

2nd Steps - Creating Devices and End Points / Entities

HA is installed in RPi4. I created 2 devices using ESP32 and ESP8266.

The screenshot shows the Home Assistant interface. On the left, there's a sidebar with icons for Overview, Energy, Map, Logbook, History, and ESPHome. The ESPHome icon is highlighted with a blue background. The main area has a header "ESPHome" with a logo. Below it, there are two cards: one for "esp32" labeled "ONLINE" and another for "esp8266" also labeled "ONLINE". Each card has "EDIT" and "LOGS" buttons and a three-dot menu icon.

Firmware was first uploaded by connecting ESP device to the RPi4, then install addons from Configuration Menu:

The screenshot shows the "Configuration" section of the Home Assistant configuration menu. On the left, there's a sidebar with Logbook, History, ESPHome, File editor, Node-RED, Media, Developer Tools, Configuration (which is selected and highlighted with a blue background), Notifications, and Jibon. The main area lists several add-ons with icons and descriptions:

- Automations & Scenes: Manage automations, scenes, scripts and helpers
- Blueprints: Pre-made automations and scripts by the community
- Add-ons, Backups & Supervisor: Create backups, check logs or reboot your system
- Dashboards: Create customized sets of cards to control your home
- Tags: Trigger automations when an NFC tag, QR code, etc. is scanned
- People & Zones: Manage the people and zones that Home Assistant tracks
- Settings: Basic settings, server controls, logs and info

Following

The screenshot shows the "Add-ons" page. At the top, there are tabs for Add-ons (which is selected and highlighted with a blue background), Backups, and System. Below are four cards representing installed add-ons:

- ESPHome: ESPHome add-on for intelligently managing all your ESP8266/ESP32 devices
- File editor: Simple browser-based file editor for Home Assistant
- Node-RED: Flow-based programming for the Internet of Things
- Z-Wave JS: Control a ZWave network with Home Assistant Z-Wave JS

Enter to ESPHome:

ESPHome

Current version: 2022.1.4 ([Changelog](#))

6 Rating

Host

Auth

Ingress

ESPHome add-on for intelligently managing all your ESP8266/ESP32 devices.
Visit the [ESPHome](#) page for more details



ESPHome

Start on boot

Make the add-on start during a system boot



Hostname

a0d7b954-esphome

Watchdog

This will start the add-on if it crashes



Add-on CPU Usage

0 %



Show in sidebar

Add this add-on to your sidebar



Add-on RAM Usage

3.2 %



STOP

RESTART

Created by Paint S

OPEN WEB UI UNINSTALL

Click to OPEN WEB UI:

ESPHome Dash will appear. At the bottom-right corner there is Add new Device Button. Click on this button to start the wizard. It is important to select the board type in the wizard or edit in script later. See below the initial 2 scripts:

[For ESP32](#)

```
# declare script name
esphome:
    name: esp32

# define parameters for this script
esp32:
# board type is ok because we're getting input
    board: nodemcu-32s
    framework:
        type: arduino

# Enable logging
logger:

# Enable Home Assistant API
api:

ota:
    password: "517275055223d8eae7de0835606a68e1"

wifi:
    ssid: !secret wifi_ssid
    password: !secret wifi_password
```

```

# Enable fallback hotspot (captive portal) in case wifi connection
fails
ap:
  ssid: "Esp32 Fallback Hotspot"
  password: "ETvJcTZP4S1z"

captive_portal:

binary_sensor:
- platform: gpio
  pin:
    number: 32
    mode: INPUT_PULLUP
    inverted: True
  name: "Living Room Window"
  device_class: window
  filters:
    - delayed_on: 10ms

# Example configuration entry
light:
- platform: binary
  name: "Desk Lamp"
  output: light_output

output:
- id: light_output
  platform: gpio
  pin: 23

# Code for remote

#remote_transmitter:
#  pin: GPIO33
#  carrier_duty_percent: 50%

# Individual switches
#switch:
#  - platform: template
#    name: "LG TV ON/Off"
#    turn_on_action:
#      remote_transmitter.transmit_lg:
#        data: 0x20DF10EF # power on/off
#        nbits: 32

```

For ESP8266 Nodemcu:

```
esphome:  
    name: esp8266  
  
esp8266:  
    board: nodemcuv2  
  
    # Enable logging  
    logger:  
  
    # Enable Home Assistant API  
    api:  
  
    ota:  
        password: "db8b1bd647ea4d4599a7a1fe8ee498d3"  
  
    wifi:  
        ssid: !secret wifi_ssid  
        password: !secret wifi_password  
  
        # Enable fallback hotspot (captive portal) in case wifi connection  
        fails  
        ap:  
            ssid: "Esp8266 Fallback Hotspot"  
            password: "Ib7giTom64VN"  
  
    captive_portal:  
  
    binary_sensor:  
        - platform: gpio  
            pin:  
                number: D3  
                mode: INPUT_PULLUP  
                inverted: True  
                name: "Living Room Window"  
                device_class: window  
                filters:  
                    - delayed_on: 10ms  
  
    # Example configuration entry  
    light:  
        - platform: binary  
            name: "Desk Lamp"  
            output: light_output  
  
    output:  
        - id: light_output  
            platform: gpio  
            pin: D2
```

Automations

In Devices and Services section under Configuration menu, enter a device in ESPHome. Click on n Devices (ex 1 Devices), There is a section called Automation. We can add multiple automations. In each automation there are 4 section:

NAME OF AUTOMATION Triggers Conditions Actions

Concept of a state machine. It can build Trigger Action network between multi tier devices.