1st Draft

openHAB2 Raspberry beginner's walkthrough – (Using Raspberry Pi 3 with openHAB2 and Z-Wave, WiFi LED, Samsung TV and YahooWeather bindings for a home automation project)

Contents

Version index

0.1 DRAFT	This is the initial version of the document with any feedback of the community. This version was written with the focus on initial hardware and software setup which should already have a good consistency. The openHAB2 configuration and real home automation part still needs some work done! Anyway this part will be the part where you have to apply individual changes and go to the forum for further help since no home automation project is similar to another. You will find not all the bindings utilized and not all the things added to the openHAB2 configuration of the used testbed system since I am building the testbed system again from scratch. So in some places you find pictures missing or description not being completed. I am also using a parallel system (my current live home automation system) to check different hardware configuration, especially when it comes to the displays or to Z-Wave controllers. *Testbed system:* Raspberry Pi 3 MicroSD Sandisk Ultra 32GB Power supply goobay 3,1A USB Mouse basic logitech USB Keyboard basic Microsoft qwertz German layout Display HDMI connection to 24” computer display Case no name incl. ventilation Z-Wave controller UZB Z-Wave PLUS USB stick by Z-Wave.Me *Live home automation system:* *Testbed system:* Raspberry Pi 3 MicroSD no name 16GB Power supply no name 3,1A incl. cable swith Bluetooth Keyboard Rapoo Bluetooth Keyboard qwertz German layout Display 7” Raspberry display Case premium case for Raspberry Pi 7" Touch-Display (the closed version) Z-Wave controller Aeotec by Aeon Labs Z-Stick Gen5 *Live Z-Wave devices:* Z-Wave Fibaro Double Switch 2 Z-Wave Aeotec by Aeon Labs ZW100 Multisensor Z-Wave Aeon Labs ZW088 Z-Wave Key Fob, Gen5 Z-Wave NodOn Smart Plug

Chapter 1: Before you start

Is openHAB2 the right choice for my home automation project?

Be aware that openHAB2 is an OPEN home automation solution which is strongly living from a very supportive community. If you want to have a plug and play solution with supplier guaranteed service level and a high likeliness that all the features are working and all the hardware is compatible, you might be better off in getting a ready to use home automation kit including the designated controller (like e.g. devolo or homematic IP). Consider this especially if you are planning to do safety related automation or emergency detection like fire alarm.

If you, on the other hand, are willing to spend a few hrs/days in learning how to do a little installation and coding yourself and have no problems with the service level of a Raspberry Pi 3 (it is not as failsafe as other controllers) you might find a perfect environment with openHAB2 for your low cost, very flexible and continuously improving home automation environment.

Introduction:

This tutorial is targeting beginners like me to get a step by step guideline to get all the things installed. Since I am no coding expert and have no experience in Raspberry and Raspbian I am trying to go through the things step by step, so you should be able to get everything done, even without exactly having to go into all the details. That is one of the reasons I am also using the graphical GUI PIXEL for Raspbian since I thing it makes it easier for the beginners to get started (and you might want to use PIXEL anyway when you are using the Raspberry 7" display as interface for your home automation controller)

This tutorial is also based on **having a Windows PC** to support the setup process. You might be able to completely do it without the support of an extra PC, if you can get a MicroSD card with a pre-installed Raspbian OS and use the display options (the Raspberry 7" display or HDMI Display) for the Raspberry.

DISCLAIMER:

This tutorial might contain some typos, errors or ways of setting up, which can be done in a better way. I am just reflecting my process of starting from scratch and slowly working my way through hundreds of online tutorials, manuals, forum threads etc. and on the way, highlighting the issues I had in getting things working. There will be no guarantee that the given instructions are working for your project as well.

Anyway I hope this tutorial will help some beginners to enjoy home automation with openHAB2.

A few words about the 2 in openHAB2:

The 2 in openHAB2 is important! The tutorial is based on the openHAB2 and will not go into all the details of the old version.

You just have to be aware, that a lot of online documentation is still for the openHAB version and will **not** be applicable for openHAB2!

So the best thing is always to go to the official webpage of openHAB2 and start from there, and only if you really can't find the information or the link there, go to google and search for other solutions. I was always using the search setting (last year) so it was more likely to the results considering openHAB2 and not openHAB.

Chapter 2: Preparation

Shopping list:

As mentioned before, I am basing this tutorial on the graphical GUI of Raspbian named PIXEL so the shopping list is also containing parts for this optional setup:

Minimal setup of the controller:

Raspberry Pi 3	
MicroSD card 16GB (minimal to have some buffer for the future) Make sure you have the right card reader to plug the MicroSD card into your computer!	
Designated Raspberry power supply (min. 2,5A 5V, I recommend 3A) *Do not use other USB chargers since insufficient power supply (shown in GUI as lightening symbol in the upper right corner) will result in serious issues like e.g. Bluetooth not working*) A cable switch might be a good thing since you might have to hard-reset your Pi in the early days more often and the Pi itself does not have a power switch	
USB Mouse	
USB Keyboard	
HDMI cable (full size to whatever your display needs)	

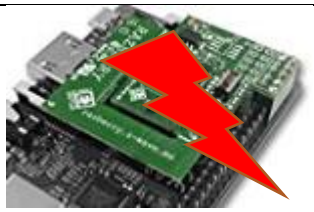
Ethernet cable (optional, if you not want to use WiFi to connect the Raspberry to your gateway)	
Raspberry case (optional, will **not** be needed if you are going for the 7” Raspberry display setup)	
Display with HDMI input (optional, will **not** be needed if you are going for the 7” Raspberry display setup)	

Additional hardware for optional setup of the controller with 7” Raspberry display:
(I found it very useful to have one permanent GUI interface mounted on your controller, you can also use this touchscreen interface directly to interact with your home automation):

Raspberry Pi 7" Touch-Display	
Premium case for Raspberry Pi 7" Touch-Display (closed version) often sold in bundle with Touch-Display, should be available in black, white and transparent This is a very good case if you want to place the controller on a table or counter since it is protecting the Raspberry from the back.	
Alternative: Cases for Raspberry Pi 7" Touch-Display. You will find a wide range of other cases. The open versions might give you a better access to the Pi GPIO pins or for changing SD card. Please consider: since you can rotate the image of the GUI on the display you can also choose to switch from landscape to portrait orientation	
Bluetooth keyboard (optional, since the optional on screen touch keyboard for Raspbian PIXEL was not working without errors, I decided to go for a Bluetooth keyboard which makes the typing much easier)	

Z-Wave Controller

(be aware that the details serial numbers or item names may vary since you have to always make your you get the hardware which is allowed in your country!)**:**

Aeotec by Aeon Labs Z-Stick Gen5 Pros: - Allows offline inclusion of Z-Wave devices which makes it very easy since you only have to take the stick to the mounted device, not the entire Raspberry Cons: - Including battery powered devices into openHAB2 requires a special process (see tutorial)	
Alternative: UZB Z-Wave PLUS USB stick by Z-Wave.Me Pros: - Cheapest controller Cons: - For inclusion, the controller has to be plugged into the Raspberry, so for mounted devices like wall switches , you have to take the Raspberry in close proximity of the device or do the inclusion before you mount the switch inside the wall	
NOT REALLY an Alternative: RaZberry2 Z-Wave Plus Daughter Card for Raspberry Pi Home Automation *(not plug and play compatible with optional setup of the controller with 7" Raspberry display!)* Pros: - will be mounted directly on the Raspberry so it is not using a USB port Cons: - will be mounted directly on the Raspberry which is blocking the GPIO pins for e.g. the Display power supply or additional cooling fans, so you have to manually solder the power wires at the back of the razberry - is using the i/o port of the Raspberry Pi 3 on board Bluetooth, so a lot of additional configuration is needed to get the razberry and the Bluetooth running in parallel. - most expensive controller	

Z-Wave sensors, switches and actuators and other home automation devices
(be aware that the details serial numbers or item names may vary since you have to always make your you get the hardware which is allowed in your country! So it is just a selection of what I was using in my project in Germany and some products might not be available in other countries):

Z-Wave Fibaro Double Switch 2, Z-Wave Plus Smart Switch (comes at the same costs than the single switch and gives you 2 channels. Only reason to go for single switch is you need the full power range of the single switch since the double switch has slightly lower range)	
Z-Wave Aeotec by Aeon Labs ZW100 Multisensor	
Z-Wave Aeon Labs ZW088 Z-Wave Key Fob, Gen5	
Z-Wave NodOn Smart Plug (not available e.g. in the US)	
WiFi XCSOURCE Magic UFO-WiFi LED-Controller Type LD382 (other brand names might work as well, but you have to make sure it is Type LD382, LD382A or LD686) I was using a WiFi controller on purpose since: - it is only about half the price of a Z-Wave WiFi controller - you can control the device as well via smart phone (like light to music feature of the app) But some things you have to be aware of using WiFi LED instead of Z-Wave LED: - you have to have a WiFi network to which your Raspberry and your WiFi LED controller is connected - you have to manually install a beta / snapshot version of openHAB2 or manually install the WiFi LED Binding on top of the package based installation of openHAB2 (see tutorial)	

NOTE: if you want to by other Z-Wave devices make your they are listed in the Z-Wave device list of the openHAB2 Z-Wave binding to make sure they are supported in the context of openHAB2:

http://www.cd-jackson.com/index.php/zwave/zwave-device-database/zwave-device-list

Software list:

My tutorial is using a MS-Windows windows machine for the PC part (You should be able to get it done with Mac or Linux PCs as well, but you have to go online to look up the differences and do some adaptions on the tutorial e.g. mounting the Raspberry file system to PC)

Windows Download list:

The latest **Raspbian** (Raspberry OS) image You have to download the "Raspbian Jessie with PIXEL - Image with PIXEL desktop based on Debian Jessie" since this tutorial is using PIXEL	https://www.Raspberrypi.org/downloads/Raspbian/
Etcher (to write the Raspbian image to the SD-Card)	https://etcher.io/
Eclipse Smart HomeDesigner (optional but strongly recommended for easy editing of OpenHAB2 configuration files; incl. syntax highlighting) You have to choose the right version for your PC	https://www.openhab.org/downloads.html
To use Eclipse Smart HomeDesigner you need **Java Runtime Environment JRE** (if not already installed on your PC)	https://java.com/
PuTTY or **KiTTY** portable to access the Raspberry console from your PC	http://www.putty.org/ https://portableapps.com/apps/internet/kitty-portable
WinSCP portable to access Raspberry file System directly from your PC (might become obsolete if you use a SAMBA server on your Raspberry, see tutorial)	https://winscp.net/eng/download.php

Raspberry downloads:

How to download software will be explained in the tutorial, but as a reference you will use

openHAB2 Package repository based installation or manual installation (be aware that the file locations on the Raspberry will be different based on which kind of installation you choose)

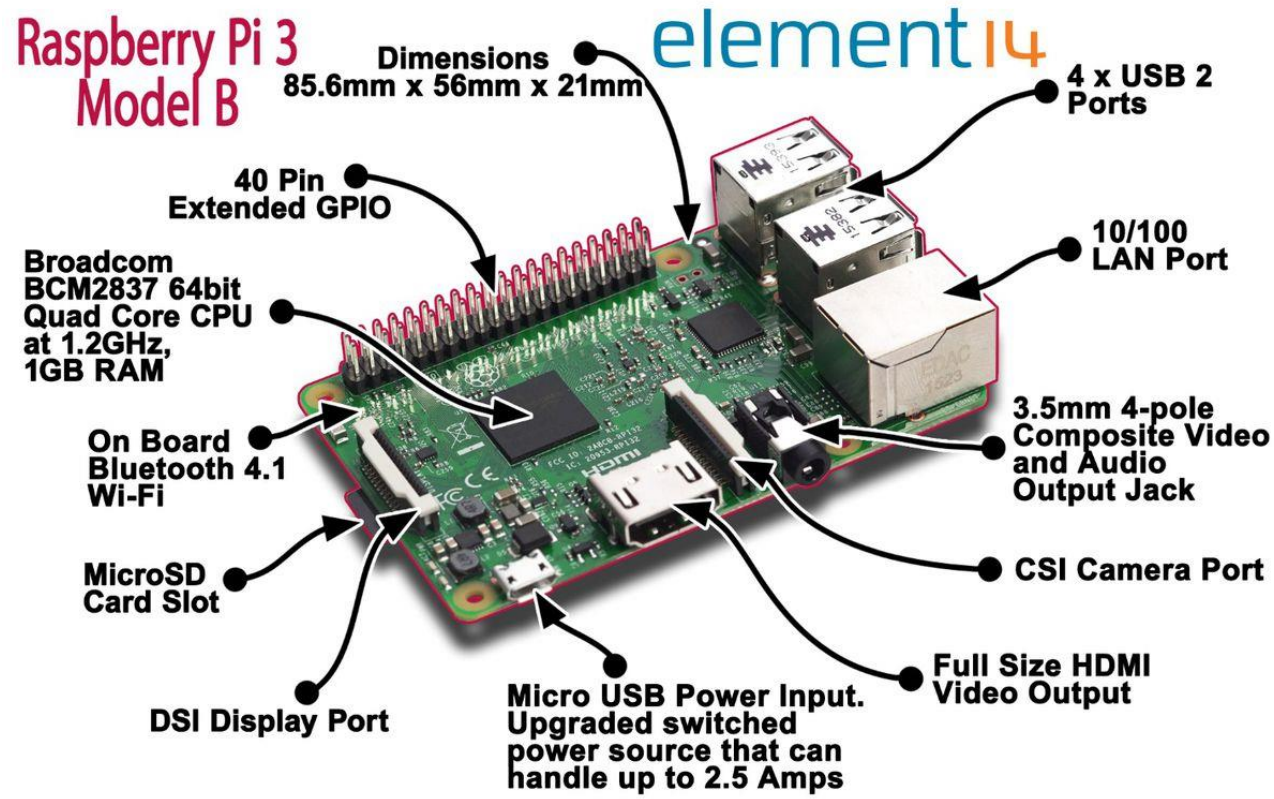
Samba server(for access of Raspberry files from Windows machine; needed for Eclipse Smart HomeDesigner)

xscreensaver (optional if you are using the display setup, to easy control screen blackening or screen savers)

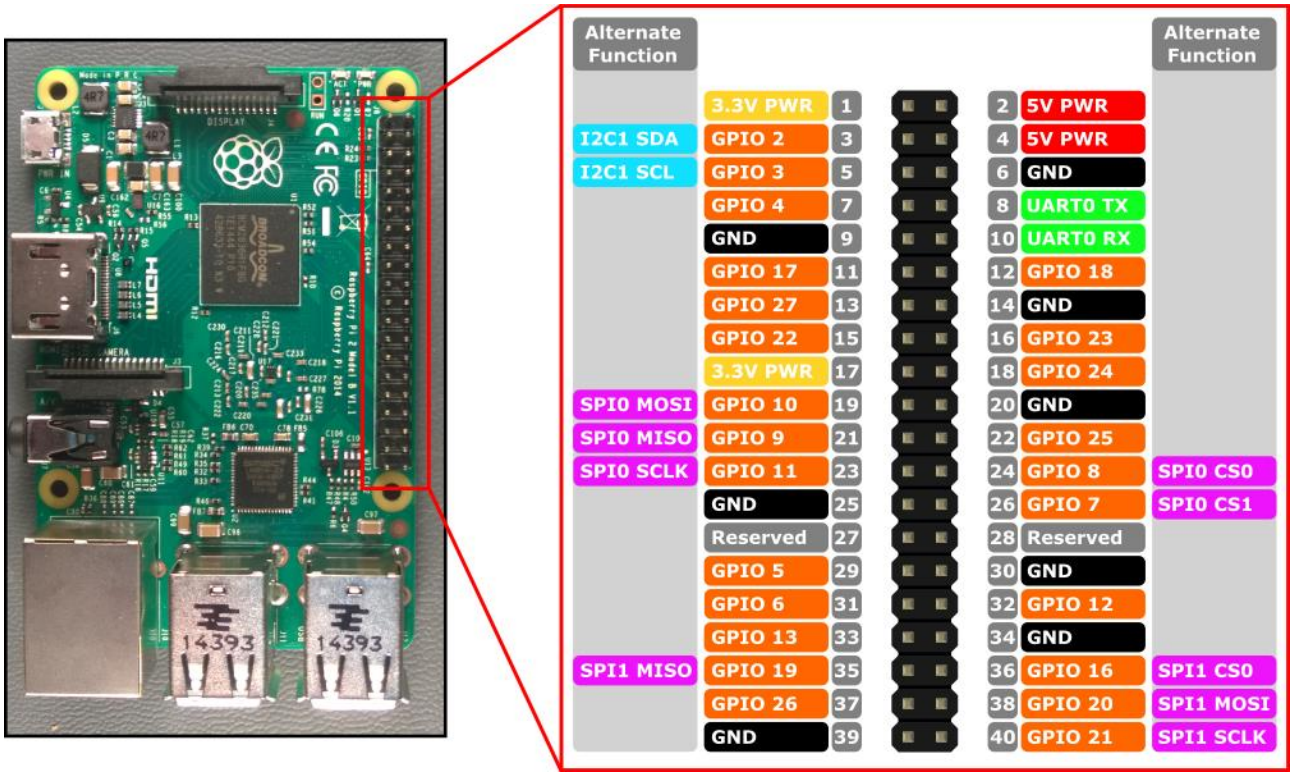
Chapter 3: Raspberry hardware and Raspbian OS installation

General information about Raspberry interfaces and GPIO pins:

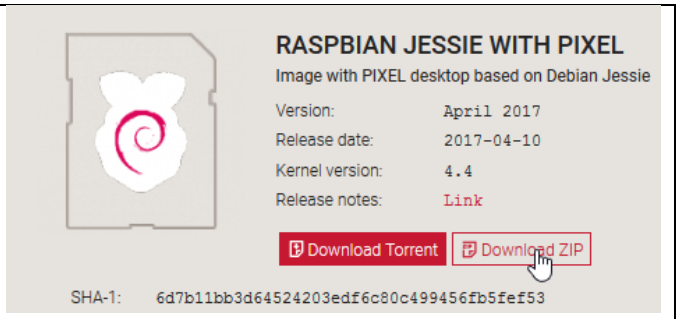
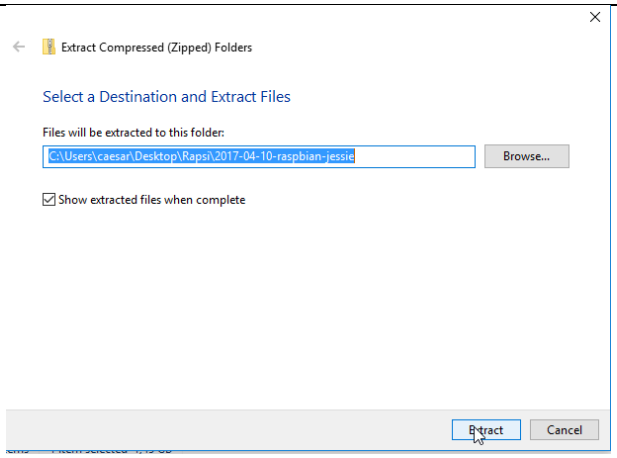
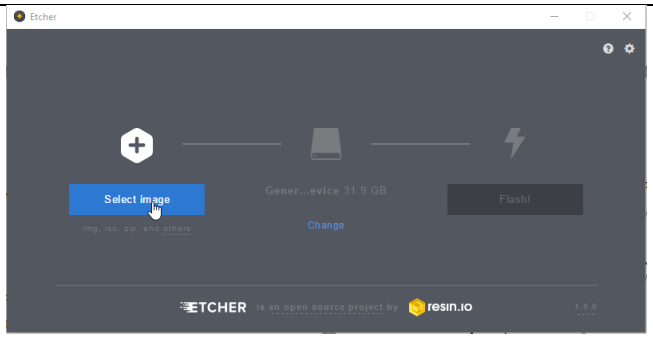
Raspberry input Overview:



Raspberry GPIO pin Overview:

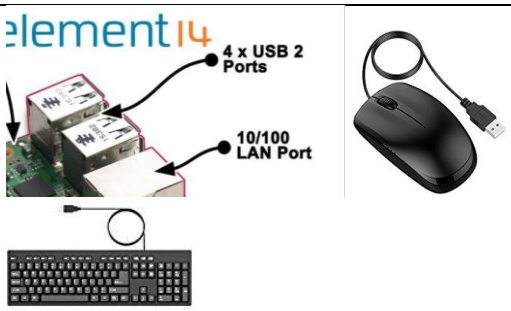
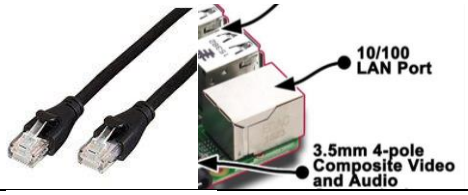
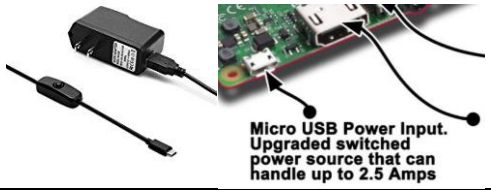


Preparing MicroSD card - writing Raspbian image to MicroSD card (PC required):

Download latest Raspbian Release (*.zip file) to a Windows folder	
Extract *.zip file to receive *.img file	
Use Etcher to write image to a MicroSD card 1. select image 2. select drive with MicroSD card plugged in to 3. start flashing	

Connecting the hardware to the Raspberry

Basic hardware setup:

Connect keyboard and mouse to the USB ports	
Insert the MicroSD card (pins facing the circuit board)	
Connect Raspberry with display using HDMI (optional, will **not** be needed if you are going for the 7” Raspberry display setup)	
Connect the Raspberry with Ethernet cable to your gateway (optional)	
Connect the power supply to the micro USB power input **Make sure that you have everything plugged in and the Raspberry is clear of any metal items since this step is already powering up your Raspberry.**	

Optional: Installation of 7” Raspberry display and display case:

The full tutorial will be found on:

https://www.element14.com/community/docs/DOC-78156/l/Raspberry-pi-7-touchscreen-display

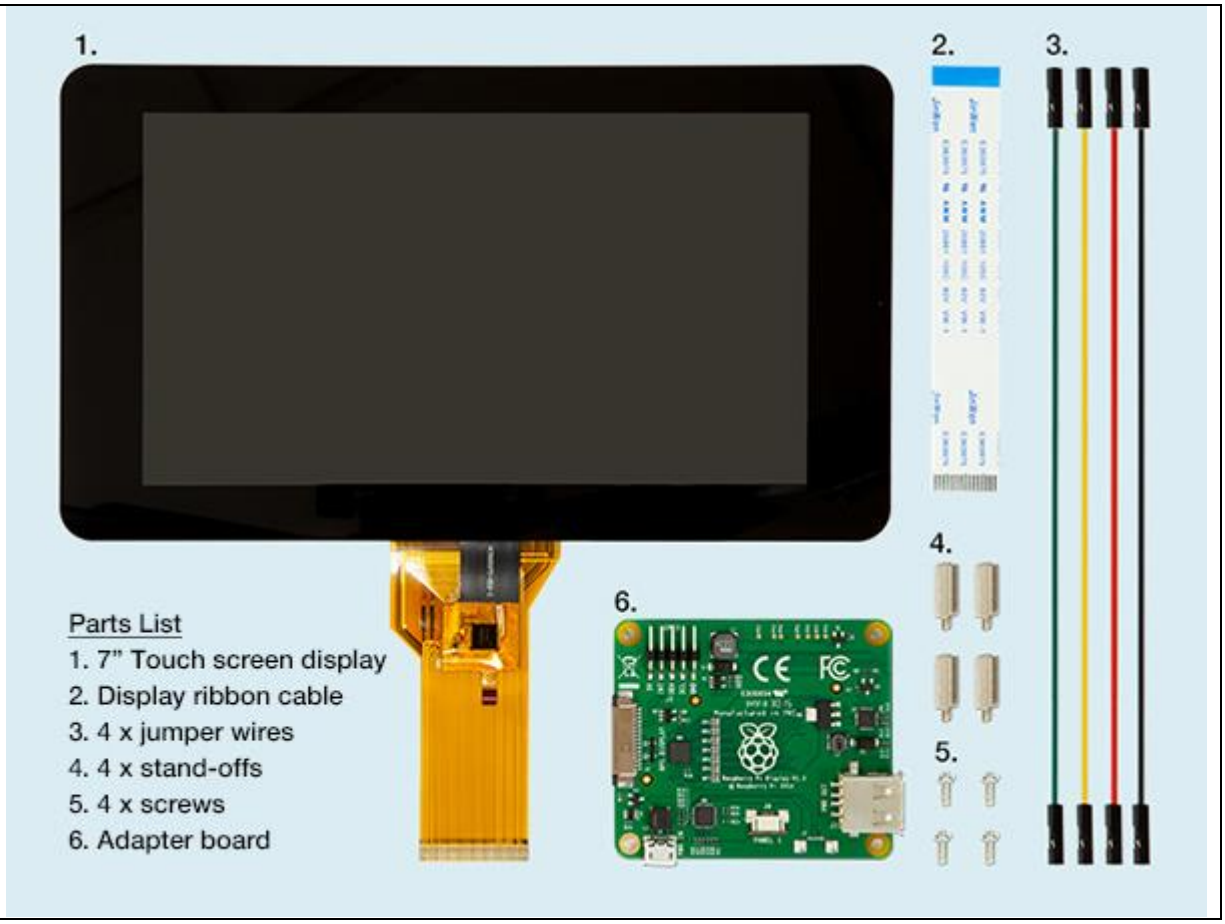
and a clip on YouTube:

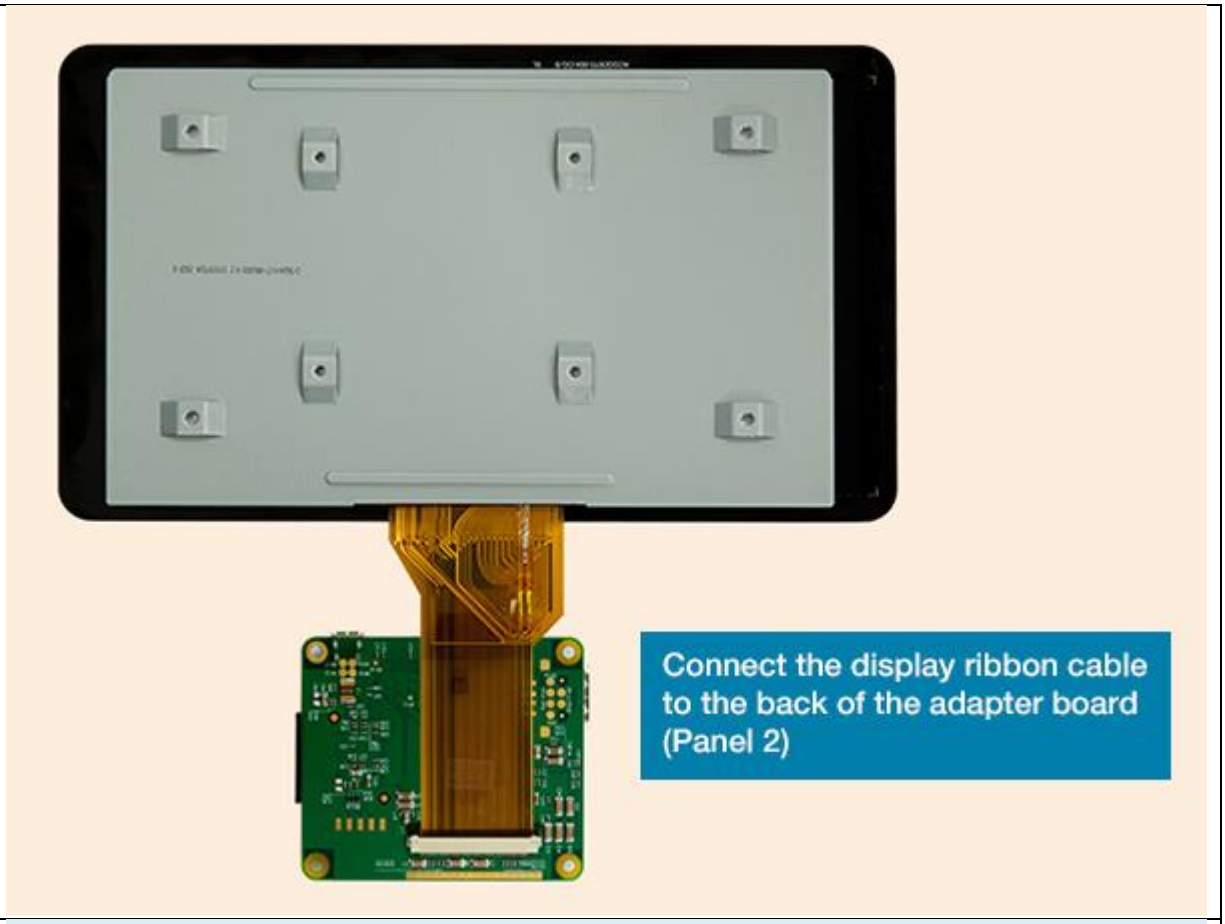
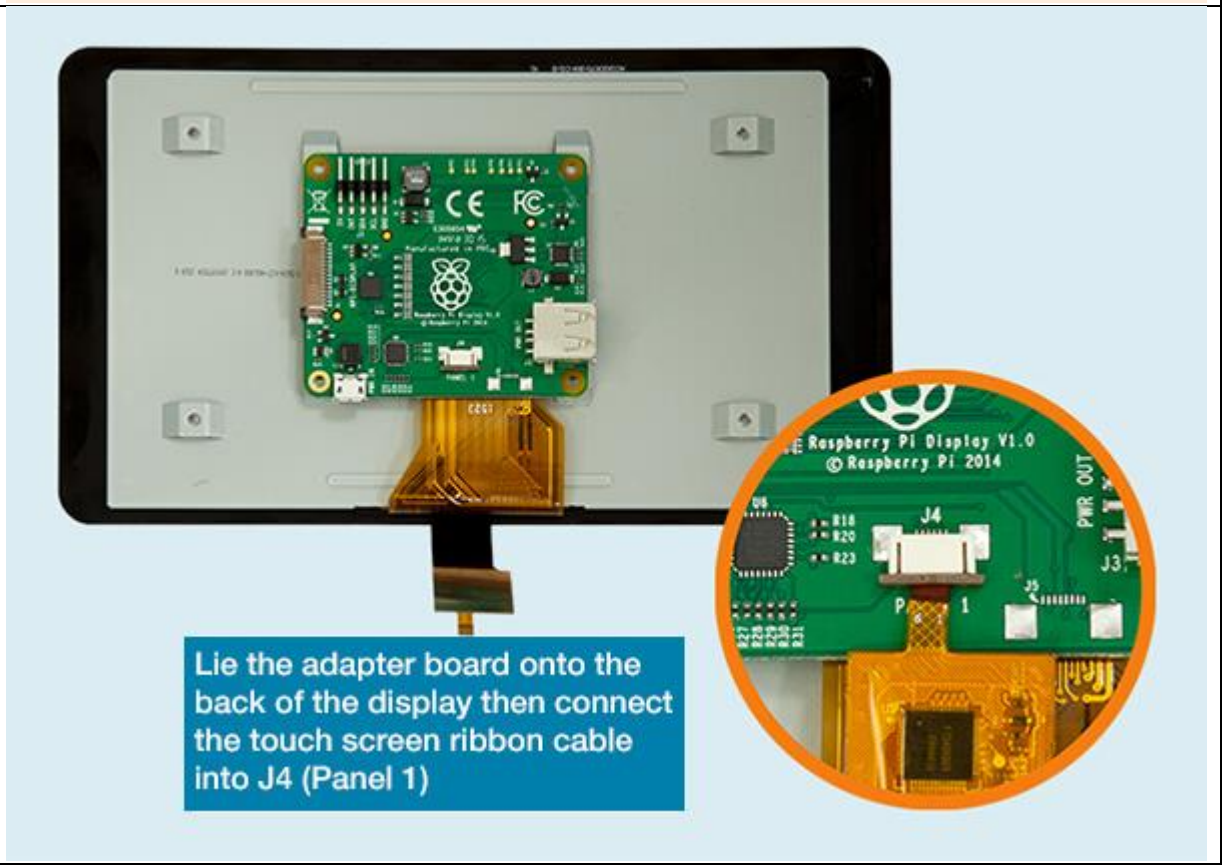
https://www.youtube.com/watch?v=tK-w-wDvRTg

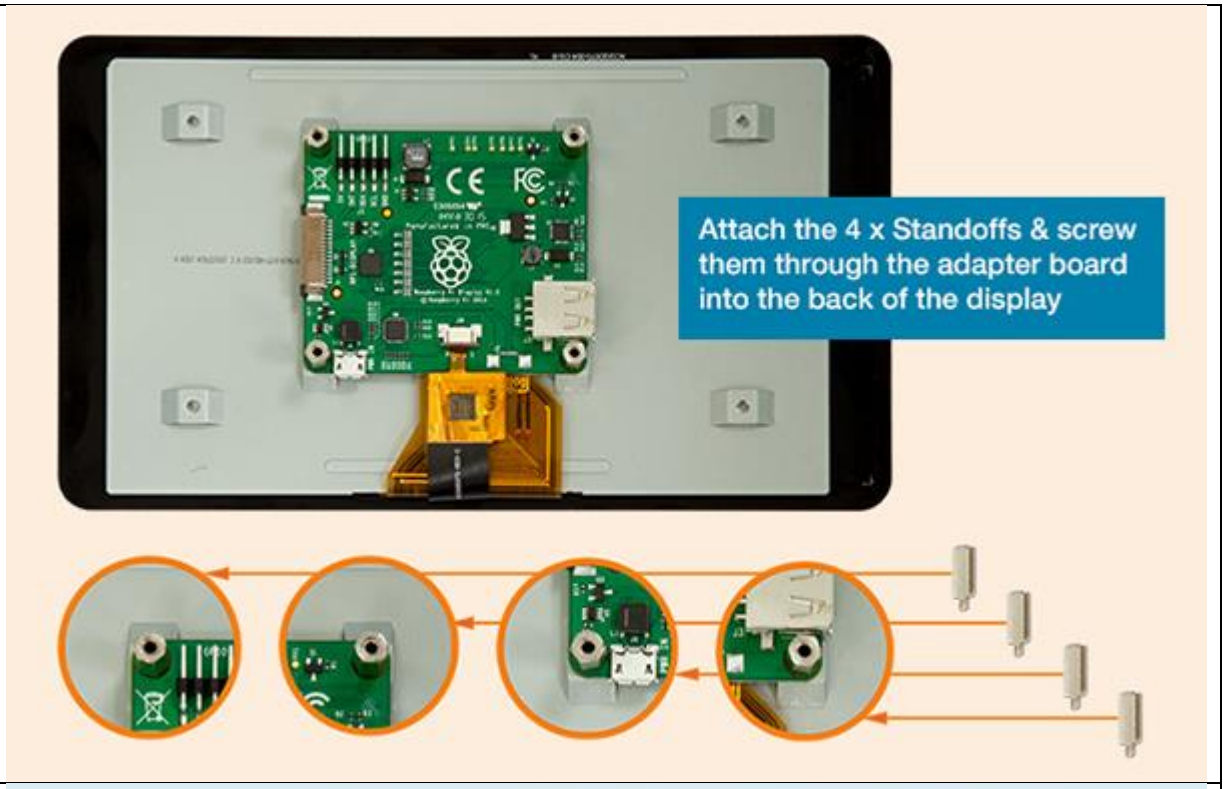
Remark: I had an issue with plugging in the power supply to the micro USB power input on the circuit board of the display (like shown in the video). The Raspberry was still showing me the low power symbol (lightening symbol on the upper right corner) SOLUTION: I had to plug in the power supply to the micro USB power input on the Raspberry itself. The display is now powered via the jumper cables. The standard display case is also allowing for both micro USB power inputs to be used.

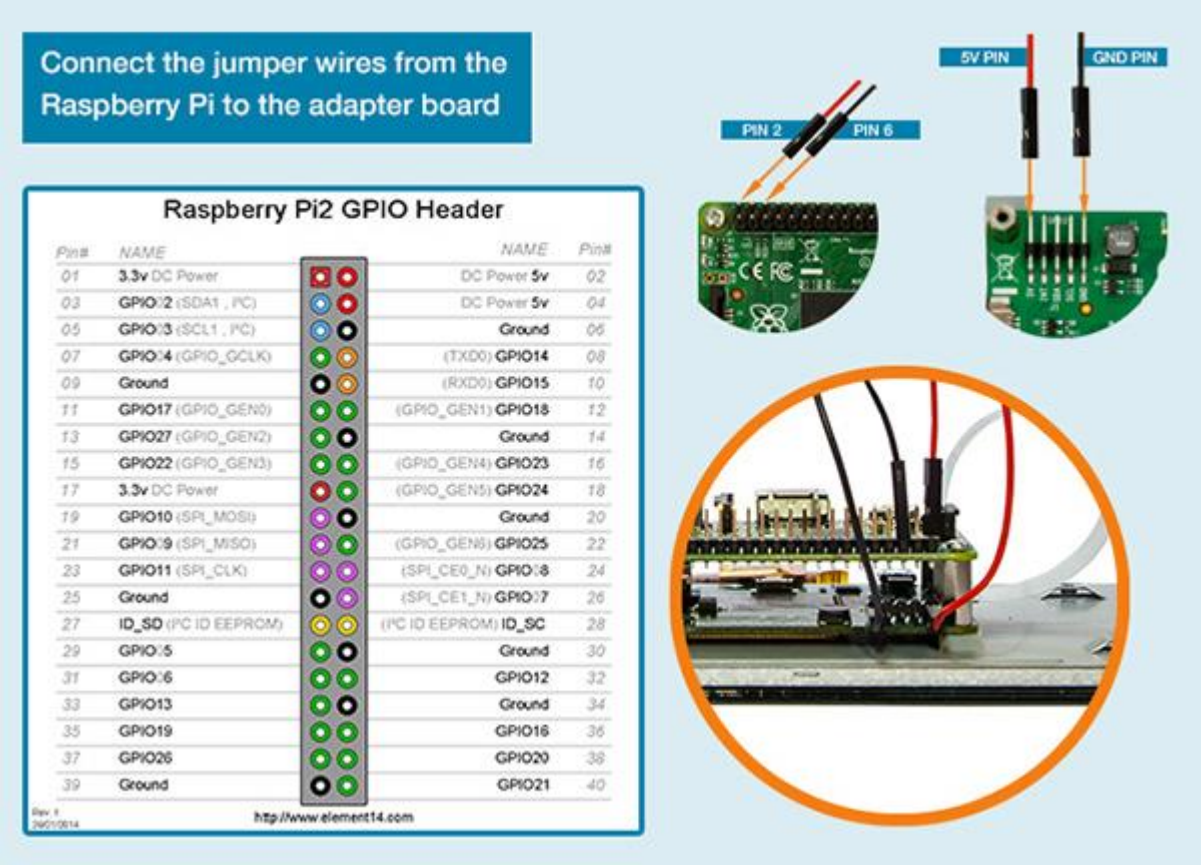
NOTE: If the image on the display is having the wrong orientation, you can rotate the image by changing the configuration of Raspbian (see tutorial section Initial configuration of Raspbian)

Display installation pictures:

Step 1	

Step 2	 Connect the display ribbon cable to the back of the adapter board (Panel 2)
Step 3	 Lie the adapter board onto the back of the display then connect the touch screen ribbon cable into J4 (Panel 1) Raspberry Pi Display V1.0 © Raspberry Pi 2014 PWR OUT J4 J3 J5 P 1

Step 4	 Attach the 4 x Standoffs & screw them through the adapter board into the back of the display
Step 5	 Attach the DSI ribbon cable to the Adapter Board

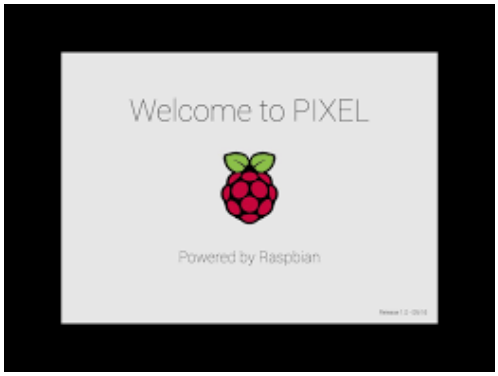
Step 6	
Step 7	 The two other path cables (green and yellow can be ignored)
Step 8	Assembly of the standard display. (Make sure you have inserted the MicroSD card since you won't have access to the slot as soon as you mounted the case!) Just pull the back plate off the case, insert the display including the mounted Raspberry (make sure that the path cables and the display cables are not crushed between case and board), tighten it with the 4 screws and put the back plate into place Here is a good clip on YouTube: https://www.youtube.com/watch?v=wpSxibZOmoo

Chapter 4: Raspbian basic configuration

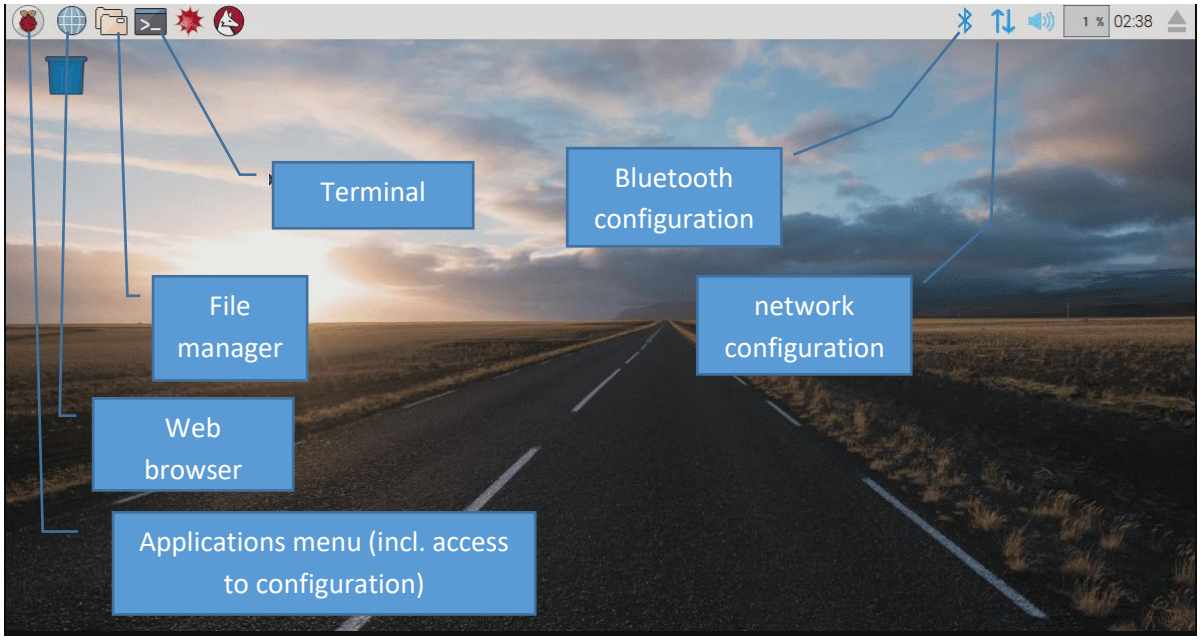
Starting up Raspberry or the first time – Raspbian PIXEL desktop

Since this tutorial is focussing on using the PIXEL GUI here are a few basic tips

Raspberry start-up screen

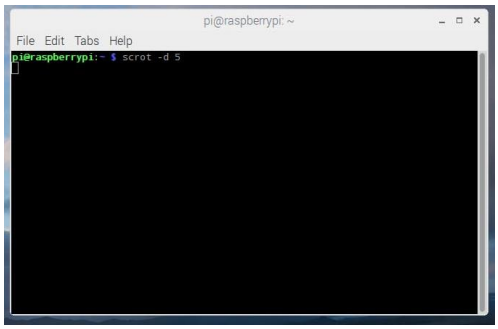


PIXEL basic desktop (including the programs used in this tutorial) not unlike other PC OS desktops:



NOTE: If you are working with the 7" Raspberry display setup you might need to flip/rotate the display orientation. Just check the section" Optional: Change display orientation" later in this chapter

Working with the Terminal:



NOTE: As soon as you have connected the Raspberry to the network you might find it easier to open the Terminal remotely using PuTTY. This also allows you to directly paste command lines from this tutorial into the Terminal. (Right click in PuTTY terminal is pasting the content of the clipboard into the terminal)

Basic terminal commands and functions:
The full list can be found on:

https://www.Raspberrypi.org/documentation/linux/usage/commands.md

`help`	Is showing you basic commands
`sudo othercommand`	is allowing you to run other commands as super user aka root user
`ls -la`	Shows the files in a directory incl. additional information
`cd` `cd` `cd ..` `cd directory` `cd /directory/directory`	Is changing the shell working directory. It can be used with attributes: No attribute => working directory is changed to user root directory. working directory is changed to directory one level above working directory is changed to the named `directory` inside the current directory working directory is changed to the directory defined by the full path `/directory/directory` .
`nano filename` `(sudo nano filename)`	Is stating a basic editor in the terminal to open or create a simple text or configuration file, mostly you have to add a `sudo` if you want to be able to write the files with root user rights. Closing the editor is done by ctrl+x and then choosing whether you want to save your changes or not

Initial configuration of Raspbian

The following steps make sure, that basic Raspbian configuration is done.

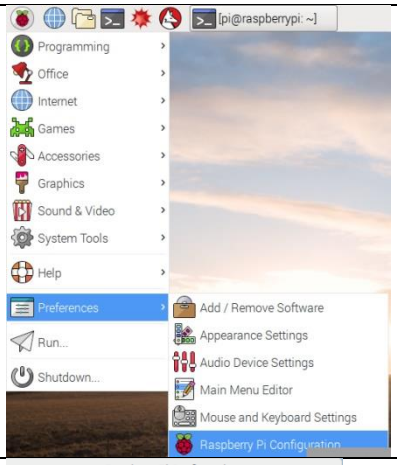
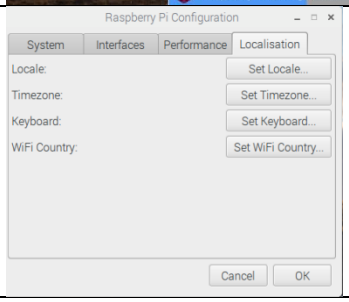
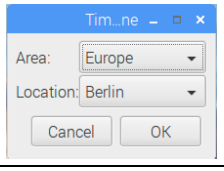
NOTE: There may be many tutorials in how to set-up and configure Raspbian and going into more details about user rights and other Raspbian features. This tutorial is showing the way which worked for my project aiming to run openHAB2 on the Raspberry.

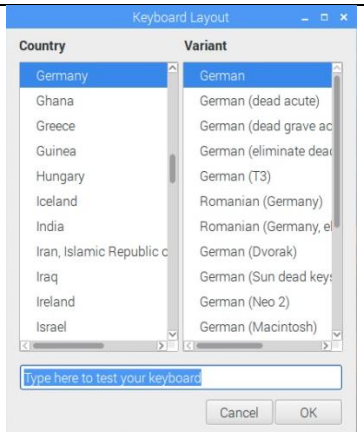
Since this tutorial is using the PIXEL GUI I always refer to the PIXEL way of configuring and only go back to the terminal way (text only) if it is required.

Localisation:

The 1.st thing you want to do is changing the localisation settings to make sure your keyboard layout and WiFi settings are matching.

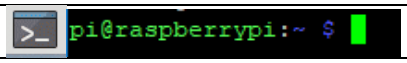
NOTE: Do not change the password before you have changed the keyboard layout since you might put in a different password than you expect (e.g. US qwerty vs. German qwertz results in “Raspberrz” instead of “Raspberry”)

Open Raspberry Pi Configuration Application menu -Preferences --Raspberry Pi configuration	
Go to tab Localisation	
Select Locale, Timezone, Keyboard and WiFi Country `Locale`: choose your language and country `Timezone`: choose timezone	 

Keyboard: choose keyboard language and specific keyboard layout WiFi Country: choose country specific WiFi settings	 
Accept the reboot	

Optional: Change display orientation

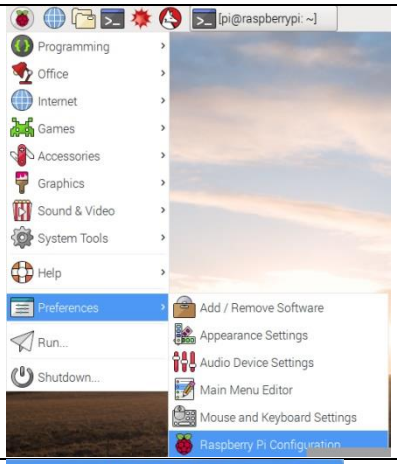
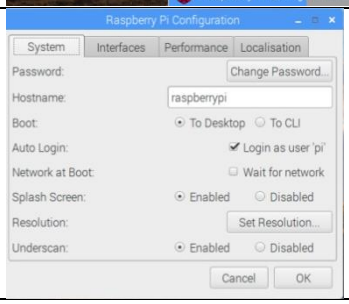
If you are working with the 7” Raspberry display setup you might need to flip/rotate the display orientation for specific cases

Open Terminal	
Open boot config.txt file in nano editor	`sudo nano /boot/config.txt`
Add the line at the bottom of the file: (This will flip the display orientation)	`lcd_rotate=2`
Optional: You can choose from different angles 0 degrees rotation or 90 degrees rotation or 180 degrees rotation or 270 degrees rotation or horizontal flip or vertical flip	 `display_rotate=0` or `display_rotate=1` or `display_rotate=2` or `display_rotate=3` or `display_rotate=0x10000` or `display_rotate=0x20000`
Exit and save the file	`<ctrl+x>` `<y>` `<Enter>`
Reboot the Raspberry for the changes to take effect	`sudo reboot`

Changing Password:

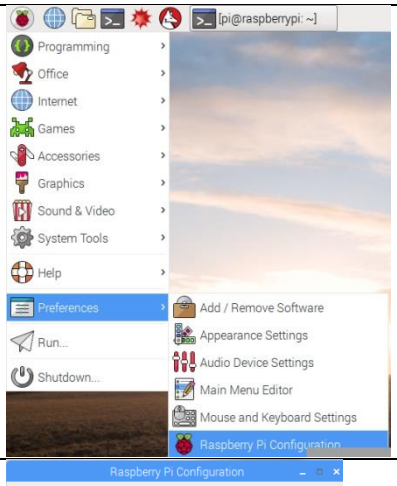
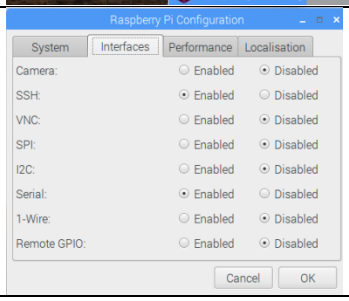
This is important to secure your standard Raspberry user “pi” before you connect the Raspberry to the network.

NOTE: Make sure you have changed the keyboard layout to your requirements before change the password.

Open Raspberry Pi Configuration Application menu -Preferences --Raspberry Pi configuration	
Go to tab System and select Change Password (Remark: you might not have the option Expand Filesystem as shown in the picture)	
Enter initial (for standard user “pi” it is “Raspberry” and your new password	Raspberry *yourpassword* *yourpassword*

Enabling interfaces:

This is required for the communication to the PC (SSH) and to the Z-Wave stick (Serial)

Open Raspberry Pi Configuration Application menu -Preferences --Raspberry Pi configuration	
Go to tab Interfaces	
Enable SSH (to access the Raspberry via Network) Enable Serial (to enable Serial Port for Z-Wave controllers)	SSH: Enable Serial: Enable

Connect Raspberry to network:

Either by plugging in a Ethernet cable or by connecting to a WiFi network:

Click on the network symbol	3 lines and 2 red crosses if no connection is available
Select WiFi network:	*yourwifi*
Enter WiFi password	*yourwifipassword*

Check the IP address of the Raspberry:

To do so you have to check the IP address of the Raspberry in the terminal

Start terminal by clicking on the icon	
use the command	`ifconfig`
Result: the terminal shows you the ip configuration and the IP addresses for the different connections	Ethernet cable: `eth0` *xxx.xxx.xxx.xxx* or WiFi: `wlan0` *xxx.xxx.xxx.xxx*

NOTE: You might want to set your IP address of the Raspberry to static, if you get problems with the lease time setting of your gateway (IP address is changing whenever you reconnect to the network)

Update / Upgrade Raspbian:

Raspbian is proving online updates so make sure that you have the latest installed before you go further in the configuration.

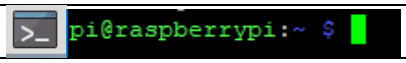
Start terminal by clicking on the icon	
use the command (be aware that the upgrade function will take several minutes to complete if you run it for the first time)	`sudo apt-get update` `sudo apt-get upgrade`

Optional Raspberry settings and configuration:

The following settings and configuration is just for additional information and might not be needed to setup openHAB2. Some of the settings and configuration might still be useful.

Check partition size on MicroSD card:

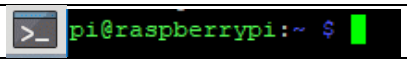
Make sure Raspbian is using the full capacity of the MicroSD card (normally while starting up Raspbian for the first time, it is done automatically and the Raspberry will restart automatically):

Open Terminal	
use the command	`sudo fdisk -l`
Result: the terminal shows you the partition size of the two partitions on the MicroSD card summing up to the total capacity	Example for 16 GB:
If the capacity is not completely used (e.g. you were using not a plain Raspbian image) you have to expand the partitions manually in the terminal configuration	
Start terminal configuration with command	`sudo raspi-config`
Select Option (Be aware that the option numbers might change in newer Raspbian releases)	`7 Advanced Options`
Select Option	`A1 Expand Filesystem` Prompt will tell you that the file system has been increased
Now select to exit the configuration	`<Finish>`
Allow reboot	`<Yes>`

Create a Desktop icon and link it to a application

To be able to create a Icon you have to first create a *.desktop file

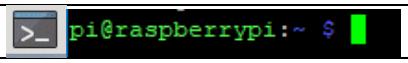
NOTE: This example is creating the desktop icon for the user “pi”

Open Terminal	
Go to the directory desktop for your “pi” user	`cd /home/pi/Desktop`
Create a specific desktop file using nano editor	`sudo nano yourdesktopfile.desktop`
Enter parameters into the file accordingly Name: *YourShortcutName* Comment: *Your Shortcut Comment* Icon: *YourIcon*.png Application for shortcut: YourShortcutApp	`Desktop Entry]` `Name=YourShortcutName` `Comment=Your Shortcut Comment` `Icon=/usr/share/pixmaps/YourIcon.png` `Exec=/usr/bin/YourShortcutApp` `Type=Application` `Encoding=UTF-8` `Terminal=false`
Exit and save the file	`<ctrl+x>` `<y>` `<Enter>`

Enabling root user:

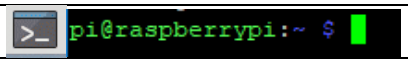
Since by default the “root” disabled it can’t be used. You might want to enable it for certain purposes like e.g. enabling the root user for samba file server to get full access to the directories from a PC (see chapter setup samba server)

NOTE: There is a reason for the “root” being disabled! Enabling the user is allowing full access to the Raspbian and therefore creating a security risk. Please always consider whether you really want to enable this user!

Open Terminal	
Since the user already exists you just have to set the password NOTE: you can also use the command to change the password later on	`sudo passwd root`
Just enter twice the new password for the “root” user	*`rootpassword`* *`rootpassword`*

Enabling remote SSH access for root user:

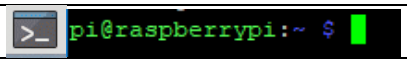
NOTE: There is a reason for the “root” not being enabled for SSH! Enabling the user for SSH is allowing full remote access to the Raspbian and therefore creating a significant security risk. Please always consider whether you really want to enable this user for SSH!

Open Terminal	
Open sshd.config file in nano editor	`sudo nano /etc/ssh/sshd_config`
Find the section `# Authentication` in the file	`# Authentication:` `LoginGraceTime 120` `PermitRootLogin without-password` `StrictModes yes`
And change the `PermitRootLoing` line to	`PermitRootLogin yes`
Exit and save the file	`<ctrl+x>` `<y>` `<Enter>`
Reboot the Raspberry for the changes to take effect	`sudo reboot`

Optional: Raspbian PIXEL screensaver (xscreensaver)

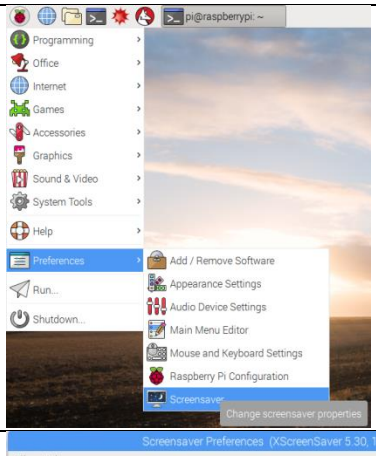
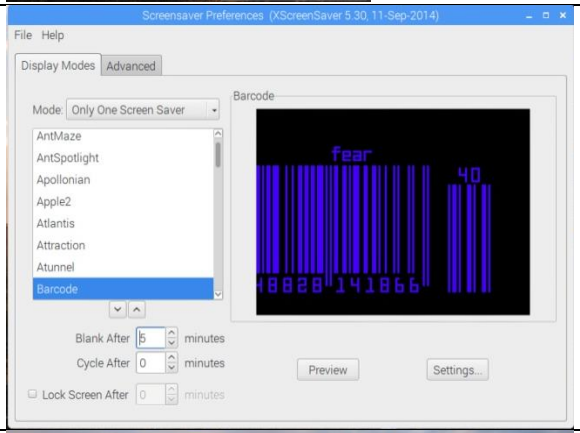
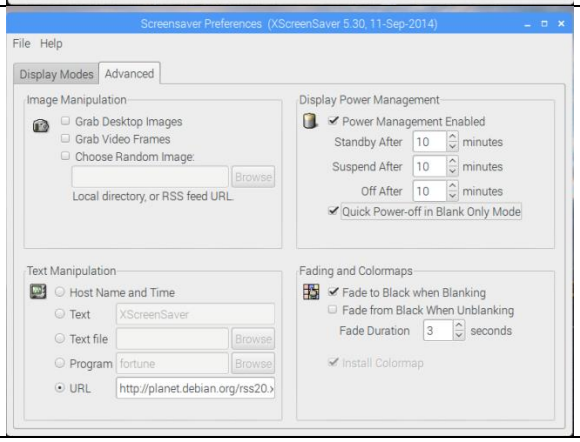
If you are working with the 7” Raspberry display setup you might want to use a screensaver as well.

Installation of xscreensaver:

Open Terminal	
Install xscreensaver and some additional screen saver themes	`sudo apt-get install xscreensaver xscreensaver-data-extra xscreensaver-gl-extra` `<y>`

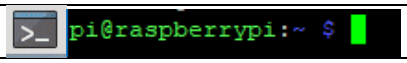
Configuration of xscreensaver:

I am showing an example configuration which is first switching on a screensaver and then turning off the display completely.

Open Screensaver Preferences Application menu -Preferences --Screensaver	
Configure Display Modes Mode: `<Only One Screen Saver>` Screensaver: `<Barcode>` Blank After: `<5>` minutes Cycle After: `<0>` minutes NOTE: This is just a sample configuration selecting one screensaver after 5 minutes	
Configure Advanced Display Power Management `<check>` Power Management Enabled Standby After `<10>` minutes Suspend After `<10>` minutes Off After `<10>` minutes `<check>` Quick Power-off in Blank Only Mode (Display Power Management) NOTE: This is just a sample switching off the screen after 10 minutes	
Close the Screensaver Preferences	

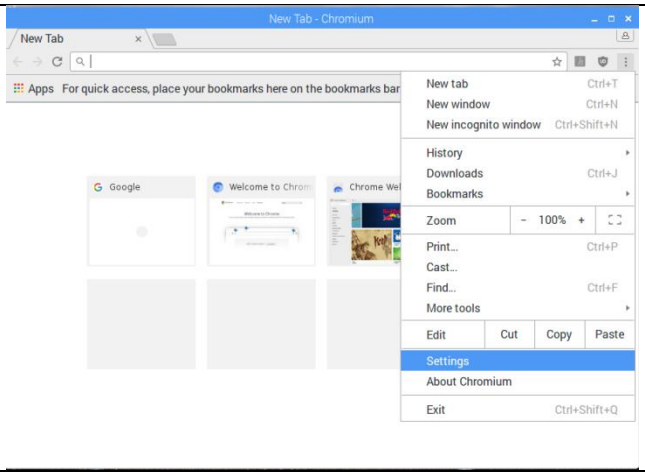
Optional: Start Chromium Web server on Raspbian boot

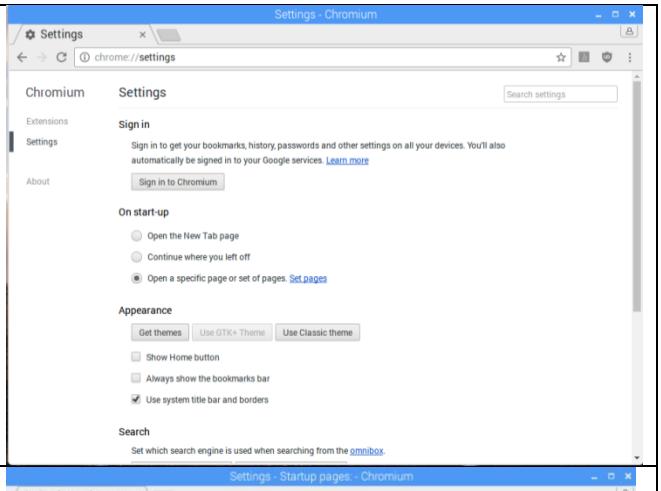
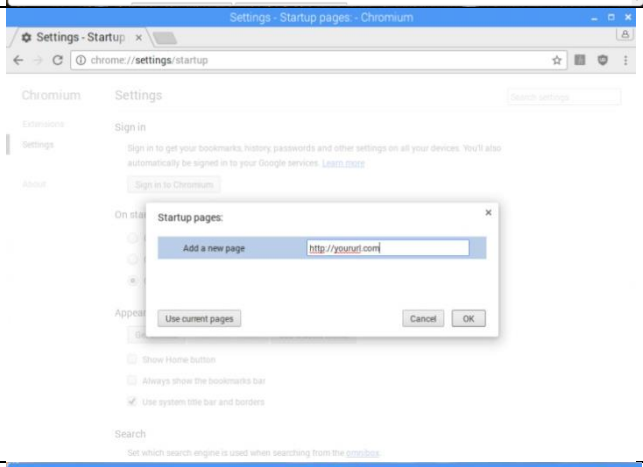
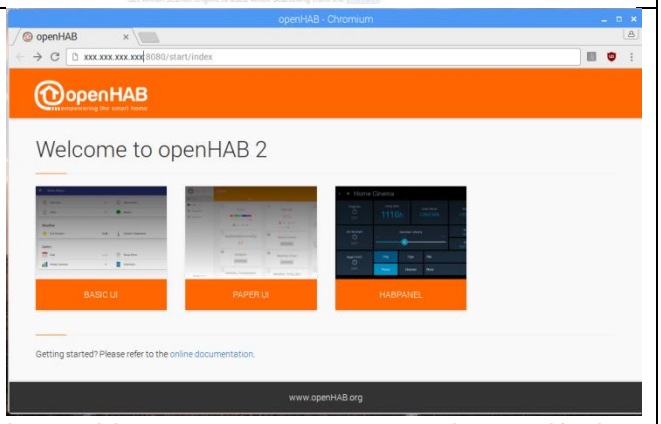
Configure the autostart file:

Open Terminal	
Open the autostart configuration file with nano editor	`sudo nano /home/pi/.config/lxsession/LXDE-pi/autostart`
Add the lines at the end of the file (ignoring error dialogs)	`@unclutter` `@chromium-browser --noerrdialogs`
Optional parameters: --kiosk (for full screen mode. NOTE: to exit full screen mode you have to press “Alt+F4” on the keyboard of the Raspberry, so you have to have a keyboard installed to exit this mode!) --incognito (for incognito mode of the browser) http://yourul.com (for selecting the URL directly in the configuration file. NOTE: selecting the URL via Chromium settings might be easier)	`@chromium-browser --noerrdialogs --kiosk --incognito http://yourul.com`
Exit and save the file	`<ctrl+x>` `<y>` `<Enter>`
Check if the browser is coming up after reboot	`sudo reboot`

Select the start URL for Chromium web browser:

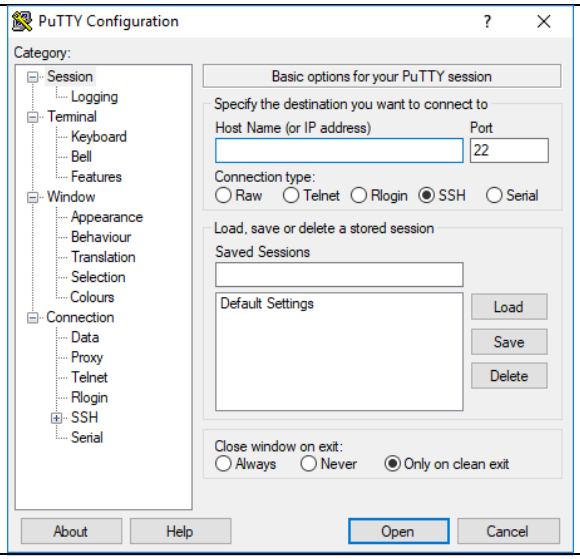
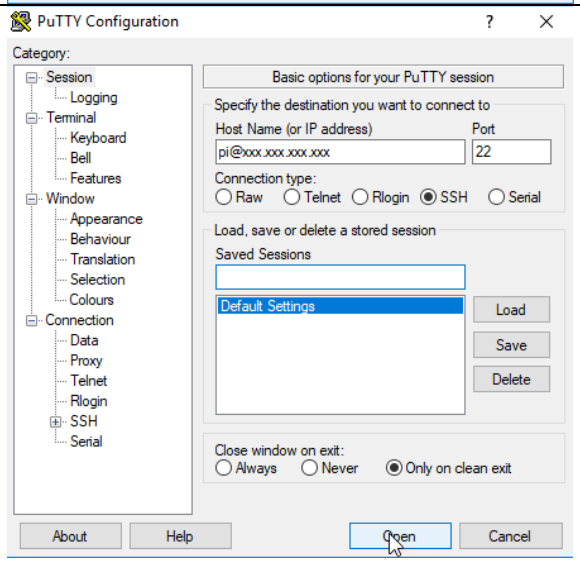
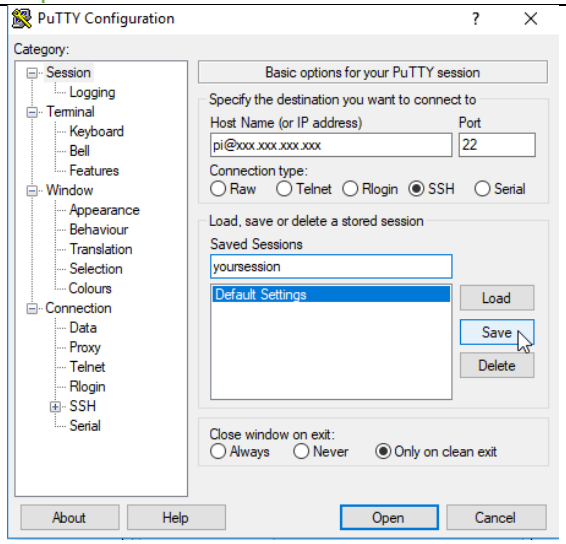
NOTE: You can also select the URL in the autostart file, but using the browser functionality is giving you a simpler access (no terminal) and you can check the result without rebooting

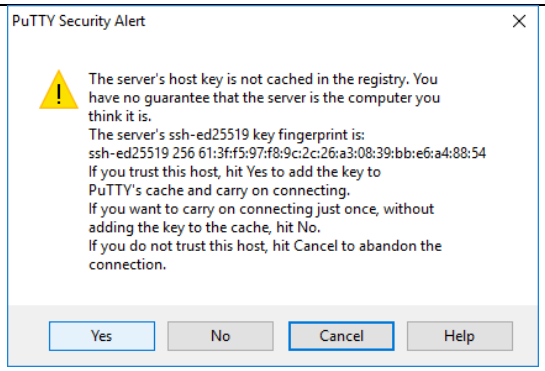
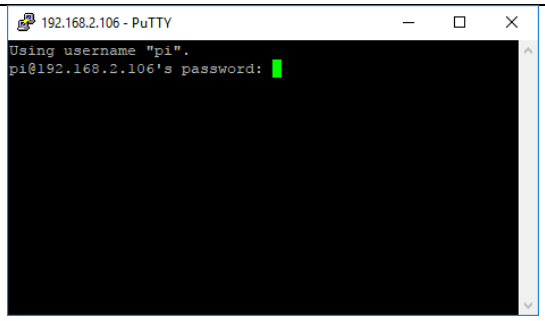
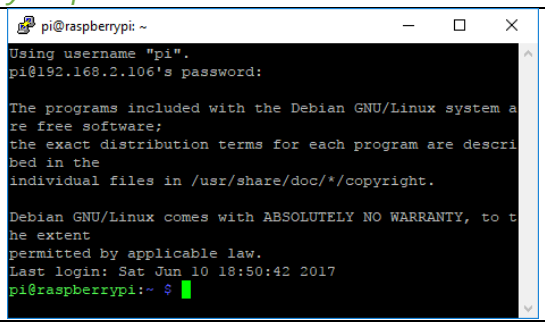
Open Chromium and go to the Settings (3 bullets icon)	

Select in the On start-up chapter the option <check> Open a specific page or set of pages And click on the link <Set pages> to enter the requested start URL	
Enter the requested start URL *http://yoururl.com* <OK> your URL Now Chromium is allowing you to enter an additional URL which you can ignore	
To check if it is working Close Chromium browser Open Chromium browser Result: *http://yoururl.com* should be loaded on startup (in my example I selected the openHAB2 GUI start page) Optional: If you have completed your openHAB2 configuration and want to use HABPanel as GUI you can just use the URL http://xxx.xxx.xxx.xxx:8080/habpanel/index.html#/ or even start specific pages in the HABPanel GUI (just use the URL shown in the browser when you access the HABPanel page)	 *http://xxx.xxx.xxx.xxx:8080/start/index*

Chapter 5: Setting up Raspbian for access via PC

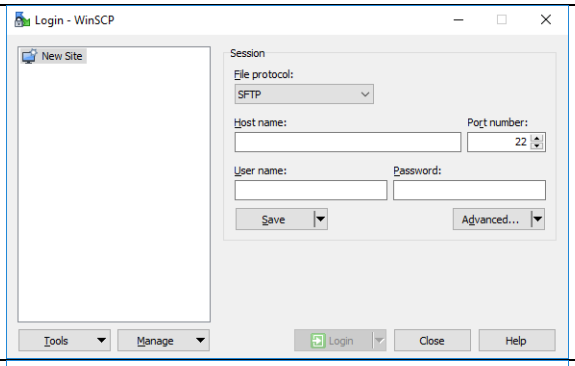
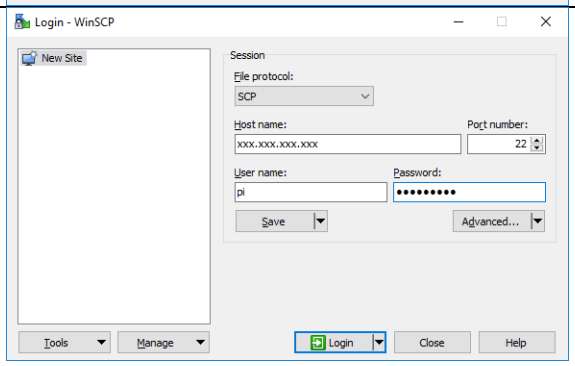
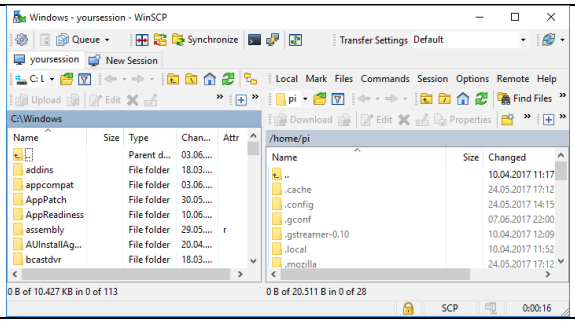
Connect to the Raspberry terminal your windows system using, KiTTY or PuTTY:

Open KiTTY or PuTTY on your PC	
Enter Hostname (pi@ in front of the IP is giving the user you want to use for connecting, in this case the standard user “pi”), Port and Connection type Select Open to launch the terminal	 Hostname: pi@xxx.xxx.xxx.xxx Port:22 Connection type: SSH <Open>
Optional save the session	 Saved session *yoursessionname*

	<Save>
On first connection an security alert is coming which you have to accept	 Yes
Now a terminal window is opening on our PC asking you to enter the “pi” user password	 *yourpassword*
The terminal window is now starting up in the user home directory	
You can now use the PC terminal window the same way you us the terminal on the Raspberry itself	

Optional: Connect to the Raspberry file system from your windows system using WinSCP:

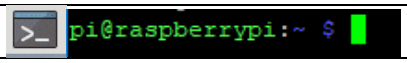
NOTE: The connection can only access the rights of the Raspberry user. So the standard user “pi” will not have the writing rights for multiple directories. For full access you have to use the user “root” (user needs to be enabled since it is disabled in standard setup, procedure shown later in the tutorial), but enabling this user for SSH access is opening up a significant security risk, so it is recommended to use as Raspberry based Samba server for full access to specific directories (shown later in the tutorial).

Open WinSCP on your computer	
Select: New Site Select File protocol: SCP Enter Host name: *xxx.xxx.xxx.xxx* Enter port: 22 Enter User name: pi (standard user with limited access to the file system) Enter Password for “pi”	 File protocol: SCP Host name: *xxx.xxx.xxx.xxx* Port: 22 User name: pi Password *yourpassword*
Optional save the Session	<Save>
WinSCP is launched showing the windows directory on the left side and the “pi” user home directory of the Raspberry on the right side of the window	

Setup Samba server

To have access to the Raspberry file system using the PC file explorer (it is needed to run Eclipse Smart Home Designer your PC) you have to setup a Samba server on the Raspberry first.

NOTE: You can also follow the alternative instructions for a openHAB2 centric installation at: http://docs.openhab.org/installation/linux.html#network-sharing

Open Terminal	
Make sure Raspberry is updated (optional)	`sudo apt-get update`
Download samba server to Raspbery (confirm with enter)	`sudo apt-get install samba samba-common-bin` `<Enter>`
Open the samba server configuration file in nano editor	`sudo nano /etc/samba/smb.conf`
Go to the end of the file and add following lines	`[RaspberryPiDirectories]` `comment = Your full access to Raspberry Pi directories` `path = /` `read only = no`
Optional Change the workgroup name if needed, otherwise uncomment and enable WINS support in the section	`# Windows Internet Name Serving Support Section:` `# WINS Support - Tells the NMBD component of Samba to enable its WINS Server` `#   wins support = no` `wins support = yes`
Exit and save the file	`<ctrl+x>` `<y>` `<Enter>`
Check the syntax of the samba configuration file. Result: there should be no error message(red) in the prompt	`testparm` `<Enter>`
Now you have to restart the services to reload the config file	`sudo systemctl restart smbd.service` `sudo systemctl restart nmbd.service`
Make sure that the services are running again without errors	`sudo systemctl status smbd.service` `sudo systemctl status nmbd.service`

Common samba server commands:

`sudo systemctl status smbd.service` `sudo systemctl status nmbd.service`	Check if all the services are running
`sudo systemctl restart smbd.service` `sudo systemctl restart nmbd.service`	Restart the samba services
`sudo systemctl stop smbd.service` `sudo systemctl stop nmbd.service`	Manually stop the samba services
`sudo smbpasswd -d sambausr`	If needed: disable a user for samba
`sudo smbpasswd -e sambausr`	If you need to enable a user for samba

Optional: Generic samba user setup

NOTE: Skip this section if you only want to use samba for openHAB2

Create a special user *sambausr* for the samba server so you not have to use the root or pi user to allow access to the directories on the Raspberry	`sudo adduser sambausr`
You have to enter your password *sambausrpassword* and optional information you can just leave empty and finally save with y	`Adding user `sambausr' ...` `Adding new group `sambausr' (1001) ...` `Adding new user `sambausr' (1001) with group `sambausr' ...` `Creating home directory `/home/sambausr' ...` `Copying files from `/etc/skel' ...` `Enter new UNIX password:` `Retype new UNIX password:` `passwd: password updated successfully` `Changing the user information for sambausr` `Enter the new value, or press ENTER for the default` `Full Name []:` `Room Number []:` `Work Phone []:` `Home Phone []:` `Other []:` `Is the information correct? [Y/n] y`
Map the user for Samba with	`sudo smbpasswd -a sambausr`
Enter the password *sambausrpassword*	`New SMB password:` `Retype new SMB password:` `Added user sambausr.`

NOTE: Instead of allowing the *sambausr* to have full access on the Raspbian file system to some specific folders by applying the `chown` command, you might consider enable the root user and use the root user to connect to the Raspberry from the file system (see chapter Optional Raspberry settings and configuration). But be aware that enabling the “root” user is creating a security risk!

Map the user for Samba with	`sudo smbpasswd -a root`
Enter the password *rootpassword*	`New SMB password:` `Retype new SMB password:` `Added user root.`

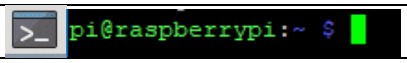
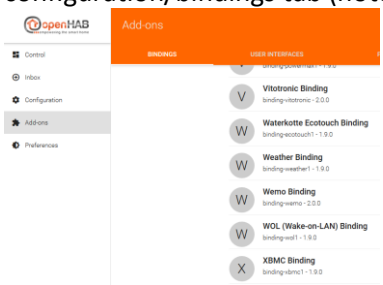
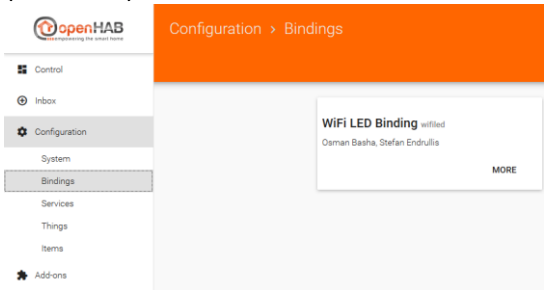
Optional: Mapping Raspbian samba directories to Windows (IOS and Linux mapping process can be found online):

NOTE: Skip this section if you only want to use samba for openHAB2

One time map the Raspberry folder to a windows drive (in this case Z) enter in the CMD Prompt (just put CMD in the search of Windows 10 to open the command prompt)	`net use Z: \\xxx.xxx.xxx.xxx\RaspberryPiDirectories /user:sambausr sambausrpassword /persistent:no`
Persistent map the Raspberry folder to a windows drive (in this case Z) enter in the CMD Prompt (just put CMD in the search of Windows 10 to open the command prompt)	`net use Z: \\xxx.xxx.xxx.xxx\RaspberryPiDirectories /user:sambausr sambausrpassword /persistent:yes`
You can also create a simple *.bat file for easy double clicking. Open the editor by just putting notepad in the search of Windows 10 Enter the line Save as *yourmapping*.bat	`net use Z: \\xxx.xxx.xxx.xxx\RaspberryPiDirectories /user:sambausr sambausrpassword /persistent:no`

Chapter 6: Installation of openHAB2 on Raspberry

This tutorial is only focussing on the package repository installation of the stable version and only on the add-ons for the listed hardware. All other installations are described on the openhab.org site installation for Linux: (http://docs.openhab.org/installation/linux.html#package-repository-installation) For the Raspbian you have to go for the “Apt Based Systems” part of it.

Open Terminal	
First, add the openHAB2 bintray repository key to your package manager and allow Apt to use the HTTPS Protocol	`wget -qO - 'https://bintray.com/user/downloadSubjectPublicKey?username=openhab' | sudo apt-key add -` `sudo apt-get install apt-transport-https`
I choose the stable Official (Stable) build The stable builds contain the latest official release with tested features.	`echo 'deb https://dl.bintray.com/openhab/apt-repo2 stable main' | sudo tee /etc/apt/sources.list.d/openhab2.list`
Next, resynchronize the package index:	`sudo apt-get update`
Now install openHAB2 with the following command:	`sudo apt-get install openhab2`
Optional but recommended: When you choose to install an add-on, openHAB2 will download it from the internet on request. If you plan on disconnecting your machine from the internet, then you will want to also install the add-ons package.	`sudo apt-get install openhab2-addons`
Since we were installing the stable version, we have to manually add the binding WIFILED used for the WiFi LED controller manually to the system. First you have to change to the add-ons directory. Than you have to download the latest version of the binding directly from the online repository NOTE: Later, this binding will not be available in the PAPER UI GUI under the Add-ons/Bindings tab, but will show up in the configuration/bindings tab (note here) 	`cd /usr/share/openhab2/addons` `sudo wget https://openhab.ci.cloudbees.com/job/openHAB2-Bundles/lastSuccessfulBuild/org.openhab.binding%24org.openhab.binding.wifiled/artifact/org.openhab.binding/org.openhab.binding.wifiled/2.1.0-SNAPSHOT/org.openhab.binding.wifiled-2.1.0-SNAPSHOT.jar` (but here!) 
If everything went well, you can start openHAB2 and register it to be automatically executed at system startup.	`sudo systemctl start openhab2.service` `sudo systemctl status openhab2.service` `sudo systemctl daemon-reload` `sudo systemctl enable openhab2.service`

Common openHAB2 service commands:

`sudo systemctl status openhab2.service`	Shows the status of openHAB2
`sudo systemctl start openhab2.service`	Start the service of openHAB2
`sudo systemctl stop openhab2.service`	Stops the service of openHAB2
`sudo systemctl restart openhab2.service`	Restarts the service of openHAB2
`sudo apt-get purge openhab2` `sudo rm /etc/apt/sources.list.d/openhab2.list`	This commands uninstall openHAB2 from your Raspbian

openHAB2 configuration for the samba server:

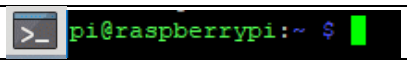
This is required to grant the PC based Eclipse Smart Home Designer access to the requested configuration folder on your Raspbian.

The shares are configured to be not open for guests nor to the public. Let's activate the "openhab" user as a samba user	`sudo smbpasswd -a openhab`
Enter the password *openhabpassword*	`New SMB password:` `Retype new SMB password:` `Added user openhab.`
Be aware, that creating and later using a specific user will ensure that permissions are honoured. Make sure, the "openhab" user has ownership and/or write access to the openHAB2 configuration files. This can be accomplished by executing:	`sudo chown -hR openhab:openhab /etc/openhab2`
Restart the samba service to allow the changes to be utilized	`sudo systemctl restart smbd.service`
One time map the Raspberry folder to a windows drive (in this case `Z`) enter in the CMD Prompt (just put `CMD` in the search of Windows 10 to open the command prompt)	`net use Z: \\`*`xxx.xxx.xxx.xxx`*`\RaspberryPiDirectories /user:openhab `*`openhabpassword`*` /persistent:no`
Optional: Persistent map the Raspberry folder to a windows drive (in this case `Z`) enter in the CMD Prompt (just put `CMD` in the search of Windows 10 to open the command prompt) NOTE: Now every time you boot up your PC it will try to connect to the Raspbian samba server.	`net use Z: \\xxx.xxx.xxx.xxx\RaspberryPiDirectories /user:openhab openhabpassword /persistent:yes`
Optional: You can also create a simple *.bat file for easy double clicking. Open the editor by just putting `notepad` in the search of Windows 10 Enter the line Save as *`yourmapping`*`.bat`	`net use Z: \\xxx.xxx.xxx.xxx\RaspberryPiDirectories /user:openhab openhabpassword /persistent:no`

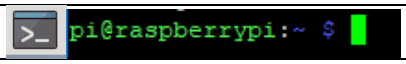
openHAB2 Privileges for Common Peripherals

An openHAB2 setup will often rely on hardware like a modem, transceiver or adapter to interface with home automation hardware. Examples are a Z-Wave, Enocean or RXFcom USB Stick or a Raspberry Pi add-on board connected to the serial port on its GPIOs. In order to allow openHAB2 to communicate with additional peripherals, it has to be added to corresponding Linux groups. The following example shows how to add Linux user openHAB2 to the often needed groups `dialout` and `tty`. Additional groups may be needed, depending on your hardware and software setup.

Adding openhab user to groupds dialout an tty

Open Terminal	
Enter command (This is adding the `openhab` user to the group `dialout`)	`sudo adduser openhab dialout`
Enter command (This is adding the `openhab` user to the group `tty`)	`sudo adduser openhab tty`
Optional: Enter command (f you are looking to enable sound privileges for openHAB2, it will also be necessary to add openHAB2 to the “audio” group.)	`sudo adduser openhab audio`

Granting java environment access to serial ports

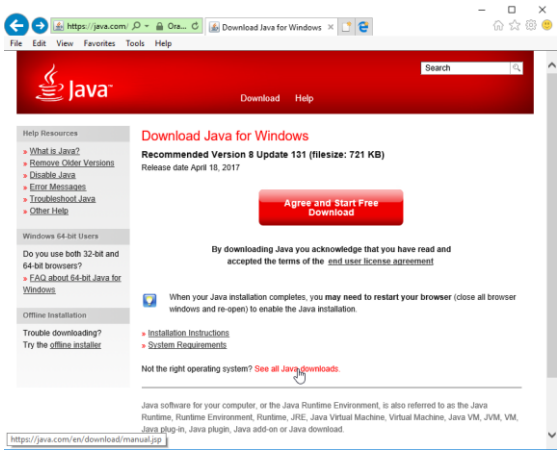
Open Terminal	
Change to directory	`cd /etc/default/`
Open openhab2 file in nano editor	`sudo nano openhab2`
Change the text from (nothing between the “”) To (something between the “”)	`EXTRA_JAVA_OPTS=""` `EXTRA_JAVA_OPTS="-Dgnu.io.rxtx.SerialPorts=/dev/ttyUSB0:/dev/ttyS0:/dev/ttyS2:/dev/ttyACM0:/dev/ttyAMA0"`
Exit and save the file	`<ctrl+x>` `<y>` `<Enter>`
Make sure the changes take effect by rebooting the Raspberry	`sudo reboot`

Chapter 7: Installation of Eclipse Smart Home Designer

(Optional but strongly recommended for easy editing of openHAB2 configuration files; incl. syntax highlighting)

The complete installation guide can be found on :
http://docs.openhab.org/installation/designer.html#setup

Installation guide for windows (Eclipse Smart Home Designer and Java Runtime Environment):

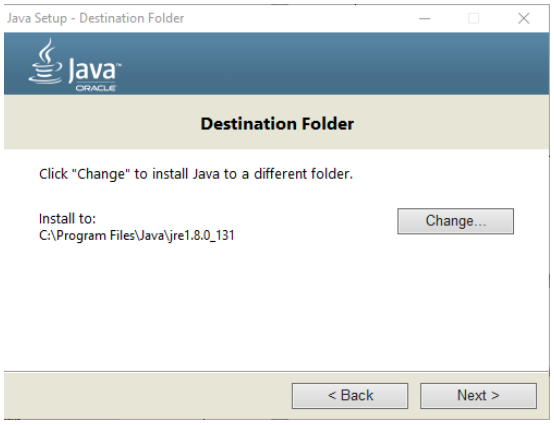
Download the Windows 64 bit version	http://eclipse.org/downloads/download.php?file=/smarthome/releases/0.8.0/eclipsesmarthome-incubation-0.8.0-designer-win64.zip
Optional: Download the 32 bit version since there are reports about the “stable” 64 bit version running not stable	http://eclipse.org/downloads/download.php?file=/smarthome/releases/0.8.0/eclipsesmarthome-incubation-0.8.0-designer-win.zip
Optional, but not recommended for beginner: Download a snapshot version	https://github.com/eclipse/smarthome/blob/master/docs/documentation/community/downloads.md#designer-builds
Unzip the downloaded file to a destination of your choice	*yourpcdrive:\yourdestionation*\eclipsesmarthome
Download the offline Java Runtime Environment. Go to the java homepage download section Select “See all Java downloads” Or directly go to: Download the 64bit version (something like jre-8u131-windows-x64.exe) or the 32bit depending on your system	https://java.com/en/download/  (https://java.com/en/download/manual.jsp) 

Install the Java Runtime Environment to the Eclipse Smart Home Designer folder
Start the Java installer
Select on the first screen “Change destination folder”

<Install>

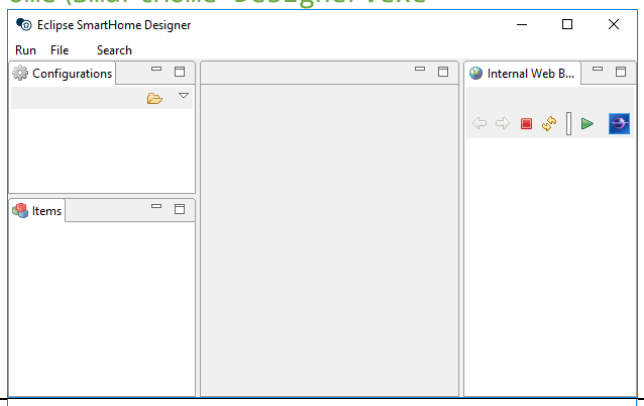
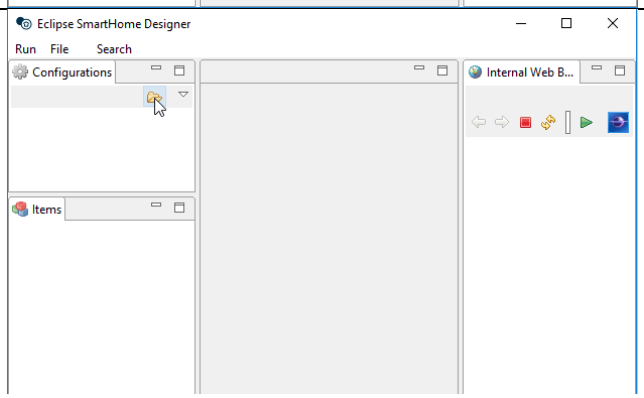
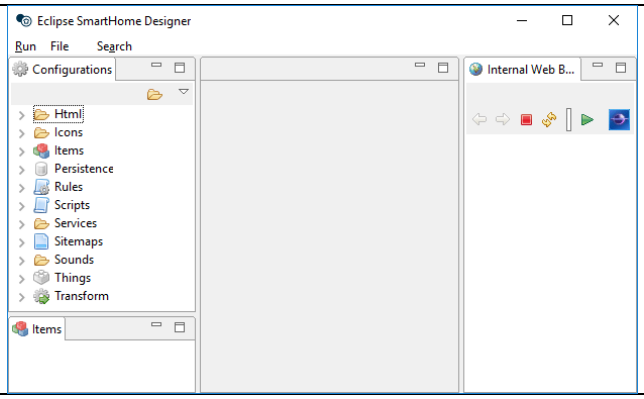
Change the folder to \jre inside your Eclipse Smart Home Designer folder *yourpcdrive:\yourdestionation\eclipseSmart Home* since this is the loaction your Eclipse Smart Home Designer is expecting the JRE.

NOTE: This is now delivering you a “portable” version of the Eclipse Smart Home Designer which just can be copied/moved to different locations or machines without the need of reinstalling.



<Change>
yourpcdrive:\yourdestionation\eclipsesmarthome\jre

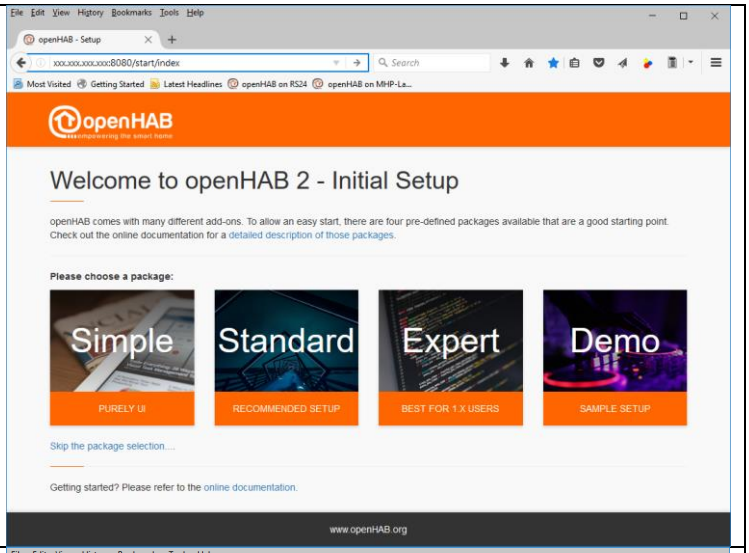
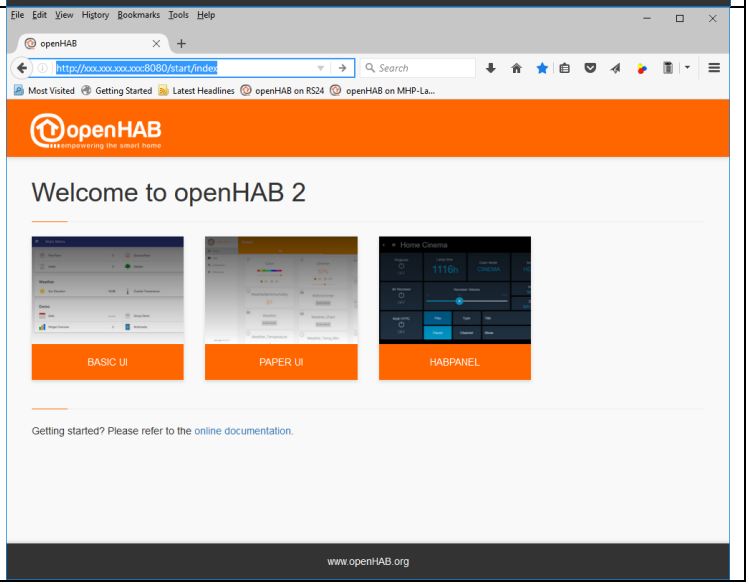
Launching first time:

Execute the SmartHome-Designer.exe	`yourpcdrive:\yourdestionation\eclipsesmarthome\SmartHome-Designer.exe` 
Link the Eclipse Smart Home Designer to the configuration folder on your Raspberry NOTE: Make sure that you have mapped the samba drive before (see openHAB2 setup for the samba server) If you have chosen a different drive letter in the mapping, just replace the Z:	 `Z:\etc\openhab2`
Eclipse Smart Home Designer should now recognize the file structure inside the configuration folder and augment the different folders with different icons	

Chapter 8: Initializing openHAB2 (finally: first startup)

To open the openHAB2 you have to access the web GUI with a browser on your PC or directly from your Raspberry, depending on your setup.

The URL will be `http://xxx.xxx.xxx.xxx:8080/start/index` filling in the IP of your Raspberry

First the GUI will ask you to select the initial setup configuration. This will install a standard set of GUIs in openHAB2	
After a few minutes of installation the standard openHAB2 start GUI will come up, showing you the icons for the pre-installed GUIs: `<BASIC UI>` `<PAPER UI>` `<HABPANEL>`	

Now you are ready for the configuration of your home automation project in openHAB2!

Chapter 10: General information about configuring openHAB2

Make sure to double check with the official website of openHAB2 since in the end this is where you will find the correct answers if something in this tutorial is not working:

http://www.openhab.org/

As a start you should read the beginners tutorial form beginning to end so you understand the basic concept of things, items, rules etc. and how they are mend to work together;

http://docs.openhab.org/introduction.html

openHAB (no 2) vs. openHAB2

One of the most important things I had to learn is that there is also an openHAB (no 2)! So here are a few things if found out to be considerable if you are using openHAB2:

- Always make sure, that you are looking at the right version of openHAB when it comes down to documentation, since a lot of older online documentation is refereeing to the openHAB (no 2). When a few things might be done the same/similar way in openHAB2, other things have changed and will not work in openHAB2
- You will also encounter two different ways of storing configuration in openHAB.
    - In openHAB (no 2).configuration was stored in files only
    - In openHAB2 you can still use the files, but also can use database storage for certain typs when you do the configuration with PAPER UI. This now might lead to some confusion since you will not be able to change e.g. items in PAPER UI which were configured using a text file. Also you would not have the correct syntax highlighting in the Eclipse Smart Home Designer is expecting file configuration only an items configured in PAPER UI will show up as errors.
    - But be aware that some configuration in openHAB2 still has to be done via text file like e.g. rules. You might find already some progress in the snapshot release of openHAB2, but I decided to base this tutorial on the stable release with limited functionality in PAPER UI database
- Regularly check the website of openHAB2 for news since a lot of new features are expected to be implemented.
- Meanwhile be not afraid to go and sign up the openHAB community: https://community.openhab.org and ask your questions there. I got replies to my problems within days, sometimes even within hrs. There is also a designated area for beginners.

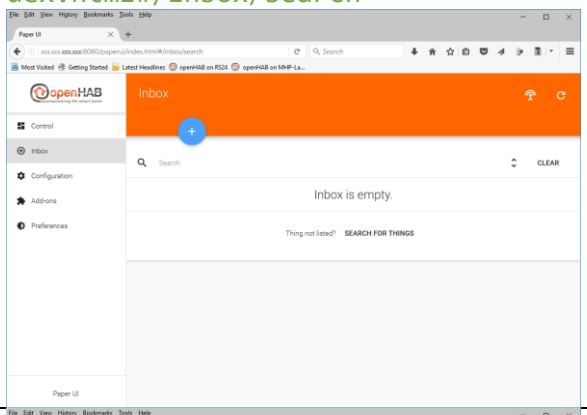
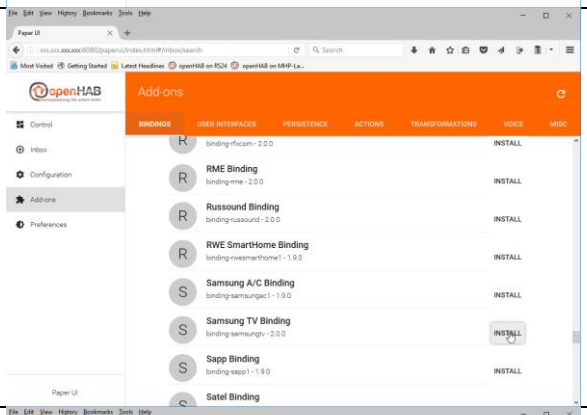
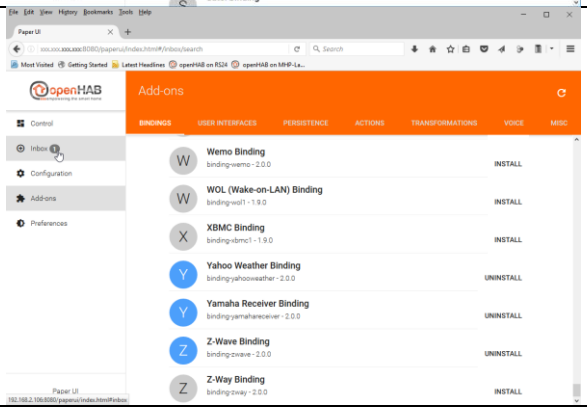
Chapter 10: Configuring openHAB2 using PAPER UI

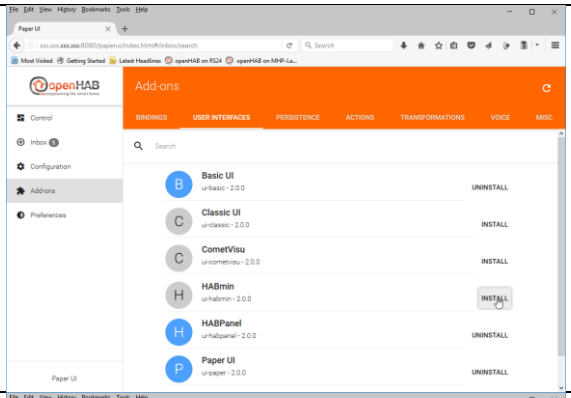
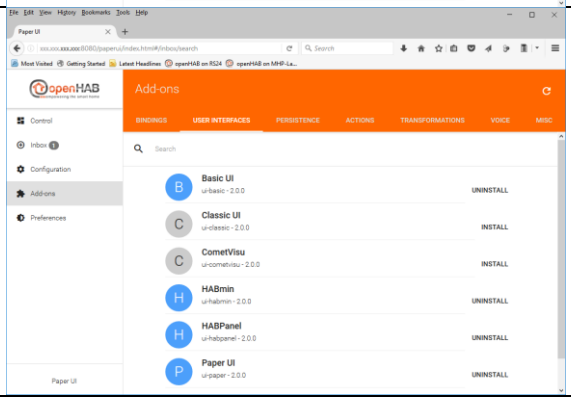
Since the online documentation is mostly referring to PAPER UI GUI I will try to stick to this GUI as long as possible.

- HABmin GUI will be needed for some Z-Wave installation
- HABPANEL GUI will be used to create the final user frontend for this project

NOTE: Since you will be regularly starting and switching the GUIs I highly recommend creating quick links in your browser for each GUI

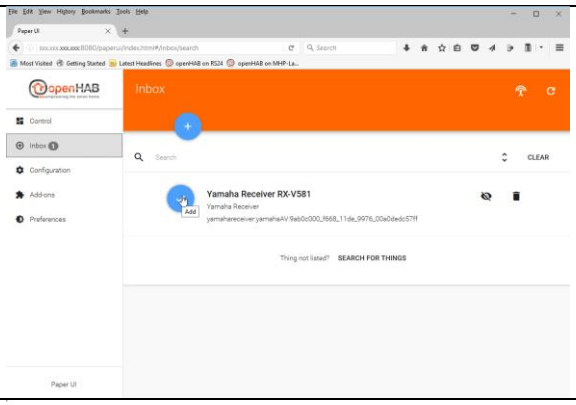
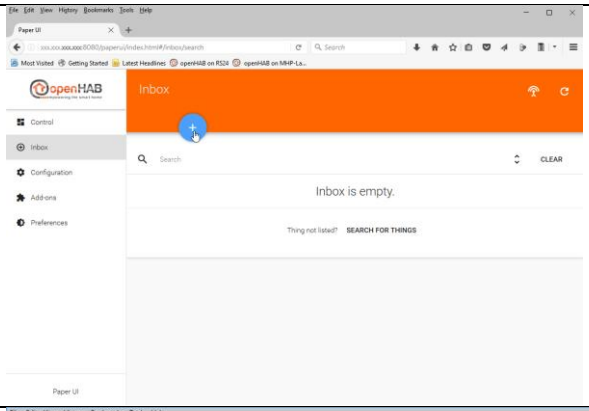
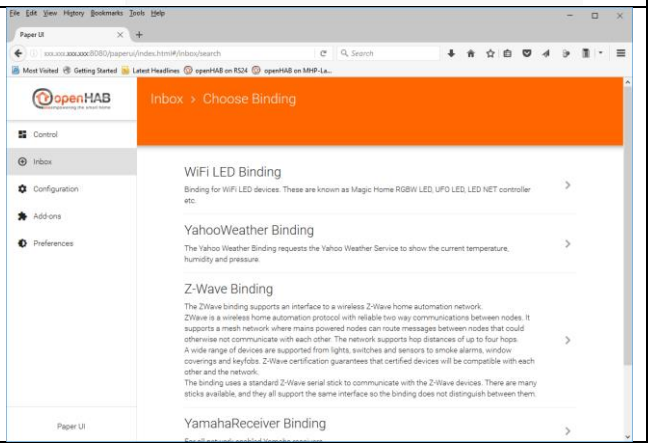
Installing Add-ons

Start PAPER UI	http://xxx.xxx.xxx.xxx:8080/paperui/index.html#/inbox/search 
Select -Add-ons --Bindings and install the Bindings: <Samsung TV Binding> <YahooWeather Binding> <YamahaReceiver Binding> <Z-Wave Binding>	
Result: The icons of the bindings should change to blue The install option should change to uninstall (sometimes you have to reload the page if the update is not coming up for a few minutes) NOTE: Since in my project case, the Yamaha Receiver is already connected to the same network than my Raspberry, I do already get a message in the inbox which is telling me, that a new thing was found	

Select -Add-ons --User Interfaces and install the User Interface HABmin which we will need for some Z-Wave stuff	
Result: HABmin GUI is installed	

General process of adding new things to the configuration

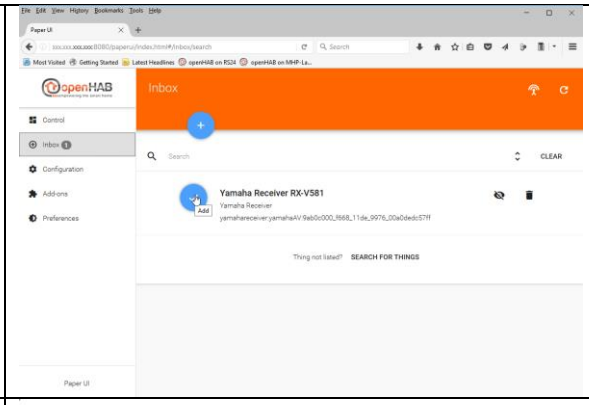
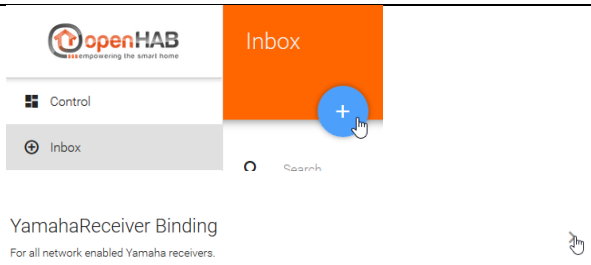
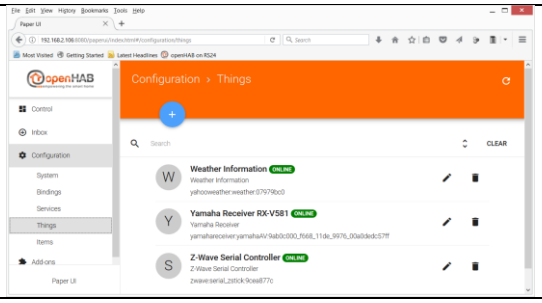
NOTE: If you can add new things to openHAB2 depends on whether they are connected to the network (if you use IP), whether they are included in the Z-Wave network of the Z-Wave controller or whether your Raspberry is online if you use online sourced like YahooWeather

Since in my project case, the Yamaha Receiver is already connected to the same network than my Raspberry, I do already get a message in the inbox which is telling me, that a new thing was found Now just click on the blue icon with the check mark to add this thing	
You can now change the name of the thing if you want to	 <ADD AS THING>
If the thing is not found automatically, you have to add it using the blue add icon (+) to manually add a thing.	
This will now show you all the installed Bindings which can be used to add more things. NOTE: You should find all bindings there which we installed in the step Installing Add-ons. Additionally you will find the <WiFi LED Binding>, which we manually installed in “Chapter 6: Installation of openHAB2 on Raspberry” since this is a snapshot Binding which we managed manually	

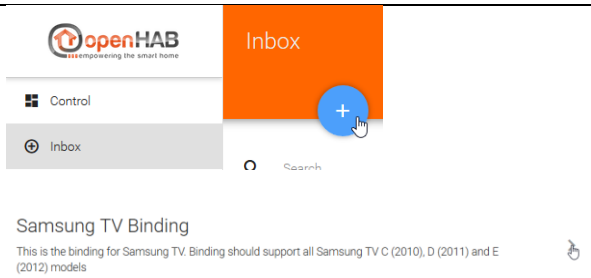
Adding local things connected via network (IP)

NOTE: Make sure that the device is connected to the Raspberry network via IP.

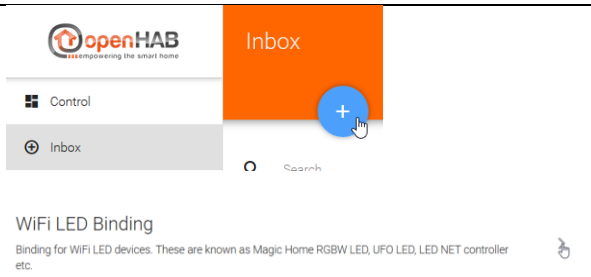
Adding Things using YamahaReceiver Binding:

The thing should come up automatically in you inbox as soon as you connect it to the network	
You can now change the name of the thing if you want to You can also change the name of the thing if you are planning to run with your own naming convention	 <ADD AS THING>
Optional: Select in your PAPER UI inbox the add blue add icon (+) Search for the thing manually using the YamahaReceiver Binding	
Switch to -Configuration --Things tab and check whether the now thing is available. The green icon online next to the name is indicating that openHAB2 is connected to the thing and information can be exchanged	

Adding Things using Samsung TV Binding:

The thing should come up automatically in you inbox as soon as you connect it to the network	Picture missing
You can now change the name of the thing if you want to and add the thing.	Picture missing <ADD AS THING>
Optional: Select in your PAPER UI inbox the add blue add icon (+) Search for the thing manually using the Samsung TV Binding	
Switch to -Configuration --Things tab and check whether the now thing is available. The green icon online next to the name is indicating that openHAB2 is connected to the thing and information can be exchanged	Picture missing

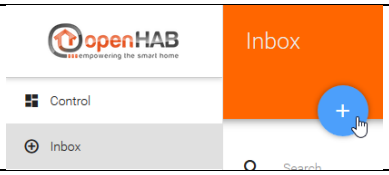
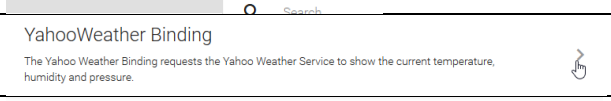
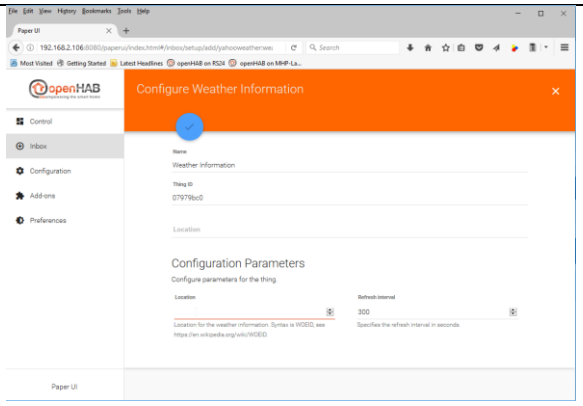
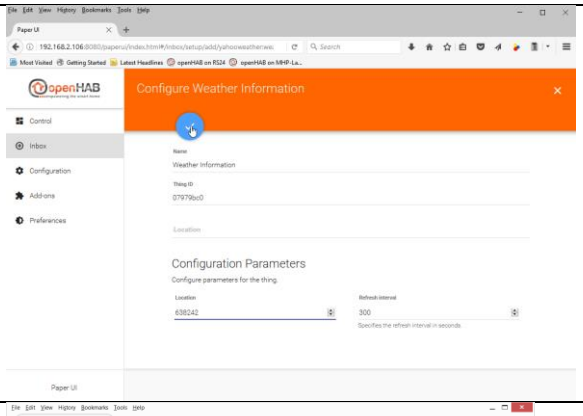
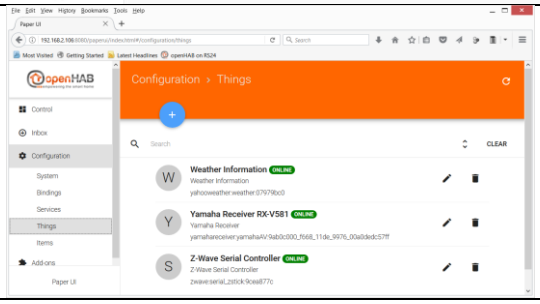
Adding Things using WiFi LED Binding:

The thing should come up automatically in you inbox as soon as you connect it to the network	Picture missing
You can now change the name of the thing if you want to and add the thing.	Picture missing <ADD AS THING>
Optional: Select in your PAPER UI inbox the add blue add icon (+) Search for the thing manually using the Samsung TV Binding	
Switch to -Configuration --Things tab and check whether the now thing is available. The green icon online next to the name is indicating that openHAB2 is connected to the thing and information can be exchanged	Picture missing

Adding online things connected via network (IP)

NOTE: Make sure the Raspbian does have online access pages.

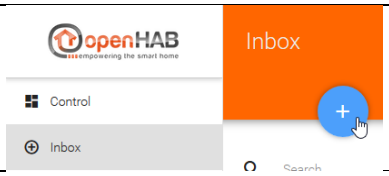
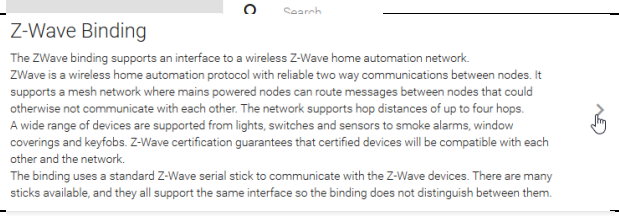
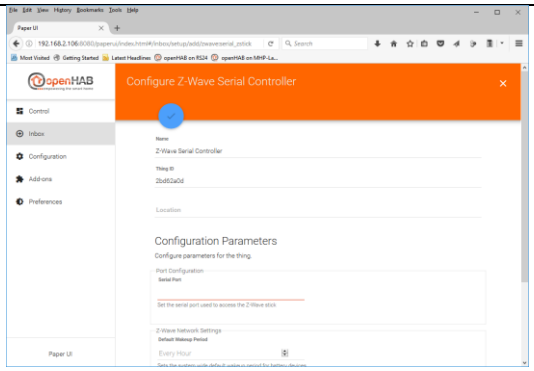
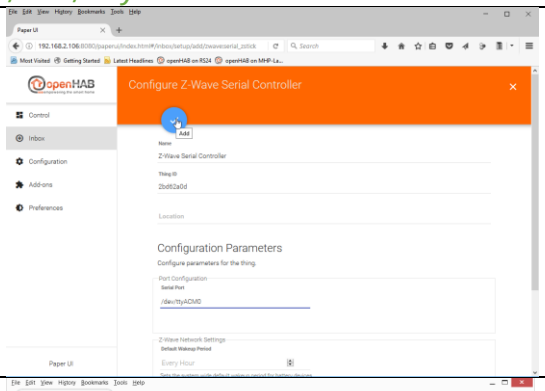
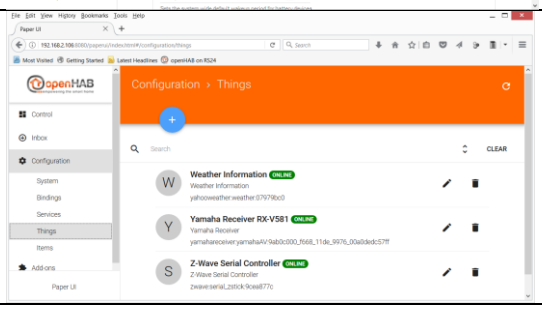
Adding Things using YahooWeather Binding:

Select in your PAPER UI inbox the add blue add icon (+)	
Select the Binding `<YahooWeather Binding>`	
Select the thing `<Weather Information>`	
Now you have to configure the thing In this case you have to enter the WOEID which is a 32-bit reference identifier of the location you want to see the weather information for. You can look up the WOEID on e.g. http://www.woeidlookup.com/ In this case we choose Berlin WOEID: 638242 More information on WOEID on: https://en.wikipedia.org/wiki/WOEID	 638242
Then just add the thing by clicking on the blue check icon	
Switch to -Configuration --Things tab and check whether the now thing is available. The green icon online next to the name is indicating that openHAB2 is connected to the thing and information can be exchanged	

Adding things connected via Z-Wave controller

NOTE: first you have to connect the Z-Wave controller as a thing. After this you will use HABmin to further include things into the Z-Wave network. These things should show up automatically in the inbox of PAPER UI.

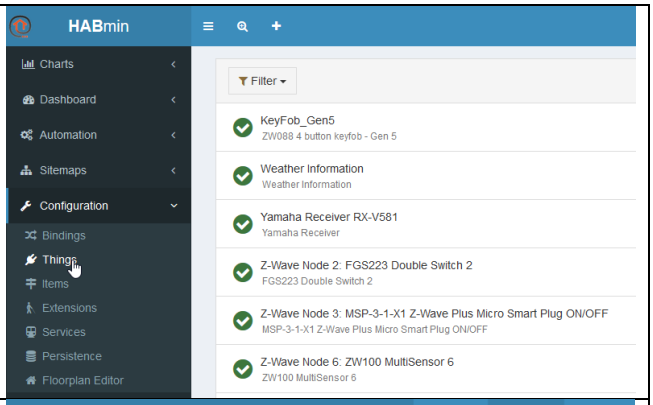
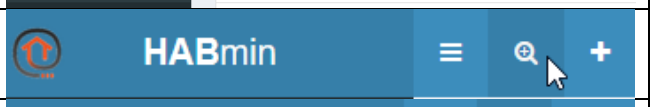
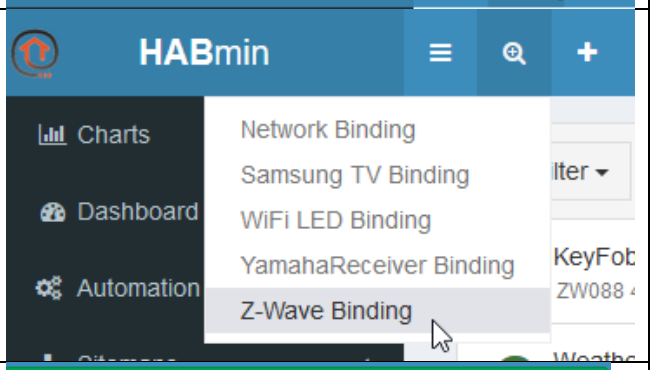
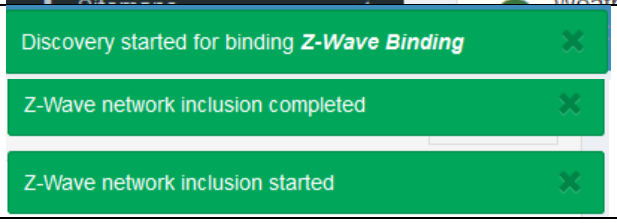
Adding Z-Wave controller

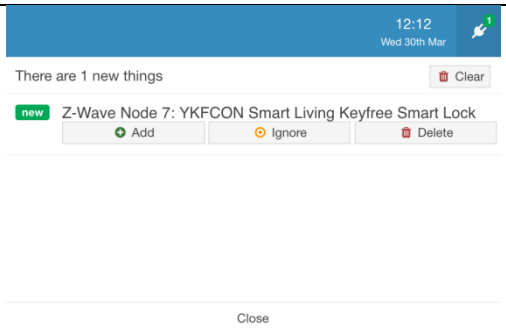
Select in your PAPER UI inbox the add blue add icon (+)	
Select the Binding `<Z-Wave Binding>`	
Select the thing `<Z-Wave Serial Controller>`	
Now you have to configure the thing In this case you have to enter the serial Port of the Z-Wave controller The standard port where the UBS-Z-Wave controller should come up it: `/dev/ttyACM0`	 `/dev/ttyACM0`
Then just add the thing by clicking on the blue check icon	
Switch to -Configuration --Things tab and check whether the now thing is available. The green icon online next to the name is indicating that openHAB2 is connected to the thing and information can be exchanged	

Adding Z-Wave things using HABmin UI

To include further devices into your Z-Wave network and make the available as openHAB2 things you have to use the HABmin inclusion functionality.

Please also consult the online documentation for general information about Z-Wave :
https://github.com/openhab/org.openhab.ui.habmin/wiki/ZWave-Device-Installation

Start HABmin Direct URL: `http://xxx.xxx.xxx.xxx:8080/habmin/index.html#/home` The select the Things tab -Configuration --Things HABmin will show you all the things which are available in PAPER UI as well	
To start the inclusion in HABmin you have to select the add thing icon of HABmin (magnifying glass)	
Then select the Z-Wave binding	
This now is triggering the inclusion mode of your Z-Wave controller	
Now you have to put your Z-Wave devices in inclusion mode to be detected for online detection (applicable when using UZB Z-Wave PLUS USB stick by Z-Wave.Me as a Z-Wave controller	Please refer to the Z-Wave devices manual how to put them in inclusion mode
Optional: Using the Aeotec by Aeon Labs Z-Stick Gen5 which is allowing for offline inclusion Since this Z-Wave controller stick is allowing for offline inclusion, you have to unplug the stick before you start the inclusion on HABmin and use the inclusion button on the stick to start the inclusion mode inclusion mode of your Z-Wave controller The plug in the stick again do the HABmin inclusion. You do not have to put your Z-Wave devices again in inclusion mode.	Please refer to the Aeotec by Aeon Labs Z-Stick Gen5 documentation for further details

And when a device is found it will be listed Just select the <Add> button to include this device as a thing for openHAB2 NOTE: some devices may come up as **"unknown device"**. You can still add the device and "cure" the information later. The normal reason for this is that the device was not able to finish the communication with HABmin during the inclusion process This might be very likely for battery powered devices since the will go to sleep mode after a certain amount of time. To cure this you just have to manually wake up the device (see device manual) multiple times (up to 10 times) before the communication is completed and the device is recognized as a proper thing in HABmin ATTENTION: Using the **Aeotec by Aeon Labs Z-Stick Gen5** almost every battery powered item will come up as **"unknown device"** since in the timespan you include the battery device to the stick and you do the HABmin inclusion after you plugged the stick back in will be to long an most of the devices will go back to sleep. You might prevent this by manually waking up the device again just seconds before you start the inclusion on HABmin. The other reason is that the device is not supported by the openHAB2 Z-Wave binding. You can check the supported devices at: http://www.cd-jackson.com/index.php/zwave/zwave-device-database/zwave-device-list	
The newly added thing should now be visible in the HABmin GUI things tab Now you can switch to the PAPER UI and find the new things in the inbox or check the new things in the things tab	

For details documentation on the Z-Wave Binding usage in HABmin consult:
http://www.cd-jackson.com/index.php/openhab/habmin/10-habmin-zwave-binding-initialisation

Creating items form things

The concept of Things, Channels, Items and Links

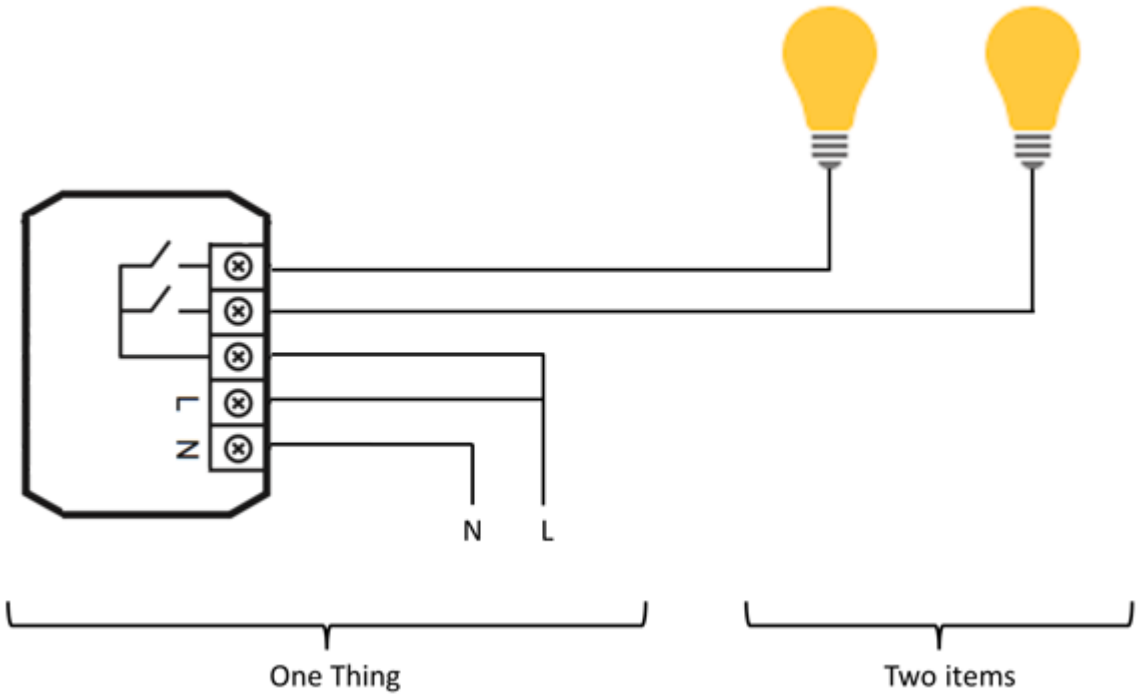
Before we start please read first the concept of Things, Channels, Items and Links which is directly tanken from the openHAB2 User Manual page http://docs.openhab.org/concepts/index.html#things-channels-items-and-links

Things are the entities that can be physically added to a system and which can potentially provide many functionalities at once. It is important to note that things do not have to be devices, but they can also represent a web service or any other manageable source of information and functionality. Things provide their functionality through a set of Channels. Channels are “passive” and can be regarded as a declaration of a Thing, what it can offer. It is up to the individual setup, which of the Channels are actively used through Items (see below).

Items represent (fine-grained) functionality that is used by applications - as user interfaces or automation logic. Items have a state and they can receive commands.

The glue between Things and Items are Links. Links are associations between exactly one Thing Channel and one Item. If a Channel is linked to an Item, it is “enabled”, which means that the functionality that the Item represents is handled through the given Channel. Channels can be linked to multiple Items and Items can be linked to multiple Channels.

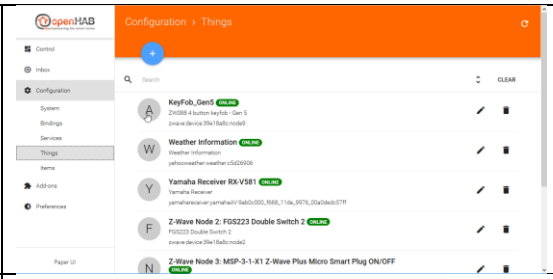
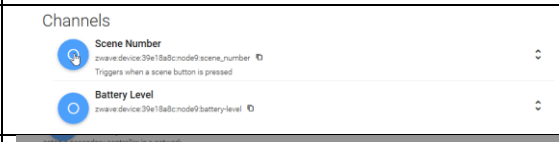
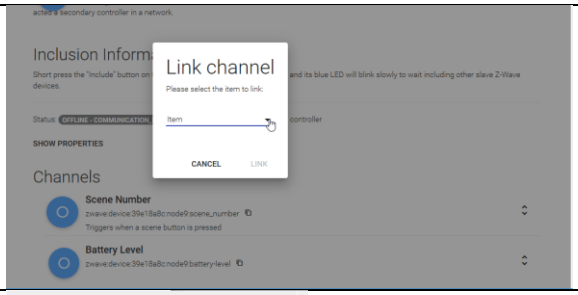
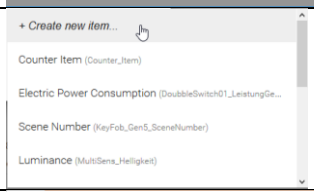
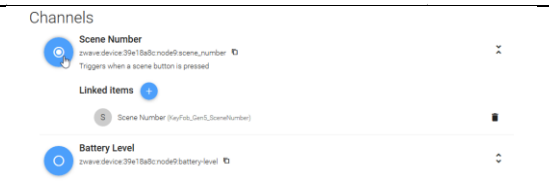
To illustrate these concepts, take a two-channel actuator that controls two lights:



The actuator is the Thing. This might be installed in the electrical cabinet, it has a physical address and needs to be setup and configured in order to be used. The user is instead interested in the two lights, which are located at different locations in his home. These lights are the desired functionality, thus the Items and they are linked to the Channels of the actuator. A Link can be regarded like a physical wire in this example.

The basic process of in PAPER UI is explained on:
http://docs.openhab.org/tutorials/beginner/configuration.html

Creating the required items for my project

Switch to -Configuration --Things and select the KeyFob_Gen5 thing	
A list of all the available channels of the thing will be shown	
Now click on the blue icon in front of the channel number to link this channel to an item	
A link channel window will come up Select the pull down option for the item	
The next window coming up will allow you to select already existing items or: In our case <+Create new item…>	
The next window will allow you to configure the item you want to link to the thing channel You can also change the name of the item if you are planning to run with your own naming convention Then select <LINK> to create your new item	
Now the blue icon to the left of the name will change (white dot in the center) By clicking on this icon the channel will expand and show you the linked items to this cannel	
You can now find the new item in -Configuration --Items	

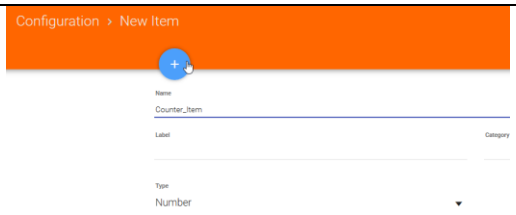
Now go on creating the items according to the list below

Thing	Channel	Item
KeyFob_Gen5	zwave:device:39e18a8c:node9:scene_number	KeyFob_Gen5_SceneNumber
Weather Information	yahooweather:weather:c5d26906:temperature	Yahoo_Temperature
Weather Information	yahooweather:weather:c5d26906:humidity	Yahoo_Humidity
Yamaha Receiver RX-V581	yamahareceiver:yamahaAV:9ab0c000_f668_11de_9976_00a0dedc57ff:power	YamahaReceiverRXV581_Power
Yamaha Receiver RX-V581	yamahareceiver:yamahaAV:9ab0c000_f668_11de_9976_00a0dedc57ff:volume	YamahaReceiverRXV581_Volume
Yamaha Receiver RX-V581	yamahareceiver:yamahaAV:9ab0c000_f668_11de_9976_00a0dedc57ff:mute	YamahaReceiverRXV581_Mute
Z-Wave Node 2: FGS223 Double Switch 2	zwave:device:39e18a8c:node2:meter_watts	DoubbleSwitch01_LeistungGesamt
Z-Wave Node 2: FGS223 Double Switch 2	zwave:device:39e18a8c:node2:switch_binary1	DoubbleSwitch01_Relais1
Z-Wave Node 3: MSP-3-1-X1 Z-Wave Plus Micro Smart Plug ON/OFF	zwave:device:39e18a8c:node3:switch_binary	SchuKo01
Z-Wave Node 3: MSP-3-1-X1 Z-Wave Plus Micro Smart Plug ON/OFF	zwave:device:39e18a8c:node3:meter_watts	SchuKo01_Leistung
Z-Wave Node 6: ZW100 MultiSensor 6	zwave:device:39e18a8c:node6:sensor_relhumidity	MultiSens_Luftfeuchte
Z-Wave Node 6: ZW100 MultiSensor 6	zwave:device:39e18a8c:node6:sensor_temperature	MultiSens_Temperatur
Z-Wave Node 6: ZW100 MultiSensor 6	zwave:device:39e18a8c:node6:sensor_luminance	MultiSens_Helligkeit

NOTE: If you want to delete items or links between thing channels and items again you have to be aware of the following behavior of openHAB2

Deleting the item with existing link between thing and channel will result in: - “Item deleted” prompt - Item still be visible in the items tab - Channel still showing link to the item To completely delete the item you have to delete the link between thing and channel in the thing tab.
Deleting the link between thing and channel in the thing tab will result in - Channel is showing no link in thing tab (blue icon without with center) - Item still available in the item tab, but is having not link to the original channel This is as designed since you might want to link multiple channels to on item or use the item to link it to another channel

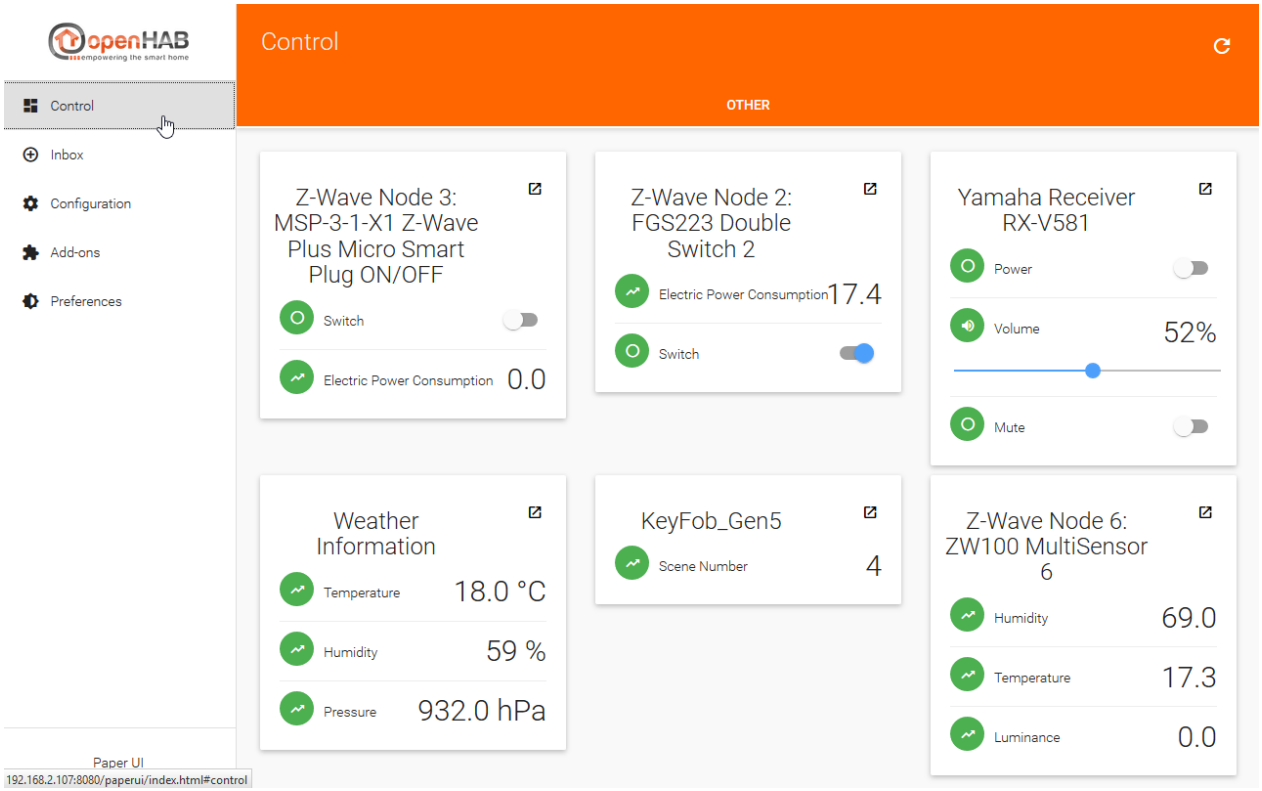
Also create a stand-alone item which we will use for the rules later on:

Switch to -Configuration --Items And press the blue icon (+) Enter Name `Counter_Item` Select Typ `<Number>` And confirm the creation by clicking again on the blue icon (+)	 `Counter_Item` `<Number>`
Result: A new item called `Counter_Item` should be visible in the item tab	

Chapter 11: Creating user interfaces for your home automation project

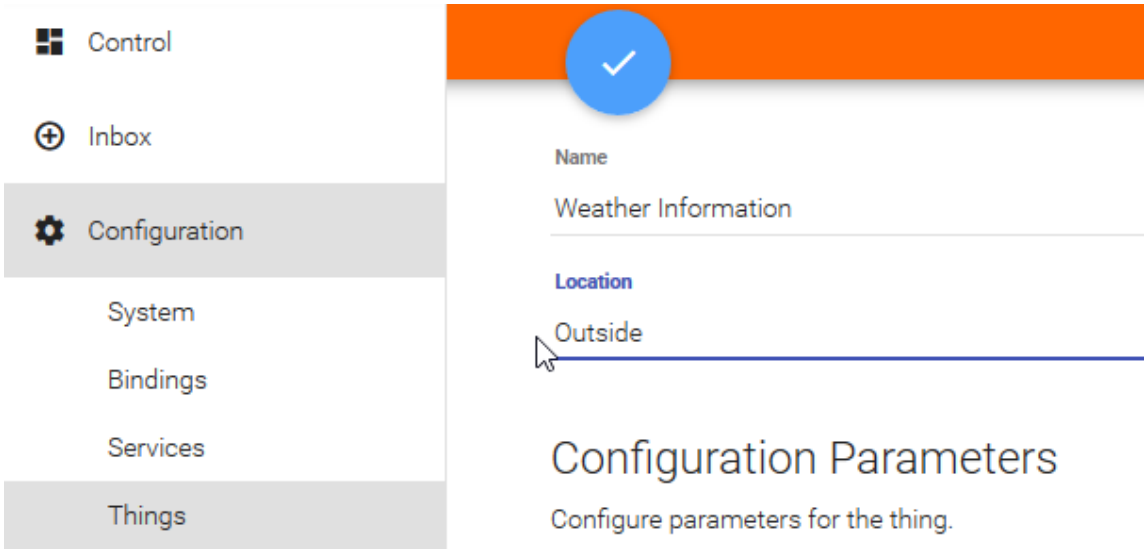
PAPER UI user interface Control

The most simple way of creating an interactive user interface is in just clicking on the control tab to PAPTER UI. Now you should see all the items grouped by the things they belong to



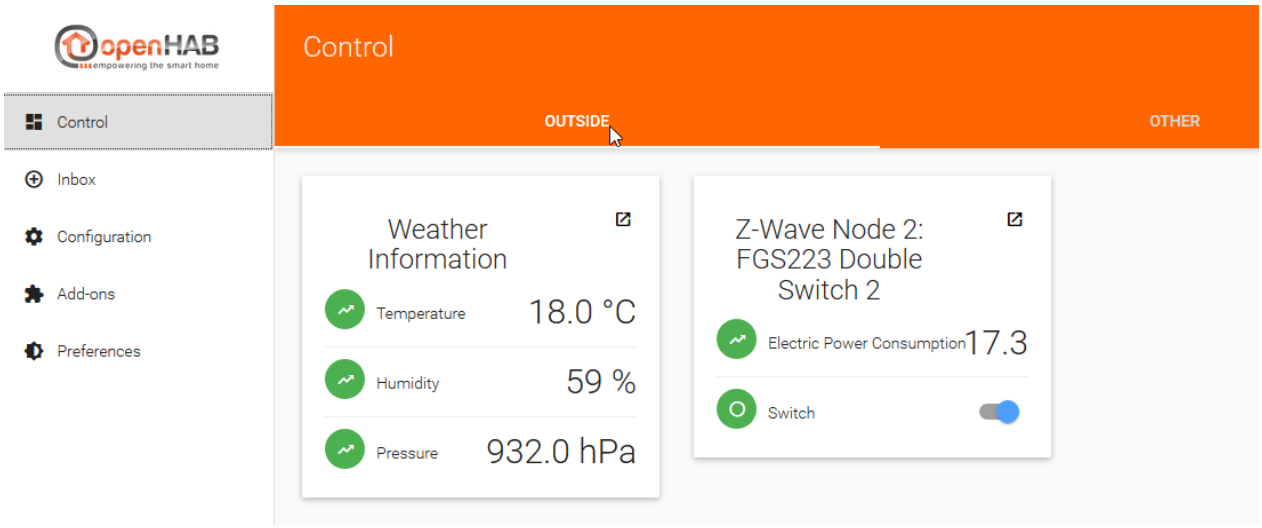
You can also easily create different tabs of things by adding `<Location>` information to the thing in the thing tab:

NOTE: I had issues in updating a few things, so it will be good to it while creating the things

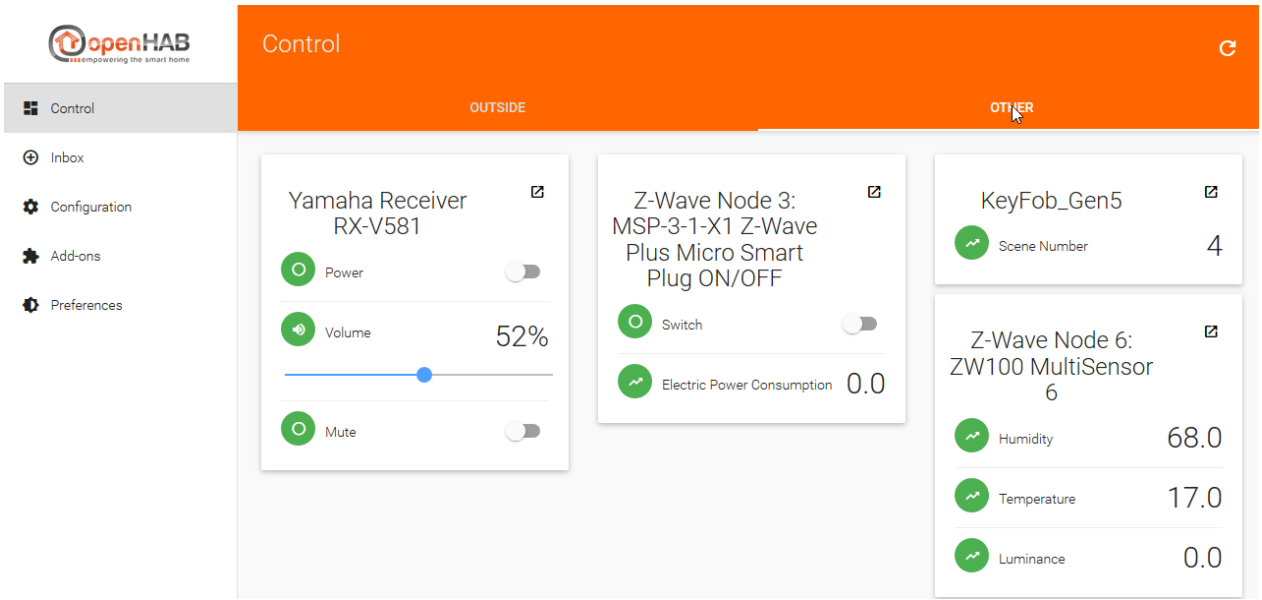


Now you will have Control panel giving you various tabs with items grouped by their things

Tab OUTSIDE



Standard tab OTHER

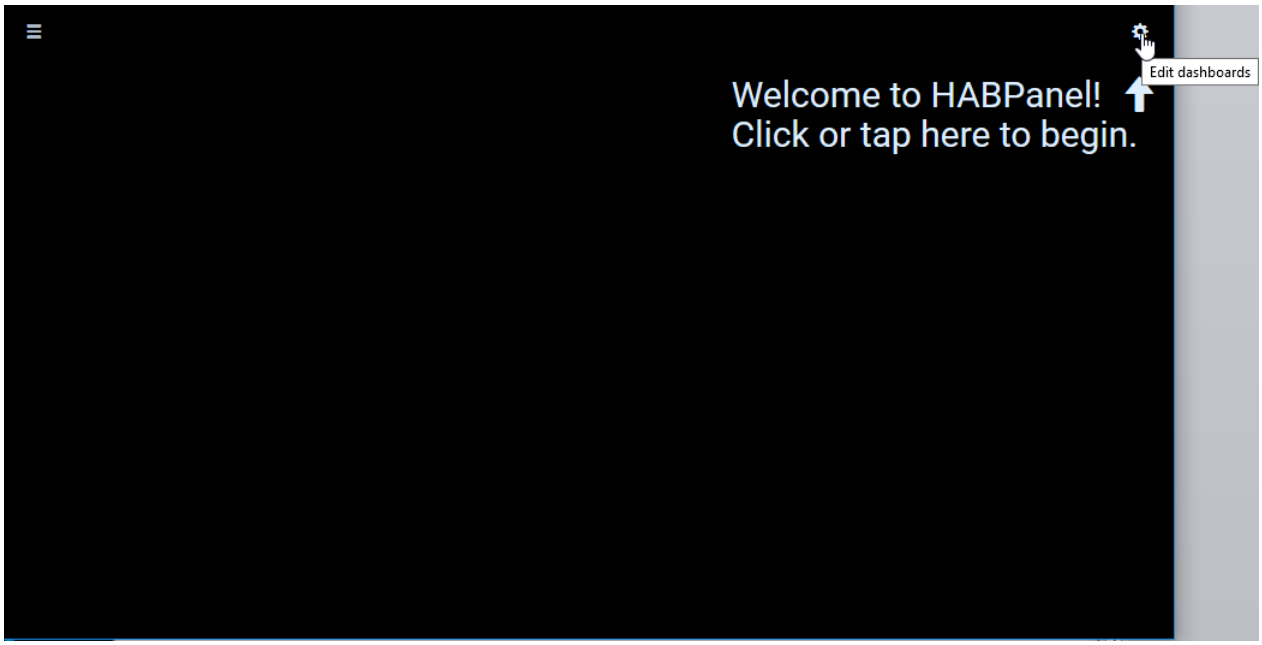


HABPanel UI user interface

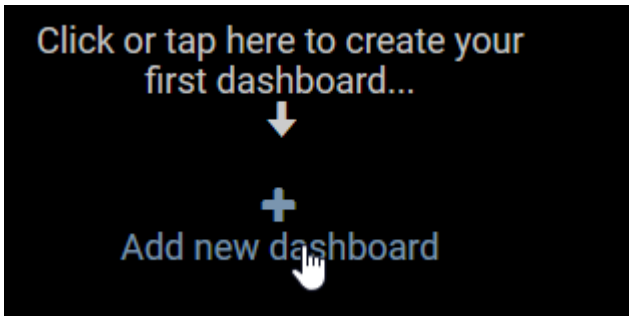
The HABPanel UI is all about creation a user interface for your home automation project. Therefore you can not to do all the configuration work for openHAB2 in this UI. The configuration hast to be done in the other UIs

It can be reached directly using this URL: http://xxx.xxx.xxx.xxx:8080/habpanel/index.html#/

It will come up a complete blank panel asking you to start configuration:



You now can choose to <Add new dashboard>



How it is done is explained online on:

http://docs.openhab.org/addons/uis/habpanel/readme.html

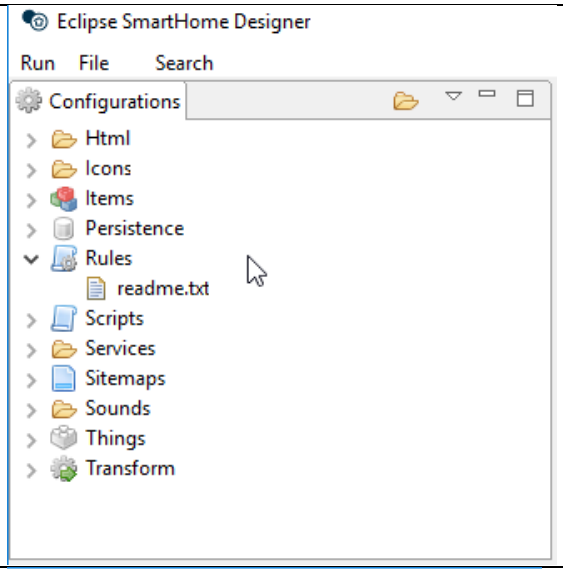
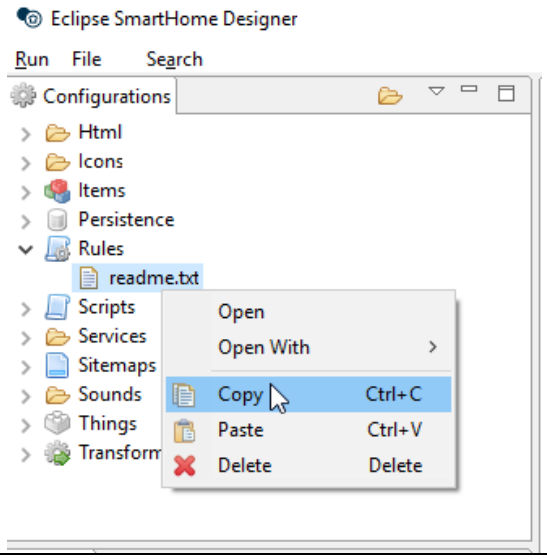
To come: Setting up the dashboard for my home automation project

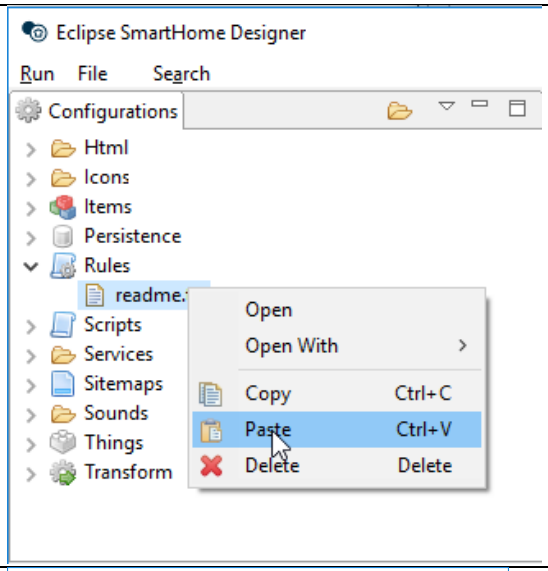
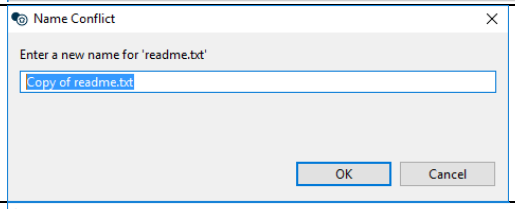
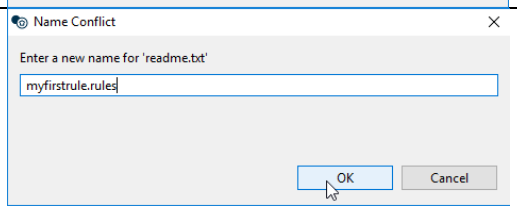
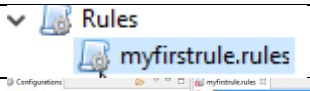
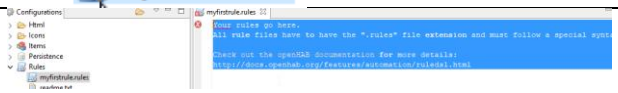
Chapter 12: Creating rules

Since rules can't be configured using the PAPER UI (in the stable version of openHAB2 at least) you now have to go to the text files.

For this we will use the Eclipse Smart Home Designer since it creates at least some syntax highlighting.

Creating your first rule

Start Eclipse Smart Home Designer on your PC Make sure you have mapped the Raspbian samba drive to your PC and Start Eclipse Smart Home Designer is set to the right folder in this samba drive (see Chapter 7: Installation of Eclipse Smart Home Designer -: part launching first time) You should now the augmented icons for the different folders If you check the Rules folder you will only find a readme.txt file	
You need to create a file with the ending .rules to store your rules there. The fastest way of doing it is using the basic file management functionality of Eclipse Smart Home Designer Right click on the readme.txt file and select copy	

Then right click again and select Paste	
A new window will ask you to enter a new file name.	
Change the file name to `myfirstrule.rules` an press `<OK>`	 `myfirstrule.rules` `<OK>`
A new file will appear in the Rules folder showing the rule icon	
Now double click on the file to open it and delete the old content to have a plain rule rile And save the rule file again	 <ctrl+a> <del> <ctrl-s>
Now you can paste the following rules into your rule file to get your first automation done The rule is designed to switch on/off the `Z-Wave Node 2: FGS223 Double Switch 2` based on the `illumination measured by the Z-Wave Node 6: ZW100 MultiSensor 6` The trigger value is 10 lumen To prevent von switching on/off if the lamination is around 10 lumen an e.g. a cloud is casting a temporarily shadow, There is a counter included which is	``` // this is the rules file import org.openhab.core.library.types.* import org.openhab.model.script.actions.* import java.lang.Math import java.util.Calendar import java.util.Date import java.util.TimeZone import java.text.SimpleDateFormat import org.joda.time.* var Number loop_counter = 0 rule "check_illumination" // using the loop_counter to ensure that it is // 10 times in a row darker/lighter before triggering switch ```

measuring multiple times the illumination and only triggering the switch if there was a repeating amount of lamination reading above or below the trigger value	when // every x seconds "0/x" the value is checked // 0/30 means every 30 sec the value is checked Time cron "0/30 * * ? * * *" then // reset loop_counter if required (counter outside -5 +5 range) if (loop_counter >= -5 && loop_counter <= 5){ // <= 10 is defining the LUX value when ligth is swiched on/off if (MultiSens_Helligkeit.state <= 10) { if (loop_counter > -5) { loop_counter = loop_counter -1 postUpdate(Counter_Item, loop_counter) } else { if (DoubbleSwitch01_Relais1.state == OFF) sendCommand (DoubbleSwitch01_Relais1, ON) } } else if (loop_counter < 5) { loop_counter =loop_counter +1 postUpdate(Counter_Item, loop_counter) } else { if (DoubbleSwitch01_Relais1.state == ON) sendCommand (DoubbleSwitch01_Relais1, OFF) } } else { loop_counter = 0 } end
The second rule in this file is to trigger the two switches using the KeyFob remote control. The KeyFob is offering 8 SceneNumber settings. 1=Button one shot, 2=Button one long, 3=Button two short and so on. So I selected each button short is switching on a switch and button long is switching off the switch. If you are adding more switches just copy the rule section for one button and change the switch item and the scene number	rule "KeyFob" when Item ZWaveNode9ZW0884ButtonKeyfobGen5_SceneNumber received update then //Scene number 1 - Button 1 (up left) pressed short if (ZWaveNode9ZW0884ButtonKeyfobGen5_SceneNumber.state == 1) { sendCommand (DoubbleSwitch01_Relais1, ON) }

	//Scene number 2 - Button 1 (up left) pressed long if (ZWaveNode9ZW0884ButtonKeyfobGen5_SceneNumber.state == 2) { sendCommand (DoubbleSwitch01_Relais1, OFF) } //Scene number 3 - Button 2 (up left) pressed short if (ZWaveNode9ZW0884ButtonKeyfobGen5_SceneNumber.state == 3) { sendCommand (SchuKo01, ON) } //Scene number 4 - Button 2 (up left) pressed long if (ZWaveNode9ZW0884ButtonKeyfobGen5_SceneNumber.state == 4) { sendCommand (SchuKo01, OFF) } //Scene number 5 - Button 3 (up left) pressed short //Scene number 6 - Button 3 (up left) pressed long //Scene number 7 - Button 4 (up left) pressed short //Scene number 8 - Button 4 (up left) pressed long end

Further rule documentation:

For further rules documentation incl. the rule syntax please refer to the online documentation:

http://docs.openhab.org/configuration/rules-dsl.html