



ZB-Shield with Home Assistant

This document describes how to use POPP ZB- Shield with existing Home Automation platform called Home Assistant (Hass.io) (<https://www.home-assistant.io/>).

POPP ZB-Shield firmware version, referenced in this guide: **6.3.0**

Home Assistant (Hass.io) software version, referenced in this guide: **0.112.4**

This guide focuses on:

- Connect POPP ZB-Shield to your Raspberry Pi
- Setup Zigbee Home Automation component in Home Assistant
- Zigbee devices Pairing and Removal
- Zigbee device example

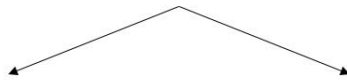
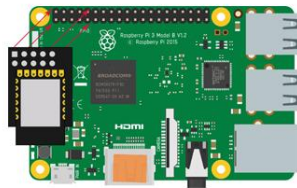
This guide DOES NOT focus on Home Assistant (Hass.io) installation and initial configuration. Please follow the official instructions <https://www.home-assistant.io/hassio/installation/>.

Introduction

POPP ZB-Shield can be used with a Raspberry Pi with Home Assistant installed.



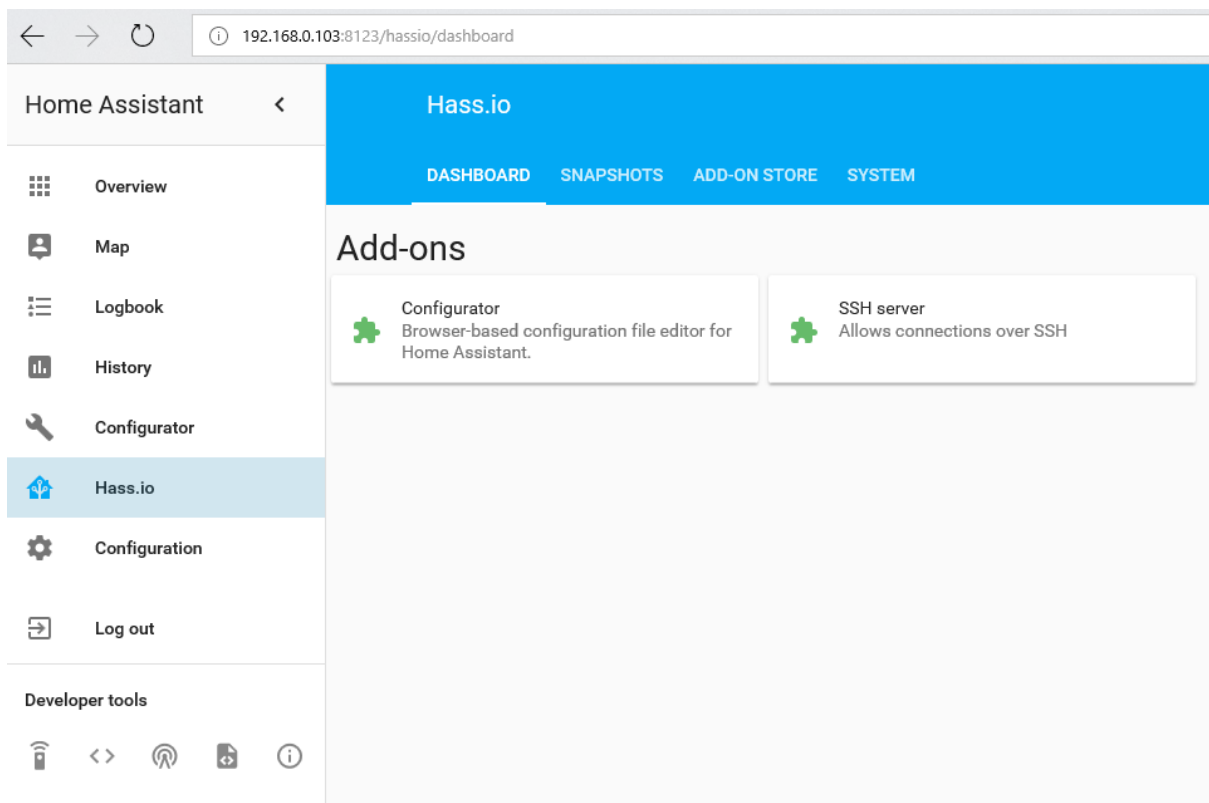
Home Assistant installed and running



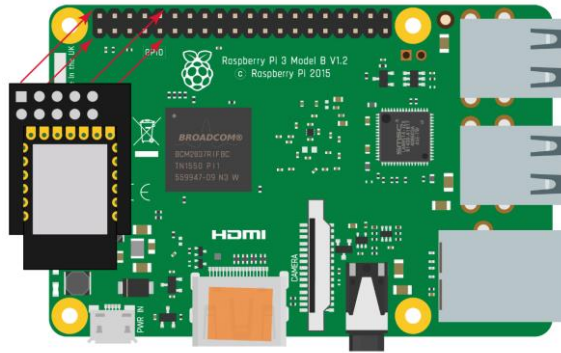
Initial setup and connection

This guide does not cover Home Assistant (Hass.io) installation. We assume, that you can access Home Assistant Web Interface using your browser.

It is highly recommended to have SSH Server and Configurator addons installed, at least during the Zigbee setup phase. To install them just follow the Hass.io installation guide (<https://www.home-assistant.io/hassio/installation/>).



Connect POPP ZB-Shield to your Raspberry Pi



Disable Serial Console

By default, the Raspberry Pi (1,2,3,4) has the Serial Console enabled on the UART, which is used by the POPP ZB-Shield. This results in conflict, so we need to disable the Serial Console.

To do it:

- Insert the SD card with installed Hass.io into your PC/laptop.
- Find the **cmdline.txt** file in the SD card file system

hassos-boot (G:)

| Name | Date modified | Type | Size |
|--------------------------|--------------------|-------------|----------|
| overlays | 7/3/2020 5:06 PM | File folder | |
| bcm2710-rpi-3-b.dtb | 7/3/2020 5:06 PM | DTB File | 26 KB |
| bcm2710-rpi-3-b-plus.dtb | 7/3/2020 5:06 PM | DTB File | 27 KB |
| bcm2710-rpi-cm3.dtb | 7/3/2020 5:06 PM | DTB File | 25 KB |
| boot.scr | 7/3/2020 5:06 PM | SCR File | 3 KB |
| bootcode.bin | 7/3/2020 5:06 PM | BIN File | 52 KB |
| cmdline.txt | 7/21/2020 11:04 PM | TXT File | 1 KB |
| config.txt | 7/21/2020 11:05 PM | TXT File | 2 KB |
| fixup.dat | 7/3/2020 5:06 PM | DAT File | 7 KB |
| start.elf | 7/3/2020 5:06 PM | ELF File | 2,818 KB |
| u-boot.bin | 7/3/2020 5:06 PM | BIN File | 384 KB |

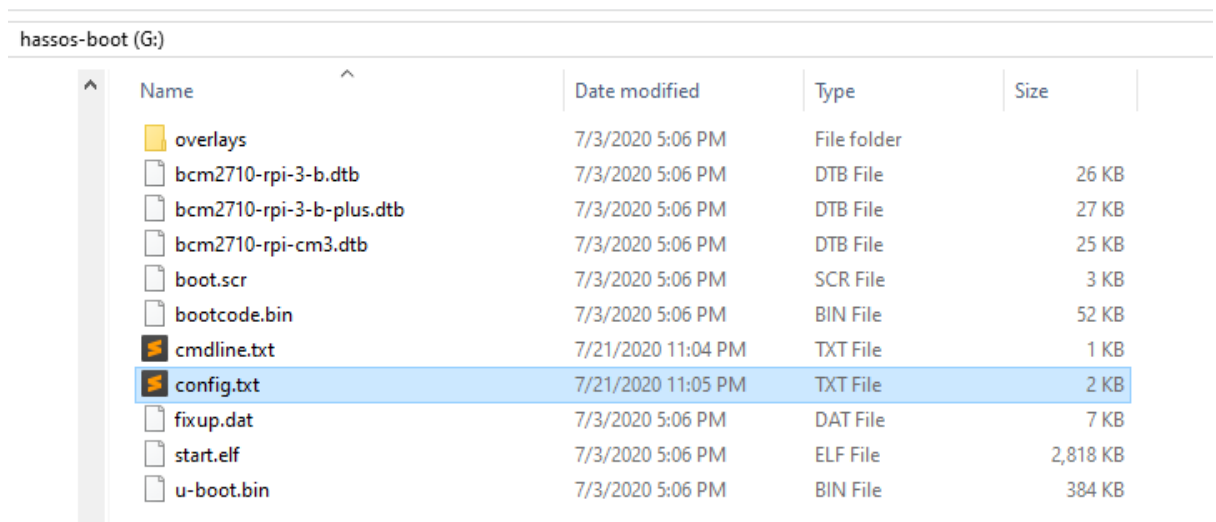
- Modify this file and remove `console=tty1`:

```
cmdline.txt
dwc_otg.lpm_enable=0
```

Enable UART Bluetooth overlay (RPi 3 and Rpi 4)

If you are using Raspberry Pi 3 you need to perform an additional step of enabling the Bluetooth UART overlay configuration. To do it:

- Insert the SD card with installed Hass.io into your PC/laptop.
- Find the **config.txt** file in the SD card file system

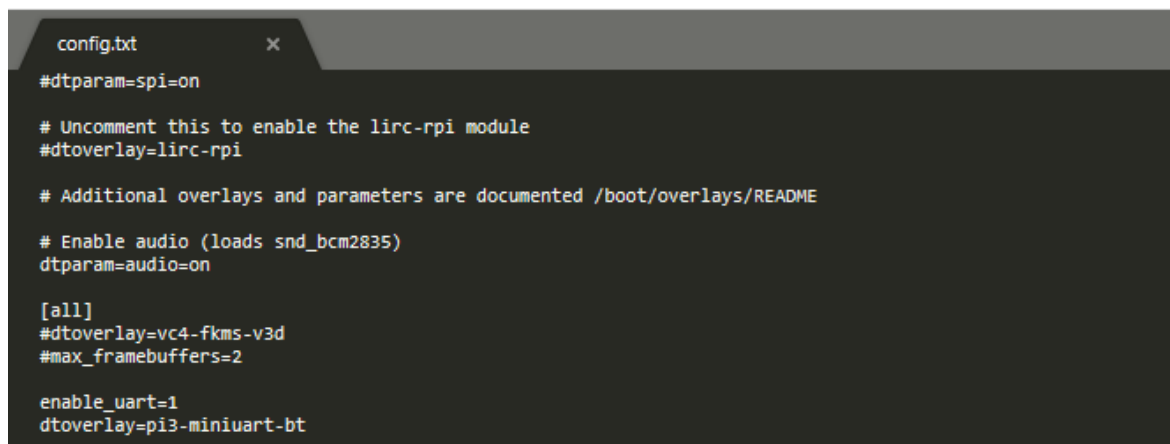


| Name | Date modified | Type | Size |
|--------------------------|--------------------|-------------|----------|
| overlays | 7/3/2020 5:06 PM | File folder | |
| bcm2710-rpi-3-b.dtb | 7/3/2020 5:06 PM | DTB File | 26 KB |
| bcm2710-rpi-3-b-plus.dtb | 7/3/2020 5:06 PM | DTB File | 27 KB |
| bcm2710-rpi-cm3.dtb | 7/3/2020 5:06 PM | DTB File | 25 KB |
| boot.scr | 7/3/2020 5:06 PM | SCR File | 3 KB |
| bootcode.bin | 7/3/2020 5:06 PM | BIN File | 52 KB |
| cmdline.txt | 7/21/2020 11:04 PM | TXT File | 1 KB |
| config.txt | 7/21/2020 11:05 PM | TXT File | 2 KB |
| fixup.dat | 7/3/2020 5:06 PM | DAT File | 7 KB |
| start.elf | 7/3/2020 5:06 PM | ELF File | 2,818 KB |
| u-boot.bin | 7/3/2020 5:06 PM | BIN File | 384 KB |

- Modify this file and add an extra line at the end:
enable_uart=1
dtoverlay=pi3-miniuart-bt

Raspberry 4: dtoverlay=disable-bt

enable_uart=1



```
config.txt
#dtparam=spi-on

# Uncomment this to enable the lirc-rpi module
#dtoverlay=lirc-rpi

# Additional overlays and parameters are documented /boot/overlays/README

# Enable audio (loads snd_bcm2835)
dtparam=audio-on

[all]
#dtoverlay=vc4-fkms-v3d
#max_framebuffers=2

enable_uart=1
dtoverlay=pi3-miniuart-bt
```

Check configuration in the Web Interface

Once you have modified the hardware setup, boot up the Raspberry Pi and check the configuration.

- Insert your SD card back inside the Raspberry Pi
- Wait for Hass.io to boot
- Connect to the web interface using your favourite browser
- Navigate to the **Supervisor -> System -> Hardware** menu

- Confirm the UART serial ports are visible for you (/dev/ttyAMA0 and /dev/ttyS0)

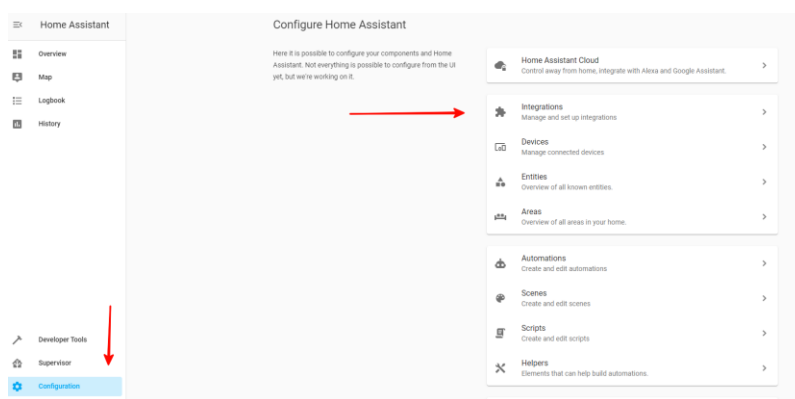
Zigbee HA Component configuration

To work with POPP ZB-Shield from Home Assistant we are using Zigbee Home Automation Component (<https://www.home-assistant.io/components/zha/>). It comes preinstalled into Hass.io so we only need to configure it properly to get it working.

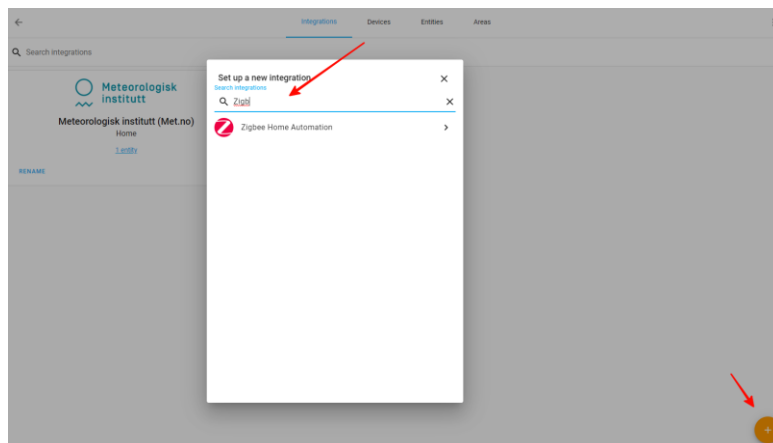
Configure ZHA component

To setup Zigbee Home Automation component to work properly with POPP ZB-Shield we need to:

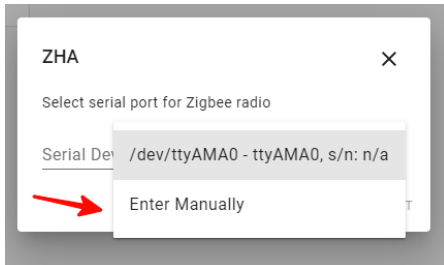
- Navigate to the **Configuration -> Integrations**



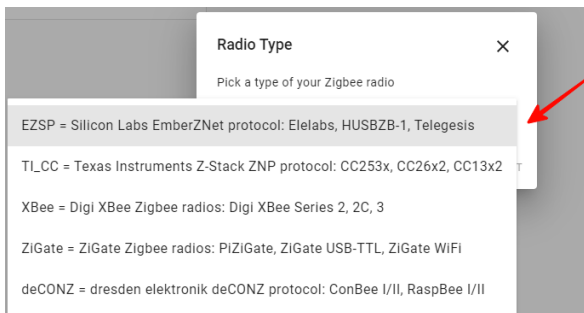
- Add new Integration and search for **Zigbee Home Automation**



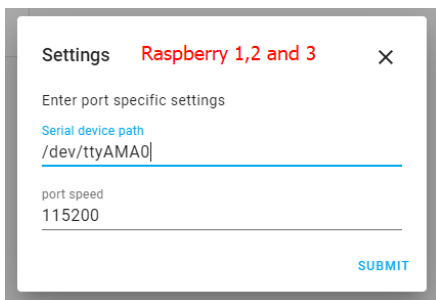
- Select **Enter Manually**



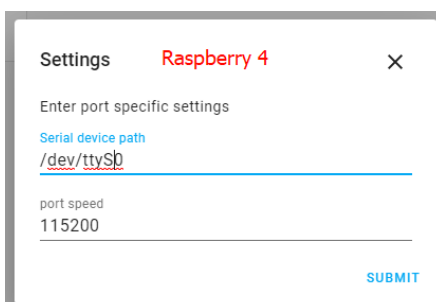
- Select **EZSP** radio type



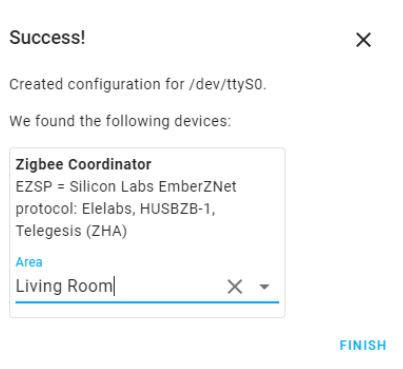
- For Raspberry Pi 1,2,3 enter **/dev/ttyAMA0** as Serial Port and select Baud rate 115200



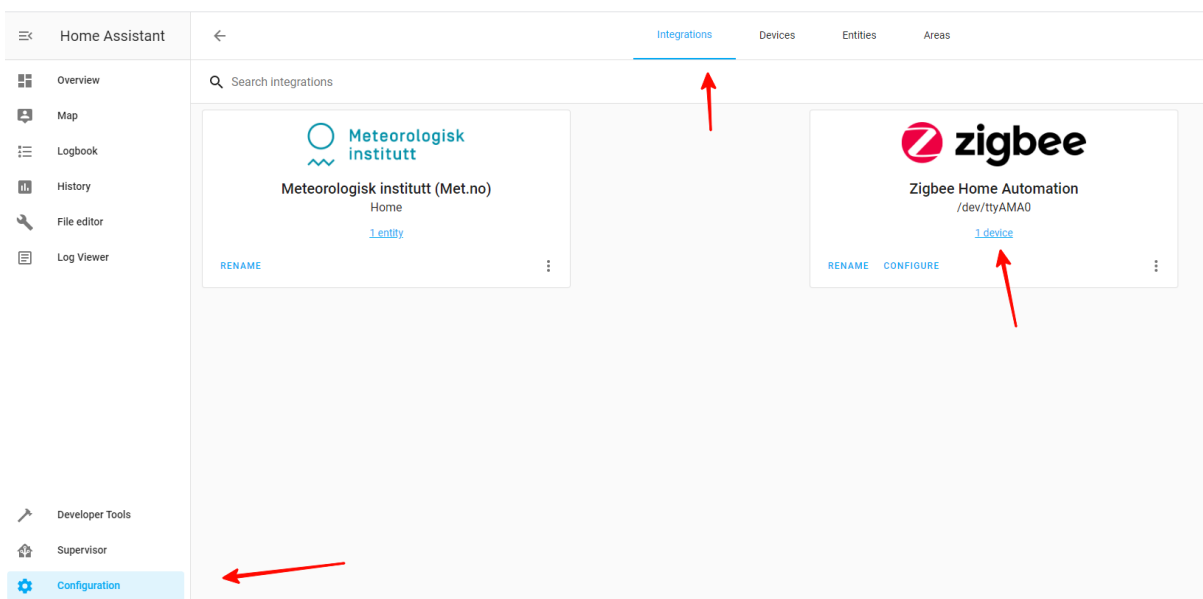
- For Raspberry Pi 4 enter **/dev/ttyS0** as Serial Port and select Baud rate 115200



- The installation should be successful



Now the Zigbee Integration should appear:



Setup Logging (optional)

To spot any potential issues it's good practice to enable logging, at least during the setup and installation period. To do it, just add the following lines to the configuration file

/config/configuration.yaml:

logger:

default: warn

logs:

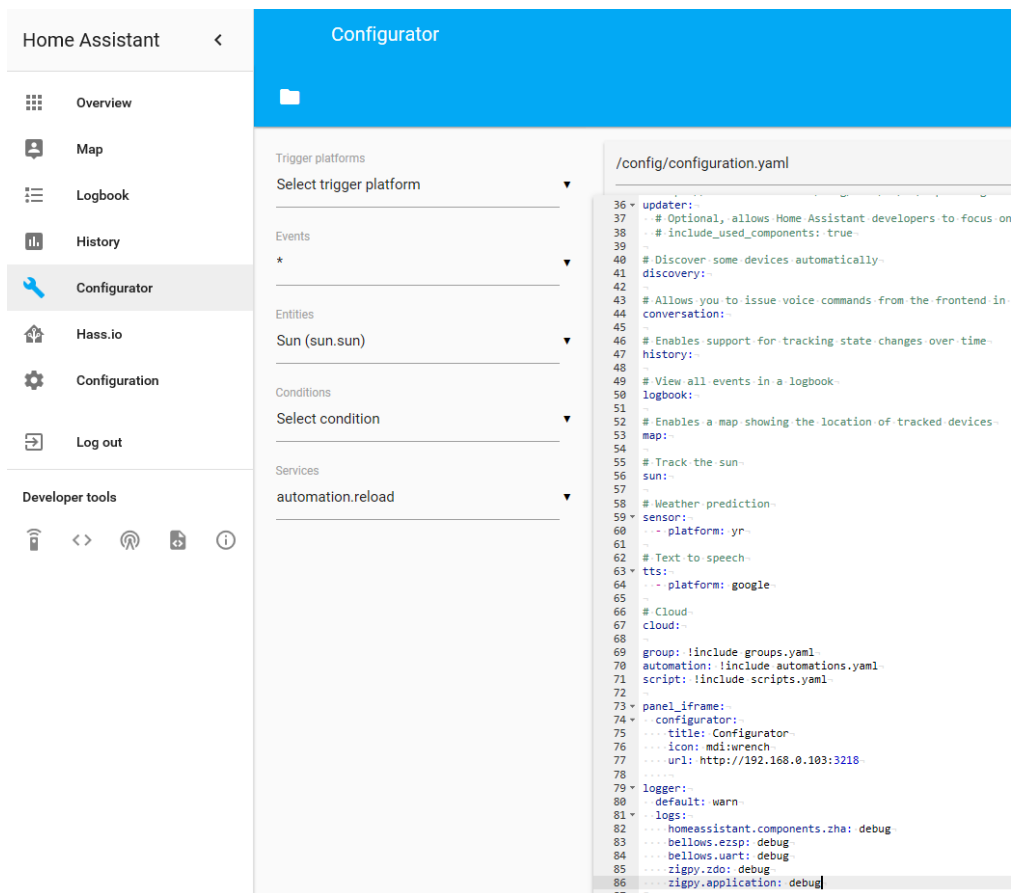
homeassistant.components.zha: debug

bellows.ezsp: debug

bellows.uart: debug

zigpy.zdo: debug

zigpy.application: debug

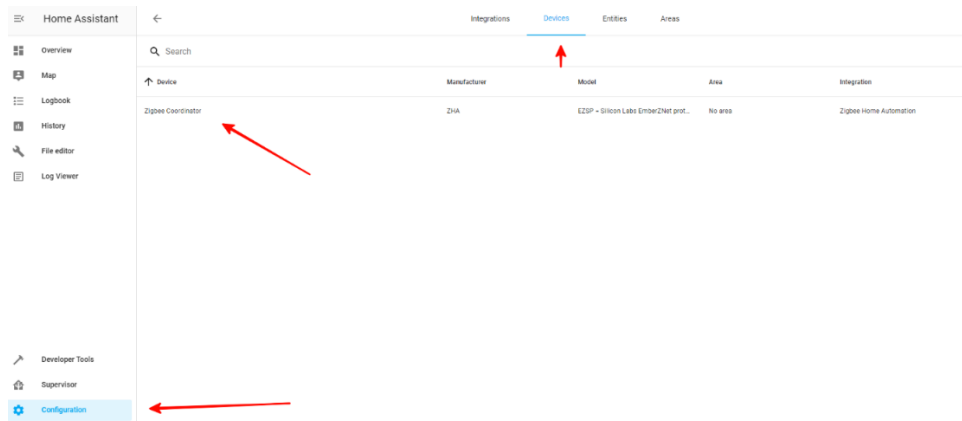


Zigbee HA Component Usage

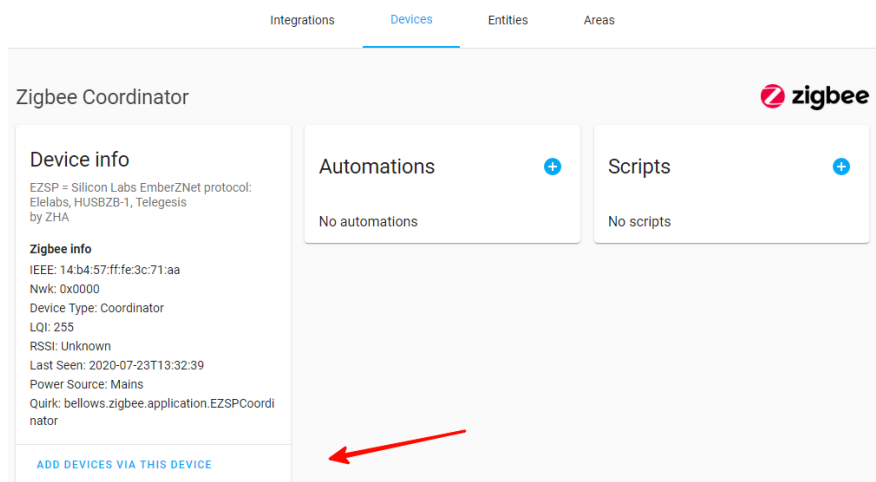
Once Zigbee Component is added and configured properly you can start to use it.

Add your devices to the Home Assistant

Open **Configuration** and go to **Devices** and **Zigbee Coordinator**



Start “Add Devices via this device”



When you will call it, you have 60 seconds to add the device.

Searching for ZHA Zigbee devices...



Make sure your devices are in pairing mode. Check the instructions of your device on how to do this.

Devices will show up here once discovered.

During this period, you need to follow Device manual to put it in Pairing mode. Sometimes you just need to give it power.

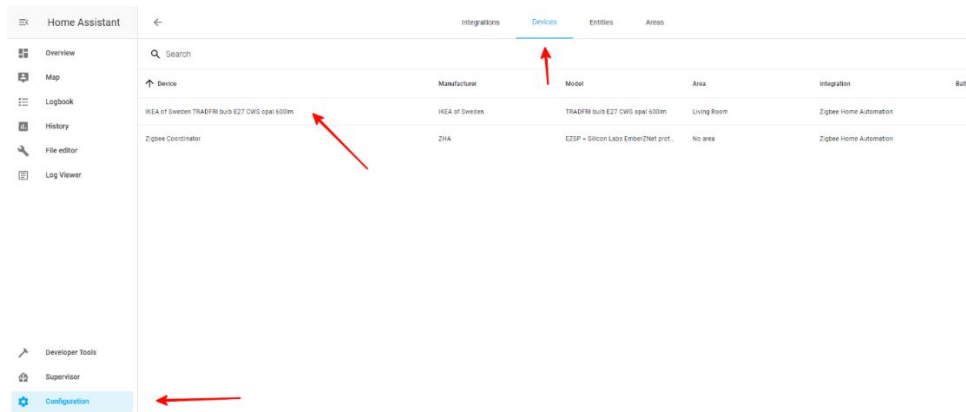
If the device is found, you will be able to see it in the logs (example device)

The screenshot shows the POPP interface during a ZHA Zigbee device search. At the top, it says "Searching for ZHA Zigbee devices..." with a small blue dot and a loading spinner. Below this, a device card is displayed for "IKEA of Sweden TRADFRI bulb E27 CWS opal 600lm". The card includes the manufacturer name, a lightbulb icon, and a "Change device name" field with the current name "IKEA of Sweden TRADFRI bulb E27 CWS opal 600lm". Below the name field is an "Area" dropdown menu. At the bottom of the screenshot, a log window shows the following messages:

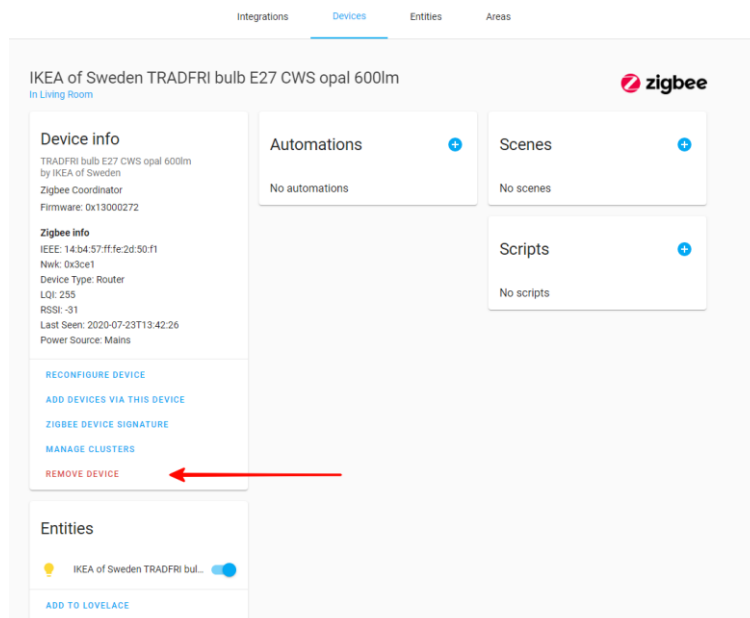
```
[0x3ce1:1:0x0000]: 'async_initialize' stage succeeded
[0x3ce1:1:0x1000]: 'async_initialize' stage succeeded
[0x3ce1:1:0x0019]: 'async_initialize' stage succeeded
[0x3ce1:1:0x0005]: 'async_initialize' stage succeeded
[0x3ce1](TRADFRI bulb E27 CWS opal 600lm): power source: Mains
[0x3ce1](TRADFRI bulb E27 CWS opal 600lm): completed initialization
[0x3ce1:1:0x0006]: attempting to update onoff state - from cache: False
[0x3ce1:1:0x0006] ZCL deserialize: <ZCLHeader frame_control=<FrameControl frame_type=GLOBAL_COMMAND manufacturer_specific=False is_reply=True disable_default_response=True> manufacturer=None tsn=63
command_id=Command.Read_Attributes_rsp>
None: polling current state - from cache: True
```

Remove your device from Home Assistant

Open **Configuration** and go to **Devices**.



Select the device, which you would like to remove



Once you call this service you can verify in the logs, that the device has left the network.

Example: Philips Hue Bulb

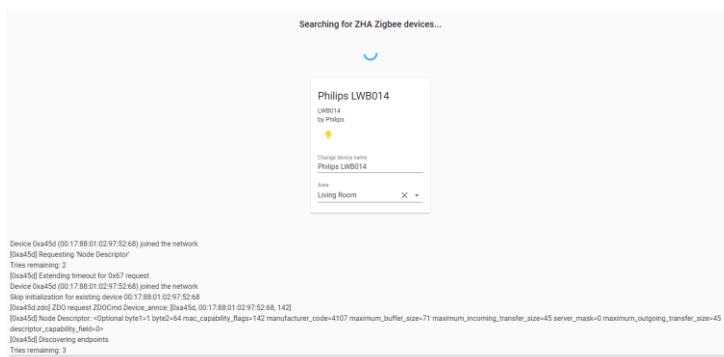
This example is done with Hue White Single bulb E26 but is applicable to other products as well.



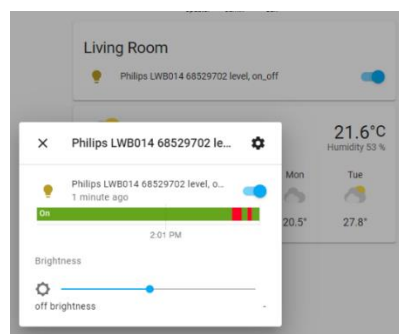
To control Philips Hue Light bulb using Home Assistant, one first needs to reset it.

Once it is reset, you can follow the regular process to Add it to the Home Assistant.

- Call **Add Devices** service as explained in Add devices section of this guide
- Power ON the Lightbulb
- Confirm it is added to the Home Assistant



Now you can control it directly or use in the scenarios.





Troubleshooting

If your issue is not described here or you need help resolving it, please contact support at info@popp.eu.