Inteligentne budynki

mgr inż. Jerzy Sobczyk

Oprogramowanie

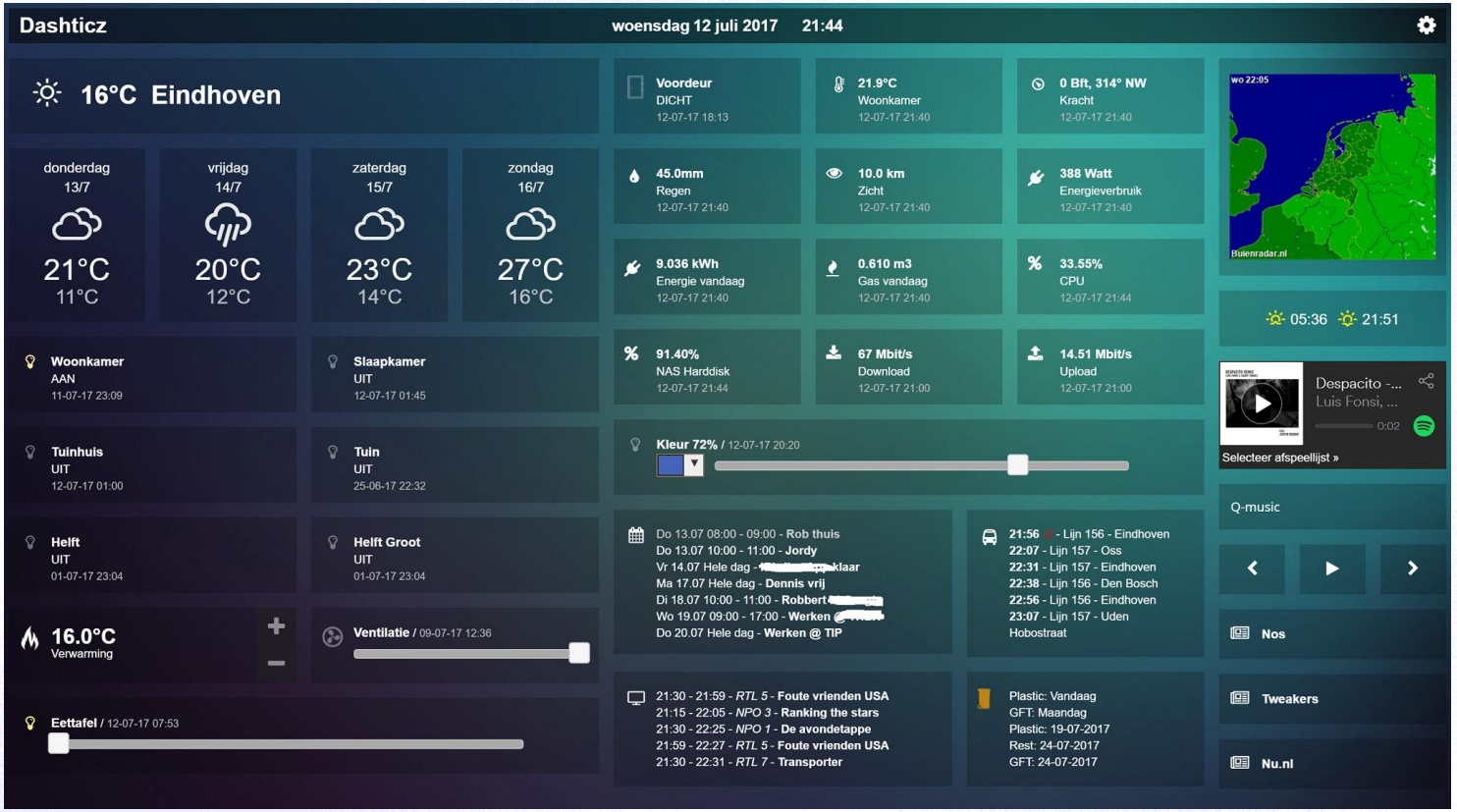
Plan wykładu

- Domoticz https://www.domoticz.com/
- OpenHab https://www.openhab.org/
- HomeAssistant https://www.home-assistant.io/
- Other platforms

Domoticz

- Domoticz to bardzo lekki system automatyki domowej pozwalajacy na monitorowanie i konfigurowanie różnych urządzeń takich jak oświetlenie, przełaczniki, rozmaite czujniki/mierniki temperatury, deszczu, wiatru, ultrafioletu, zużycia energii elektrycznej, zużycia gazu, zużycia wody i wielu innych. Powiadomienia/alarmy moga być wysyłane na urządzenia mobilne.
- Domoticz ma otwarte źródła i jest darmowy!
- Moze działac w systemach:
  - Rapberry Pi
  - Windows
  - Linux
  - Mac OSX
  - Synology NAS
  - FreeBSD
  - FreeNAS

Domoticz - panel

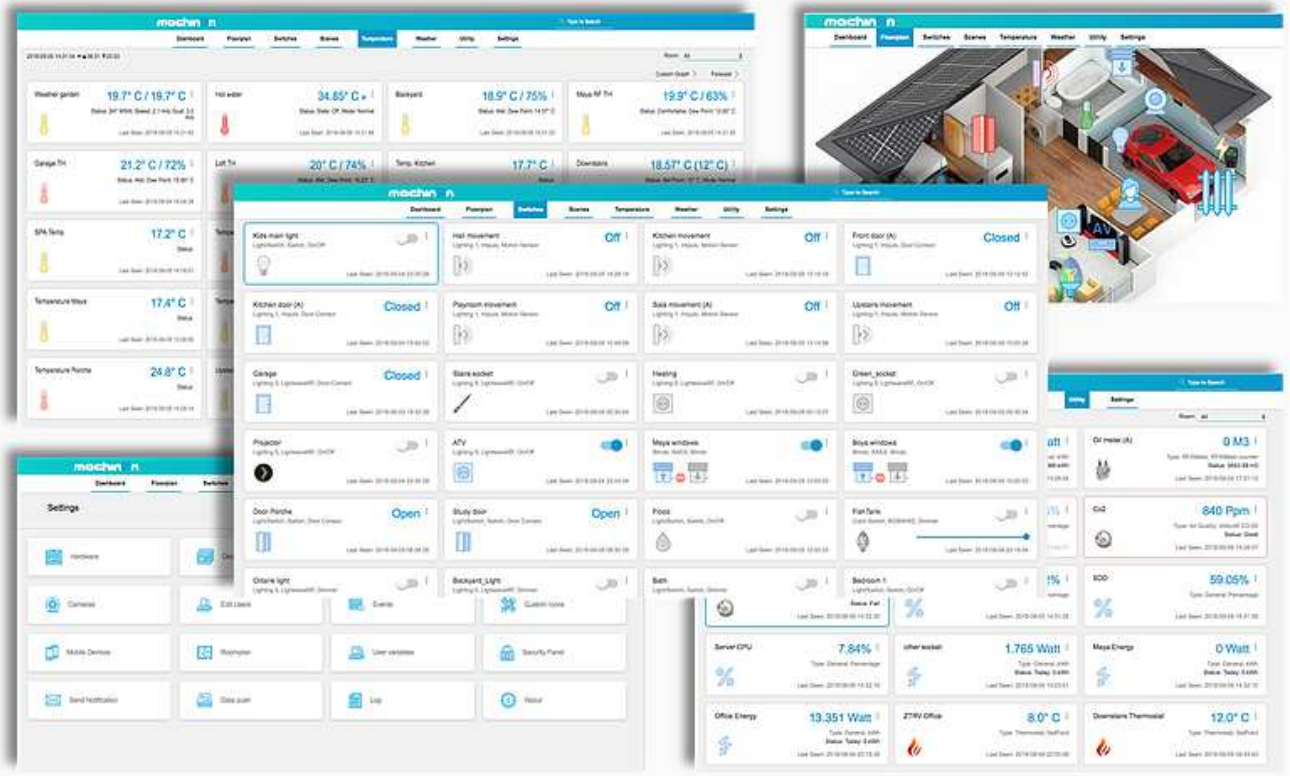


https://github.com/Dashticz/dashticz

Domoticz - panel w stylu Aurora

https://www.domoticz.com/wiki/Aurora_Theme

Domoticz - styl Machinon



https://www.domoticz.com/wiki/Machinon_Theme

Domoticz - skrypty

- Blocky to graficzny interfejs do JavaScriptu wygladający podobnie do Scratcha.
- Skrypty Lua i DzVents moga być zapisane w bazie lub w systemie plików.
- Skrypty w językach Bash, Python, Perl, ... działaja na zewnątrz programu Domoticz i wukorzystują JSON API do pobierania danych z Domoticza i wysyłania powiadomień, zgłaszania zdarzeń, modyfikowania zmiennych, ...
- JSON API
  `http://`[*username*:*password*@]*domoticz-ip*[:*port*]`/json.htm?`*api-call*

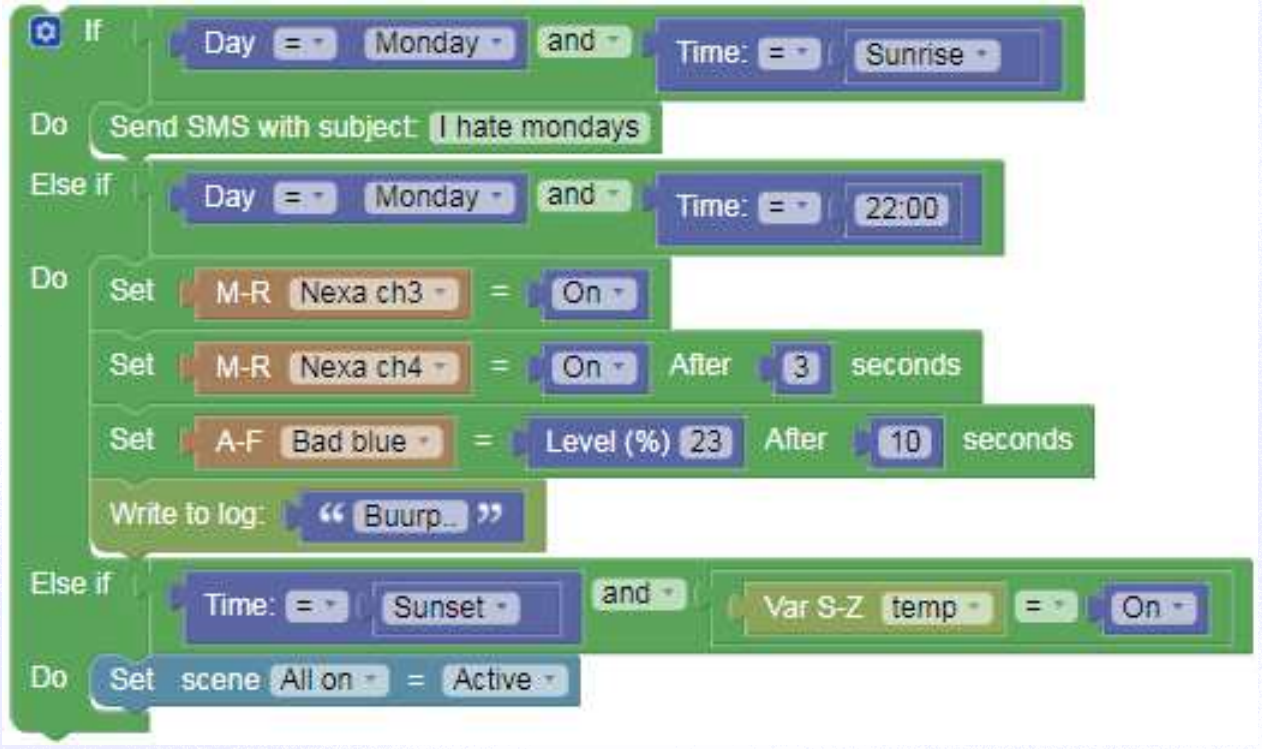
  {
  ”status” : ”OK”
  }

Domoticz - Blockly

- Blocky to rówież biblioteka dla JavaScriptu do tworzenia blokowych, graficznych języków programowania i edytorów.
- Zwykle jest uruchamiany w przegladarce i przypomina język Scratch.
- Jest projektem Google.
- Jest darmowy i ma otwarte źródła.
- Jest udostepniany na licencji Apache 2.0.

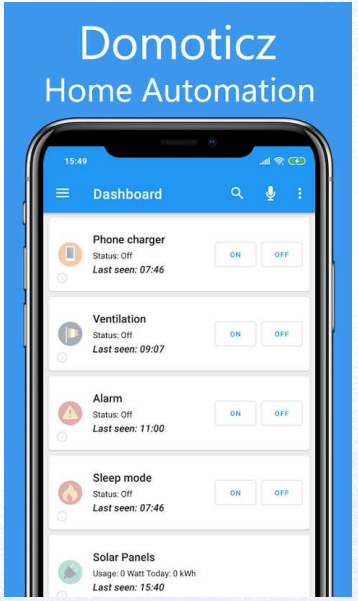
https://en.wikipedia.org/wiki/Blockly

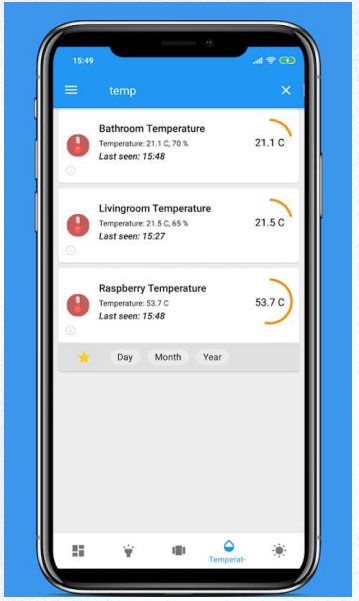
Domoticz - Blockly



https://www.domoticz.com/wiki/Blockly

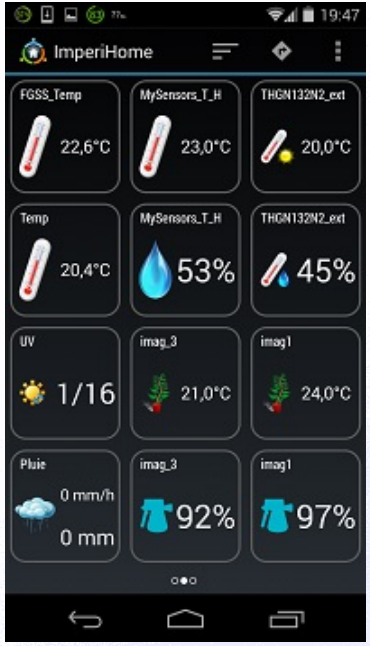
Domoticz - Android App





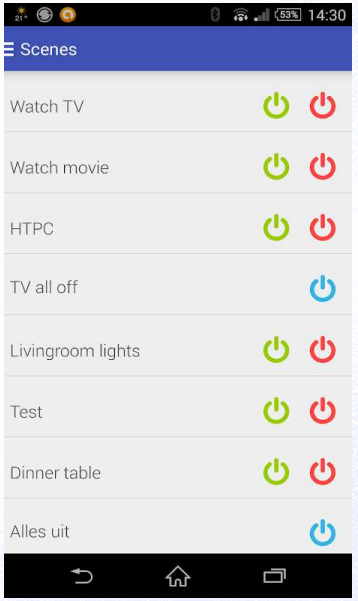
https://play.google.com/store/apps/details?id=nl.hnogames.domoticz

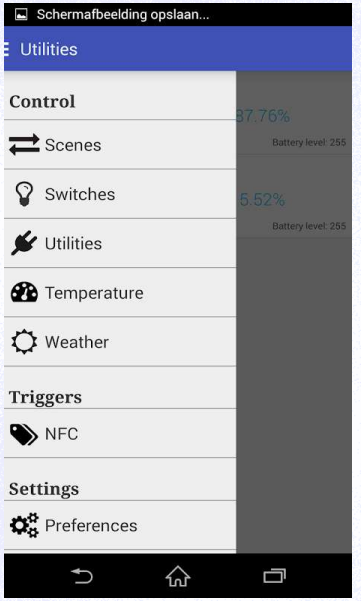
Domoticz - Android Imperi Home

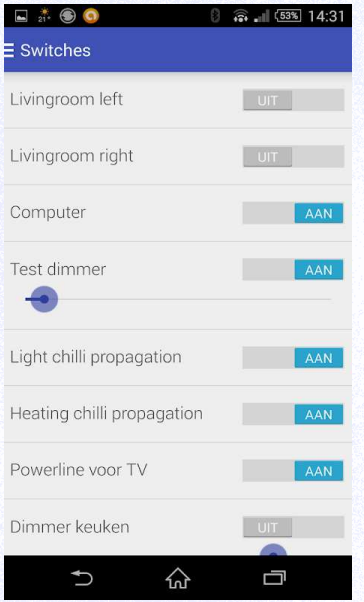


https://www.domoticz.com/wiki/ImperiHome

Domoticz - Android Dromotica







https://play.google.com/store/apps/details?id=me.sebastiaanschimmel.dromotica

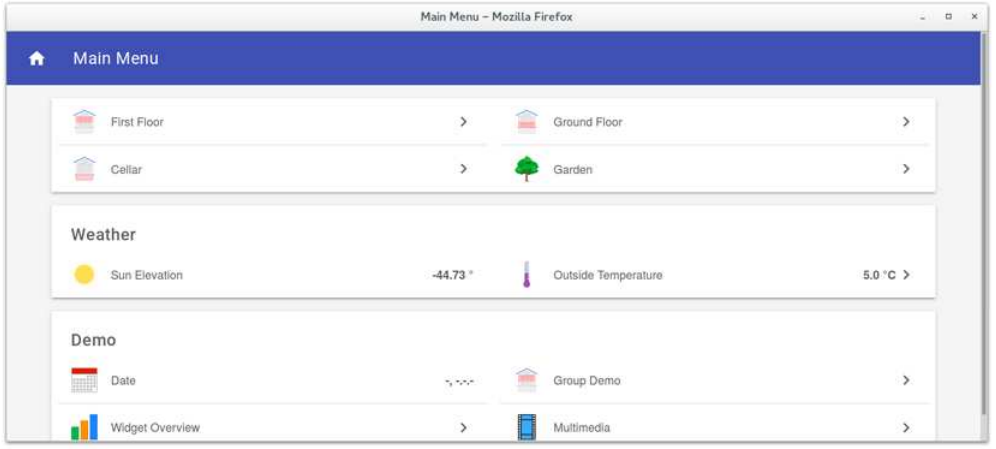
OpenHAB

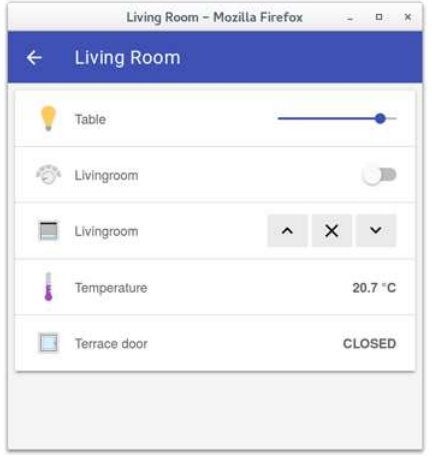
- OpenHAB czyli "The open **H**ome **A**utomation **B**us" to technologia o otwartym źródle niezalezna od rodzajów systemów automatyki domowej działajaca w centrum inteligentnego domu!
- OpenHAB 2 jest tworzony w jezyku Java.
- OpenHAB wykorzystuje OSGi dla modularności.
- OpenHAB używa serwera Apache Karaf jako pojemnika wraz z Eclipse Equinox jako środowiska wykonania OSGi. Jako serwer HTTP jest dołaczony Jetty.
- OpenHAB mozna zainstalowac na:
  - Rapberry Pi
  - Windows
  - Linux
  - Mac OS
  - Armbian
  - openHABian
  - Synology DiskStation
  - QNAP NAS

OpenHAB - UI

- The Paper UI
  - Add-on management,
  - Thing discovery,
  - Linking,
  - Items, sitemaps, presistence configurations and rules have to be defined in configuration files.
- The Basic UI
  - This interface is used to present the different sitemaps.
  - Web interface.
  - AJAX navigation,
  - Live update.
- The Classic UI
  - The Classic UI offers the same services that the Basic UI but with a different look and feel, close to an old iOS one.

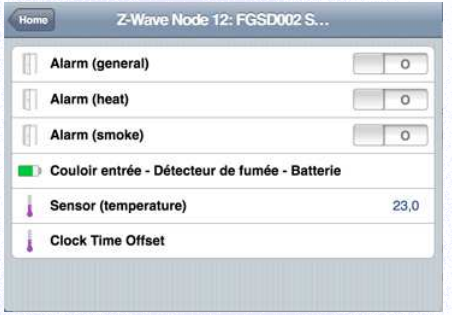
OpenHAB - Basic UI





https://www.openhab.org/docs/tutorial/uis.html

OpenHAB - Classic UI



https://www.openhab.org/docs/tutorial/uis.html

OpenHAB - Bindings

- Bindings integrate physical hardware, external systems and web services in openHAB. There are over 390 bindings.
- Popular bindings include:
  - Astro
  - Bluetooth
  - D-Link Smart Home
  - HUE
  - KNX
  - KODI
  - MQTT
  - nest
  - Network
  - Gardena
  - HTTP
  - Mi
  - RFXCOM
  - SONOS
  - IKEA
  - ZigBee
  - Z-Wave

OpenHAB - Things

- Things are the starting point for configuring physical entities in openHAB.
- Cofiguring the Thing:
  1. Identify the binding required for the Thing
  2. Install the binding if it has not already been installed
  3. Define and configure the Thing
  4. Identify the Channels provided by the Thing
  5. Add Items and link them to the Thing's Channels
  6. At this point Items can be used to control the Thing or consume its information in e.g. Sitemaps or Rules

OpenHAB - Items, sitemaps

- Items represent all properties and capabilities of the user's home automation.
- A collection of Things and Items represent physical or logical objects in the user's home automation setup.
- Sitemaps are used to select and prepare these elements in order to compose a user-oriented presentation of this setup for various UI's.
- Sitemap concepts:
  - Elements - Elements present information, allow interaction and are highly configurable based on the system state.
  - Parameters - A certain set of parameters can be configured to customize the presentation of an element.
  - Blocks - Multiple elements can be nested inside or behind others.
  - Dependencies - OpenHAB supports dependencies for system state and possible interactions by providing parameters for dynamic behavior.

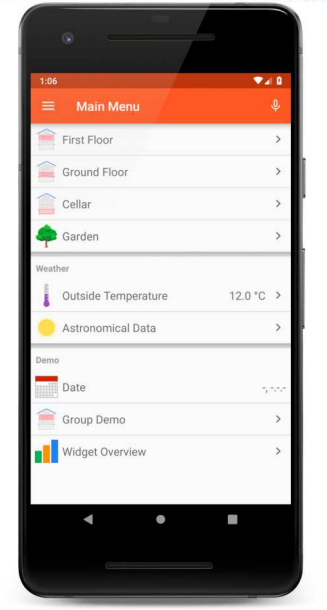
OpenHAB - Persistences, Rules

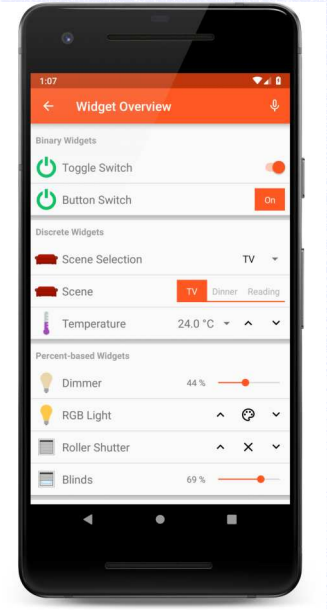
- Persistence is a data store used to keep data over the time.
- Rules are used for automating processes. Each rule can be triggered, which invokes a script that performs any kinds of tasks. physical or logical objects in the user's home automation setup.
- Transformations are used to translate data from a cluttered or technical raw value to a processed or human-readable representation.
- Actions are predefined methods that are called from openHAB rules and scripts. They can be used to execute openHAB-specific operations or to send commands or data to external hardware or services.

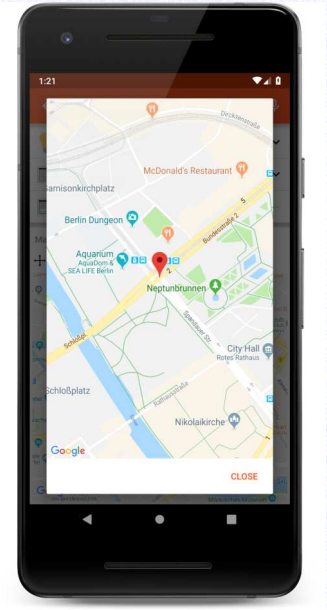
OpenHAB - Scripting

- Languages are known to work well for openHAB scripting:
  - JSR223 - standard scripting API for Java Virtual Machine,
  - Jython (Python on the JVM),
  - Nashorn Javascript (ECMAScript implementation included in JDK8 through 10, deprecated in 11),
  - Apache Groovy (JVM scripting language).

OpenHAB - Android App







https://www.openhab.org/docs/apps/android.html

Home Assistant

- Home Assistant to system automatyki domowej o otwartym źródle stawiający na sterowanie lokalne i prywatość.
- Jest rozwijany przez ogólnoswiatową społeczność programistów i majsterkowiczów.
- Zalecany sprzęt: Raspberry Pi 4 model B.
- Jest kofigurowany w języku YAML.
- Wykorzystuje bazę danych SQLite.
- Współpracuje z:
  - Amazon Alexa
  - ecobee
  - ESPHome
  - IFTTT
  - IKEA TRÅDFRI
  - MQTT
  - Philips Hue
  - Plax Media Server
  - SmartThings
  - SONOS
  - Z-Wave
  - ZigBee

Home Assistant - User Interface

https://www.home-assistant.io/lovelace/

Home Assistant - config

```
homeassistant:
  name: Home
  latitude: 52.15
  longitude: 20.98
  elevation: 106
  unit_system: metric
  time_zone: Europe/Warsaw
  external_url: "https://www.example.com"
  internal_url: "http://homeassistant.local:8123"
  allowlist_external_dirs:
    - /usr/var/dumping-ground
    - /tmp
  allowlist_external_urls:
    - "http://images.com/image1.png"

automation: !include automation.yaml
```

Home Assistant - device config

Two styles of configuration.

```
# Collect every entity under the \parent"
sensor:
  - platform: mqtt
    state_topic: "home/bedroom/temperature"
    name: "MQTT Sensor 1"
  - platform: mqtt
    state_topic: "home/kitchen/temperature"
    name: "MQTT Sensor 2"
  - platform: rest
    resource: http://IP_ADDRESS/ENDPOINT
    name: "Weather"

switch:
  - platform: vera
```

```
# List each device separately
sensor bedroom:
  platform: mqtt
  state_topic: "home/bedroom/temperature"
  name: "MQTT Sensor 1"

sensor kitchen:
  platform: mqtt
  state_topic: "home/kitchen/temperature"
  name: "MQTT Sensor 2"

sensor weather:
  platform: rest
  resource: http://IP_ADDRESS/ENDPOINT
  name: "Weather"

switch 1:
  platform: vera

switch 2:
  platform: vera
```

Home Assistant - rules

```
# Turns on lights 1 hour before sunset if people are home
# and if people get home between 16:00-23:00
  - alias: 'Rule 1 Light on in the evening'
    trigger:
      # Prefix the first line of each trigger configuration
      # with a '-' to enter multiple
      - platform: sun
        event: sunset
        offset: '-01:00:00'
      - platform: state
        entity_id: all
        to: 'home'
    condition:
      # Prefix the first line of each condition configuration
      # with a '-'' to enter multiple
      - condition: state
        entity_id: all
        state: 'home'
      - condition: time
        after: '16:00:00'
        before: '23:00:00'
    action:
      # With a single service call, we don't need a '-' before service - though you can if you want to
      service: homeassistant.turn_on
      entity_id: group.living_room
```

```
# Turn off lights when everybody leaves the house
  - alias: 'Rule 2 - Away Mode'
    trigger:
      platform: state
      entity_id: all
      to: 'not_home'
    action:
      service: light.turn_off
      entity_id: all
```

Home Assistant - triggers

```
automation:
  trigger:
    platform: numeric_state
    entity_id: sensor.temperature
    # Optional
    value_template: "{{ state.attributes.battery }}"
    # At least one of the following required
    above: 17
    below: 25

    # If given, will trigger when condition has been for X time,
    # can also use days and milliseconds.
    for:
      hours: 1
      minutes: 10
      seconds: 5
```

```
automation:
  trigger:
    platform: state
    entity_id: device_tracker.paulus, device_tracker.anne_therese
    # Optional
    from: "not_home"
    # Optional
    to: "home"

    # If given, will trigger when state has been the to state for X time.
    for: "01:10:05"
```

Home Assistant - actions

```
automation:
  # Change the light in the kitchen and living room
  # to 150 brightness and color red.
  trigger:
    platform: sun
    event: sunset
  action:
    service: light.turn_on
    data:
      brightness: 150
      rgb_color: [255, 0, 0]
      entity_id:
        - light.kitchen
        - light.living_room
```

```
automation 2:
  # Notify me on my mobile phone of an event
  trigger:
    platform: sun
    event: sunset
    offset: -00:30
  action:
    # Actions are scripts so can also be a list of actions
    - service: notify.notify
      data:
        message: Beautiful sunset!
    - delay: 0:35
    - service: notify.notify
      data:
        message: Oh wow you really missed something great.
```

Home Assistant - scenes

```
scene:
  - name: Romantic
    entities:
      light.tv_back_light: "on"
      light.ceiling:
        state: "on"
        xy_color: [0.33, 0.66]
        brightness: 200
  - name: Movies
    entities:
      light.tv_back_light:
        state: "on"
        brightness: 125
      light.ceiling: off
      media_player.sony_bravia_tv:
        state: "on"
        source: HDMI 1
        state: "on"
```

Inne platformy

- OpenMotics https://www.openmotics.com/
- Eventghost http://www.eventghost.net/
- ioBroker https://www.iobroker.net/
- AGO Control https://www.agocontrol.com/
- FHEM http://fhem.de/fhem.html
- Calaos https://calaos.fr/en/
- Pimatic https://pimatic.org/
- Homebridge https://homebridge.io/
- Smatrhomatic https://www.smarthomatic.org/
- Jeedom https://www.jeedom.com/site/en/index.html
- MyController https://www.mycontroller.org/#/home
- PiDome https://pidome.org/
- HomeGenie http://homegenie.it/
- *Samsung SmartThings* https://www.smartthings.com/
- *HomeSeer* https://homeseer.com/
- ...

Pozycje wypisane w kolorze pomarańczowym nie udostępniaja źródeł.

Powtórzenie

- Jakie elementy sa niezbędne w platformach automatyki domowej?
- Co to są elementy (Items), wyzwalacze (Triggers), akcje (Actions) i sceny (Scenes)?

Pytania?