

# Integracja z protokołem ZigBee

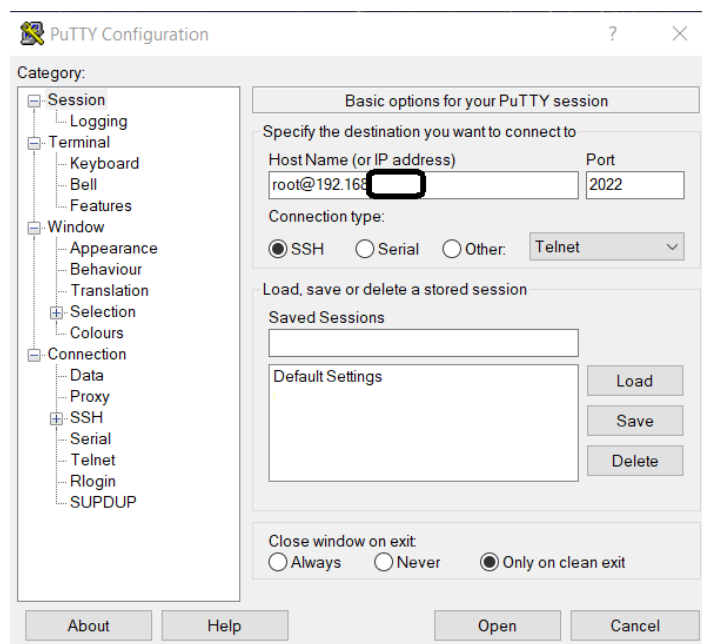
Numer dokumentu: PO-197    Wersja: 2.1    Data publikacji: 17 czerwca 2024

## Wstęp

Integracja urządzeń wspierających protokół ZigBee z systemem Ampio możliwa jest na przykład dzięki dopięciu bramki do modułu M-SERV-s. W celu wykonania połączenia niezbędne jest użycie platformy Node-RED. W poniższym przykładzie jako bramka służy moduł ZBDongle-E firmy Sonoff z dedykowaną anteną.

## Podłączenie bramki

Aby podłączyć bramkę, odłączamy M-SERV od zasilania, wpinamy moduł w dowolne złącze USB i zasilamy ponownie serwer. Po kilku minutach, poprzez interfejs www aktywujemy połączenie SSH (wskazówki dostępne w poradniku [Konfiguracja serwera](#)). Logujemy się przy użyciu utworzonego hasła na konto *root* na serwerze np. poprzez aplikację *putty*.



## Wyszukanie portu z bramką

Po wpisaniu hasła wyszukujemy urządzenia z użyciem komendy:

```
dmesg | grep tty
```

Mostek prawdopodobnie zostanie dodany jako *ttyACM0*.

## Konfiguracja dla obrazów serwera od wersji numer 400

### Instalacja

Przechodzimy do folderu, w którym możemy wprowadzać zmiany:

```
cd /root
```

Klonowanie repozytorium zigbee2mqtt:

```
wget https://github.com/Koenkk/zigbee2mqtt/archive/refs/heads/master.zip
```

Rozpakowanie zigbee2mqtt:

```
unzip master.zip
```

Zmiana nazwy folderu:

```
mv /root/zigbee2mqtt-master/ /root/zigbee2mqtt/
```

Zmiana aktualnego folderu

```
cd /root/zigbee2mqtt
```

Instalowanie zawartości:

```
npm ci
```

## Modyfikacja pliku konfiguracyjnego

Otwieramy plik do edycji

```
nano /root/zigbee2mqtt/data/configuration.yaml
```

Pole *server* ustawiamy na *mqtt://localhost*.

Pole *port* zgodnie z tym co wyszukane zostało powyżej np. */dev/ttyACM0*.

Pola połączenia MQTT zgodnie z naszymi ustawieniami serwera, *user* to najczęściej *admin* oraz odpowiednie hasło (dane logowania jak do blozków *mqtt* w Node-RED). Należy pamiętać, aby usunąć znak *#* oznaczający komentarz linii.

```
GNU nano 3.2
# Home Assistant integration (MQTT discovery)
homeassistant: false

# allow new devices to join
permit_join: true

# MQTT settings
mqtt:
  # MQTT base topic for zigbee2mqtt MQTT messages
  base_topic: zigbee2mqtt
  # MQTT server URL
  server: 'mqtt://localhost'
  # MQTT server authentication, uncomment if required:
  user: admin
  password: my_password

# Serial settings
serial:
  # Location of CC2531 USB sniffer
  port: /dev/ttyACM0
```

Po zmianie zapisujemy i zamykamy plik konfiguracyjny. W putty robimy to poprzez *Ctrl+x*, następnie *y* i *Enter*.

## Pierwsze uruchomienie

Wpisujemy komendę

```
npm start
```

## Konfiguracja dla starszych obrazów

### Tworzenie i konfigurowanie folderu

Tworzymy folder

```
sudo mkdir /ampio/rw/zigbee2mqtt
```

Nadajemy uprawnienia

```
sudo chown -R ${USER}: /ampio/rw/zigbee2mqtt
```

### Klonowanie repozytorium zigbee2mqtt

```
git clone --depth 1 https://github.com/Koenkk/zigbee2mqtt.git /ampio/rw/zigbee2mqtt
```

### Instalowanie zawartości

Zmieniamy aktualny folder

```
cd /ampio/rw/zigbee2mqtt
```

Następnie instalujemy

```
npm ci
```

### Modyfikacja pliku konfiguracyjnego

Otwieramy plik do edycji

```
nano /ampio/rw/zigbee2mqtt/data/configuration.yaml
```

Pole `server` ustawiamy na `mqtt://localhost`.

Pole `port` zgodnie z tym co wyszukane zostało powyżej np. `/dev/ttyACM0`.

Pola połączenia MQTT zgodnie z naszymi ustawieniami serwera, `user` to najczęściej `admin` oraz odpowiednie hasło (dane logowania jak do blozków `mqtt` w Node-RED).

```
GNU nano 3.2
# Home Assistant integration (MQTT discovery)
homeassistant: false

# allow new devices to join
permit_join: true

# MQTT settings
mqtt:
  # MQTT base topic for zigbee2mqtt MQTT messages
  base_topic: zigbee2mqtt
  # MQTT server URL
  server: 'mqtt://localhost'
  # MQTT server authentication, uncomment if required:
  user: admin
  password: my_password

# Serial settings
serial:
  # Location of CC2531 USB sniffer
  port: /dev/ttyACM0
```

Po zmianie zapisujemy i zamykamy plik konfiguracyjny. W putty robimy to poprzez *Ctrl+x*, następnie *y* i *Enter*.

## Pierwsze uruchomienie

Wpisujemy komendę

```
npm start
```

## Dodanie urządzenia podrzędnego

Urządzenie dodawane w poradniku to czujnik temperatury i wilgotności SNZB-02 firmy Sonoff. Instrukcje dla urządzeń dostępne są na stronach odpowiednich producentów. W przypadku opisywanego czujnika, w celu jego dodania należy przytrzymać przycisk na obudowie przez 5 sekund. Po poprawnym dodaniu urządzenia w terminalu powinno pokazać się kilka komunikatów w tym np. taki:

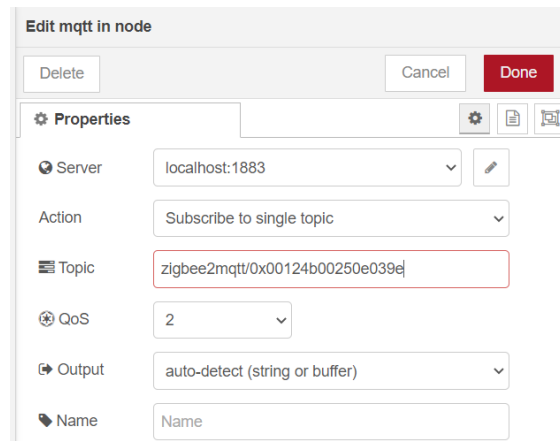
```
zigbee2MQTT:info 2023-02-28 08:58:06: Successfully interviewed '0x00124b00250e039e', device has successfully been paired
```

Po dodaniu czujnika, można poprzez platformę Node-RED reagować na dane lub z użyciem bloczków (np. *mqtt out*) wysłać dane do magistrali CAN.

**UWAGA!** Aby zachować bezpieczeństwo, po dodaniu wszystkich urządzeń należy zatrzymać proces np. *Ctrl+c*, ponownie wejść w plik *configuration.yaml* i ustawić *permit\_join:false*. Potem uruchamiamy proces ponownie.

## Konfiguracja w Node-RED

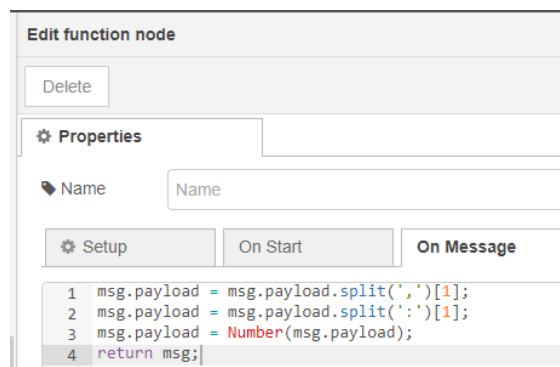
Poradnik opisujący podstawy Node-RED w systemie Ampio dostępny jest pod adresem: [Integracja systemu Ampio z Node-RED](#). Po dodaniu urządzeń podrzędnych można już nasłuchiwać danych z brokera MQTT Ampio. Topic na którym urządzenie nadaje można podejrzeć w terminalu poprzez połączenie SSH. W tym przykładzie jest to topic główny i ID dodanego urządzenia: *zigbee2mqtt/0x00124b00250e039e*.



Dane można podejrzeć po dodaniu bloczka *debug*.

```
28.02.2023, 09:05:19 node: 3604926b8f956599
zigbee2mqtt/0x00124b00250e039e : msg.payload :
string[85]
"
{"battery":100,"humidity":33.31,"link
quality":220,"temperature":22.97,"vol
tage":3300}"
```

W celu np. odczytania liczbowo wilgotności z tego czujnika przepuszczamy informację przez bloczek *function* z zawartością:



Różne urządzenia końcowe mogą nadawać informację w różnych postaciach dlatego warto podejrzeć dane w oknie *debug* przed napisaniem funkcji przesyłającej informację.

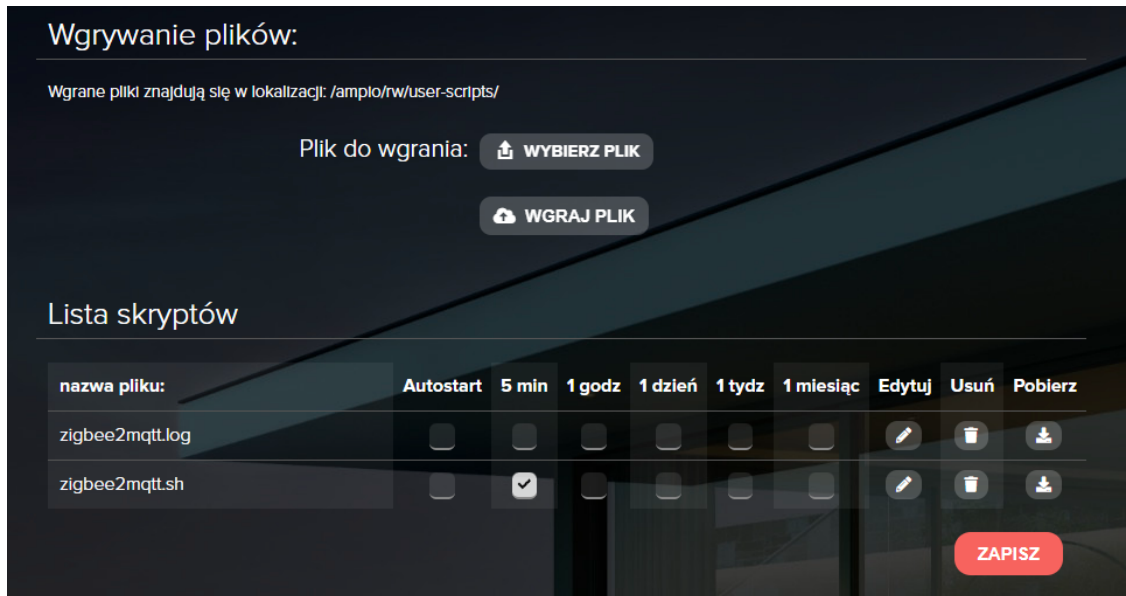
## Automatyczne uruchamianie dla obrazów serwera od wersji numer 400

Logujemy się ponownie poprzez SSH, będąc w głównym folderze root pobieramy i uruchamiamy skrypt komendą:

```
curl https://dist.ampio.pl/scripts/zigbee2mqtt400.sh | bash -s
```

## dla starszych obrazów

Aby aplikacja została uruchomiona automatycznie po restarcie zasilania, należy dodać odpowiedni skrypt. W interfejsie www modułu M-SERV wchodzimy w zakładkę **SYSTEM** a następnie **SKRYPTY**. Pobieramy załączony plik o nazwie `zigbee2mqtt.sh`. Wgrywamy go poprzez **WYBIERZ PLIK**, następnie **WGRAJ PLIK**. W kolejnym kroku zaznaczamy przy skrypcie pole `5 min` i naciskamy **ZAPISZ**.



## Test działania

W celu sprawdzenia poprawności konfiguracji resetujemy zasilanie serwera i po kilku minutach sprawdzamy działanie poprzez Node-RED np. w oknie *debug*.

Plik do pobrania:

- [zigbee2mqtt.sh](#)