



Integracja nadajnika radiowego Elmes TX-RS2 z systemami inteligentnego domu: „Home Assistant” i „OpenHab”

Nadajnik TX-RS2 można podłączyć do minikomputera *Raspberry*® lub dowolnego komputera z wyjściem szeregowym (ang. „serial port”), a także do urządzeń, które nie posiadają takiego interfejsu, ale mają złącze USB. W tym drugim przypadku połączenie wykonuje się za pomocą specjalnego adaptera, który zamienia transmisję USB na transmisję portu szeregowego (ang.: „USB-serial adapter”). Takie połączenie pozwala na integrację urządzeń odbiorczych „Elmes Elektronik” z systemami inteligentnego domu, typu np. „Home Assistant” lub „OpenHab”.

Dla uproszczenia procedurę instalacji opiszemy na przykładzie minikomputera *Raspberry*. Instalacja na innym komputerze może się minimalnie różnić.

Instalacja systemu.

Opis samej instalacji systemu inteligentnego domu: „Home Assistant” lub „OpenHab” wykracza poza zakres tego dokumentu. Można tu tylko wspomnieć, że są dostępne obrazy obu tych systemów przygotowane na *Raspberry*. Wystarczy więc tylko nagrać obraz na kartę SD w czytniku kart podłączanym do PC-ta, a następnie włożyć kartę do *Raspberry* i podłączyć zasilanie. System zainstaluje się automatycznie.

Można również najpierw zainstalować na *Raspberry* system operacyjny „Raspberry Pi OS”, dawniej „Raspbian”, a potem doinstalować oprogramowanie inteligentnego domu.

Podłączenie nadajnika TX-RS2 do minikomputera *Raspberry*.

Nadajnik TX-RS2 jest wyposażony w 3-żyłowy przewód, a każda żyła jest zakończona osobnym pojedynczym złączem pasującym do pojedynczego pinu złącza kołkowego znajdującego się na płycie *Raspberry*. Połączenie wykonujemy przy wyłączonym zasilaniu. Właściwe podłączenie jest bardzo ważne, aby uniknąć uszkodzenia nadajnika TX-RS2 lub samego komputera *Raspberry*:

| TX-RS2 nazwa sygnału – kolor przewodu | Raspberry nazwa sygnału - numer na złączu |
|--|--|
| GND - czarny | GND - 6 |
| 3,3V - czerwony | 3,3V - 1 |
| RX - niebieski | TXD - 8 |

Konfiguracja systemu „Home Assistant”

Konfiguracja wymaga edycji plików w systemie Linux, który jest zainstalowany na *Raspberry*. Musimy więc mieć dostęp do terminala. Domyślna instalacja „Home Assistant” go nie zawiera, ale można go łatwo doinstalować jako jeden z wielu dostępnych dodatków („ADD-ONS”). W tym celu otwieramy w przeglądarce interfejs graficzny „Home Assistant” poprzez wpisanie: <http://homeassistant:8123>, a następnie klikamy [Supervisor](#) -> [Add-on store](#) -> [Terminal & SSH](#). Wówczas dostęp do terminala będzie z poziomu interfejsu graficznego „Home Assistant” - zakładka [Terminal](#).

Do edycji i tworzenia plików możemy użyć edytora tekstu wbudowany w system: [nano](#).

Krok 1

W katalogu zawierającym pliki konfiguracyjne „Home Assistant” - w typowej instalacji będzie to katalog [/config](#) – tworzymy plik o nazwie: [wyslij_komende.sh](#). W tym celu wpisujemy w terminalu:

```
nano /config/wyslij_komende.sh
```

i w pliku tym wpisujemy dwie linijki:

```
#!/bin/bash  
printf "$1" >> /dev/ttyAMA0
```

zapisujemy plik i wracamy do terminala. Teraz musimy zmienić atrybuty tego pliku na „wykonywalny”:

```
chmod 775 /config/wyslij_komende.sh
```

Teraz już możemy przetestować działanie nadajnika TX-RS2 wpisując w terminalu np:

```
/config/wyslij_komende.sh "##030100"
```

co spowoduje wysłanie do portu szeregowego *Raspberry* o nazwie: [/dev/ttyAMA0](#) ciągu znaków: [##030100](#) (więcej o porcie szeregowym w uwagach na końcu tekstu). Po odebraniu tego polecenia, na-

dajnik TX-RS2 wyśle transmisję do odbiornika odpowiadającą naciśnięciu pierwszego przycisku (numeracja przycisków: 00..08) w kanale nr 01 (numeracja: 00..1D). Transmisja będzie trwała 1 s i towarzyszyć jej będzie pulsowanie diody LED.

Krok 2

Do pliku `/config/scripts.yaml` dopisujemy:

```
wyslij_komende_1 # nazwa skryptu
mode: queued # zbyt szybko napływające polecenia czekają w kolejce
sequence:
- service: shell_command.wyslij_komende2 # wywołaj polecenie wyslij_komende_2 z pliku configuration.yaml
  data_template:
    komenda_2: "{{ komenda }}" # i jako parametr: komenda_2, przekaz mu parametr: komenda
# przypisany do przycisku z kroku 4
```

Uwaga! Tekstu na prawo od znaku # i samego znaku # nie wpisujemy – to komentarz.

Krok 3

Do pliku `/config/configuration.yaml` dopisujemy:

```
shell_command:
  wyslij_komende_2: '/config/wyslij_komende.sh {{ komenda_2 }}
```

Krok 4

Na końcu możemy już utworzyć własną stronę z przyciskami. Przykładowo, dla sterowania trzema role-
tami, mogłaby ona wyglądać tak:



Najprościej jest utworzyć ją korzystając z graficznego interfejsu „Home Assistant”. Opis, jak to zrobić, wykracza poza zakres tego poradnika i został pominięty. Następnie w trybie edycji kodu należy dopisać kilka linijek do definicji każdego przycisku. Poniżej przedstawiono definicję jednego przycisku: ze strzałką w górę:

```
- type: buton
  hold_action:
    action: more-info
  icon: 'hass:arrow-up-bold' # przycisk będzie wyświetlany w postaci grubej strzałki w górę
  show_icon: true
  show_name: false
  show_state: false
  tap_action
  action: call-service # następne linijki określają akcję wywoływaną po naciśnięciu przycisku
  service: script.wyslij_komende_1 # niech tą akcją będzie
  service_data:
    komenda: '##030102' # wywołanie skryptu o nazwie: wyslij_komende_1 z pliku scripts.yaml
# i przekazanie mu parametru komenda równego ##030102
```

Wyjaśnijmy teraz sekwencję zdarzeń, które nastąpią po naciśnięciu przycisku:

- 1) Zostanie wywołany skrypt o nazwie `wyslij_komende_1` z pliku `scripts.yaml` z parametrem `komenda` równym `##030102`.
- 2) Skrypt `wyslij_komende_1` wywoła polecenie `wyslij_komende_2` znajdującą się w pliku `configuration.yaml` z parametrem: `komenda_2` równym `##030102`, który otrzymał jako parametr gdy sam był wywoływany.
- 3) Polecenie `wyslij_komende_2` wywoła skrypt linuxowy `wyslij_komende.sh` przekazując mu jako parametr ciąg: `##030102`, który otrzymał jako parametr gdy sam był wywoływany.
- 4) Skrypt linuxowy `wyslij_komende.sh` wywoła polecenie: `printf`, które wyśle ten ciąg do portu szeregowego.

Konfiguracja systemu „OpenHab”

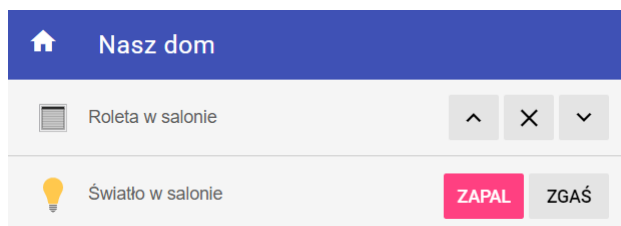
W tym systemie, podobnie jak w „Home Assistant”, konfiguracja wymaga edycji plików w systemie Linux, który jest zainstalowany na Raspberry. W tym celu musimy mieć dostęp do terminala. Możemy skorzystać z popularnego *Putty*, łącząc się z adresem *openhab* protokołem SSH na domyślnym porcie 22. Ale zanim to zrobimy, musimy wykonać krok 1.

Krok 1

Po zainstalowaniu systemu „OpenHab”, aby pozwolić na komunikację na porcie szeregowym, należy doinstalować tzw. *Serial Binding*. W tym celu otwieramy w przeglądarce interfejs graficzny „OpenHab” poprzez wpisanie: <http://openhab:8080>, a następnie klikamy na *PAPER UI* -> *Add-ons* -> *BINDINGS*. Pojawi nam się lista kilkudziesięciu dostępnych *Bindings*. Wybieramy *Serial Binding* i klikamy instaluj.

Krok 2

Teraz stworzymy dwie kontrolki, jedna służąca do sterowania roletą, a druga - światłem, jak na rysunku poniżej:



Z poziomu terminala przechodzimy do katalogu, który zawiera pliki konfiguracyjne „OpenHab”. Domyślnie będzie to: */etc/openhab2*. Edytujemy plik */items/default.items* – w tym celu wpisujemy w terminalu:

```
nano items/default.items
```

i w pliku tym wpisujemy definicję rolety (w jednej linii):

```
Rollershutter Roleta_Salon „Roleta w salonie” { serial="/dev/ttyAMA0, UP(##030100),  
DOWN(##030101), STOP(##030102) }
```

co spowoduje, że po naciśnięciu przycisku „GÓRA”, na port szeregowy */dev/ttyAMA0* zostanie wysłany ciąg znaków *##030100*, po naciśnięciu „DÓŁ” - ciąg *##030101*, a po naciśnięciu „STOP” - ciąg *##030102*. Więcej o porcie szeregowym w uwagach na końcu tekstu.

Następnie definiujemy wyłącznik światła:

```
Switch Swiatlo_Salon „Światło w salonie” { serial="/dev/ttyAMA0, ON(##030200), OFF(##030201) }
```

co oznacza, że po naciśnięciu przycisku „ON”, na port szeregowy */dev/ttyAMA0* zostanie wysłany ciąg znaków *##030200*, a po naciśnięciu przycisku „OFF” - ciąg znaków *##030201*.

Krok 3

Żeby zdefiniowane w kroku 2 elementy pojawiły się na ekranie, należy jeszcze dopisać kilka linijek do pliku: *sitemaps/default.sitemap*, który powinien wyglądać tak:

```
sitemp default label="Nasz dom"  
{  
  Switch item=Roleta_Salon  
  Switch item=Swiatlo_Salon mappings=[ON="Zapal", OFF="Zgaś"]  
}
```

Krótkie wyjaśnienie zapisu: *mappings=[ON="Zapal", OFF="Zgaś"]*. Powoduje on, że zamiast domyślnego, przesuwanego wyłącznika, pojawią się dwa przyciski oznaczone *"Zapal"*, i *"Zgaś"*

Uwagi

Port szeregowy.

Występujący w tekście powyżej ciąg */dev/ttyAMA0* to nazwa pierwszego portu szeregowego w komputerze Raspberry. Nazwa portu szeregowego może być inna niż */dev/ttyAMA0*. Np. może to być: */dev/ttyS0*, a w przypadku, kiedy nadajnik TX-RS podłączamy do portu USB za pomocą kabla-adaptera USB-serial, może to być np. */dev/ttyUSB0*. Listę wszystkich dostępnych portów można sprawdzić poleceniem:

```
ls /dev
```

Porty szeregowy w komputerze Raspberry są domyślnie zablokowane. Żeby odblokować pierwszy port szeregowy na pinach 8 –TX i 10 – RX złącza GPIO, należy dopisać do pliku */boot/config.txt* dwie linijki:

```
enable_uart=1
dtoverlay=miniuart-bt
```

Czwarta wersja komputera Raspberry Pi posiada, oprócz standardowego portu na pinach 8 i 10, 4 dodatkowe: *ttyAMA1..ttyAMA4*. Aby włączyć wszystkie 4, w */boot/config.txt* należy dopisać:

```
dtoverlay=uart2           # otwiera port /dev/ttyAMA1, z TXD2 na pinie 27 złącza GPIO
dtoverlay=uart3           # otwiera port /dev/ttyAMA2, z TXD3 na pinie 7 złącza GPIO
dtoverlay=uart4           # otwiera port /dev/ttyAMA3, z TXD4 na pinie 24 złącza GPIO
dtoverlay=uart5           # otwiera port /dev/ttyAMA4, z TXD5 na pinie 32 złącza GPIO
```

Uwaga. W domyślnej instalacji systemu „Home Assistant” dostęp do pliku */boot/config.txt* może być utrudniony – nie jest on dostępny z poziomu terminala wywoływanego z interfejsu graficznego „Home Assistant”. W takim przypadku, edycję tego pliku można wykonać na PC, podłączając czytnik kart SD z włożoną kartą z zainstalowanym systemem „Home Assistant”.