

Návod na integraci Smarwi do systému Home Assistant

Návod na integraci WiFi otevírače oken do systému Home Assistant (HA) pomocí MQTT.

1. Nastavení komunikace mezi Smarwi a MQTT brokerem

Tento návod uvažuje použití lokálního MQTT brokeru, který zprostředkovává MQTT komunikaci mezi Smarwi otevíračem oken a HA v lokální síti. Nejsnazší způsob je instalace Mosquitto MQTT broker addonu přímo v HA (<https://bit.ly/3pyi4IU>). Pozor – toto funguje pouze v případě, že používáte Hassio, neboli připravenou instalaci operačního systému Linux s předinstalovaným HA, např. pro Raspberry Pi.

Pokud provozujete HA jako samostatnou aplikaci v rámci daného operačního systému, bude potřeba rozběhnout samostatnou instanci Mosquitto brokeru, a to ideálně na stejném zařízení, jako běží HA, např. pomocí technologie Docker (<https://bit.ly/3nyjECf>).

Alternativně by mělo být možné využít veřejný MQTT broker od Vektivy (broker.vektiva.com), který je přednastaven v konfigurátoru Smarwi, nicméně tato varianta nebyla autorem otestována.

Nastavení MQTT brokeru se provede v konfiguraci Smarwi sekci Settings/Advanced, kam se zadá IP adresa lokálního MQTT brokeru (v tomto případě 192.168.1.196):

< Settings

Okno ložnice

Basic **Advanced** Finetune Calibration Time Update Info

Broker server: 192.168.1.196

WiFi Off hours

0	1	2	3	4	5
6	7	8	9	10	11
12	13	14	15	16	17
18	19	20	21	22	23

Latitude (N/S) °: 50.088001

Longitude (E/W) °: 14.420000

WiFi PHY Mode: 802.11b

Save Apply

71

2. Konfigurace v systému Home Assistant

2.1 Konfigurace MQTT

V souboru `configuration.yaml` je nutné podobně jako v předchozím případě nakonfigurovat spojení na MQTT broker. HA má vlastního embedovaného MQTT klienta, který se s brokerem propojí a není tedy nutné instalovat žádný addon. To se provede takto:

```

mqtt:
  discovery: true
  discovery_prefix: homeassistant
  broker: 192.168.1.196
  birth_message:
    topic: 'hass/status'
    payload: 'online'
  will_message:
    topic: 'hass/status'
    payload: 'offline'

```

Alternativně lze místo IP adresy lokálního brokera zadat `broker.vektiva.com`.

2.2 Konfigurace Smarwi

Dále se v konfiguračním souboru HA nadefinuje binární MQTT sensor a MQTT switch, které jsou pomocí MQTT propojené s otevíračem Smarwi.

Binární sensor zaznamenává stav okna, tj. má hodnoty 'on' pokud je okno otevřeno, nebo 'off' pokud je zavřeno. Definice binárního sensoru v `configuration.yaml` vypadá takto:

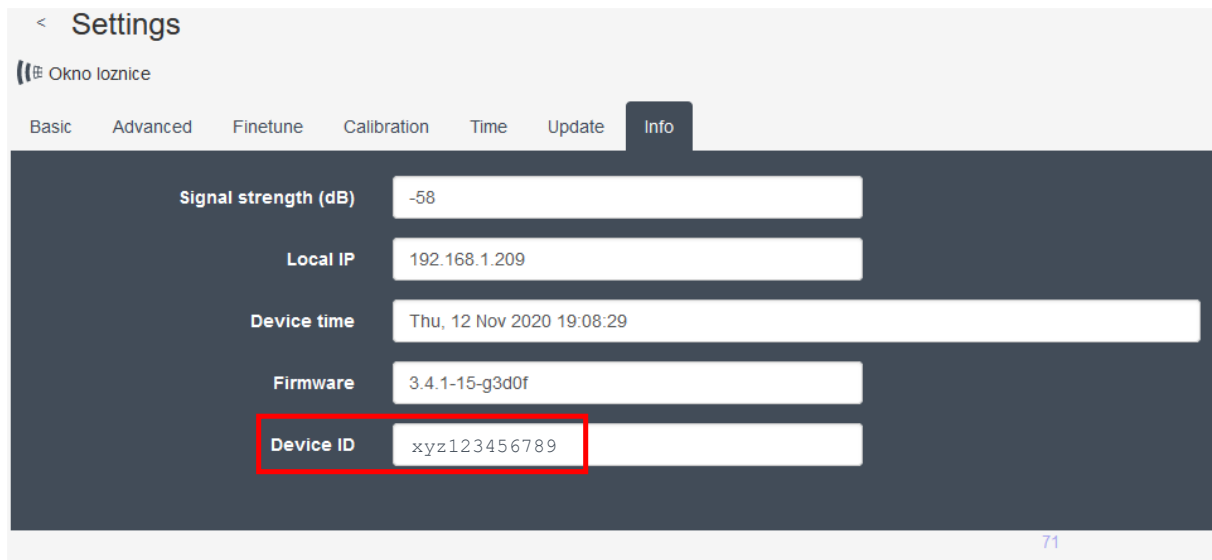
```

binary_sensor:
  - platform: mqtt
    name: okno loznice stav
    state_topic: "ion/remoteid/%deviceid/status"
    value_template: >-
      {% if 'pos:c' in value %}
        {{'OFF'}}
      {% else %}
        {{'ON'}}
      {% endif %}
    device_class: window

```

V parametru `state_topic` je potřeba změnit hodnoty `remoteid` a `deviceid` za ty, které jsou nastavené ve Smarwi jako **Remote ID** a **Device ID**. Tyto hodnoty lze buď získat v nastavení Smarwi v sekcích Basic resp. Info (viz screenshoty), nebo nalézt v sekci User account po kliknutí na šipku v pravém horním rohu na <https://vektiva.online>.

The screenshot shows the 'Settings' page for 'Okno loznice'. The 'Basic' tab is selected. The 'Remote ID' field is highlighted with a red box and contains the value 'abcdefghij'. Other fields include 'Device name' (Okno loznice), 'Select Wifi network', 'Wifi Password', 'Wifi Mode' (Client), and 'Remote KEY'. There are 'Save' and 'Apply' buttons at the bottom left.



V tomto konkrétním případě by řádka začínající `state_topic` vypadala takto:

```
state_topic: "ion/abcdefghij/%xyz123456789/status"
```

Pozn: znak `%` před `deviceid` tam musí zůstat.

Dále je potřeba nakonfigurovat přepínač, který ovládá otevírání a zavírání okna. Přepínač aktualizuje svůj stav na základě hodnoty sensoru z předchozího kroku, tj. podle stavu entity `'binary_sensor.okno_loznice_stav'`. Pokud se tedy okno otevře např. pomocí fyzického tlačítka Smarwi, přepne se i automaticky přepínač v HA.

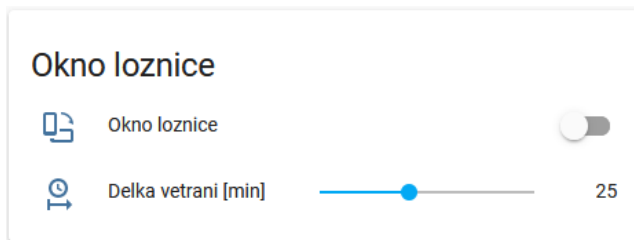
Konfigurace vypadá takto:

```
switch:
  - platform: template
    switches:
      smarwi_okno_loznice:
        friendly_name: "Okno loznice"
        value_template: "{{ is_state('binary_sensor.okno_loznice_stav', 'on') }}"
        turn_on:
          - service: mqtt.publish
            data:
              payload: 'open;100'
              topic: 'ion/remoteid/%deviceid/cmd'
        turn_off:
          - service: mqtt.publish
            data:
              payload: 'close;100'
              topic: 'ion/remoteid/%deviceid/cmd'
```

I zde je nutné změnit hodnoty `remoteid` a `deviceid` tak, jako v předchozím kroku. Hodnota 100 v části `payload` znamená otevření/zavření okna na maximum (100%). Její změnou tak lze snadno otevřít okno třeba jen na 50%.

2.3 Konfigurace uživatelského rozhraní HA

Přepínač pro vládání Smarwi je možné jednoduše přidat do uživatelského rozhraní HA. To pak může vypadat např. takto:



Pro konfiguraci je nutné vložit tento kód do souboru `ui-lovelace.yaml` (pokud nepoužíváte konfiguraci UI pomocí yaml souboru, bude jistě možné přidat přepínač definovaný v předchozím kroku přímo v editoru UI):

```
- type: entities
  title: Okno loznice
  entities:
    - entity: switch.smarwi_okno_loznice
      icon: 'mdi:phone-rotate-landscape'
    - entity: input_number.delka_vetrani
```

Parametr `delka_vetrani` určuje, jak na jak dlouho se má okno otevřít (viz dále). Tuto proměnnou je nutné zdefinovat v souboru `configuration.yaml` takto:

```
input_number:
  delka_vetrani:
    name: Delka vetrani [min]
    icon: mdi:clock-start
    initial: 30
    min: 0
    max: 60
    step: 5
```

Pak je možné vytvořit automatizaci, která např. každou celou hodinu okno otevře a za dobu, určenou předchozím parametrem, jej zase zavře a takto zajistit pravidelné větrání celou noc. Dobu otevření okna lze snadno měnit posuvníkem, např. podle venkovní teploty. Konfigurace automatizace v souboru `automation.yaml` vypadá takto:

```
- id: '1603834682347'
  alias: Vetrani loznice 23 00
  description: Vetrani loznice ve 23:00
  trigger:
    - at: '23:00:00'
      platform: time
  action:
    - service: switch.turn_on
      entity_id: switch.smarwi_okno_loznice
    - delay: 00:{{ states('input_number.delka_vetrani') | int }}:00
    - service: switch.turn_off
      entity_id: switch.smarwi_okno_loznice
```

Zkopírováním této automatizace je možné vytvořit další, spouštěné v jiném čase. Tj. v řádku `- at: '23:00:00'`, zaměnit např. za `- at: '00:00:00'`, `- at: '01:00:00'`, atd.

3. Ovládání Smarwi externím tlačítkem

Pro jednodušší otevírání okna lze využít některé z externích tlačítek integrovatelných do HA, které je možné umístit třeba vedle nočního stolku a ovládat okno pohodlně z postele. Autorovi tohoto návodu se osvědčila tlačítka IKEA z řady výrobků pro chytrou domácnost TRÅDFRI (<https://bit.ly/2IKdPCB>) a to především díky velmi nízké ceně.

Vzhledem k tomu, že tlačítko komunikuje přes Zigbee, je potřeba zprovoznit domácí Zigbee síť. K tomu je potřeba si pořídit Zigbee controller, např. Texas Instruments CC2531, ideálně s externí anténou pro zvýšení dosahu signálu (<https://bit.ly/3lDtp1t>) a ten zapojit do USB portu v zařízení, kde běží HA. Pozn: není potřeba kupovat Zigbee bránu od IKEA, komunikaci obstará tento controller, a to nejen s výrobky IKEA, ale s jakýmkoliv jiným Zigbee zařízením.

Dále je potřeba nainstalovat knihovnu Zigbee2MQTT, která překládá Zigbee zprávy do MQTT, které pak přeposílá do MQTT brokera z kapitoly 1 (v tomto případě již není možné použít broker od Vektivy). Opět je možné použít addon do Hassio, nebo bude nutné stejně jako v případě MQTT brokeru spouštět Zigbee2MQTT pomocí Dockeru. Návod na instalaci a nastavení naleznete zde: <https://www.zigbee2mqtt.io>

Pokud se vše zdárně podaří rozběhnout, lze začít s párováním tlačítka se Zigbee controllerem (<https://bit.ly/3pwZNW1>). Tlačítko je potřeba mít během párování velmi blízko antény controlleru (2-3 cm). Po spárování se vytvoří nový záznam v sekci `devices`: ve vlastním konfiguračním souboru `configuration.yaml` knihovny Zigbee2MQTT (pozor – nejde o konfigurační soubor HA). Ten pak může vypadat např. takto:

```
frontend:
  port: 8088
experimental:
  new_api: true
homeassistant: true
permit_join: true
mqtt:
  base_topic: zigbee2mqtt
  server: 'mqtt://localhost'
serial:
  port: /dev/ttyACM0
devices:
  '0x123456789abcdef':
    friendly_name: Vypinac1
```

Poslední 2 řádky jsou nově spárované tlačítko, pod `friendly_name` pak bude vidět jako zařízení v HA. Všechna připojená MQTT zařízení lze snadno zobrazit v sekci Nastavení/Integrace/MQTT/Zařízení.

Otevření okna na stisk tlačítka pak provede tato jednoduchá automatizace (soubor `automations.yaml`)“

```
- id: '000000000000003'
  alias: Tlacitko 1 ON
  trigger:
    - platform: mqtt
      topic: zigbee2mqtt/Vypinac1
  condition:
    - condition: template
      value_template: '{{ "on" == trigger.payload_json.click }}'
  action:
    - service: switch.turn_on
      entity_id: switch.smarwi_okno_loznice
```

Zvýrazněné "on" na řádce `value_template` znamená krátký stisk tlačítka 1. Podobně lze udělat automatizaci pro zavření okna, záměnou za řetězec "off", který bude reagovat na stisk tlačítka 0. a změnou spouštěné akce na předposlední řádce na `switch.turn_off`.

Pozn.: kromě krátkých stisků tlačítek `0` a `1` lze odchyťávat i dlouhé stisky a to pomocí "brightness_up" resp. "brightness_down" a pomocí stejného tlačítka ovládat kromě okna třeba ještě lampičku.

Pozn.: někdy se může stát, že párování dalších tlačítek nefunguje. V tom případě je potřeba zastavit HA i službu Zigbee2MQTT, vyjmout Zigbee controller z USB portu, znovu jej zasunout a znovu nastartovat HA a Zigbee2MQTT.

Pokud je potřeba rozšířit dosahu signálu MQTT, osvědčil se autorovi tento opakovač signálu od IKEA (z nějakého důvodu teď není na českých stránkách ikea.cz): <https://bit.ly/32PpF5D>.

Dále lze velmi snadno zaintegrovat pohybové čidlo od IKEA: <https://bit.ly/38PafSl>.

Autor tuto kombinaci používá celkem kuriózním způsobem 😊 Autorův kocour, který je každou noc venku, pravidelně brzy ráno vyskočí na parapet okna v ložnici a mňoukáním se dožaduje pozornosti, čímž spolehlivě všechny probudí. Tato jednoduchá automatizace zavře okno pomocí Smarwi, jakmile je zaznamenán pohyb kocoura venku u okna mezi 22:00 večer a 7:00 ráno (další celou hodinu při pravidelném větrání se okno zase otevře):

```
id: '1603826727882'  
alias: Zavřít okno loznice pohyb  
description: ''  
trigger:  
- entity_id: binary_sensor.pohybove_cidlo  
  platform: state  
  from: 'off'  
  to: 'on'  
condition:  
  condition: time  
  after: '22:00:00'  
  before: 07:00:00  
action:  
- service: switch.turn_off  
  entity_id: switch.smarwi_okno_loznice
```