UG2: HOME ASSISTANT

This document describes how to use Elelabs ZigBee RPi Shield (https://elelabs.com/products/elelabs_zigbee_shield.html) with existing Home Automation platform called Home Assistant (Hass.io) (https://www.home-assistant.io/).

Elelabs ZigBee RPi Shield firmware version, referenced in this guide: **5.10**

Home Assistant (Hass.io) software version, referenced in this guide: **0.66.1**

This guide focuses on:

- Connect Elelabs RPi Shield to your Raspberry Pi
- Setup ZigBee Home Automation component in Home Assistant
- Troubleshooting
- ZigBee devices Pairing and Removal
- ZigBee devices examples

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/.

Table of Contents

Introduction 3

Initial setup and connection 4

ZigBee HA Component configuration 7

Setup Logging (optional) 7

Configure Zha component 8

ZigBee HA Component Usage 9

Add your devices to the Home Assistant 9

Remove your device from Home Assistant 10

Example: Philips Hue Bulb 11

Example: Xiaomi Smoke Sensor 11

Troubleshooting 13

Home Assistant Zha component couldn't be set up 13

Home Assistant does not boot at all 13

Revert using Configurator Addon 13

Revert using SSH Addon 14

How to fix 14

Zha component is loaded correctly but no zha.permit service 15

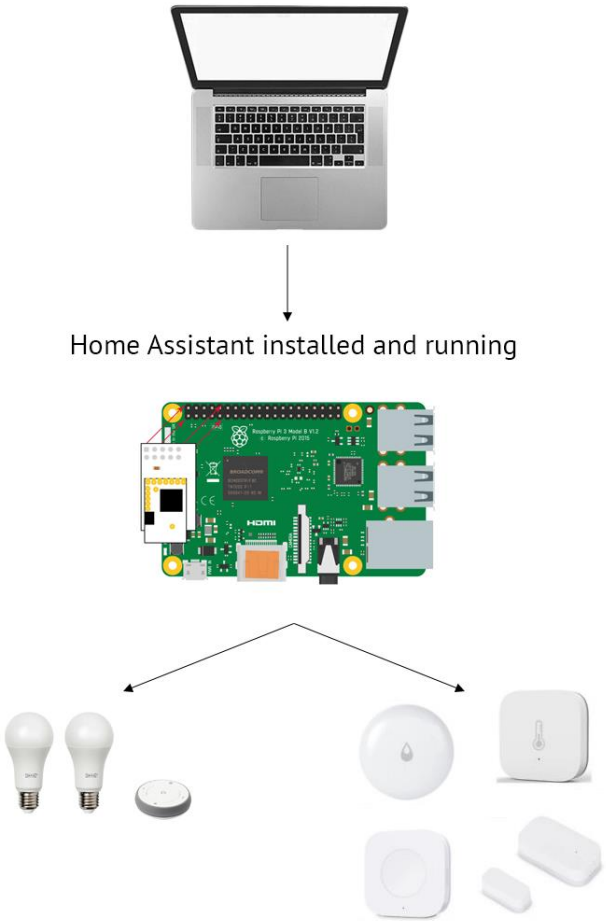
Zha component is loaded, zha.permit is started, but can't add the device 15

The device tries to be added, but an error occurs 15

There are no packets from the device at all 16

Introduction

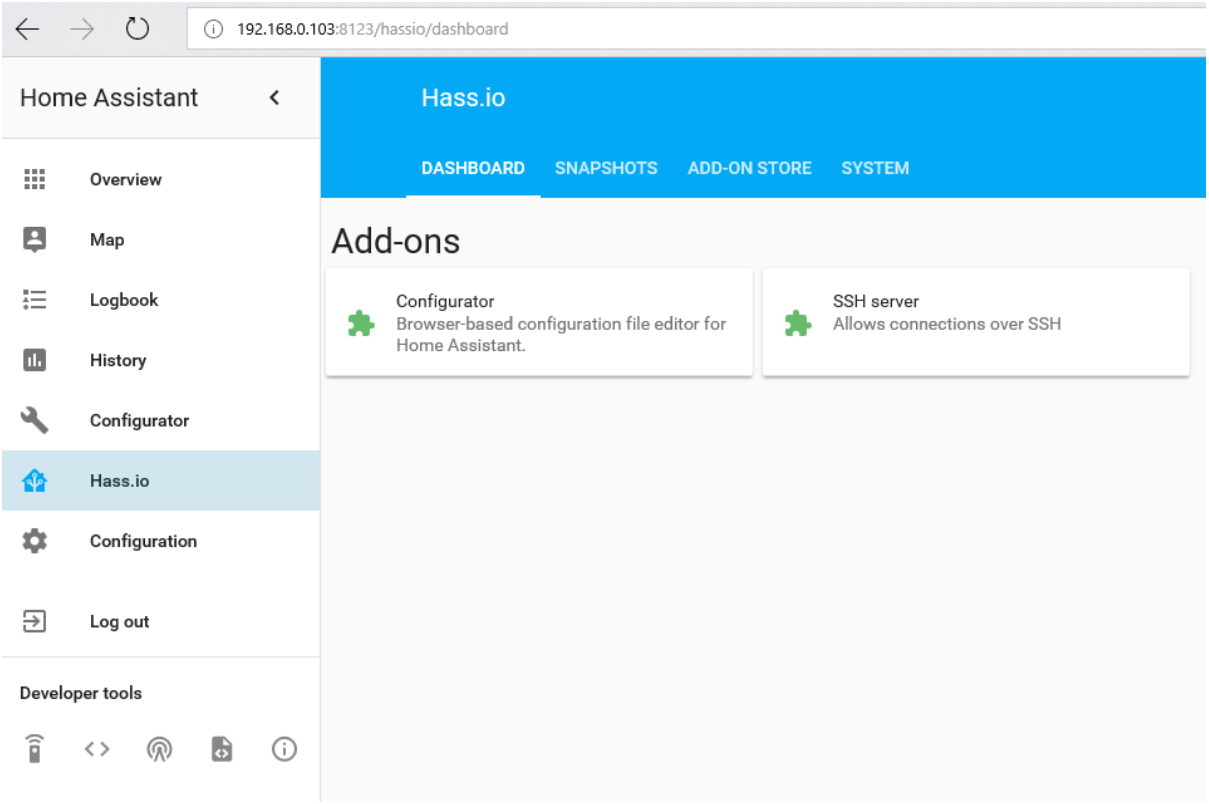
Elelabs ZigBee RPi Shield can be used with a Raspberry Pi with Home Assistant installed



Initial setup and connection

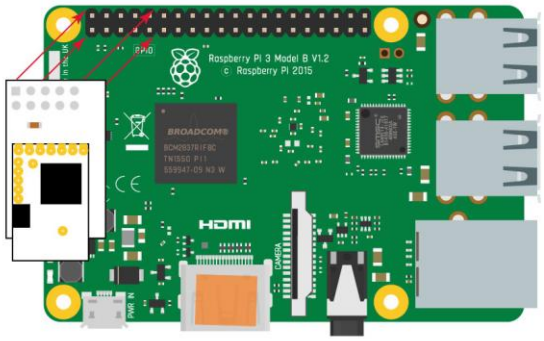
This guide does not cover Home Assistant (Hass.io) installation. We assume, that you can access Home Assistant Web Interface using your browser.

It's 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/).



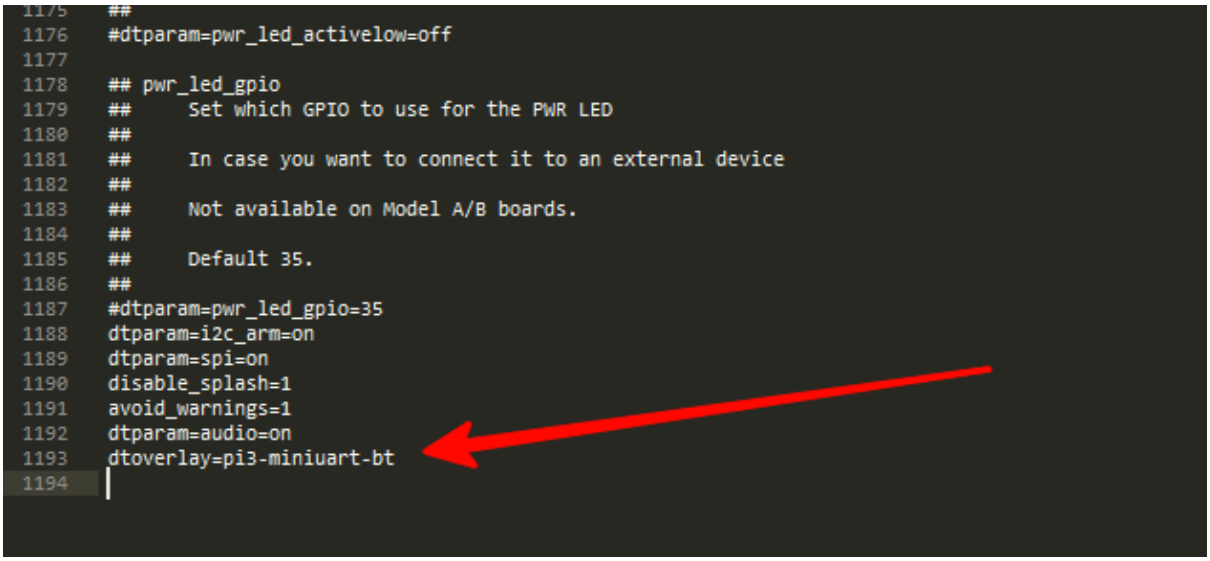
Insert the Elelabs ZigBee RPi Shield into your Host machine.

Usage with Raspberry Pi



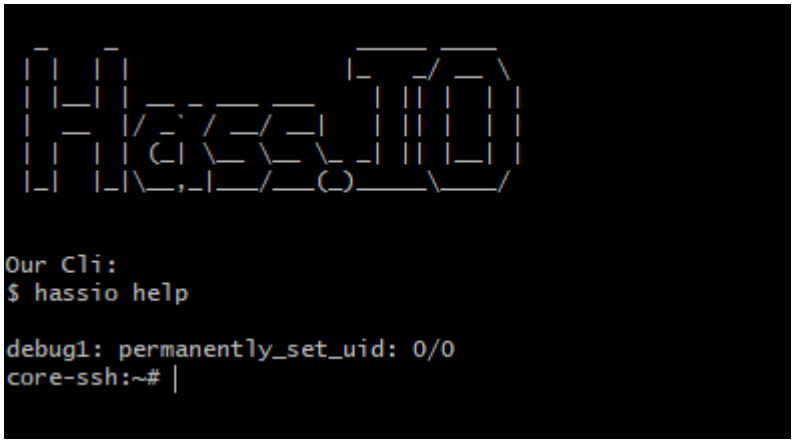
If you are using Raspberry Pi 3 you need to perform an additional step of disabling the Bluetooth UART configuration. To do it:

- *Insert the SD card with installed Hass.io into your PC/laptop.*
- *Find the **config.txt** file in the SD card file system*
- *Modify this file and add an extra line at the end:*
  `dtoverlay=pi3-miniuart-bt`



- *Insert your SD card back inside the Raspberry Pi*
- *Wait for Hass.io to boot*

Once Home Assistant is ready, please SSH into the machine using any tool available to you.



Confirm that it's visible to Home Assistant by running the following command:

`hassio host hardware`

This should give the following output:

Here you can see 2 serial ports:

- /dev/ttyAMA0 (which is the UART port of the Raspberry Pi and that's the port, which is used by Elelabs ZigBee RPi Shield)

If any of these steps failed, please check out the Troubleshooting section of this document.

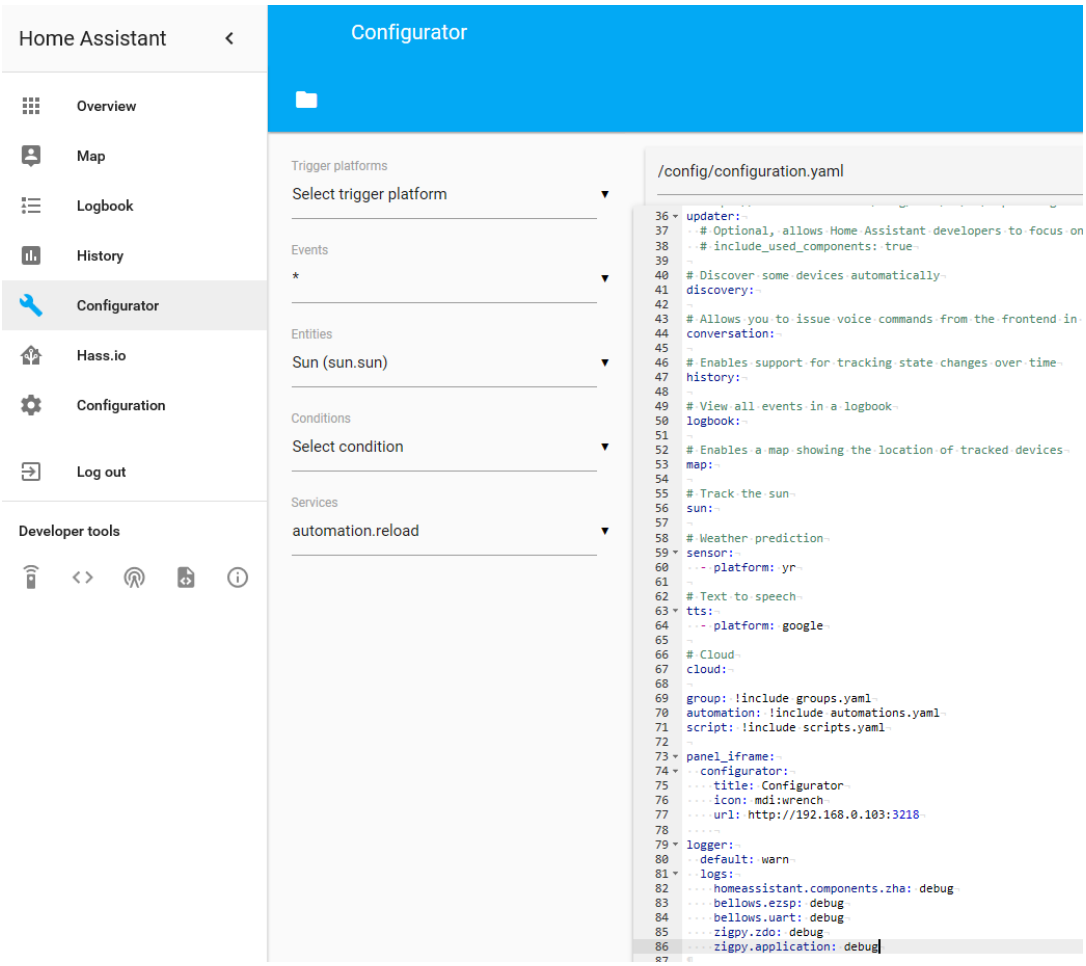
ZigBee HA Component configuration

To work with Elelabs ZigBee RPi 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.

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
```



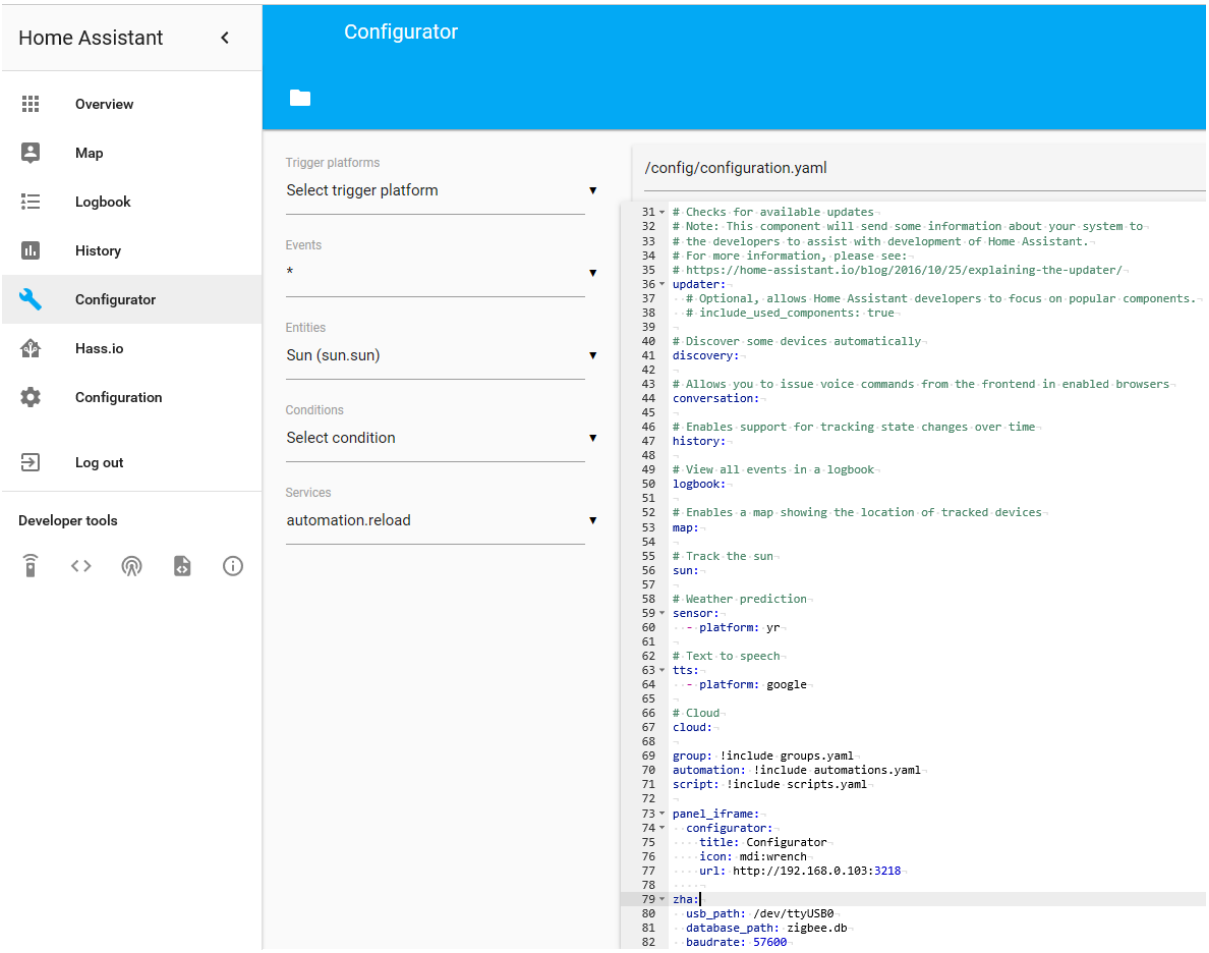
Then you can enter the following command in the SSH console to see the log output (do not forget to reload Hass.io after each modification of configuration file).

```
hassio homeassistant logs
```

Configure Zha component

To setup ZigBee Home Automation component to work properly with Elelabs ZigBee RPi Shield we need to add the following lines in the Configurator Interface to **/config/configuration.yaml**:

```
zha:
  usb_path: /dev/ttyAMA0
  database_path: zigbee.db
  baudrate: 57600
```



Here */dev/ttyAMA0* is from the output of "*hassio host hardware*" command issued in the previous chapter.

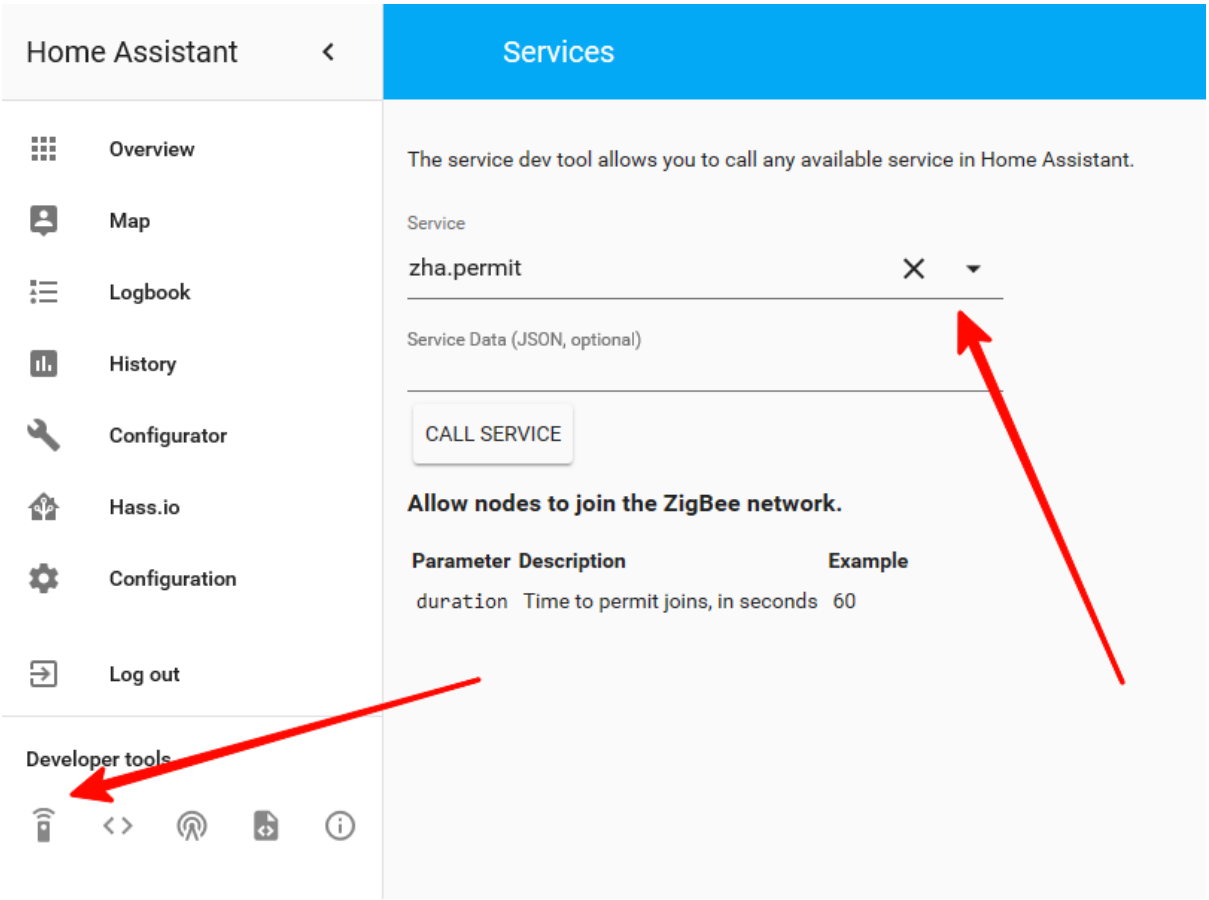
After the modification reload the Hass.io and the component would be added.

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 the Developers Tools and select *zha.permit* service.

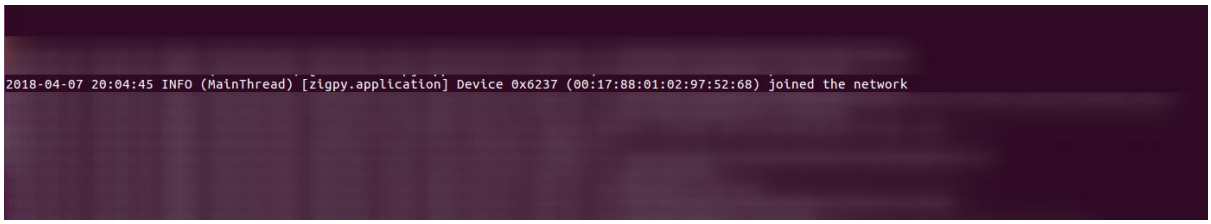


When you will call it, you have 60 seconds to add the device. It's easier to track the process in the logs, like this:

```
2018-04-07 19:47:30 DEBUG (MainThread) [bellows.uart] Data frame: b'1152a157547f15f9de7e'
2018-04-07 19:47:30 DEBUG (MainThread) [bellows.uart] Sending: b'82503a7e'
2018-04-07 19:47:30 DEBUG (MainThread) [bellows.ezsp] Application frame 85 (setPolicy) received
2018-04-07 19:47:30 DEBUG (MainThread) [bellows.ezsp] Send command getNodeId
2018-04-07 19:47:30 DEBUG (MainThread) [bellows.uart] Sending: b'125321575400ded467e'
2018-04-07 19:47:30 DEBUG (MainThread) [bellows.uart] Data frame: b'2253a157540d15b2a70f7e'
2018-04-07 19:47:30 DEBUG (MainThread) [bellows.uart] Sending: b'83401b7e'
2018-04-07 19:47:30 DEBUG (MainThread) [bellows.ezsp] Application frame 39 (getNodeId) received
2018-04-07 19:47:30 DEBUG (MainThread) [bellows.ezsp] Send command getEui64
2018-04-07 19:47:30 DEBUG (MainThread) [bellows.uart] Sending: b'235021575400c79997e'
2018-04-07 19:47:30 DEBUG (MainThread) [bellows.uart] Data frame: b'3350a157540cd129e69e4a4aa755bb087e'
2018-04-07 19:47:30 DEBUG (MainThread) [bellows.uart] Sending: b'8430fc7e'
2018-04-07 19:47:30 DEBUG (MainThread) [bellows.ezsp] Application frame 38 (getEui64) received
2018-04-07 19:56:15 INFO (MainThread) [homeassistant.components.zha] Permitting joins for 60s
2018-04-07 19:56:15 DEBUG (MainThread) [bellows.ezsp] Send command permitJoining
2018-04-07 19:56:15 DEBUG (MainThread) [bellows.uart] Sending: b'3451215754082964ae7e'
```

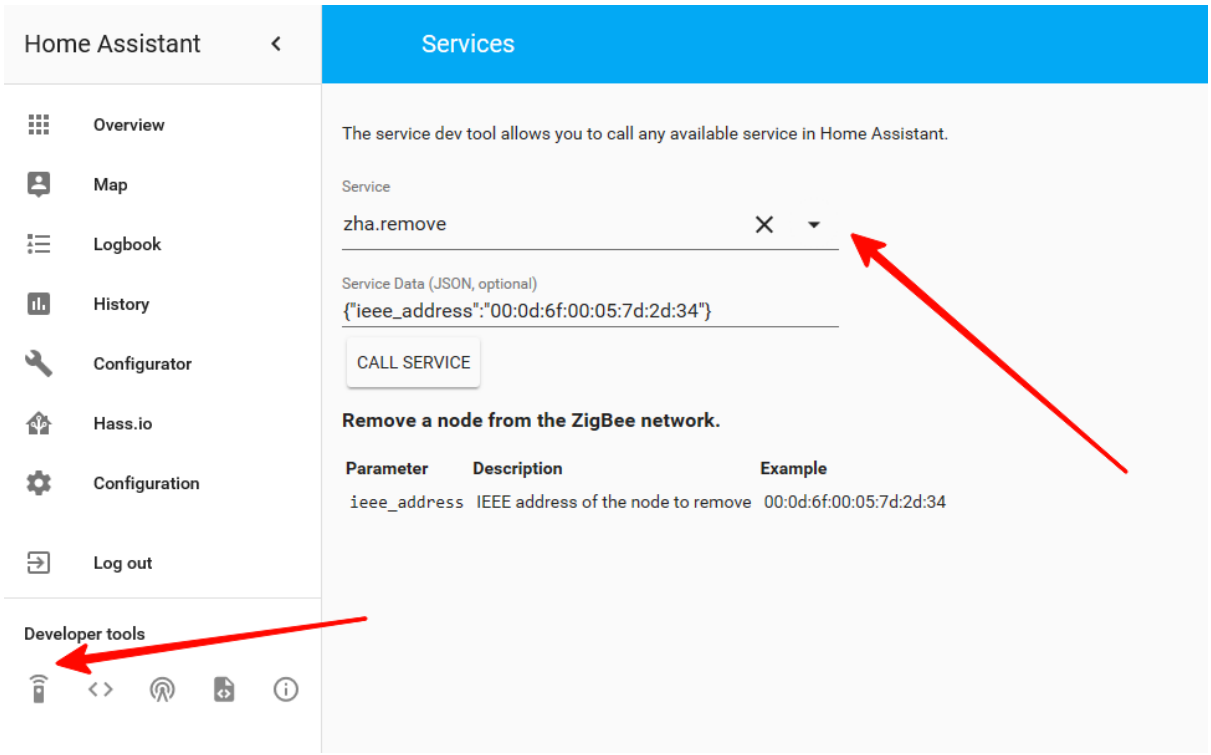
During this period, you need to follow Device manual to put it in Association mode. Sometimes you just need to give it power.

If the device is found, you will be able to see it in the logs.



Remove your device from Home Assistant

Open the Developers Tools and select *zha.remove* service.



Enter device IEEE address, which you could see in the logs during the inclusion process (another option is to download and investigate *ZigBee.db* file, which contains all the devices.

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

Example: Philips Hue Bulb

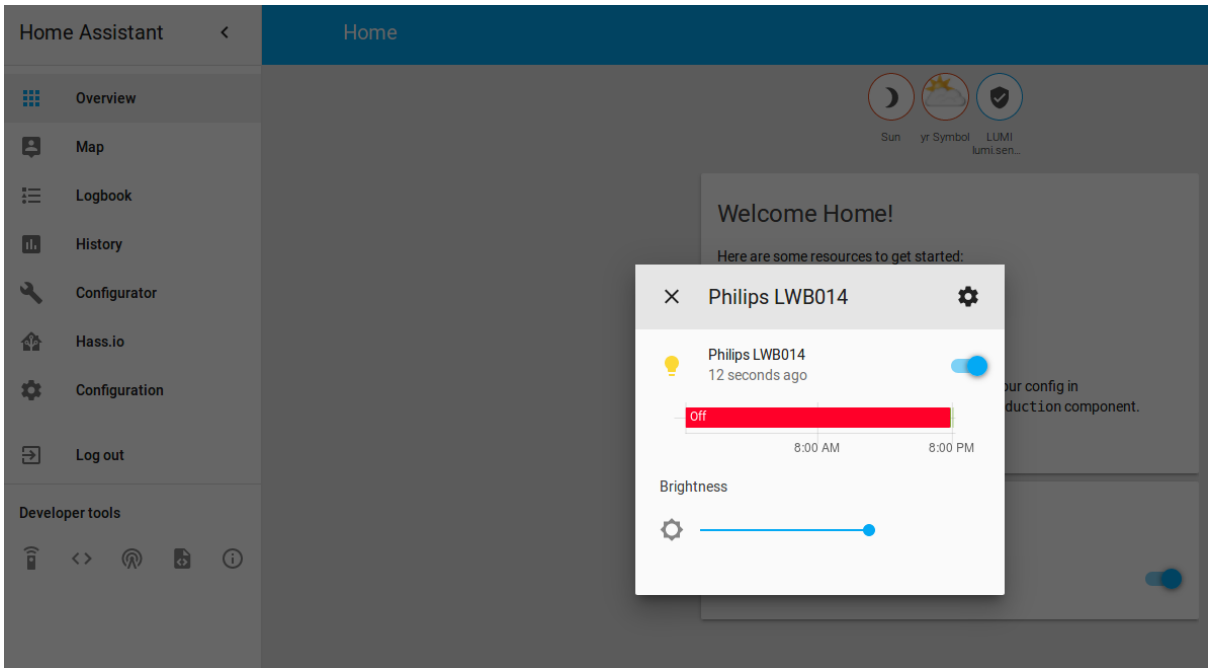
This example is done with Hue White Single bulb E26 but is applicable to other products as well.

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 permit service in the Developer Tools
- Power ON the Lightbulb
- Confirm it's added to the Home Assistant

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



Example: Xiaomi Smoke Sensor

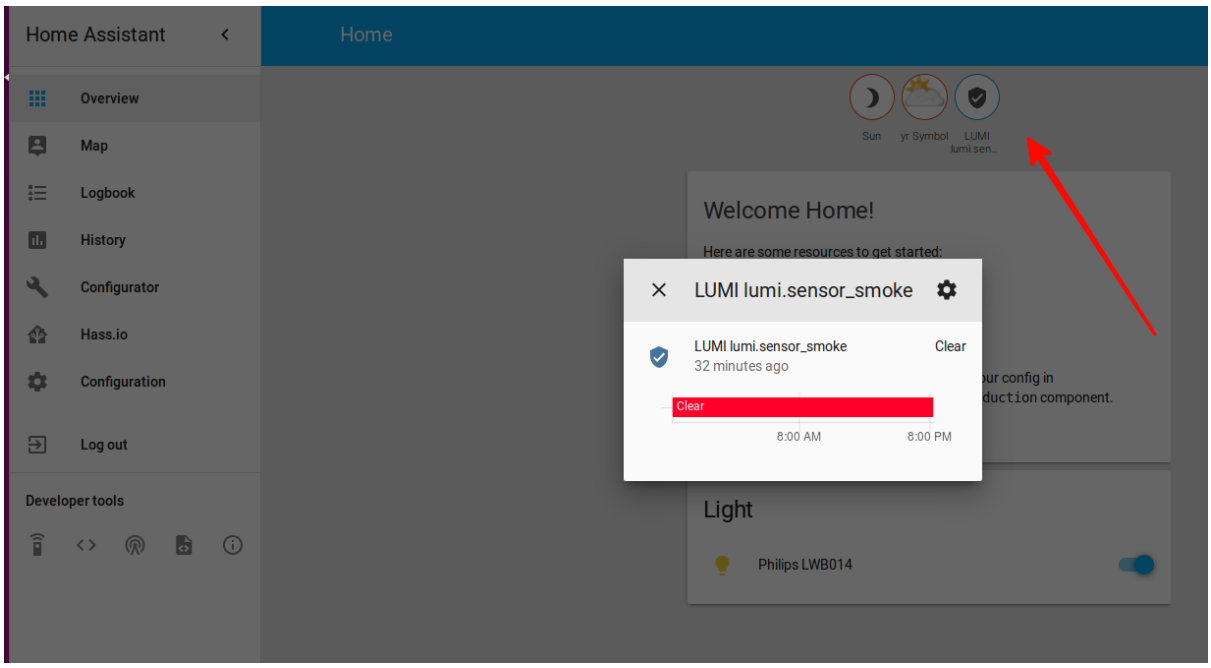
This example is done with Xiaomi Mijia Honeywell Fire Alarm Detector but is applicable to other Xiaomi ZigBee products.

To use it in Home Assistant:

- Call permit service in the Developer Tools
- Press the button on the sensor promptly within 1second
- Confirm it's added to the Home Assistant

Once included, do the test Alarm. To do it – hold the button on the sensor until it starts alarming and then release it.

The sensor icon should appear in Home Assistant:

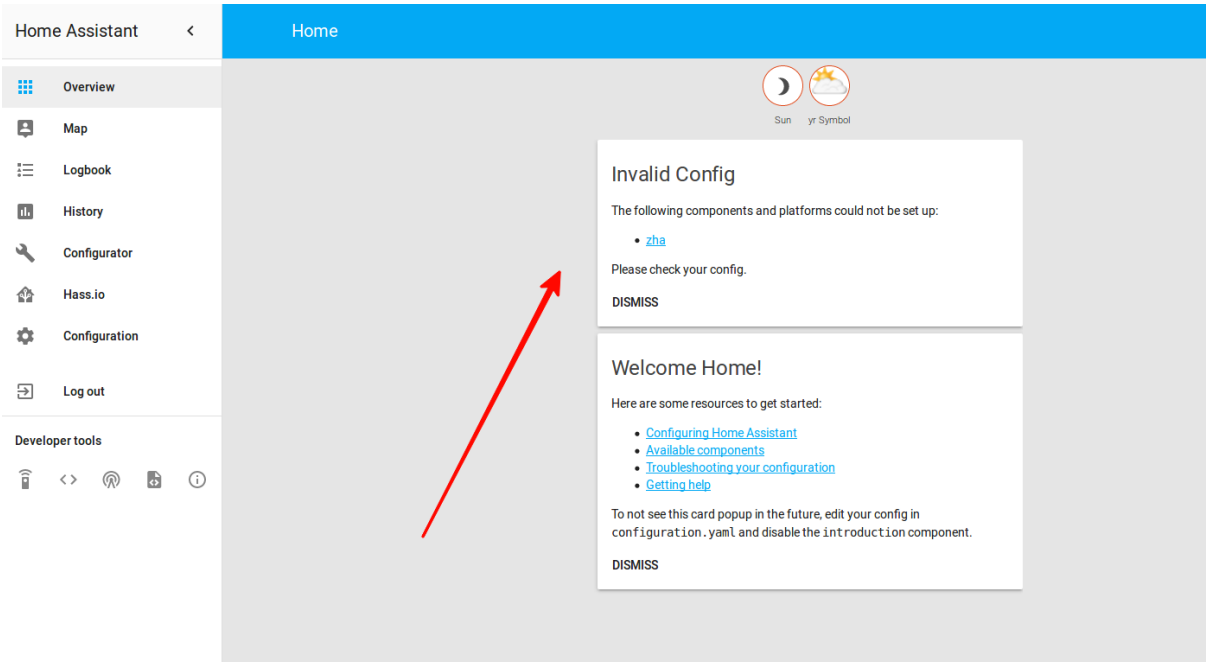


Troubleshooting

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

Home Assistant Zha component couldn't be set up

If you have modified the *configuration.yaml* file and found issues after you have restarted the Hass.io, like:



This probably means you have set up the ***usb_path*** part of the configuration wrong. Check it once again.

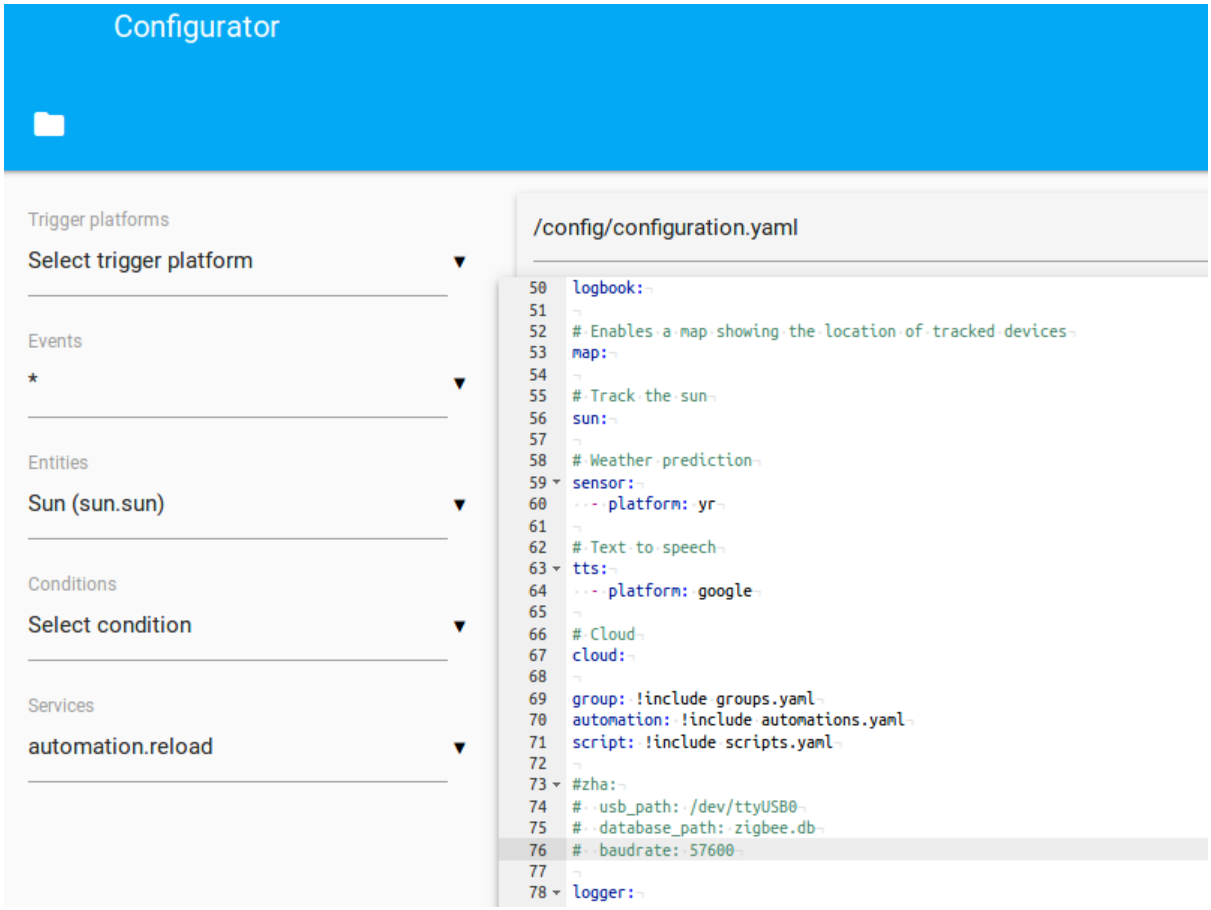
If the path is correct, then check the logs for potential issues and contact us.

Home Assistant does not boot at all

If you have modified the *configuration.yaml* file and the Home Assistant does not boot at all after the restart, first DON'T PANIC.

Revert using Configurator Addon

Just open the Configurator Addon Web UI and comment the **zha** component configuration like this:

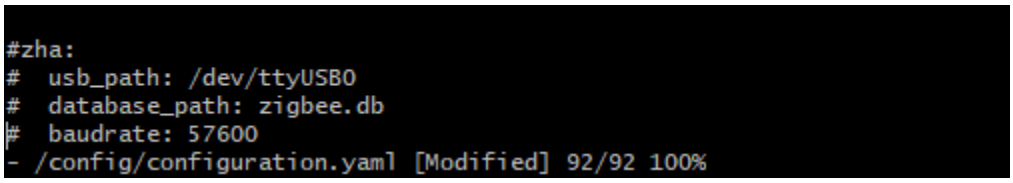


Then restart Hass.io and it will work as before.

Revert using SSH Addon

If you have the SSH Addon installed – login to the device using it. Open the *configuration.yaml* file using VI text editor and comment the **zha** component configuration, like:

```
vi /config/configuration.yaml
```



Then restart Hass.io and it will work as before.

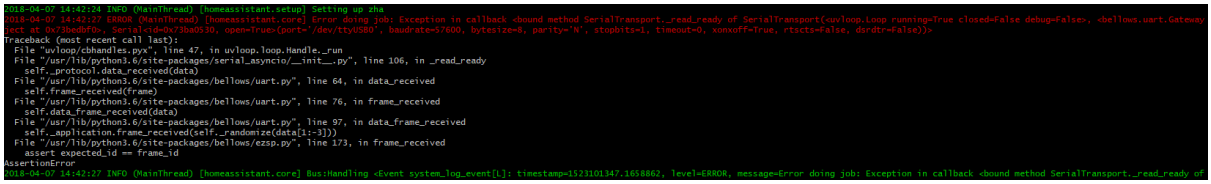
How to fix

Now once we have reverted everything back to normal let's resolve the issues.

First, we need to check the logs with the command:

```
hassio homeassistant logs
```

If you see the following issue:




```
  File "/usr/lib/python3.6/site-packages/bellows/ezsp.py", line 173, in
frame_received
    assert expected_id == frame_id
AssertionError
```

That means your Elelabs ZigBee RPi Shield has a different version, probably **v6.** While Home Assistant version is <=0.66.

First, **try to update** the Home Assistant, probably later versions will support v6 ezsp protocol and you will not have this issue.

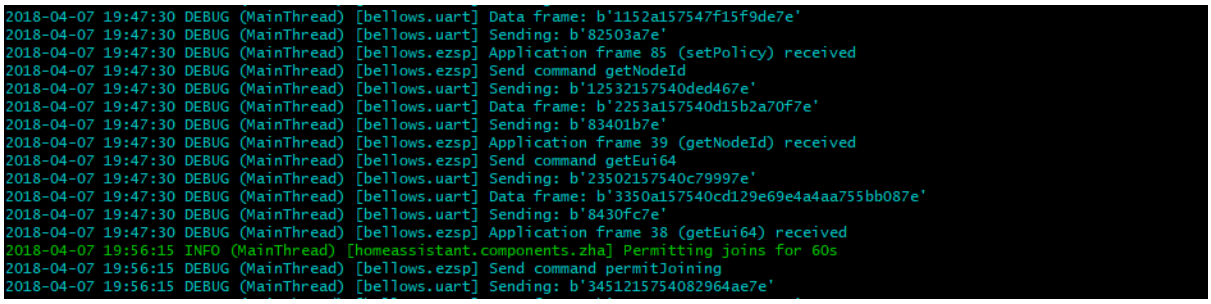
Second, try to perform the Software Downgrade on the Elelabs ZigBee RPi Shield to downgrade it to **v5**, which is supported on 0.66. To do it you better contact us at info@elelabs.com.

Zha component is loaded correctly but no zha.permit service

In this case, most probably the ZigBee component is simply not loaded. Try to investigate the logs for the issue or contact us.

Zha component is loaded, zha.permit is started, but can't add the device

First, investigate the logs and confirm you can see the start of the Permit command:



If present, then check what happens next:

The device tries to be added, but an error occurs

If you can see something like this error, when the device is added:

```
2018-04-07 19:56:55 INFO (MainThread) [zigpy.application] Device 0x27bb
(00:17:88:01:02:97:52:68) joined the network
2018-04-07 19:56:55 DEBUG (MainThread) [zigpy.application] Skip initialization for ex-
isting device 00:17:88:01:02:97:52:68
2018-04-07 19:56:55 DEBUG (MainThread) [bellows.uart] Data frame:
b'4754b157541515097e944a20aa5592099c4e276ceea867ab2b7e'
2018-04-07 19:56:55 DEBUG (MainThread) [bellows.uart] Sending: b'8520dd7e'
2018-04-07 19:56:55 DEBUG (MainThread) [bellows.ezsp] Application frame 63 (messag-
eSentHandler) received
2018-04-07 19:56:55 ERROR (MainThread) [zigpy.device] Failed ZDO request during device
initialization: Message send failure: EmberStatus.DELIVERY_FAILED
Traceback (most recent call last):
  File "/usr/lib/python3.6/site-packages/zigpy/device.py", line 51, in _initialize
    epr = yield from self.zdo.request(0x0005, self.nwk, tries=3, delay=2)
  File "/usr/lib/python3.6/site-packages/zigpy/util.py", line 52, in retry
    r = yield from func()
  File "/usr/lib/python3.6/site-packages/bellows/zigbee/application.py", line 248, in
request
    v = yield from send_fut
zigpy.exceptions.DeliveryError: Message send failure: EmberStatus.DELIVERY_FAILED
```

This means, that you are using Home Assistant <= **v0.66**. The issue is that those versions initialise the device table not as Elelabs ZigBee RPi Shield expect, so the device simply can't be added. Please try to upgrade your Home Assistant, probably later version has this issue resolved.

If not possible, please contact us at **info@elelabs.com** so we could help you with the issue.

There are no packets from the device at all

This most probably means that your device is already part of the different network and just can't join your new network. Try to reset it, using the manufacturer specific reset procedure.