

Home Automation for tinkerers

Abílio Costa
amfcalt@gmail.com

Once upon a time...

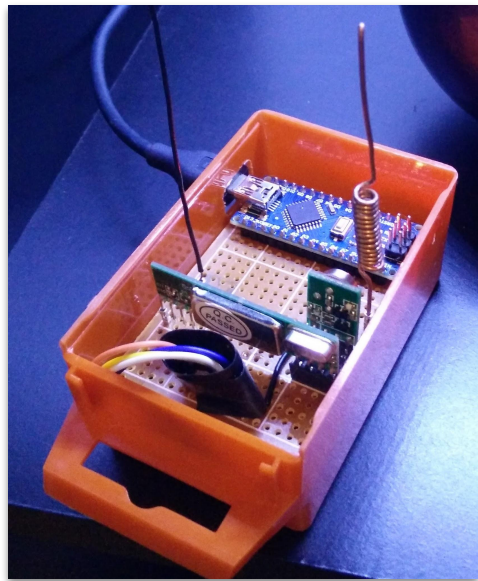
Where it all begun

- I had 3 wireless power sockets! (yay?)
- But I was using only one. Why?
 - Only a single remote:
 - How to use the other two in different rooms?
 - They were dumb. ㄟ(ˉ_ˉ)ﾉ
 - It would be nice to have one of them turn on/off on a schedule?



Poor man's solution

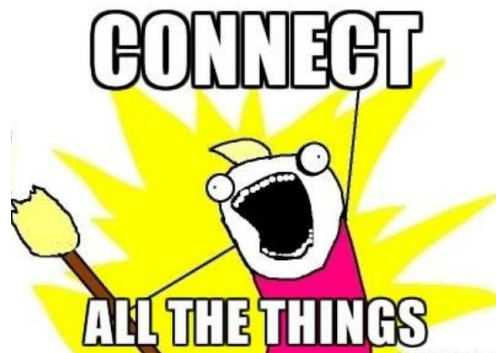
- An Arduino Nano + 433MHz RF transmitter + RF receiver.
 - Total cost: less than 5€.
 - Arduino sketch using the RC Switch library.
 - USB to a Raspberry Pi for the brains.
 - Script on the Raspberry Pi; exposing HTTP endpoints.
- My home was now so very smart!
 - Control each power socket through my phone.
 - Office desk power turns on automatically when I get home.
 - Bedroom lamp turned on automatically after the morning alarm.
 - I could buy more power sockets, even from other brands!
- The same idea can be used to interact with many other things:
 - Alarm sensors; Doorbells; Garage doors; etc.



Next step: home automation software

Why?

- Better management (my script wasn't going very far).
- Allow integrating other devices besides the power plugs.
- Make devices from different brands / protocols talk to each other.
- UI included!





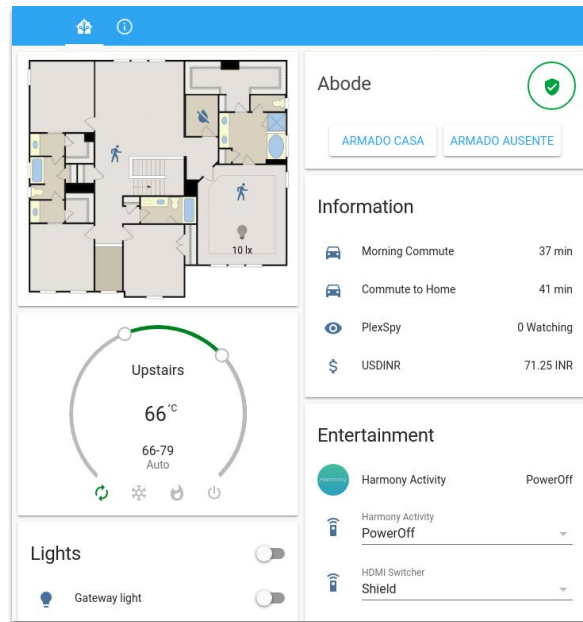
Home automation solutions

- Open-source software:
 - Home Assistant
 - Domoticz
 - OpenHAB
 - Pimatic
- Commercial hardware + software:
 - SmartThings
 - Vera
 - Xiaomi

Home Assistant

Home Assistant

- Good looking and customizable web UI (uses Google Polymer).
- Lightweight; extra functionality added with a plugin-like system.
- Very powerful automation engine (IFTTT on steroids).
- Autodiscovery: many devices will just show up without any config!
- Local: no cloud dependency!
- REST API available.
- Open source and written in Python.
- Very fast pace of development: support for new devices on the market appears quickly.





Home Assistant: typical environments

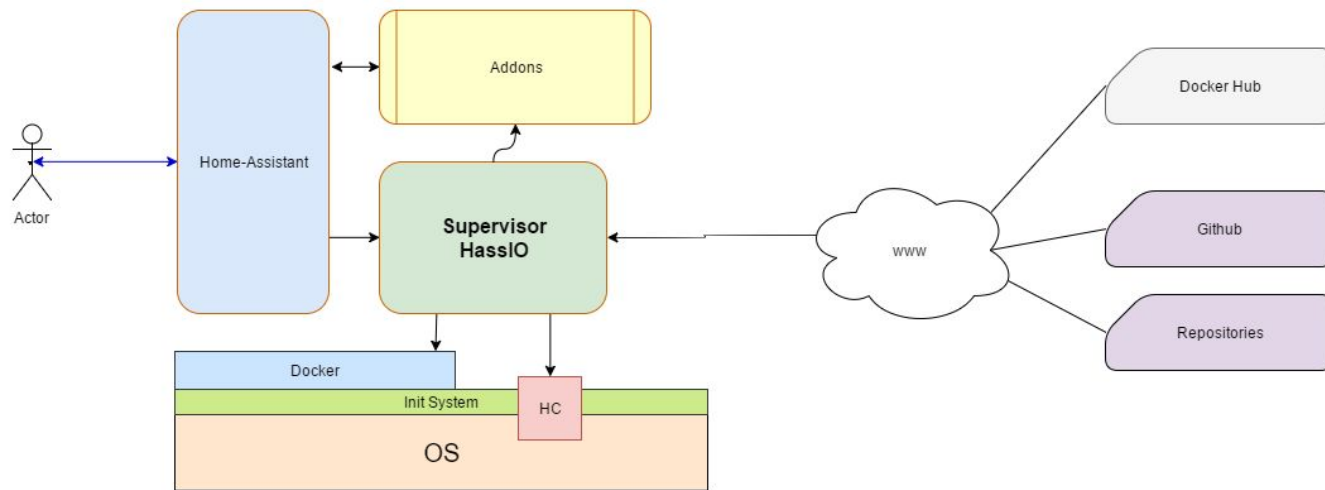
- Its Python, runs nearly “everywhere” (even on Android).
- Linux or Mac OS recommended.
- Official Docker images available.
- Hassbian: a Raspbian based distro with HA included, for the Raspberry Pi.
- Hass.io: turning a Raspberry Pi into a HA hub with UI based setup.



Hass.io

- Home Assistant-focused linux distro (like OpenELEC for HA).
- Managed through web UI (no command line required).
- Automatic backups of HA and Hass.io config.
- HA updates.
- Extended functionality with add-ons:
 - MQTT Broker; SSH Server; Samba; Duck DNS updater; Let's Encrypt manager; ...
 - Third-party add-ons allowed and available.
 - UI based installation.
- Can be manually installed on any linux distro that has Docker.

Hass.io



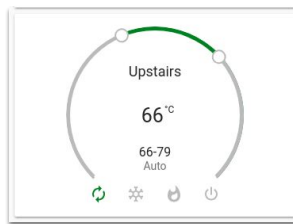
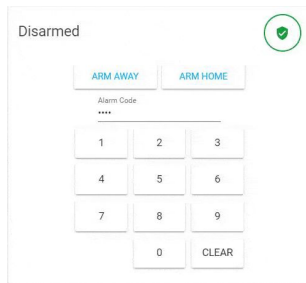


Home Assistant: components

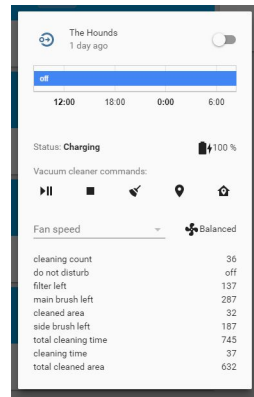
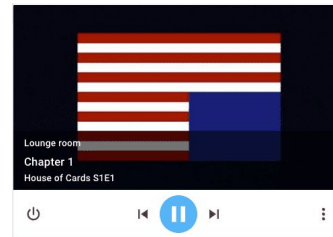
- Plugin-like system, called components, with hundreds of integrations with devices/protocols.
 - Philips Hue, IKEA Trådfri, Alexa, HomeKit, Google Assistant, Google Cast, Kodi, etc.
- Enabling components for different brands makes them talk to each other.
- Non-device components to extend capabilities:
 - Image processing/recognition (QR Codes, faces, license plates, etc).
 - Signal processing.
 - Statistics.
 - Command line integration.
- No need to manually download anything: just enable the component.

Home Assistant: entity types

- Air Quality
- Alarm Control Panel
- Binary Sensor
- Climate
- Cover
- Fan
- Light
- Lock



- Media Player
- Remote
- Sensor
- Switch
- Vacuum
- Water Heater
- Weather





Home Assistant: simple automation

```
automation:  
  alias: Turn on the lights when the sun sets  
  trigger:  
    platform: sun  
    event: sunset  
  action:  
    service: light.turn_on  
    entity_id: light.living_room
```



Home Assistant: simple automation

```
automation:
  alias: Turn on the lights when the sun sets
  trigger:
    platform: sun
    event: sunset
  condition:
    condition: state
    entity_id: group.people
    state: 'home'
  action:
    service: light.turn_on
    entity_id: light.living_room
```

Home Assistant: python scripts

- Use Python for more complex logic.
- Sandbox environment (can use only a pre-defined subset of python modules).
- Example: count how many people are at home:

```
home = 0
for entity_id in hass.states.entity_ids('device_tracker'):
    state = hass.states.get(entity_id)
    if state.state == 'home':
        home = home + 1

hass.states.set('sensor.people_home', home, {
    'unit_of_measurement': 'people',
    'friendly_name': 'People home'
})
```





Home Assistant: python scripts

- Example: notify when a light is on, ignoring the ones in exclusion list.

```
excluded = ['light.desk_light', 'light.office_light']
entities_on = []
for entity_id in hass.states.get('group.lights').attributes['entity_id']:
    if hass.states.get(entity_id).state is 'on' and entity_id not in excluded:
        entities_on.append(hass.states.get(entity_id).attributes["friendly_name"])

if len(entities_on) > 0:
    notification_title = "Home Assistant: Some lights are on"
    notification_message = "The following lights are on: " + ', '.join(entities_on)

    hass.services.call('script', 'notifications_send', {
        'title' : notification_title, 'message': notification_message})
```

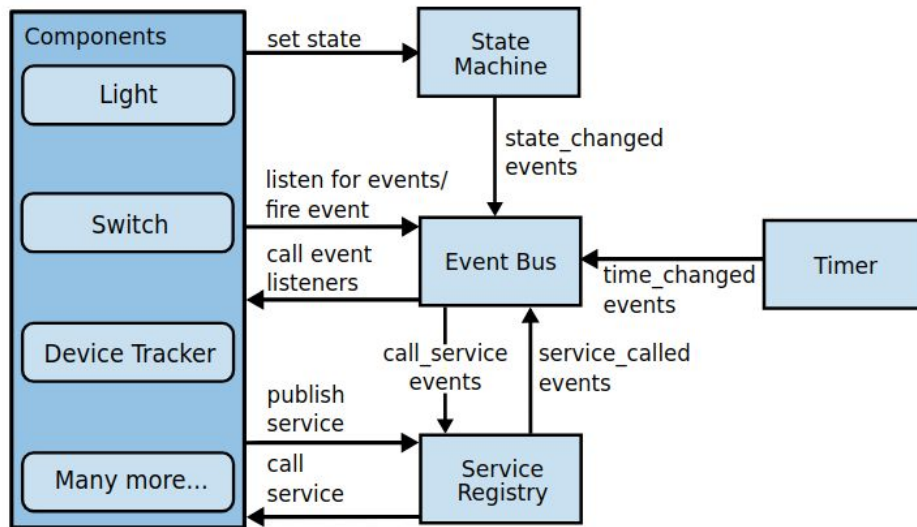


Some automation ideas

- Presence-based lights.
- Time-based lights.
- Wake up sunrise simulator.
- Wake up music.
- Arriving home routine.
- Movie mode.
- Window blind control (sun-based, weather-based).
- Automatic dehumidifier.
- Security notifier with camera snapshots to telegram.
- Robot vacuum scheduler.

Home Assistant: developing a new component

Home Assistant: architecture





Home Assistant: developing a new component

- [components/example/sensor.py](#)

```
def setup_platform(hass, config, add_devices, discovery_info=None):  
    add_devices([ExampleSensor()])
```

```
class ExampleSensor(Entity):  
    def __init__(self):  
        self._state = None  
  
    @property  
    def name(self):  
        return 'Example Temperature'  
  
    @property  
    def state(self):  
        return self._state  
  
    @property  
    def unit_of_measurement(self):  
        return TEMP_CELSIUS  
  
    def update(self):  
        self._state = 23
```

- [configuration.yaml](#)

```
sensor:  
    platform: example
```



Home Assistant: where to get help?

- “Getting started” docs: <https://www.home-assistant.io/getting-started>
- Community forums: <https://community.home-assistant.io>
- Developer docs: <https://developers.home-assistant.io>
- Discord chat: <https://www.home-assistant.io/join-chat>
 - Both user and dev channels available.

**Hardware:
its cheap (and custom)**



Common communication technologies

- Basic RF communication
- Wifi
- Bluetooth (mesh!)
- Zigbee
- Zwave

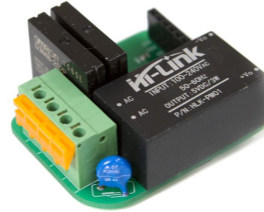
Hardware: Xiaomi ecosystem

- Xiaomi Zigbee devices:
 - Zigbee hub (HomeKit integration).
 - Many types of sensors.
 - Buttons/switches.
 - Power outlets.
 - Usually very reliable and fast.
- Yeelight tunable white and color bulbs.
- Xiaomi ecosystem (including Yeelight) allows automations.
- **Local API and Home Assistant integration.**
- Cheap, cheap, cheap!



Hardware: ESP based devices

- Many commercial devices using the Espressif ESP chips:
 - Sonoff.
 - Shelly.
 - BH Onofre.
 - Blitzwolf.
- Price range: 5€ - 20€.
- Can all be flashed with open source firmware.



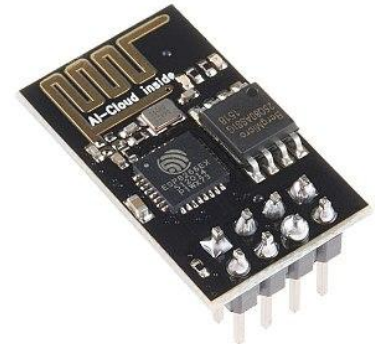
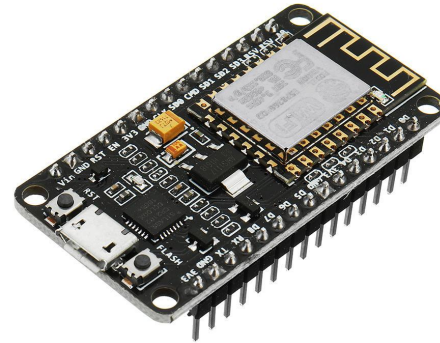


Custom firmware for ESP devices

- Many open-source firmwares available:
 - ESPHome
 - Tasmota
 - ESPurna
 - ESPEasy
- No-cloud dependency!
- Local and direct integration in Home Automation software.
- You can change and improve the firmware (its open-source!).
- Not that hard to flash on most devices.

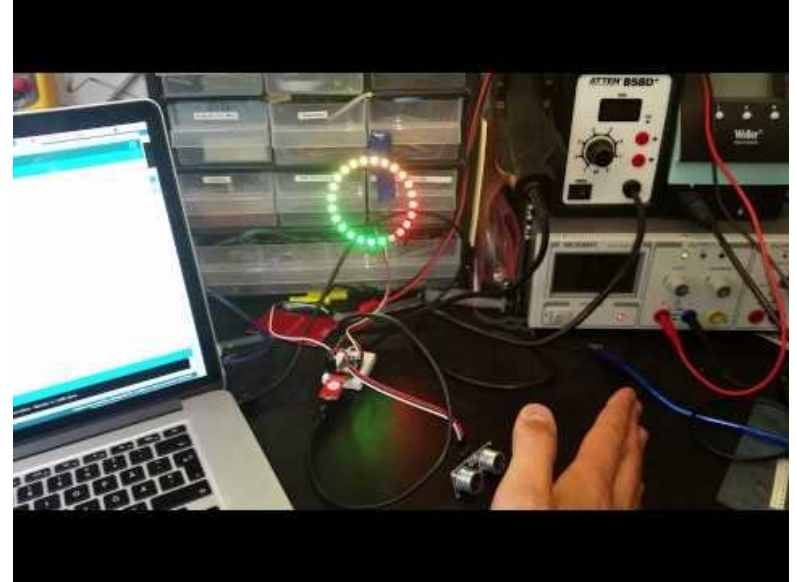
Hardware: custom modules

- ESP8266/ESP32 chips:
 - Wifi + Bluetooth (ESP32).
 - Small.
 - **Very easy to program.**
 - Many examples online + huge community.
- MySensors (mysensors.org)
 - Open source library for wireless sensors and actuators.
 - Long range.
 - Mesh network.
 - Good for battery projects.
 - Good tutorials.



Hardware: custom modules

- Bluetooth presence detection.
- Soil humidity sensor.
- Rain gauge.
- Bed occupancy sensor.
- Power/Water meter pulse sensor.
- Parking sensor.
- These can be done for a few €.





Final tips

- Go local!
- Have a plan B.
- Get a voice assistant.

Thank you.

Abílio Costa
amfcalt@gmail.com

