

Home Assistant configuration

Well... this is a long story. We will start from some basic tricks and tips that make the configuration management easier and then we will proceed with some more advanced settings.

Home Assistant use [YAML syntax](#) for its configuration files and the most important file is the configuration.yaml file that is located in /home/homeassistant/.homeassistant/

You can access all the configuration files [via samba](#) from a Windows or Linux machine, or you can edit them using a text editor like nano with the [command line interface \(CLI\)](#). In this case make sure to switch to Home Assistant user to avoid permission issues.

× **Restart** Home Assistant via webui or [via CLI](#) to apply the modification that you have done in the configuration.yaml.

```
pi@d-diot:~ $ sudo -u homeassistant -H -s
homeassistant@d-diot:/home/pi $ nano
/home/homeassistant/.homeassistant/configuration.yaml
```

configuration.yaml Change the file name according to your needs.

Configuration splitting

As stated above the configuration.yaml is the most important configuration file of Home Assistant, but to keep it simple to read is a good idea to [split your configuration](#).

By default, but it may vary with the Home Assistant version, some parts of the configuration are already split. You should see something like that in the default configuration.yaml file:

```
group: !include groups.yaml
automation: !include automations.yaml
script: !include scripts.yaml
```

Edit them and add the following lines:

[/home/homeassistant/.homeassistant/configuration.yaml](#)

```
sensor: !include sensors.yaml
binary_sensor: !include binary_sensors.yaml
switch: !include switches.yaml
input_select: !include input_select.yaml
shell_command: !include shell_command.yaml
```

Please note that in the default configuration.yaml you should already have a “sensor” entry that should appear like that:

```
# Sensors
```

```
sensor:  
  # Weather prediction  
  - platform: yr
```

Simply delete this lines, we will cover this aspect [later](#).

Create the new configuration files:

```
pi@d-diot:~ $ sudo -u homeassistant -H -s  
homeassistant@d-diot:/home/pi $ touch  
/home/homeassistant/.homeassistant/sensors.yaml  
homeassistant@d-diot:/home/pi $ touch  
/home/homeassistant/.homeassistant/binary_sensors.yaml  
homeassistant@d-diot:/home/pi $ touch  
/home/homeassistant/.homeassistant/switches.yaml  
homeassistant@d-diot:/home/pi $ touch  
/home/homeassistant/.homeassistant/input_select.yaml  
homeassistant@d-diot:/home/pi $ touch  
/home/homeassistant/.homeassistant/shell_command.yaml
```

Panel iFrame

This configuration will add an entry for lirc_web and shellinabox in the right column of Home Assistant webui. The lirc_web and shellinabox web interfaces will be integrated in Home Assistant. Edit the file configuration.yaml and add the following lines:

</home/homeassistant/.homeassistant/configuration.yaml>

```
panel_iframe:  
  lircweb:  
    title: 'Ir remotes'  
    url: 'http://d-diot.local:3000'  
    icon: mdi:radiobox-marked  
  shellinabox:  
    title: 'Shell'  
    url: 'https://d-diot.local:4200'  
    icon: mdi:powershell
```

For d-diot image v.1.1 and above, with ESPHome installed, add the following lines:

</home/homeassistant/.homeassistant/configuration.yaml>

```
panel_iframe:  
  lircweb:  
    title: 'Ir remotes'  
    url: 'http://d-diot.local:3000'  
    icon: mdi:radiobox-marked
```

```
shellinabox:  
  title: 'Shell'  
  url: 'https://d-diot.local:4200'  
  icon: mdi:powershell  
esphome:  
  title: 'ESPHome'  
  url: 'http://d-diot.local:6052'  
  icon: mdi:chip
```

Substitute the d-diot.local part of the URL with your Raspberry Pi IP address if your network does not support avahi / zeroconf. See [this](#) if you have SSL encryption enabled.

Mosquitto MQTT sever

Reference guides:

- <https://www.home-assistant.io/components/mqtt/>
- <https://www.home-assistant.io/docs/mqtt/broker#run-your-own>
- <https://www.home-assistant.io/docs/mqtt/discovery/>
- https://www.home-assistant.io/docs/mqtt/birth_will/

Add the following lines to configuration.yaml:

</home/homeassistant/.homeassistant/configuration.yaml>

```
# MQTT  
mqtt:  
  broker: localhost  
  port: 1883  
#  username:  
#  password:  
  discovery: true  
  discovery_prefix: homeassistant  
  birth_message:  
    topic: 'hass/status'  
    payload: 'online'  
  will_message:  
    topic: 'hass/status'  
    payload: 'offline'
```

Insert your credentials and uncomment username and password if you have [enabled the authentication](#) in Mosquitto.

LIRC

To enable the [LIRC component](#) add the following lines to configuration.yaml:

[/home/homeassistant/.homeassistant/configuration.yaml](#)

```
# LIRC
lirc:
```

Create the .lircrc file in /home/homeassistant:

```
pi@d-diot:~ $ sudo -u homeassistant -H -s
homeassistant@d-diot:/home/pi $ touch /home/homeassistant/.lircrc
```

MySensors

Reference guide

- <https://www.home-assistant.io/components/mysensors/>

Add the following lines to configuration.yaml

[/home/homeassistant/.homeassistant/configuration.yaml](#)

```
# Mysensors
mysensors:
  gateways:
    - device: 'localhost'
      persistence_file: '/home/homeassistant/mysensors-rfm69.json'
      tcp_port: 5003
    - device: 'localhost'
      persistence_file: '/home/homeassistant/mysensors-nrf24.json'
      tcp_port: 5004
  optimistic: false
  persistence: true
  version: '2.3'
```

RFLink

Reference guide

- <https://www.home-assistant.io/components/rflink/>

Add the following lines to configuration.yaml

[/home/homeassistant/.homeassistant/configuration.yaml](#)

```
## RFLink
#rflink:
# port: /dev/serial0
# wait_for_ack: true
```

In the d-diot image the RFLink config section is commented out because you have to [install the RFLink firmware](#) to the ATmega2560 microcontroller first; once installed simply remove the comment symbol "#".

Themes

Home Assistant support different themes that can be changed directly in the webui. A [themes repository](#) has been created by the Home Assistant community.

Reference guide:

- <https://community.home-assistant.io/t/themes-yaml/98716>
- <https://www.home-assistant.io/components/frontend/#defining-themes>
- <https://community.home-assistant.io/c/projects/themes>

Create a dir 'themes' in /home/homeassistant/.homeassistant and download all themes:

```
pi@d-diot:~ $ sudo -u homeassistant -H -s
homeassistant@d-diot:/home/pi $ mkdir
/home/homeassistant/.homeassistant/themes
homeassistant@d-diot:/home/pi $ cd /home/homeassistant/.homeassistant/themes
homeassistant@d-diot:~/homeassistant/themes $ git clone
https://github.com/maartenpaauw/home-assistant-community-themes.git
```

In the configuration.yaml add the following line:

[/home/homeassistant/.homeassistant/configuration.yaml](#)

```
# Webui Themes
frontend:
  themes: !include_dir_merge_named themes
```

Raspberry sensors

Some parameters of your Raspberry Pi can be displayed in the Home Assistant webui; add the following lines to sensors.yaml.

[/home/homeassistant/.homeassistant/sensors.yaml](#)

```
# System monitor
- platform: systemmonitor
```

```
resources:
  - type: disk_use_percent
    arg: /
  - type: disk_use
    arg: /
  - type: disk_free
    arg: /
  - type: memory_free
  - type: memory_use_percent
  - type: memory_use
  - type: swap_use_percent
  - type: swap_use
  - type: swap_free
#   - type: load_1m
  - type: load_5m
#   - type: load_15m
  - type: network_in
    arg: eth0
  - type: network_out
    arg: eth0
  - type: network_in
    arg: wlan0
  - type: network_out
    arg: wlan0
#packets_in Interface, e.g., eth0
#packets_out Interface, e.g., eth0
  - type: ipv4_address
    arg: eth0
  - type: ipv4_address
    arg: wlan0
#ipv6_address Interface, e.g., eth0
  - type: processor_use
#   - type: process
#   arg: octave-cli
  - type: last_boot

#CPU temp
- platform: command_line
  name: CPU Temperature
  command: "cat /sys/class/thermal/thermal_zone0/temp"
  # If errors occur, remove degree symbol below
  unit_of_measurement: "°C"
  value_template: '{{ value | multiply(0.001) | round(1) }}'
```

In the d-diot image this sensors are grouped in a view accessible clicking the Raspberry Pi icon in the Home Assistant webui

Outside Weather

Reference guide:

- <https://www.home-assistant.io/components/sensor.yr/>

Set the location latitude and longitude in configuration.yaml under the “homeassistant” directive and add the following lines to sensors.yaml

</home/homeassistant/.homeassistant/sensors.yaml>

```
# Outside conditions
- platform: yr
  name: Outside
# forecast: 24
  monitored_conditions:
    - temperature
    - symbol
    - precipitation
    - windSpeed
    - pressure
    - windDirection
    - humidity
    - fog
    - cloudiness
    - lowClouds
    - mediumClouds
    - highClouds
    - dewpointTemperature
```

NMAP tracker

The nmap package has been installed in the [first step of the manual installation](#).

The reference guide to learn how this component works in Home Assistant is here:

- https://www.home-assistant.io/components/device_tracker/

NMAP will work independently from your router but it uses the Raspberry Pi resources, so if you have for example a Netgear router, use the [specific component](#).

To configure the nmap component, add the following line to configuration.yaml in the “device_tacker” section, that should be already present in the default configuration.yaml:

</home/homeassistant/.homeassistant/configuration.yaml>

```
#device_tracker:
```

```
# # NMAP
# - platform: nmap_tracker
# hosts: 192.168.1.0/24
# home_interval: 10
# exclude:
# - 192.168.1.20
```

In the web interface of Home Assistant you can associate a specific nmap device to a person.

× In the d-diot image the nmap platform is disabled because the configuration depends on the characteristic of the target network. Simply remove the comment symbol (#) from the configuration.yaml lines reported above and set your network parameters to enable this tracker.

Bluetooth tracker

Reference guides:

- <https://www.home-assistant.io/docs/installation/hassbian/integrations/>
- https://www.home-assistant.io/components/device_tracker/
- https://www.home-assistant.io/components/device_tracker.bluetooth_tracker/
- <https://community.home-assistant.io/t/error-in-installing-the-bluetooth-device-tracker/914>

The required deb packages have been already installed in the [basic system setup](#), but here we need to install the python library:

```
pi@d-diot:~ $ sudo pip install pybluez
```

Add the following lines to configuration.yaml in the “device_tacker” section, under the [NMAP platform](#):

</home/homeassistant/.homeassistant/configuration.yaml>

```
#device_tracker:
# # NMAP
# - platform: nmap_tracker
# hosts: 192.168.1.0/24
# home_interval: 10
# exclude:
# - 192.168.1.20
# # Bluetooth
# - platform: bluetooth_tracker
```

In the web interface of Home Assistant you can associate a specific bluetooth device to a person.

× In the d-diot image the bluetooth platform is disabled because it slows down the Home Assistant restart. If you want to activate this functionality, simply remove the “#” symbol before “- platform: bluetooth_tracker”.

MariaDB

In the basic system setup we have installed a [MariaDB server](#). Now it's time to configure Home Assistant to use it.

As a first step we need to install the required python library in the virtual environment

```
pi@d-diot:~ $ sudo -u homeassistant -H -s
homeassistant@d-diot:/home/pi $ source /srv/homeassistant/bin/activate
(homeassistant) homeassistant@d-diot:/home/pi $ pip3 install mysqlclient
(homeassistant) homeassistant@d-diot:/home/pi $ exit
```

Stop Home Assistant:

```
pi@d-diot:~ $ sudo systemctl stop home-assistant@homeassistant
```

Add the following lines under the "recorder" section that should be already present in the default configuration.yaml.

</home/homeassistant/.homeassistant/configuration.yaml>

```
recorder:
  purge_keep_days: 3
  purge_interval: 1
  db_url: !secret mariadb_connection_url
  include:
    entities:
      - sensor.processor_use
      - sensor.memory_use_percent
      - sensor.disk_use_percent
      - sensor.load_5m
      - sensor.cpu_temperature
```

In the file secrets.yaml add the following line

</home/homeassistant/.homeassistant/secrets.yaml>

```
mariadb_connection_url: mysql://homeassistant:d-
diot.com@localhost/homeassistant
```

Remove the default SQLAlchemy DB:

```
pi@d-diot:~ $ sudo -u homeassistant -H -s
homeassistant@d-diot:/home/pi $ rm /home/homeassistant/.homeassistant/home-
assistant_v2.db
homeassistant@d-diot:/home/pi $ exit
```

Restart home assistant and see what happens:

```
pi@d-diot:~ $ sudo systemctl start home-assistant@homeassistant
```

Now Home Assistant will use the MariaDB server to store the historical data. You can use any MariaDB server instance, even a remote one, simply changing the database URL in the file secrets.yaml.

× Only the data of the entities under the “include” statement will be recorded, so you have to add your own entities manually. You can change easily this behaviour: see the Home Assistant docs to know more about the [recorder component](#).

Oled display

Now we will see how to control the [oled display](#) with Home Assistant.

Add the following lines to shell_command.yaml.

[/home/homeassistant/.homeassistant/shell_command.yaml](#)

```
start_oled: /home/homeassistant/oled/./oled.py &
stop_oled: mosquitto_pub -h localhost -t oled/set -m 'Turn off' &&
sleep 5 && killall oled.py &
```

Add the following lines to input_select.yaml.

[/home/homeassistant/.homeassistant/input_select.yaml](#)

```
oled:
  name: Mode
  options:
    - Turn off
    - Wifi
    - Ethernet
    - Clock
    - Load
    - Disk usage
    - CPU Temp
    - RAM
  initial: Clock
  icon: mdi:fullscreen
```

Add the following lines to automations.yaml

[/home/homeassistant/.homeassistant/automations.yaml](#)

```
- id: '1551909333005'
```

```
alias: Oled - Input to MQTT
trigger:
- entity_id: input_select.oled
  platform: state
condition: []
action:
- data_template:
  payload: '{{ states(''input_select.oled'') }}'
  retain: false
  topic: oled/set
  service: mqtt.publish
- id: '1551910316293'
  alias: Oled - MQTT to Input
  trigger:
  - platform: mqtt
    topic: oled/state
  condition: []
  action:
  - data_template:
    entity_id: input_select.oled
    option: '{{ trigger.payload }}'
    service: input_select.select_option
- id: '1552122444469'
  alias: Oled-Autostart
  trigger:
  - event: start
    platform: homeassistant
  condition: []
  action:
  - service: shell_command.start_oled
- id: '1552128743168'
  alias: Oled-Autostop
  trigger:
  - event: shutdown
    platform: homeassistant
  condition: []
  action:
  - service: shell_command.stop_oled
```

Add the following lines to `binary_sensors.yaml`.

/home/homeassistant/.homeassistant/binary_sensors.yaml

```
#MQTT binary sensor
- platform: mqtt
  name: "OLED status"
  state_topic: "oled/status"
  payload_on: 1
  payload_off: 0
  qos: 0
```

```
device_class: light
```

Add the following lines to switches.yaml.

</home/homeassistant/.homeassistant/switches.yaml>

```
# Template switches
- platform: template
  switches:
    oled:
      friendly_name: "Status"
      value_template: "{{ is_state('binary_sensor.oled_status', 'on')
}}"
      turn_on:
        service: shell_command.start_oled
        data:
          entity_id: binary_sensor.oled_status
      turn_off:
        service: shell_command.stop_oled
        data:
          entity_id: binary_sensor.oled_status
```

Add the following lines to groups.yaml.

</home/homeassistant/.homeassistant/groups.yaml>

```
# Oled related automations
enable_oled:
  name: "Enable Oled"
  entities:
    - automation.oled_autostart
    - automation.oled_autostop
    - automation.oled_input_to_mqtt
    - automation.oled_mqtt_to_input
  all: true
```

Duck DNS

The duck DNS component is necessary (in most cases) when you want to make your Home Assistant instance [accessible from the outside](#) respect to your local network.

Reference guide:

- <https://www.home-assistant.io/components/duckdns/>

Add the following line to configuration.yaml

</home/homeassistant/.homeassistant/configuration.yaml>

```
#duckdns:  
# domain: your_duckdns_subdomain_without_.duckdns.org  
# access_token: !secret duckdns_token
```

Add the following lines to secrets.yaml

</home/homeassistant/.homeassistant/secrets.yaml>

```
#duckdns_token: your_token_here
```

× In the d-diot image this components is disabled, if you need them simply remove the comment symbol (#) in configuration.yaml and secrets.yaml and set your own subdomain and token. See [this guide](#) to configure the remote access to your Home Assistant using Duck DNS and ssl encryption.

SSL encryption

As for the [duck DNS](#) component, the SSL encryption is necessary when you want to make your Home Assistant [accessible from the outside](#) respect to your local network.

A valid ssl certificate is issued with [dehydrated](#), that is a lightweight bash script that uses [let's encrypt](#) with the DNS-01 challenge, supported by Duck DNS, to validate your domain name.

The advantages of this setup are:

- No need to open port 80 on your router to validate or renew your domain certificate.
- A full installation of let's encrypt is not required.
- The certificate issued by let's encrypt are recognized as valid by the most common browser.

Reference guide:

- https://www.splitbrain.org/blog/2017-08/10-homeassistant_duckdns_letsencrypt
- <https://indomus.it/guide/remotizzare-home-assistant-su-raspberry-pi-in-piena-sicurezza/>

Download and install dehydrated

```
pi@d-diot:~ $ sudo -u homeassistant -H -s  
homeassistant@d-diot:/home/pi $ cd /home/homeassistant  
homeassistant@d-diot:~ $ git clone  
https://github.com/lukas2511/dehydrated.git
```

Create a "config" file inside the dehydrated directory with the following lines:

```
homeassistant@d-diot:~ $ cd dehydrated  
homeassistant@d-diot:~/dehydrated $ nano config
```

[/home/homeassistant/.homeassistant/dehydrated/config](#)

```
# Which challenge should be used? Currently http-01 and dns-01 are supported
CHALLENGETYPE="dns-01"

# Script to execute the DNS challenge and run after cert generation
HOOK="{BASEDIR}/hook.sh"
```

Create the hook script file:

```
homeassistant@d-diot:~/dehydrated $ nano hook.sh
```

Add the following lines:

[/home/homeassistant/.homeassistant/dehydrated/hook.sh](#)

```
#!/usr/bin/env bash
set -e
set -u
set -o pipefail

domain="your_subdomain.duckdns.org"
token="your_token"

case "$1" in
    "deploy_challenge")
        curl
        "https://www.duckdns.org/update?domains=$domain&token=$token&txt=$4"
        echo
        ;;
    "clean_challenge")
        curl
        "https://www.duckdns.org/update?domains=$domain&token=$token&txt=remove
d&clear=true"
        echo
        ;;
    "deploy_cert")
        sudo systemctl restart home-assistant@homeassistant.service
        ;;
    "unchanged_cert")
        ;;
    "startup_hook")
        ;;
    "exit_hook")
        ;;
    *)
        echo Unknown hook "${1}"
        exit 0

```

```
;;  
esac
```

Make the hook.sh script executable:

```
homeassistant@d-diot:~/dehydrated $ chmod 755 hook.sh
```

Add the following line to configuration.yaml:

</home/homeassistant/.homeassistant/configuration.yaml>

```
# SSL encryption and DuckDNS access  
#http:  
# base_url: your_subdomain.duckdns.org:8123  
# ssl_certificate:  
/home/homeassistant/dehydrated/certs/your_subdomain.duckdns.org/fullchain.pem  
# ssl_key:  
/home/homeassistant/dehydrated/certs/your_subdomain.duckdns.org/privkey.pem
```

× In the d-diot image the ssl encryption is disabled. Follow [this guide](#) to make your Home Assistant instance accessible through an internet connection, using Duck DNS and ssl encryption.

Configuration files

d-diot image v.1.0

</home/homeassistant/.homeassistant/configuration.yaml>

```
homeassistant:  
  # Name of the location where Home Assistant is running  
  name: Home  
  # Location required to calculate the time the sun rises and sets  
  latitude: 46.0679  
  longitude: 11.1211  
  # Impacts weather/sunrise data (altitude above sea level in meters)  
  elevation: 0  
  # metric for Metric, imperial for Imperial  
  unit_system: metric  
  # Pick yours from here:  
  http://en.wikipedia.org/wiki/List_of_tz_database_time_zones  
  time_zone: Europe/Rome  
  # Customization file  
  customize: !include customize.yaml
```

```
# Configure a default setup of Home Assistant (frontend, api, etc)
default_config:

# Show the introduction message on startup.
introduction:

# Discover some devices automatically
discovery:

# Text to speech
tts:
  - platform: google

group: !include groups.yaml
automation: !include automations.yaml
script: !include scripts.yaml

##### d-diot settings
#####

# Configuration split
sensor: !include sensors.yaml
binary_sensor: !include binary_sensors.yaml
switch: !include switches.yaml
input_select: !include input_select.yaml
shell_command: !include shell_command.yaml

# iFrame Panels
panel_iframe:
  lircweb:
    title: 'Ir remotes'
    url: 'http://d-diot.local:3000'
    icon: mdi:radiobox-marked
  shellinabox:
    title: 'Shell'
    url: 'https://d-diot.local:4200'
    icon: mdi:powershell

# Mosquitto MQTT
mqtt:
  broker: localhost
  port: 1883
# username:
# password:
discovery: true
discovery_prefix: homeassistant
birth_message:
  topic: 'hass/status'
  payload: 'online'
```



```
will_message:
  topic: 'hass/status'
  payload: 'offline'

# LIRC
lirc:

# Mysensors
mysensors:
  gateways:
    - device: 'localhost'
      persistence_file: '/home/homeassistant/mysensors-rfm69.json'
      tcp_port: 5003
    - device: 'localhost'
      persistence_file: '/home/homeassistant/mysensors-nrf24.json'
      tcp_port: 5004
  optimistic: false
  persistence: true
  version: '2.3'

# RFLink
#rflink:
# port: /dev/serial0
# wait_for_ack: true

# Frontend
frontend:
  themes: !include_dir_merge_named themes

# Device Tracker
#device_tracker:
#   # NMAP
#   - platform: nmap_tracker
#     hosts: 192.168.1.0/24
#     home_interval: 10
#     exclude:
#       - 192.168.1.20
#   # Bluetooth
#   - platform: bluetooth_tracker

# Recorder
recorder:
  purge_keep_days: 3
  purge_interval: 1
  db_url: !secret mariadb_connection_url
  include:
    entities:
      - sensor.processor_use
      - sensor.memory_use_percent
      - sensor.disk_use_percent
      - sensor.load_5m
```

- sensor.cpu_temperature

```
# Duck DNS
#duckdns:
# domain: your_duckdns_subdomain_without_.duckdns.org
# access_token: !secret duckdns_token

# SSL encryption
#http:
# base_url: your_subdomain.duckdns.org:8123
# ssl_certificate:
/home/homeassistant/dehydrated/certs/your_subdomain.duckdns.org/fullchain.pem
# ssl_key:
/home/homeassistant/dehydrated/certs/your_subdomain.duckdns.org/privkey.pem
```

[/home/homeassistant/.homeassistant/sensors.yaml](#)

```
# System monitor
- platform: systemmonitor
  resources:
    - type: disk_use_percent
      arg: /
    - type: disk_use
      arg: /
    - type: disk_free
      arg: /
    - type: memory_free
    - type: memory_use_percent
    - type: memory_use
    - type: swap_use_percent
    - type: swap_use
    - type: swap_free
#   - type: load_1m
#   - type: load_5m
#   - type: load_15m
    - type: network_in
      arg: eth0
    - type: network_out
      arg: eth0
    - type: network_in
      arg: wlan0
    - type: network_out
      arg: wlan0
#packets_in Interface, e.g., eth0
#packets_out Interface, e.g., eth0
  - type: ipv4_address
```

```
    arg: eth0
  - type: ipv4_address
    arg: wlan0
#ipv6_address      Interface, e.g., eth0
  - type: processor_use
#   - type: process
#     arg: octave-cli
  - type: last_boot

#CPU temp
- platform: command_line
  name: CPU Temperature
  command: "cat /sys/class/thermal/thermal_zone0/temp"
  # If errors occur, remove degree symbol below
  unit_of_measurement: "°C"
  value_template: '{{ value | multiply(0.001) | round(1) }}'

# Outside conditions
- platform: yr
  name: Outside
# forecast: 24
  monitored_conditions:
    - temperature
    - symbol
    - precipitation
    - windSpeed
    - pressure
    - windDirection
    - humidity
    - fog
    - cloudiness
    - lowClouds
    - mediumClouds
    - highClouds
    - dewpointTemperature
```

[/home/homeassistant/.homeassistant/binary_sensors.yaml](#)

```
#MQTT binary sensor
- platform: mqtt
  name: "OLED status"
  state_topic: "oled/status"
  payload_on: 1
  payload_off: 0
  qos: 0
  device_class: light
```

[/home/homeassistant/.homeassistant/automations.yaml](#)

```
- id: '1551909333005'
  alias: Oled - Input to MQTT
  trigger:
  - entity_id: input_select.oled
    platform: state
  condition: []
  action:
  - data_template:
    payload: '{{ states(''input_select.oled'') }}'
    retain: false
    topic: oled/set
    service: mqtt.publish
- id: '1551910316293'
  alias: Oled - MQTT to Input
  trigger:
  - platform: mqtt
    topic: oled/state
  condition: []
  action:
  - data_template:
    entity_id: input_select.oled
    option: '{{ trigger.payload }}'
    service: input_select.select_option
- id: '1552122444469'
  alias: Oled-Autostart
  trigger:
  - event: start
    platform: homeassistant
  condition: []
  action:
  - service: shell_command.start_oled
- id: '1552128743168'
  alias: Oled-Autostop
  trigger:
  - event: shutdown
    platform: homeassistant
  condition: []
  action:
  - service: shell_command.stop_oled
```

[/home/homeassistant/.homeassistant/groups.yaml](#)

```
# Oled related automations
enable_oled:
```

```
name: "Enable Oled"
entities:
  - automation.oled_autostart
  - automation.oled_autostop
  - automation.oled_input_to_mqtt
  - automation.oled_mqtt_to_input
all: true
```

[/home/homeassistant/.homeassistant/input_select.yaml](#)

```
oled:
  name: Mode
  options:
    - Turn off
    - Wifi
    - Ethernet
    - Clock
    - Load
    - Disk usage
    - CPU Temp
    - RAM
  initial: Clock
  icon: mdi:fullscreen
```

[/home/homeassistant/.homeassistant/secrets.yaml](#)

```
# Use this file to store secrets like usernames and passwords.
# Learn more at https://home-assistant.io/docs/configuration/secrets/
some_password: welcome
mariadb_connection_url: mysql://homeassistant:d-
diot.com@localhost/homeassistant
#duckdns_token: your_token_here
```

[/home/homeassistant/.homeassistant/shell_command.yaml](#)

```
start_oled: /home/homeassistant/oled/./oled.py &
stop_oled: mosquitto_pub -h localhost -t oled/set -m 'Turn off' &&
sleep 5 && killall oled.py &
```

[/home/homeassistant/.homeassistant/switches.yaml](#)

```
# Template switches
- platform: template
  switches:
    oled:
      friendly_name: "Status"
      value_template: "{{ is_state('binary_sensor.oled_status', 'on')
  }}"
      turn_on:
        service: shell_command.start_oled
        data:
          entity_id: binary_sensor.oled_status
      turn_off:
        service: shell_command.stop_oled
        data:
          entity_id: binary_sensor.oled_status
```

d-diot image v.1.1

From:
<https://wiki.d-diot.com/> - d-diot wiki

Permanent link:
https://wiki.d-diot.com/system_administration/manual_installation/8_home_assistant_configuration?rev=1556733731

Last update: 2019/05/01 18:02

