Installation of HomeAssistant with RaspBee II as a Home Automation and Alarm Hub

Mathieu ABATI (mathieu-abati.com)

27 / 10 / 2025

Sommaire

Installation of HomeAssistant with RaspBee II as a Home Automation and Alarm Hub

In this tutorial, we will see how to install HomeAssistant on a Raspberry Pi 3 as a home automation hub. Zigbee connectivity will be provided by a RaspBee II extension board. Finally, we will add a home alarm feature to this setup.

Installing HomeAssistant

Install HomeAssistant on your Raspberry Pi by following the official documentation.

Once the installation is complete, access your instance at http://homeassistant:8123.

>  **Warning:** If you have forced a DNS server address on your PC that is different from your Internet box, this may prevent you from discovering your HomeAssistant instance via Avahi.

In the user settings, enable advanced mode.

Go to **Settings**, **Add-ons**, and install **Terminal & SSH**. Check *Start on boot* and *Watchdog*.

Go to the module configuration, in the **Network** section, check *Show disabled ports*, and enter port `22`, then save.

In the **Options** section, enter either a public SSH key (recommended) or a password.

Start the module. In the **Log** tab, after a few moments, you should see that SSH is listening on port `22`.

To access a terminal, you can now either use the web interface of the module or, directly from your PC:

```
1 ssh root@homeassistant
```

>  **Note:** It is recommended to configure your Internet box so that HomeAssistant has a fixed IP.

Installing RaspBee

Connect the RaspBee II board.

From your PC, access the SD card to edit the `config.txt` file. It is located on the first partition of the SD card. If your SD card is `/dev/sda`:

```
sudo mount /dev/sda1 /mnt
```

Edit the file `/mnt/config.txt`, and add at the end:

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

Unmount the SD card:

```
sudo umount /mnt
```

Start HomeAssistant, go to its web interface, then **Settings**, **Add-ons**, and install **Mosquitto Broker**. Check *Start on boot* and *Watchdog*.

In the add-ons, use the menu at the top right then **Repositories**, to add the repository https://github.com/zigbee2mqtt/hassio-zigbee2mqtt, and then install Zigbee2MQTT. Check *Start on boot*, *Watchdog*, and *Show in sidebar*.

Complete the configuration, in the serial section, add:

```
port: /dev/ttyAMA0
adapter: deconz
```

Start Zigbee2MQTT.

Go to **Settings**, **Devices & Services**, the MQTT integration should have been auto-detected, click on its **Configure** button to complete its setup. If the MQTT integration does not appear, restart HomeAssistant.

From the icon in the sidebar, you can access the Zigbee2MQTT interface. From there you can pair your devices.

Alarm

For the home alarm, we will use Alarmo.

To install it, connect via SSH and run:

```
cd config
mkdir custom_components
cd custom_components
```

Get the link to the latest ZIP version here: https://github.com/nielsfaber/alarmo/releases, and install it (using the latest version):

```
mkdir alarmo && cd alarmo
wget https://github.com/nielsfaber/alarmo/releases/download/v1.10.9/
    alarmo.zip
unzip alarmo.zip
rm alarmo.zip
```

Restart HomeAssistant.

Go to **Settings**, **Devices & Services**, and add the Alarmo integration.

Alarmo now appears in the sidebar. Add the different sensors to the Alarmo area, and configure the various actions and notifications.

You can also connect a keypad to your automation hub to control alarm and know its status.

Accessing HomeAssistant from Outside

Configure your Internet box to expose port 443 of HomeAssistant to the external port of your choice. Below, we will use port 12345.

Create a domain on https://www.duckdns.org. We assume below that your domain is myha.duckdns.org

Install the DuckDNS add-on in HomeAssistant. Configure the token and domain, and in the **Let's Encrypt** section, modify the line as follows:

```
accept_terms: true
```

Check *Start on boot* and *Watchdog*, then start DuckDNS.

Now you need to configure HomeAssistant to use the Let's Encrypt certificate. We could configure the HomeAssistant HTTP server to use these certificates, but then we would no longer have access with the mobile app to the service locally with the local domain; we could no longer connect either via HTTP or HTTPS, as the certificate would not match. So we will configure a proxy.

Connect via SSH, and edit the file `config/configuration.yaml` to add:

```
http:
  use_x_forwarded_for: true
  trusted_proxies:
    - 172.30.33.0/24
```

Go to the developer tools in the HomeAssistant interface, at the top area we can check the configuration and restart HomeAssistant.

Install **Nginx Home Assistant SSL proxy**, and in the configuration provide the DuckDNS domain. Check *Start on boot* and *Watchdog*. Start the proxy.

Local access via HTTP is at http://homeassistant:8123, remote access via HTTPS is at https://myha.duckdns.org:12345.

In the mobile app settings, set the URL to https://myha.duckdns.org:12345. Allow the app to always have access to the location.

> **Note:** Setting the internal URL in the app causes notifications to not work well when changing networks. Leave this field empty.

Some Customizations

By editing the `configuration.yaml` file, you can remove `default_config:` and add what it includes by default but excluding what you don't want. See: https://www.home-assistant.io/integrations/default_config/.

You can find the default content of `default_config` in the HomeAssistant sources.

Troubleshooting

In Case of SD Card Corruption

After a few years, the SD card may show signs of wear due to frequent writes.

Start by reinstalling HomeAssistant on a new SD card. Use the latest backup to have a functional system again.

On your PC, mount partition #8 of the corrupted SD card.

Copy `home-assistant_v2.db` and `zigbee2mqtt` from the `supervisor/homeassistant` directory of the partition.

Also copy all the `supervisor/homeassistant/.storage/lovelace*` and `supervisor/homeassistant/.storage/local_todo*` files.

Check if the database is corrupted and try to repair it with:

```
cd /root/config
sqlite3 "home-assistant_v2.db" ".recover" | sqlite3 home-assistant_v2.db.fixed
```

On your new installation, copy this file via SSH to `/root/config`. Stop the HomeAssistant core:

```
ha core stop
```

Rename `home-assistant_v2.db.fixed` to `home-assistant_v2.db`, and copy the `lovelace*`, `local_todo*` files to `/homeassistant/.storage/`.

Restart the service:

```
1 ha core start
```

Source