**Sterbox integracja z home assistant - wprowadzenie**

Integracja pomiędzy sterbox, a home assistant odbywa się za pomocą restfull commands:

<https://www.home-assistant.io/integrations/rest_command/>

Sterbox ma dostępną opcję sterowania poprzez requesty http – szczegóły u producenta. Dzięki temu, za pomocą restfull command możemy zmieniać stan oraz odczytywać status wyjść/wejść logicznych i analogowych. Pozwala to na pełne sterowanie sterboxem – także głosowe (np HA AI Speaker brama deweloperska z wbudowanym asystentem głosowym działającym offline).

<https://allegro.pl/oferta/pierwszy-polski-inteligentny-asystent-ai-speaker-8269043586>

Większość konfiguracji musimy umieścić w pliku HA configuration.yaml

Przykład czujnika bazującego na wyjściu analogowym (jeśli posiadasz hasło, składnia wymaga dodatkowych parametrów):

sensor:

 - platform: rest

 name: Predkosc wiatru

 unit\_of\_measurement: "km/h"

 resource: http://192.168.1.182/@gca?wind

 payload: '{ "device" : "heater" }'

 value\_template: "{{ value | replace(',', '.') }}"

 scan\_interval: 1

Powyższy przykład to odczytywanie wartości z czujnika wiatru z sygnałem 0-3V, odpowiednie przeliczniki zostały ustawione już na sterbox.

Dodatkowo warto dać filtr aby ustabilizować wykresy:

- platform: filter

 name: Filtr predkosci wiatru

 entity\_id: sensor.predkosc\_wiatru

 filters:

 - filter: outlier

 window\_size: 30

 radius: 4.0

 - filter: lowpass

 time\_constant: 60

 precision: 2

Wykres:





entities:

 - entity: sensor.filtr\_predkosci\_wiatru

 name: Prędkość wiatru

 - entity: sensor.filtr\_predkosci\_wiatru\_w\_porywach

 name: Prędkość wiatru w porywach

hours\_to\_show: 1

refresh\_interval: 0

title: Prędkość wiatru

type: history-graph

Dla wyjść logicznych warto użyć binary sensor:

binary\_sensor:

 - platform: rest

 name: Status elektrozaczepu drzwi wejsciowych

 resource: http://192.168.1.182/@gcd?drzwi

 scan\_interval: 1

Graficzny obiekt:



entities:

 - entity: binary\_sensor.status\_elektrozaczepu\_drzwi\_wejsciowych

show\_header\_toggle: false

title: Elektrozaczep

type: entities

Zmianę stanu możemy realizować poprzez przełączenie na stan przeciwny (2), 0 lub 1.

W tym celu musimy zdefiniować commendy w pliku:

Np:

rest\_command:

 ustawianie\_wilgotnosci: [w tym przypadku zadajemy wartość analogową]

 url: "http://192.168.1.182/@sca?wilg\_1={{ states('input\_number.wilgotnosc\_ustawiona\_input') }}"

 otwieranie\_drzwi:

 url: 'http://192.168.1.182/@scd?dwczyt3=2'

 refleksol\_salon\_poludnie\_down:

 url: 'http://192.168.1.180/@scd?dsrpdow2=2'

 refleksol\_salon\_poludnie\_up:

 url: 'http://192.168.1.180/@scd?dsrpup2=2'

 refleksol\_salon\_zachod\_down:

 url: 'http://192.168.1.180/@scd?dsrzdow2=2'

 refleksol\_salon\_zachod\_up:

 url: 'http://192.168.1.180/@scd?dsrzup2=2'

 refleksol\_jadalnia\_poludnie\_down:

 url: 'http://192.168.1.180/@scd?djrpdow2=2'

 refleksol\_jadalnia\_poludnie\_up:

 url: 'http://192.168.1.180/@scd?djrpup2=2'

 refleksol\_jadalnia\_wschod\_down:

 url: 'http://192.168.1.180/@scd?djrwdow2=2'

 refleksol\_jadalnia\_wschod\_up:

 url: 'http://192.168.1.180/@scd?djrwup2=2'

Warto od razu dodać script dla rest command. Rest command nie w każdym wypadku zadziała bezpośrednio:

 refleksol\_salon\_poludnie\_down\_script:

 sequence:

 service: rest\_command.refleksol\_salon\_poludnie\_down

 refleksol\_salon\_poludnie\_up\_script:

 sequence:

 service: rest\_command.refleksol\_salon\_poludnie\_up

 refleksol\_salon\_zachod\_down\_script:

 sequence:

 service: rest\_command.refleksol\_salon\_zachod\_down

 refleksol\_salon\_zachod\_up\_script:

 sequence:

 service: rest\_command.refleksol\_salon\_zachod\_up

 refleksol\_jadalnia\_poludnie\_down\_script:

 sequence:

 service: rest\_command.refleksol\_jadalnia\_poludnie\_down

Przykładowy przycisk:



Przy integracji pojawia się sporo niuansów – szczególnie, gdy daną czynnością steruje także inne urządzenie wejściowe typu klawisz.

Dlatego do zmiany stanów najlepiej używać buforów. Pozwala to na sterowanie z kilku wejść tą samą zmienną w sterbox.

Na ten moment nie udało mi się uruchomić restcommand switcha, ale do sterowania wraz ze zmianą statusu można użyć np elementów warunkowych.

Definiujemy binary sensor do sprawdzenia statusu np światła. Potem definiujemy 2 przyciski z warunkowym wyświetlaniem, czyli wyświetlaj przycisk włącz światło jeśli sensor jest off i odwrotnie.

Dzięki temu status w HA zmieni się automatycznie po zmianie i nie będziemy zapalali światła już zapalonego. Dodatkowo można sobie zrobić listę z czujnikami wyświetlającymi interesujące nas wyjścia.

Powyższe informacje powinny wystarczyć do startu przygody z HA. Cała pozostała zabawa, to stricte konfiguracja automatyzacji na poziomie HA, które ma bardzo dobrą dokumentację i społeczności.

W przypadku AI Speaker jest także grupa dyskusyjna dotycząca bramki.

Samo HA daje ogromne możliwości integracji w jednym miejscu sterboxa oraz wielu innych urządzeń – sonoff, ESP home, xiaomi i wiele, wiele innych – typu fotowoltaika, modbus itp.

Przykładowy panel sterowania oparty o integrację ze sterbox:

