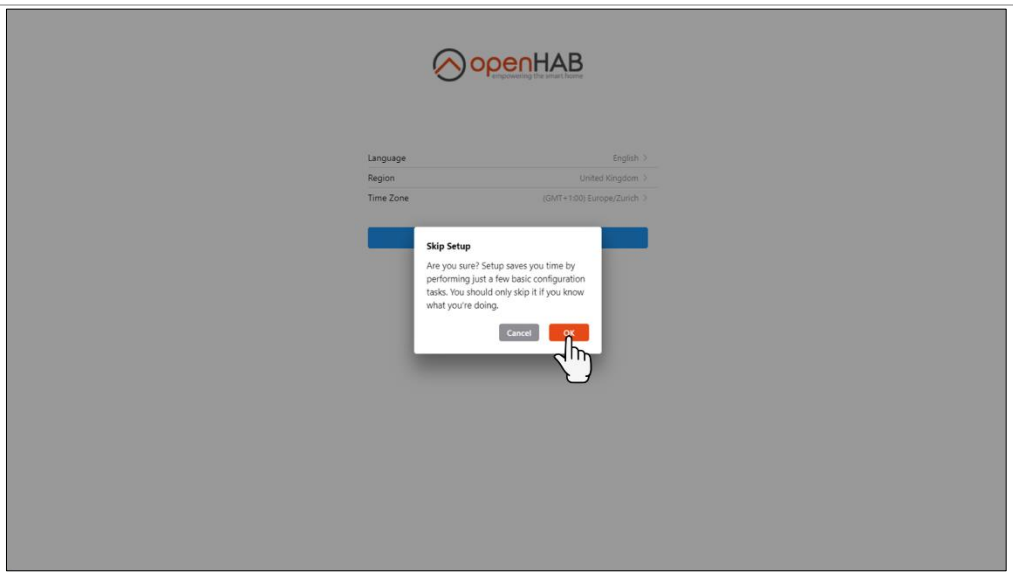
When you will be there, just follow the procedure step by step here described:



1 – Create your account with username and password

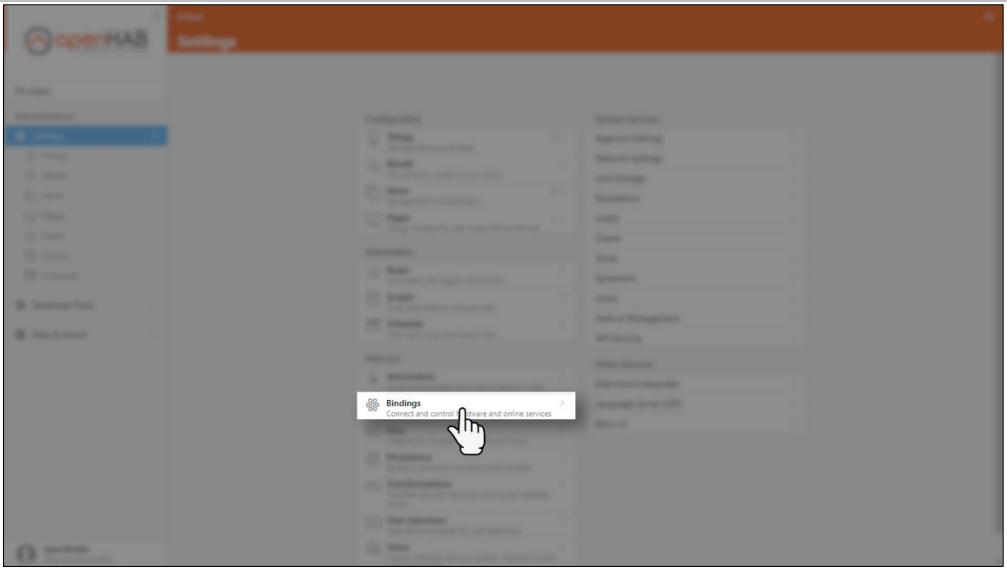


2 – Skip the setup



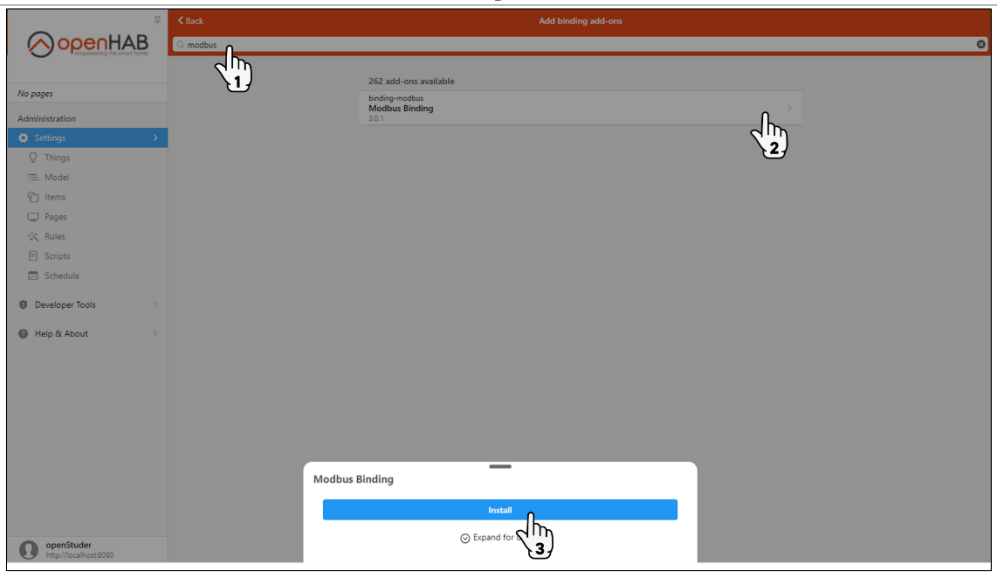
3 – Confirm the skip

4 – Go to "Settings"



5 – Look for "Bindings" section

6 – Add a new binding with the button "+"

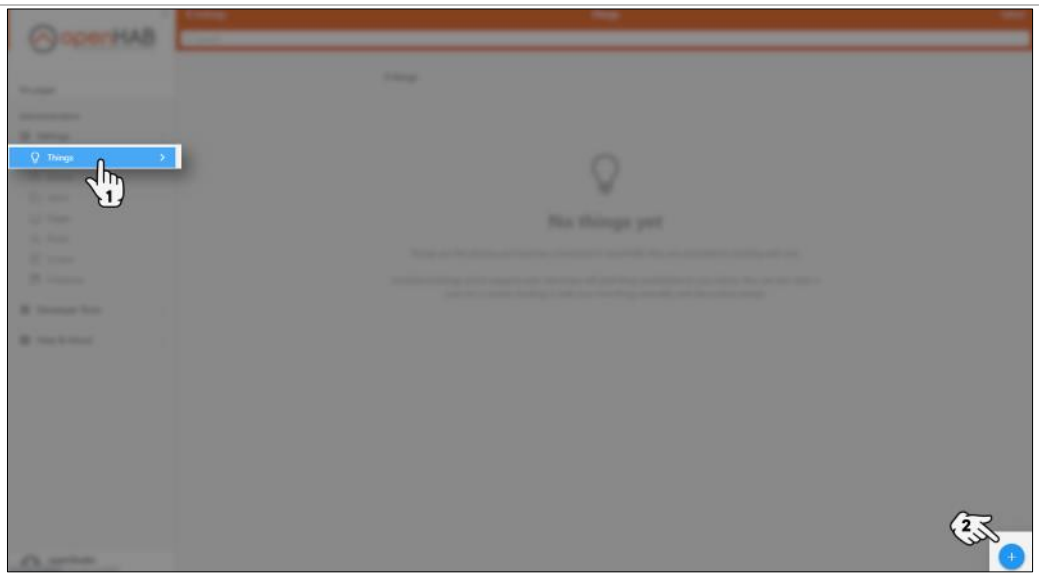


7 – Search the binding "Modbus"(1), click on it(2),install it(3)

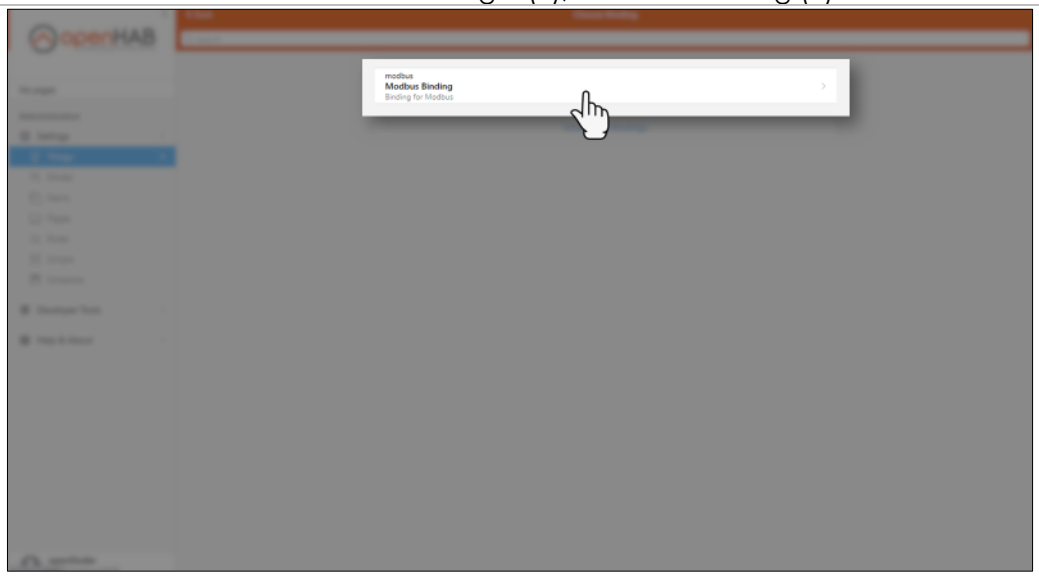
4.1 ADD THE XCOM-485I (MODBUS SERIAL SLAVE)

The first device required to configure and add into our Modbus Studer binding is the "slave" of the system also called as **bridge**. In our case it's the **Xcom-485i**.

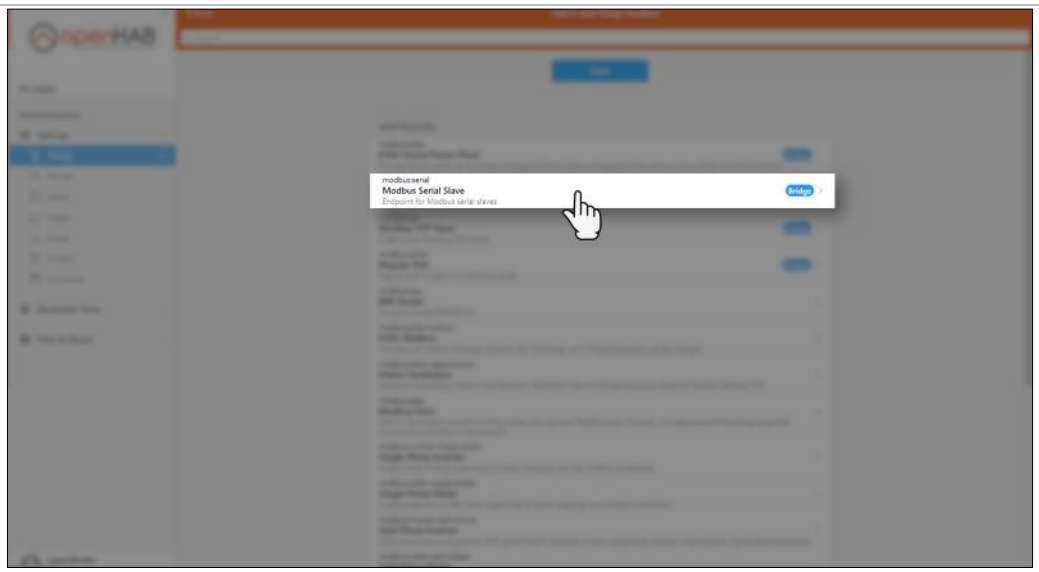
Please follow the steps below:



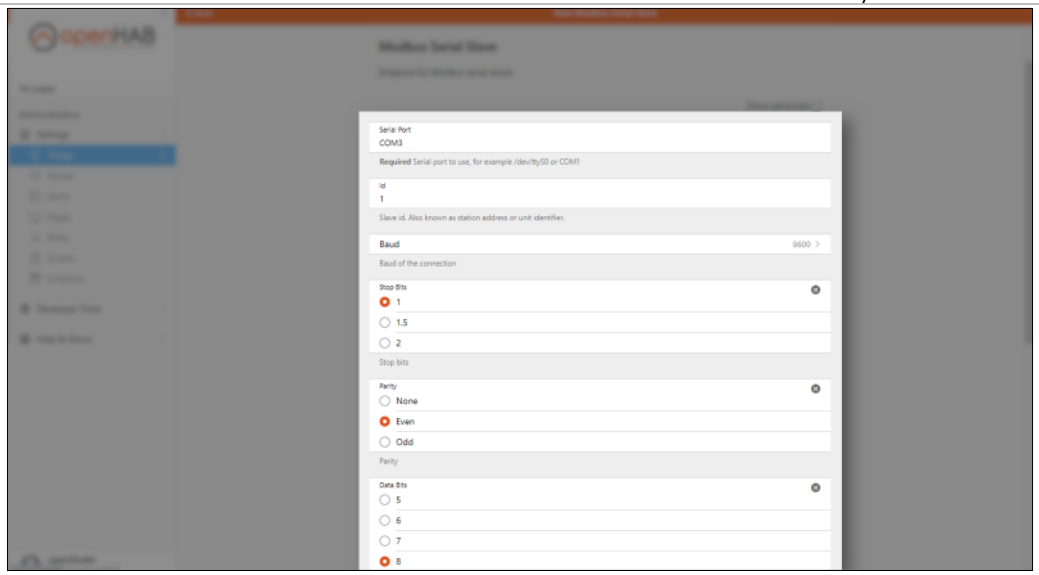
1 – Under menu "Things" (1), add a new thing (2)



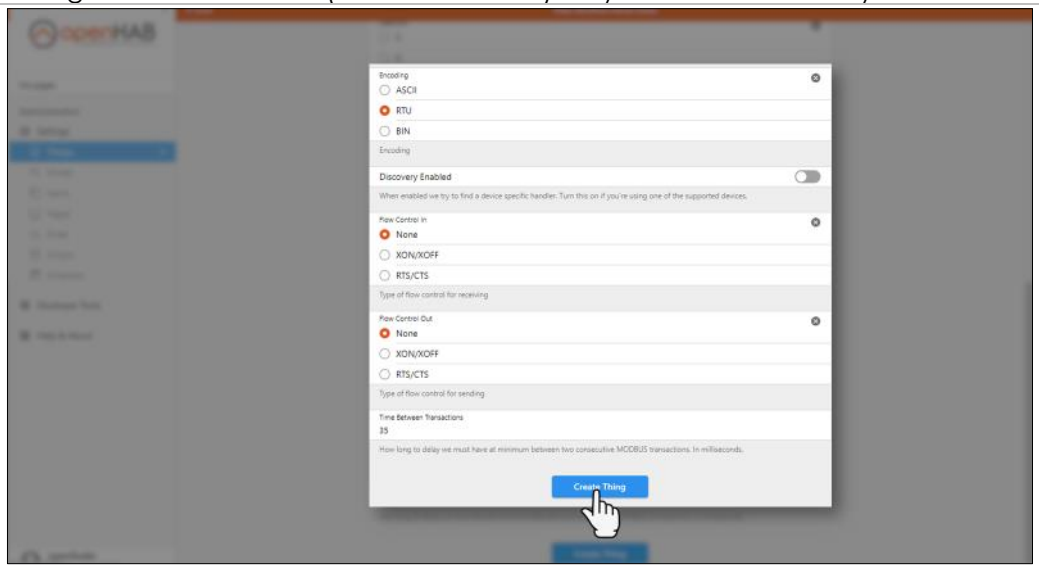
2 – Device on Modbus binding



3 – Select "Modbus Serial Slave" for the Xcom-485i only



4 – Configure the Xcom-485i (see table below). Pay attention on "Parity". Must be even.



5 – Once Configured, create thing

Serial Port*	COM3
ID	1
Baud	9600
Stop Bits	1
Parity	Even
Data Bits	8
Encoding	RTU
Flow Control IN	None
Flow Control OUT	None
Time Between Transactions	35ms

*Choose your serial port communication, this might be different in your case.
*Try to write **"/dev/ttyUSB0"** if you're using a Raspberry

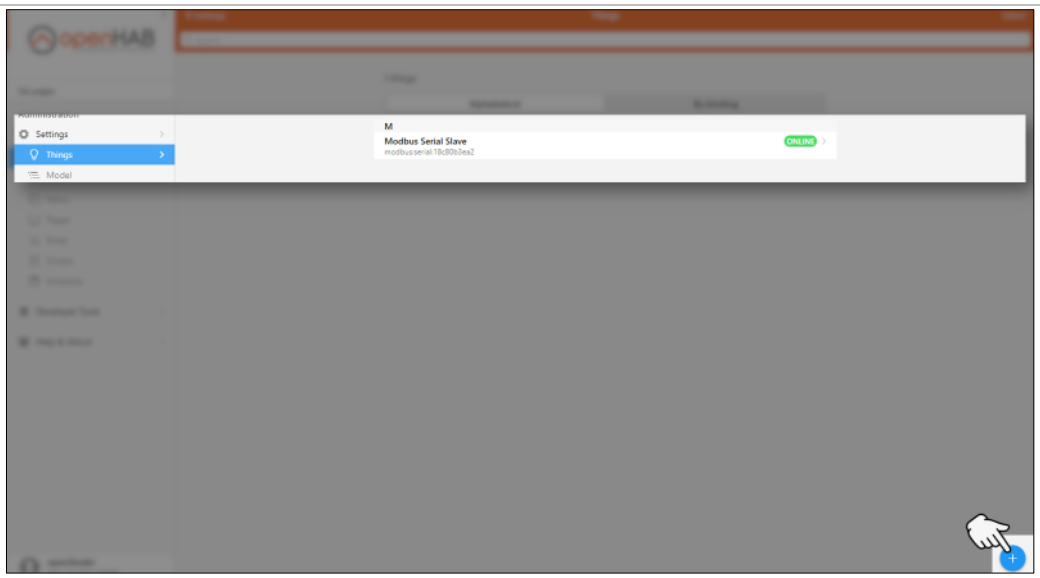
4.2 ADD A STUDER'S DEVICE (THINGS)

Each device of Studer's bus will be a single things on the modbus binding of openHAB.

This procedure can be applied for all the devices inside the Studer's bus. **Select devices type (Xtender, VarioTrack..) on Step n°3 and choose the address on Step n°5.** The address range of devices is as follow:

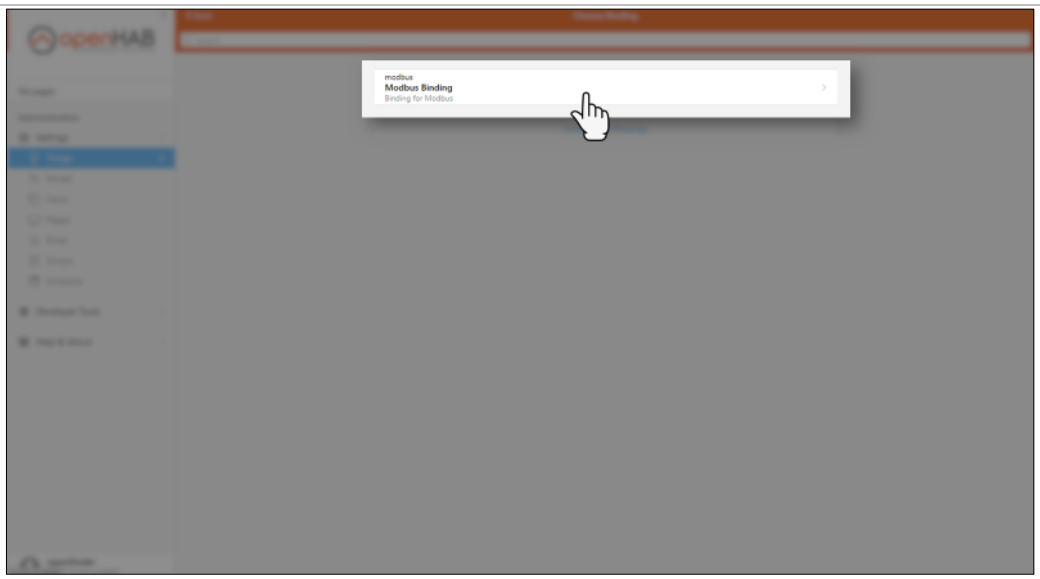
Address	Devices
1	Xcom-485i Modbus gateway (slave)
11 to 19	Each **Xtender**
21 to 35	Each **VarioTrack**
41 to 55	Each **VarioString**
61	BSP/Xcom-CAN

Table of address range for Studer devices

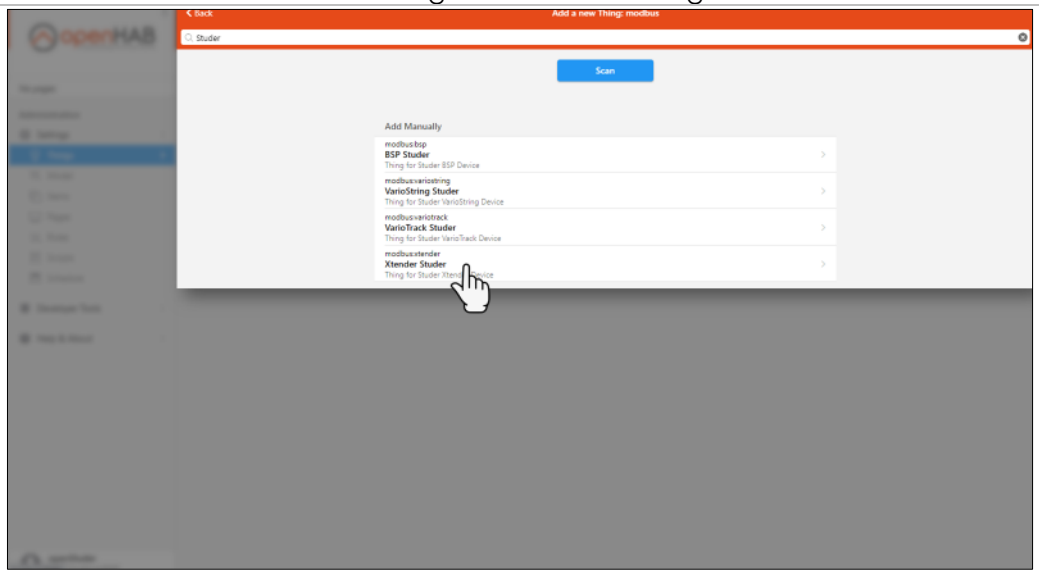


1 – Add a new thing under menu "Things"

***Tip**: Be sure the bridge (Modbus Serial Slave), our Xcom-485i is well configured and online for communicating with the other devices.*

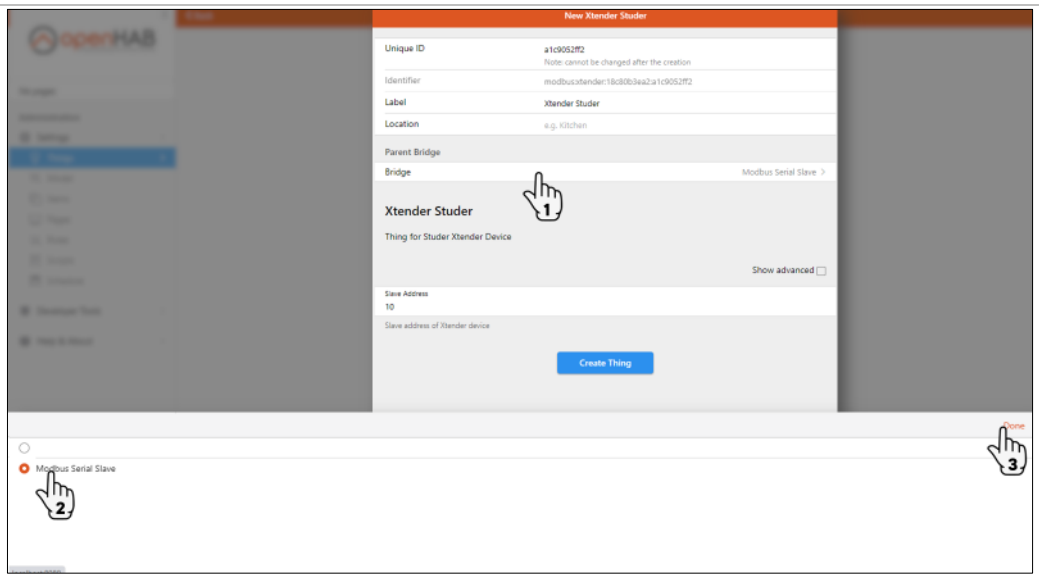


2 – Thing on Modbus binding

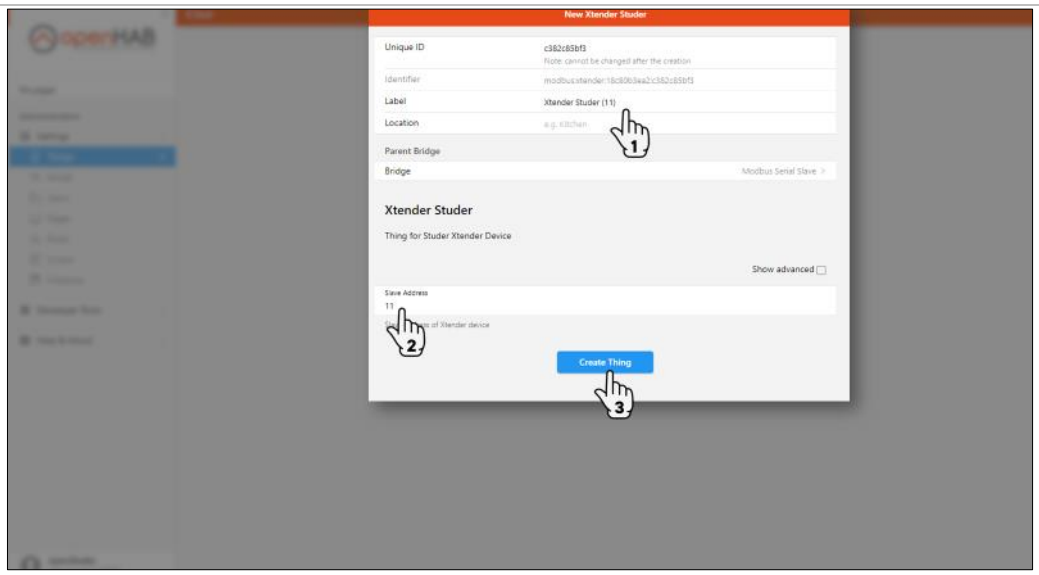


3 – Look for the Studer's devices you'd like to add. We'll add an Xtender

***Tip**: write Studer on the search bar for a faster research.*



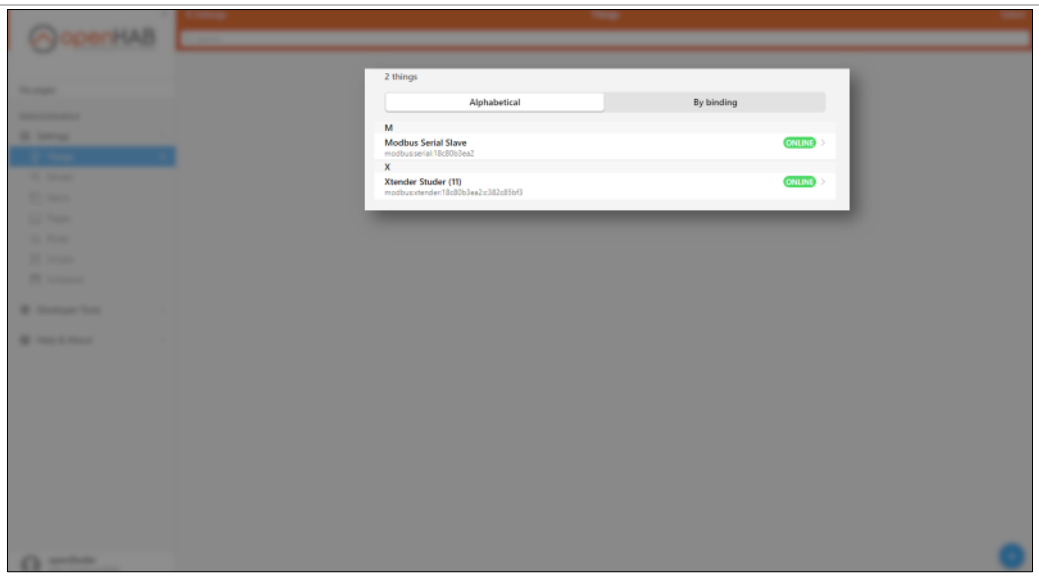
4 – Choose the bridge (1), Modbus Serial Slave "Xcom-485i" (2), Click Done (3)



5 – Rename your device (1), Select the address (2), Create the thing (3)

***Tip**: Renaming your device will help you to recognize it, useful if you have a lot of devices. I will rename it "Xtender Studer (11)".*

***Tip**: For address of device refer to table above.*



6 – Our Xtender (things) is online!

5 Visualize data (items)

Here we are on the part where you can create and personalize your monitoring solution.

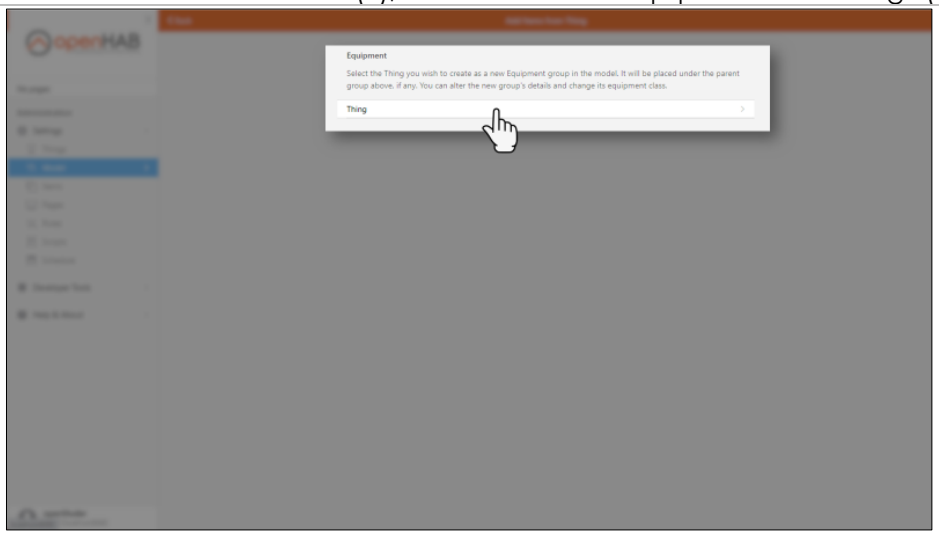
Before starting please keep in mind the table below as reference, this will help you trough the guide for better understanding the terminology.

Things	Devices, like an Xtender, VarioTrack.. etc
Items	Values, like current, voltage.. etc

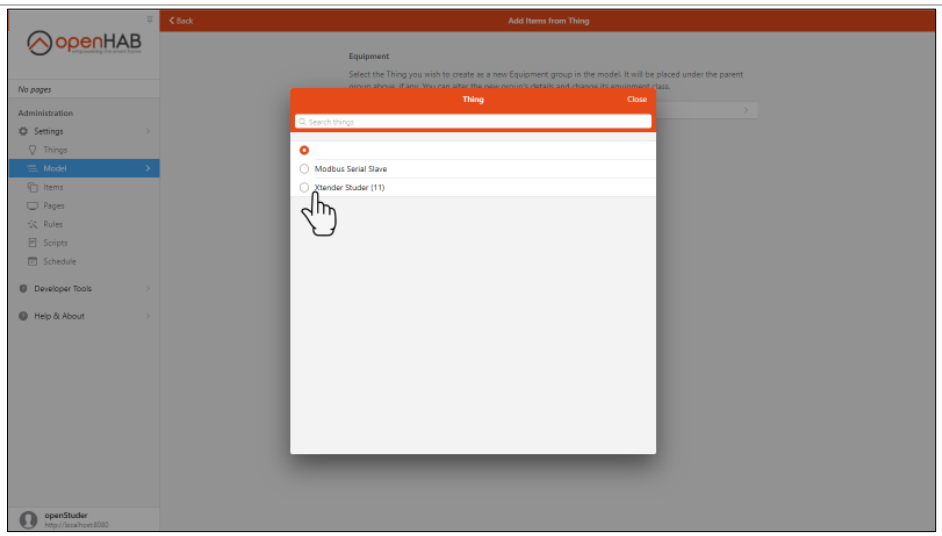
5.1 VISUALIZE VALUES FROM A STUDER'S DEVICES

We will visualize the "Input Active Power" from our Xtender we added previously. The list of items available to be visualized is present here.

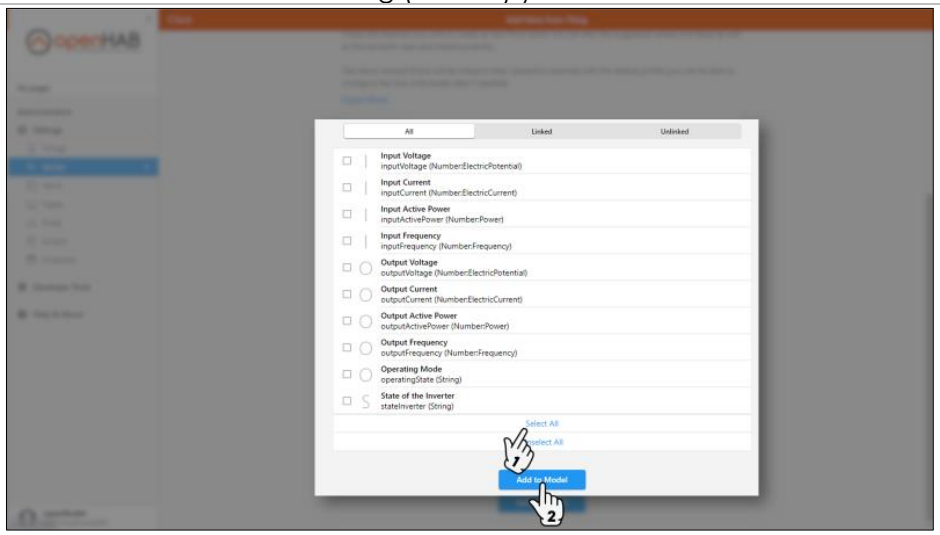
1 – Under the menu "Model" (1), click on "Create Equipment from Thing" (2)



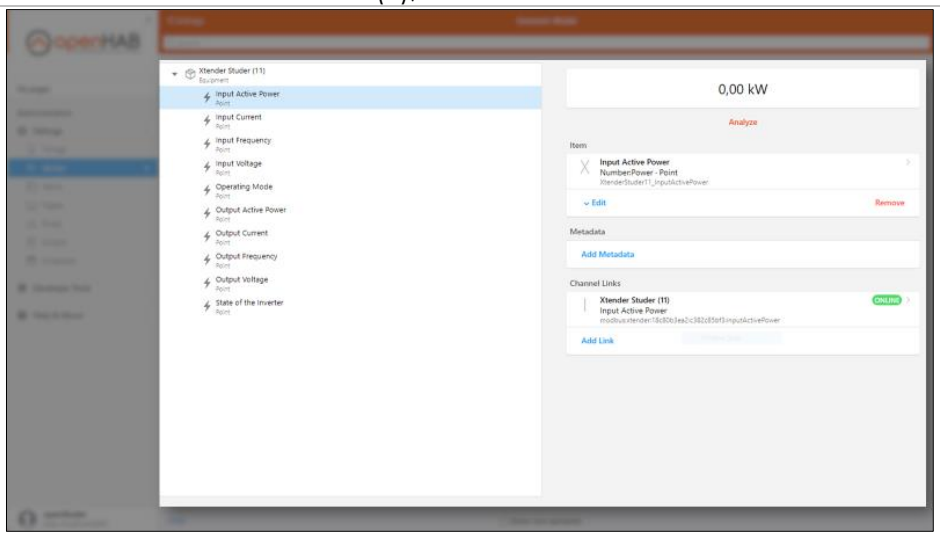
2 – Click on Thing



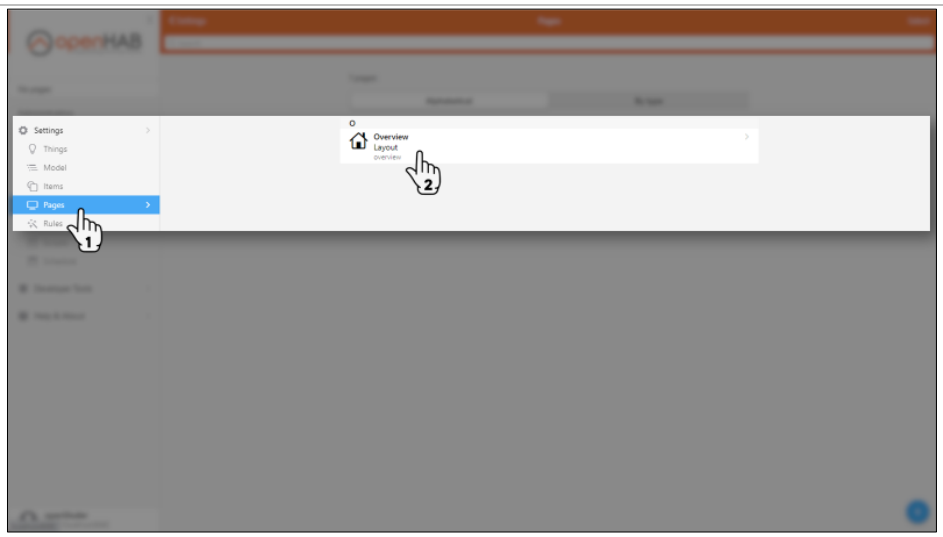
3 – Select the thing (device) you want to visualize



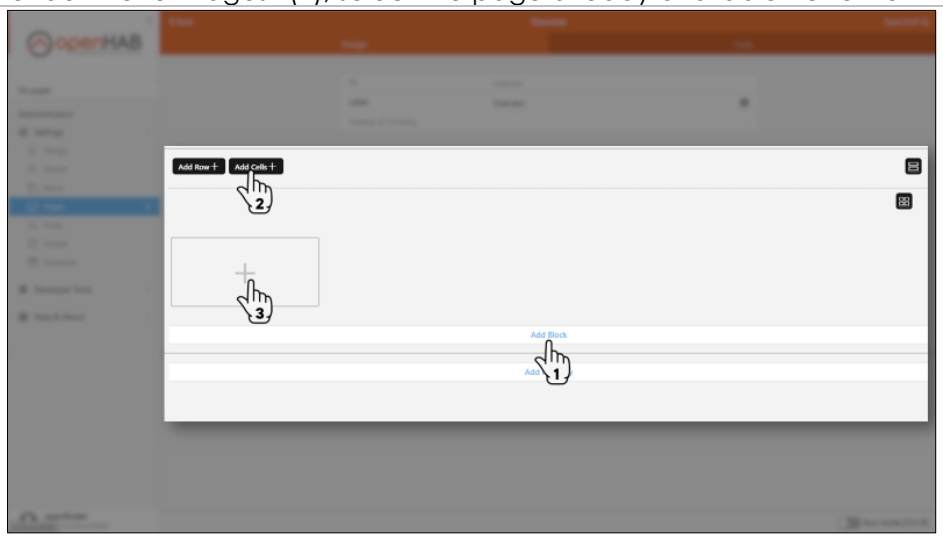
4 – Click on "Select All" (1), once done select "Add to Model"



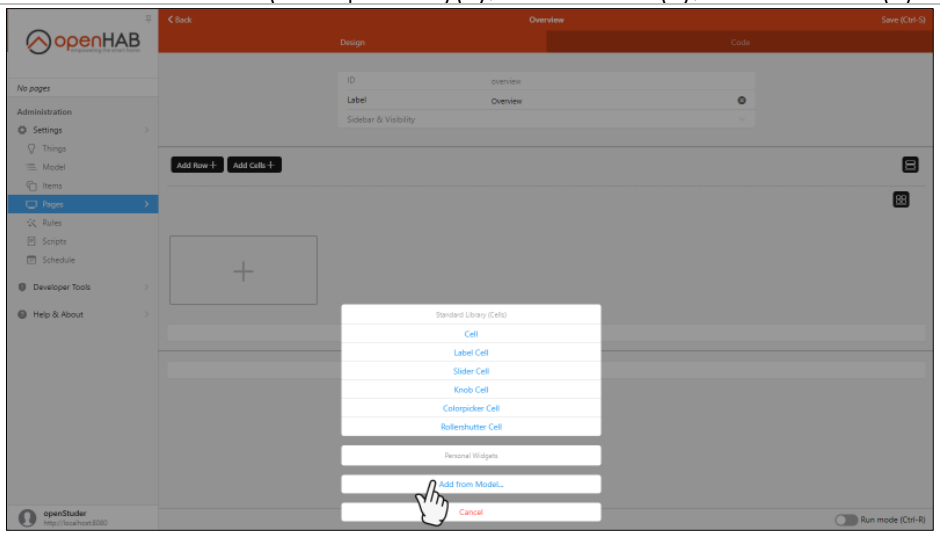
5 – Now we successfully added the values (items) from our device (thing).



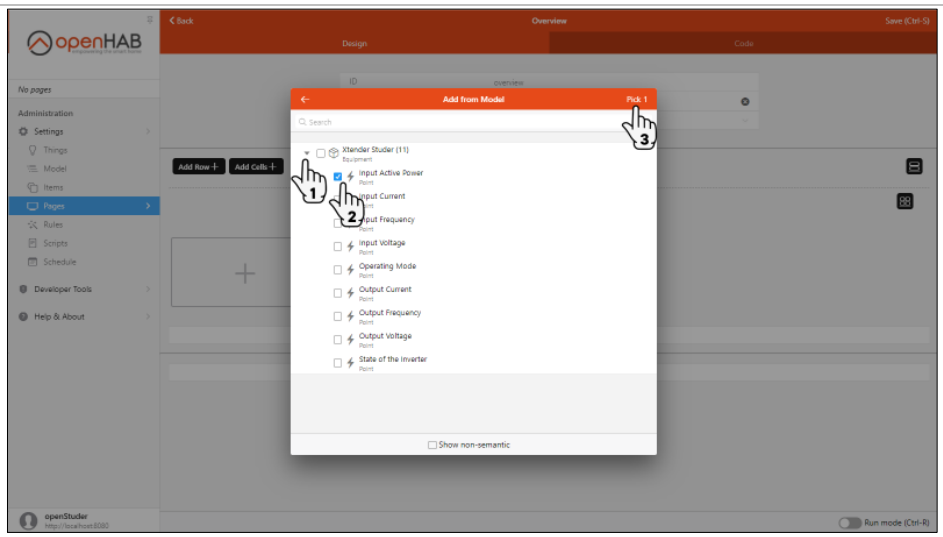
6 – Under menu "Pages" (1), select the page already available "Overview" (2)



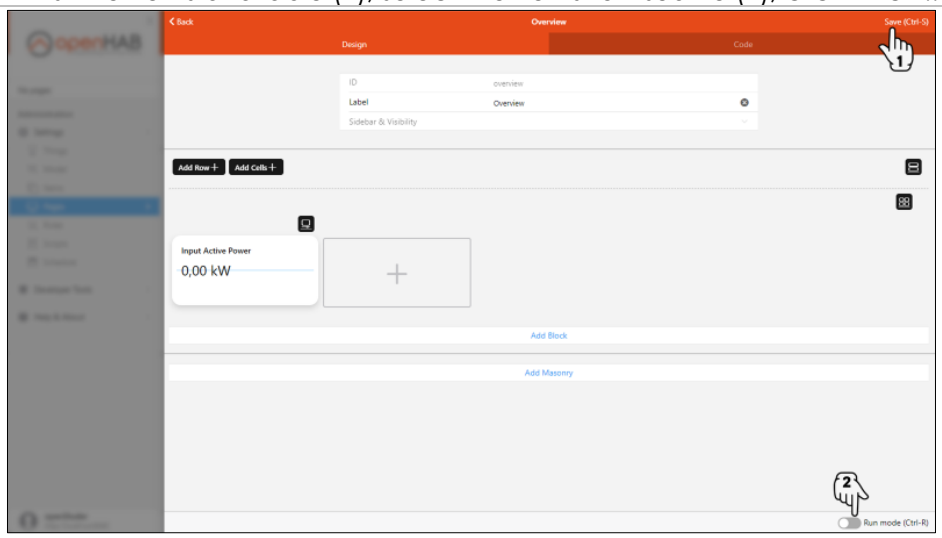
7 – Create a Block (if not present)(1), add a Cells (2), create a cell (3)



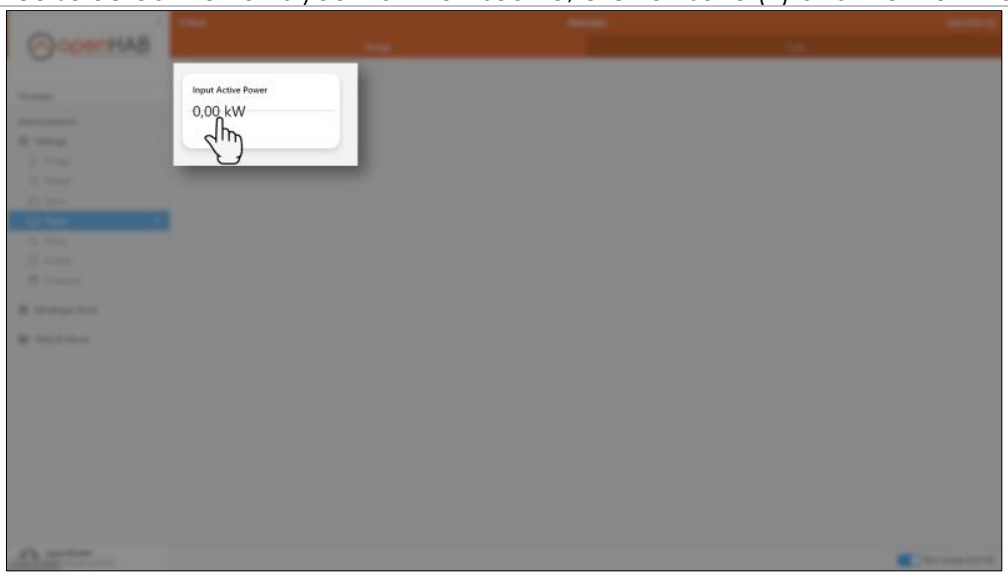
8 – Select "Add from Model"



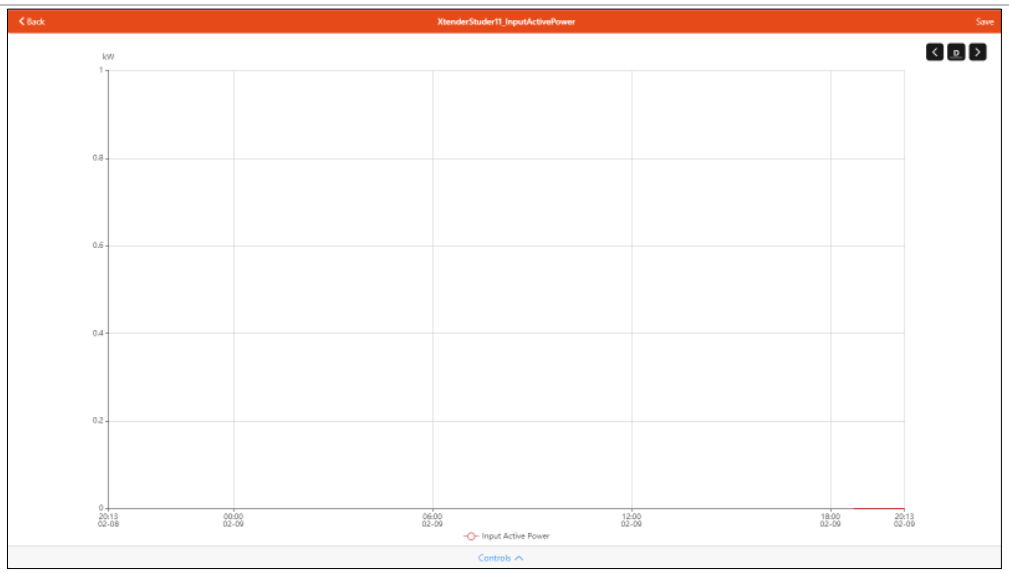
9 – List the items available (1), select the items to visualize (2), click "Pick .."



10 – Once selected the items you want to visualize, click on save (1) and then run mode (2)



11 – Here (in run mode) you will have your items. Click on it to explore the built-in graphs



12 – Here is the graph that will register your items every 5 seconds.

Conclusion

We finished the tutorial for using the Studer binding into openHAB. From now the possibilities you have are infinite, you have the white-paper for creating your own monitoring solution, adding icons, wallpaper or combine two differents bindings available on openHAB for improve your monitoring solution furthermore. We invite you to be part of the openHAB community for exploring this open source family.

The use of this manual is under the responsibility of the end-user. This manual is neither designed nor guaranteed for the supply of life support applications or any other critical application with potential risks for human beings or for the environment. We shall assume no liability for patent infringement or other third party rights involved in the use of this manual.