

Wydział Elektryczny
Zespół Automatyki (ZTMAiPC)

LABORATORIUM MODELOWANIA I SYMULACJI

Ćwiczenie 7

Modelowanie układów sterowanych zdarzeniami

1. Cel ćwiczenia

Celem ćwiczenia jest poznanie podstaw modelowania układów sterowanych zdarzeniami i posługiwania się modulem Stateflow środowiska Matlab.

2. Wprowadzenie

2.1 Układy sterowane zdarzeniami i maszyna o skończonej liczbie stanów (FSM)

Układy sterowane zdarzeniami (układy reaktywne, układy o przełączanym stanie) stanowią ważną klasę układów modelowanych w wielu zagadnieniach technicznych. W układzie tego typu następuje przechodzenie (*tranzycja*) od jednego do innego określonego stanu po wystąpieniu *zdarzenia* i spełnieniu *warunku* definiującego takie przejście. Układ sterowany zdarzeniami jest reprezentowany za pomocą modelu nazywanego *maszyną o skończonej liczbie stanów* (*Finite State Machine* – FSM). Jako przykład modelu FSM można podać opis układu przeniesienia napędu w samochodzie z automatyczną skrzynią biegów. Układ taki opisuje się za pomocą pewnej liczby stanów pracy w zależności od położenia dźwigni zmiany biegu (neutralny, drive, wsteczny itd.). Zmiana stanu następuje w wyniku zmiany położenia dźwigni, czyli zdarzenia.

2.2 Reprezentacja FSM w module Stateflow

Tradycyjne do opisu i projektowania układów sterowanych zdarzeniami wykorzystywano tablice prawdy reprezentujące zależności logiczne między wejściami, wyjściami i stanami FSM. Innym sposobem jest modelowanie zachowania się układu poprzez opisanie stanów i warunków przejść od jednego do innego. Stan aktualnie aktywny jest określany na podstawie wystąpienia określonych zdarzeń i spełnienia określonych warunków. Graficzną reprezentacją układu opartą na takim podejściu jest diagram przejść między stanami (*State-transition diagram* – STD). Analogiczny sposób zapisu algorytmu, tzw. *Sequential Flow Chart* (SFC), stosuje się w programowaniu sterowników mikroprocesorowych

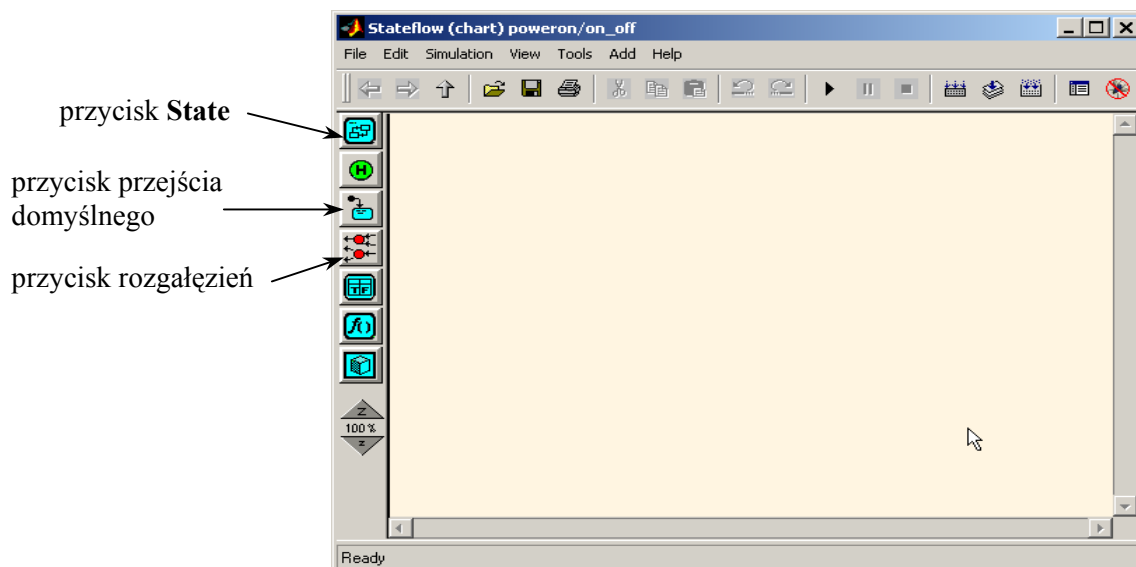
Stateflow (SF) jest specjalizowanym modulem Matlaba przeznaczonym do modelowania i symulacji FSM. Przy tworzeniu diagramu SF stosuje się wariant notacji opisu FSM opracowany przez Harela i Davida („Statecharts: A Visual Formalism for Complex Systems”, *Science of Computer Programming* 8, 1987). Graficzne bloki stanów (*states*) i strzałki przejść (*transitions*) stanowią podstawowe cegiełki diagramu Stateflow (maszyny Stateflow).

Bloki SF włącza się do schematu blokowego Simulinka, gdzie współpracują ze zwykłymi blokami Simulinka. Spełnienie przez sygnały wchodzące do bloku SF z Simulinka określonych warunków generuje zdarzenia. Sygnały wyjściowe z bloku SF mogą aktywizować lub blokować symulację części diagramu Simulinka, czyli zmieniać strukturę modelowanego układu. Pod każdym blokiem SF na schemacie Simulinka kryje się cały, często złożony, diagram Stateflow. Przy uruchamianiu symulacji SF generuje odpowiedni dla danego diagramu kod maszyny FSM.

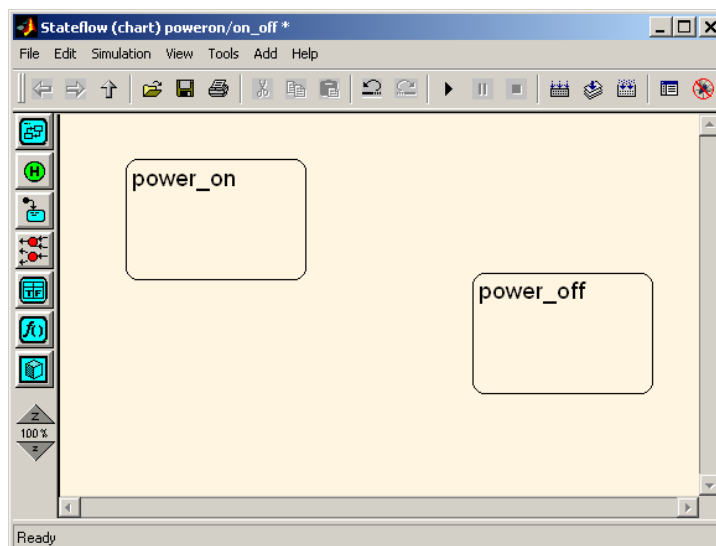
3. Program ćwiczenia

3.1. Edycja pierwszego diagramu SF

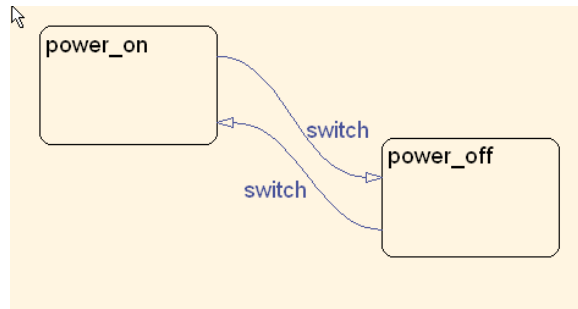
- a) Otworzyć bibliotekę Stateflow poleceniem `stateflow` z linii komend Matlab. Stateflow otwiera nowy model simlinka z pojedynczym blokiem SF `untitled`. Zmienić nazwę bloku na `on_off` edytując podpis i zapisać model poleceniem `Save as` (wybrać własną nazwę).
- b) Wywołać graficzny edytor SF klikając dwukrotnie blok `on_off`.



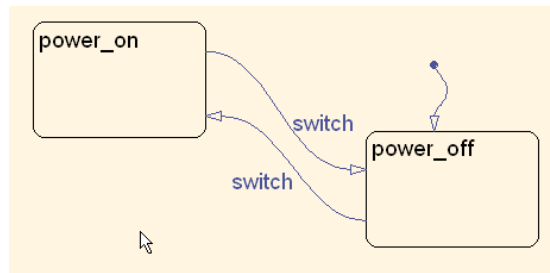
- c) Kliknąć przycisk **State** i przeciągnąć blok do obszaru diagramu SF. Powtórzyć operację i umieścić w polu edycji drugi stan. Kliknąć bloki stanów na znak „?” i nadać im nazwy jak poniżej.




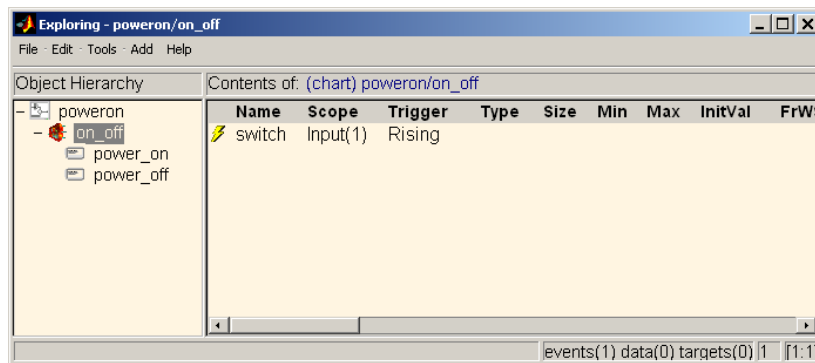
- d) Utworzyć przejścia między stanami. Umieścić kursor na boku stanu `power_on`, a kiedy zamieni się na krzyżyk przeciągnąć strzałkę do boku stanu `power_off` trzymając wciśnięty lewy przycisk myszy. Analogicznie utworzyć przejście od stanu `power_off` do `power_on`. Klikając na strzałki przejść nadać im nazwy `switch` tak jak w przypadku stanów.



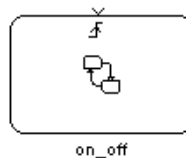
- e) Dodać przejście domyślne do stanu `power_off` klikając przycisk przejścia domyślnego i dodając strzałkę na diagramie. Przejście domyślne wskazuje stan domyślny układu uaktywniany spośród stanów wykluczających się, np. przy inicjalizacji układu.



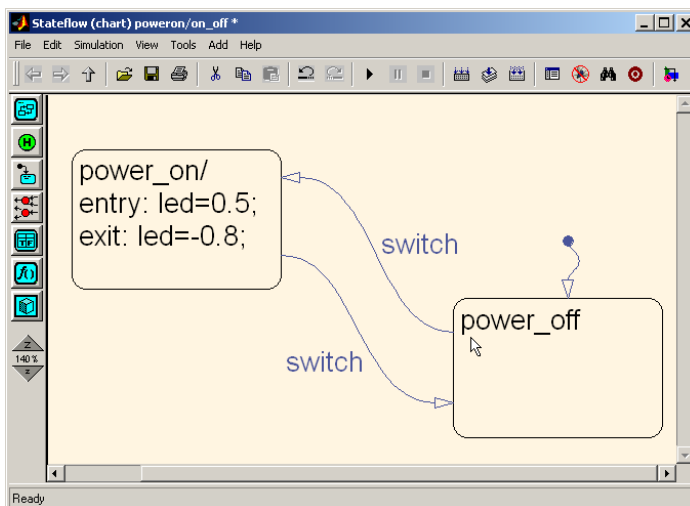
- f) Definiowanie zdarzeń wejściowych. Z menu edytora SF wybrać **Tools | Explore** i na liście hierarchii obiektów modelu Simulinka zaznaczyć blok SF. Z menu Explorer wybrać **Add | Event** (dodanie zdarzenia), a następnie zmodyfikować nazwę i właściwości zdarzenia klikając dwukrotnie znaczek  lub odpowiednie pole zdarzenia (nazwa: `switch`, scope: `Input from Simulink`, trigger: `Rising`). Zdarzenie `switch` będzie zachodzić, kiedy zbocze narastające sygnału z Simulinka dołączonego do wejścia `Input(1)` bloku SF `on_off` przekroczy zero.



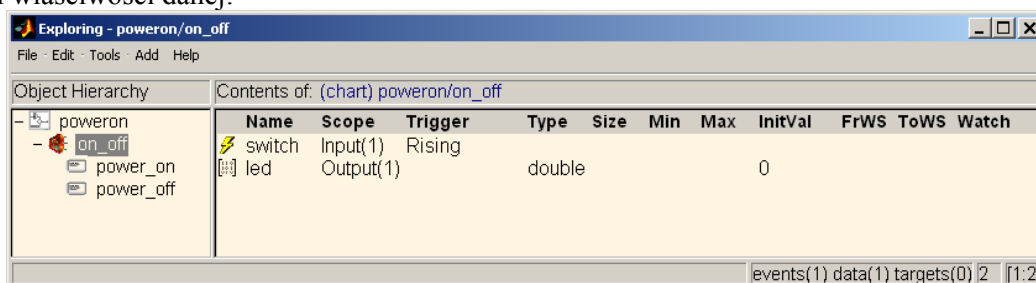
Po dodaniu wejścia z Simulinka na bloku SF pojawi się znaczek portu tego wejścia i sposobu wyzwalania.



- g) Określenie działań (*action*) stanu. Załóżmy, że przy uaktywnieniu stanu `power_on` zmienna `led` ma przyjmować wartość `+0.5`, przy wyjściu z tego stanu wartość `-0.8`. Aby to osiągnąć w edytorze Stateflow należy kliknąć w obszarze stanu i dopisać działanie przy wejściu (`entry`) i wyjściu (`exit`):

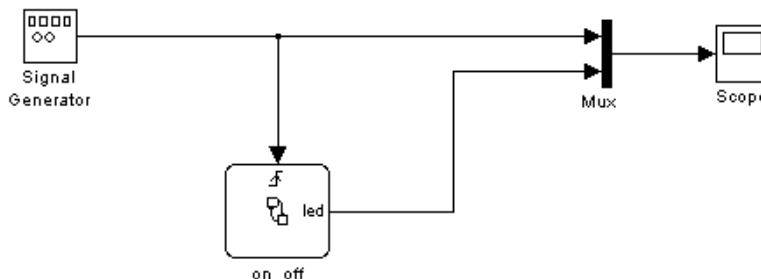


Zmienną led należy dodać jako daną bloku SF dostępną dla pozostałych bloków Simulinka. W tym celu z menu edytora SF wybrać **Tools | Explorer**, a następnie **Add | Data** i zmodyfikować nazwę i właściwości danej:



Odpowiedni port wyjściowy pojawi się na bloku SF diagramu Simulinka.

- h) Połączenie bloku SF z innymi blokami diagramu Simulinka. Rozbudować diagram Simulinka jak poniżej:

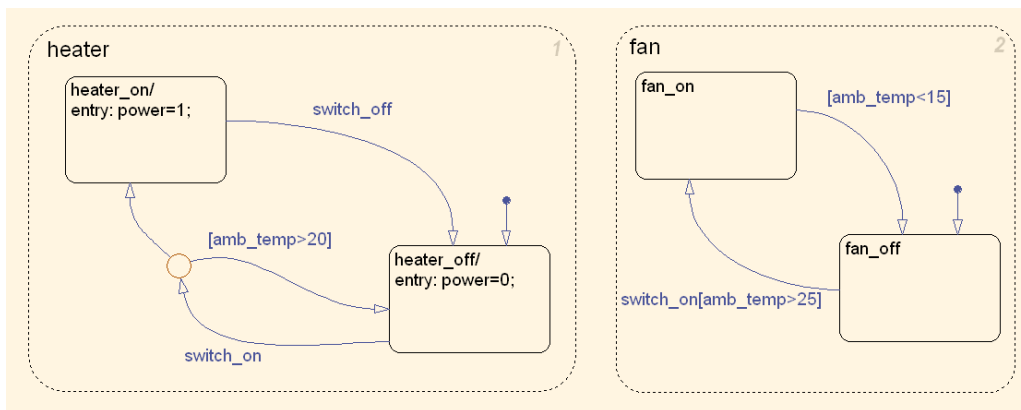


W generatorze sygnału wybrać sinusoidę o częstotliwości 0.1 Hz. W **Simulation | Parameters** ustawić czas symulacji Stop time = 100, stałokrokową metodę całkowania (Solver options - Fixed Step, metoda ode4, Fixed step size=0.01)

- i) Uruchomić symulację mając widoczne okno edytora SF i bloku Scope. Zaobserwować przejścia między stanami bloku SF i powiązać je z przebiegami bloku Scope.

3.2. Przykład: model SF działania „ręcznego” klimatyzatora

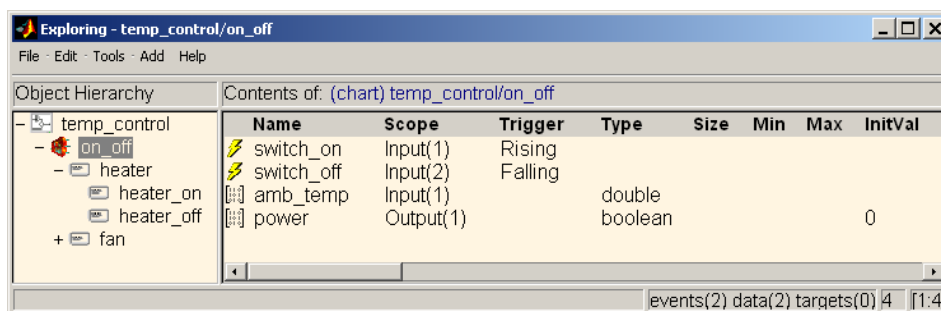
- a) Otworzyć nowy model Simulinka z blokiem SF i zapisać go pod własną nazwą. Otworzyć edytor diagramu SF i wygenerować diagram jak na rysunku poniżej:



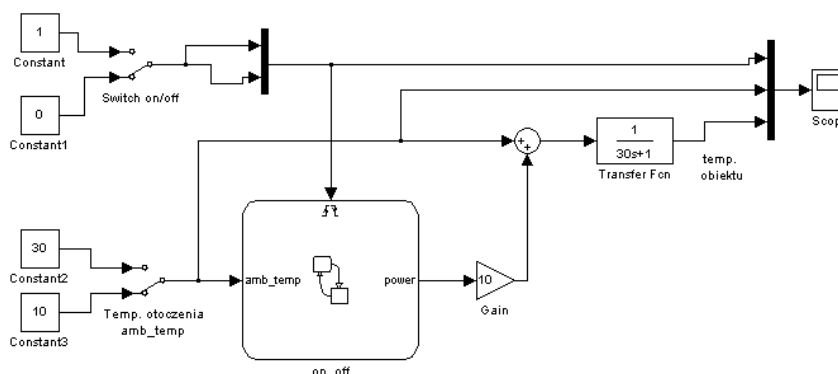
Przerywane ramki oznaczają stany, które mogą być aktywne jednocześnie (**Parallel AND**). Ramki ciągle oznaczają stany wzajemnie wykluczające się (**Exclusive OR**). Przełączanie pomiędzy tymi możliwościami uzyskuje się w edytorze SF kliknięciem prawego klawisza myszy w obszarze diagramu i wybraniem odpowiedniej opcji **Decomposition**. Stany heater (grzejnik) i fan (wentylator) są równoległe. Odpowiednie podstany on/off w ich obrębie są stanami wykluczającymi się.

Tranzycje opisuje się ogólnie w formie event[condition] (zdarzenie[warunek]). *Jeżeli zdarzenie nie jest podane, to tranzycja jest realizowana po wystąpieniu dowolnego zdarzenia.* Rozgałęzienie tranzycji (junction) pozwala na zapisanie warunku if ... elseif ... else. Tranzycja bez warunku (bez opisu) jest traktowana jako else i realizowana na końcu.

W eksploratorze edytora SF (menu **Tools | Explore**) należy zdefiniować następujące zdarzenia i dane:



b) Wygenerować część schematu Simulinka poza diagramem SF. Model powinien wyglądać jak na poniższym rysunku:

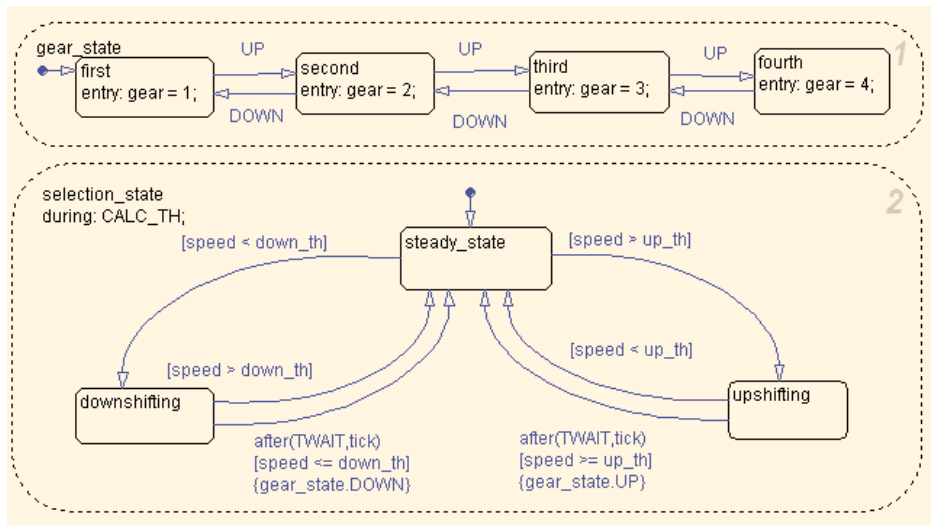
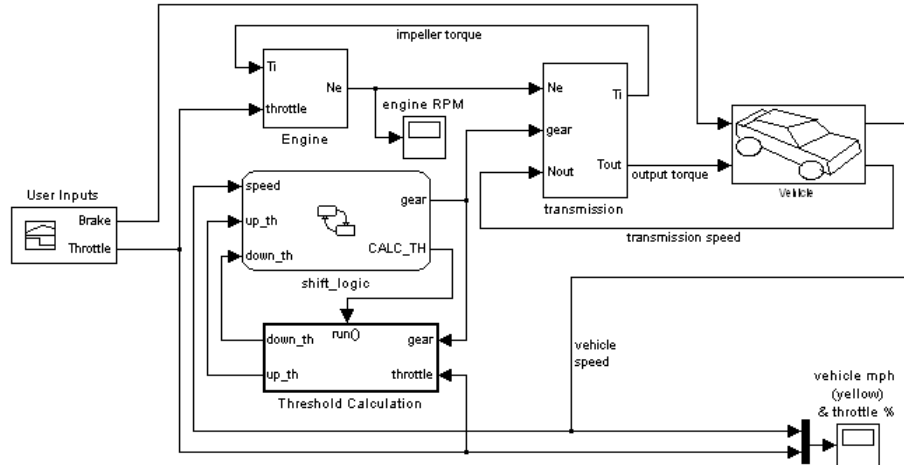


Zwrócić uwagę na konieczność wprowadzenia do bloku SF on_off wektora dwóch jednakowych zmultipleksowanych sygnałów, ponieważ w bloku SF zdefiniowane są dwa zdarzenia: switch_on i switch_off.

c) W **Simulation | Parameters** ustawić czas symulacji **Stop time = 1000**, stałokrokową metodę całkowania (**Solver options - Fixed Step**, metoda **ode4**, **Fixed step size=0.001**). Uruchomić symulację mając widoczne okno edytora bloku SF i bloku Scope. Zaobserwować działanie układu klikając przełącznikami i zmieniając ich kombinacje. Zwrócić uwagę, że zmiany temperatury otoczenia są uwzględniane dopiero po zmianie stanu przełącznika Switch, ponieważ tylko wtedy generowane są zdarzenia maszyny SF.

3.3. Opracować i/lub przeanalizować model zadany przez prowadzącego

Przykład: Model logiki układu automatycznego przeniesienia napędu (**sf_car.mdl**)



Object Hierarchy		Contents of: (chart) sf_car/shift_logic									
	Name	Scope	Trigger	Type	Size	Min	Max	InitVal	FrWS	ToWS	Watch
- sf_car	shift_logic										
+ gear_state	CALC_TH	Output(1)	Function ...								
+ selection_state	speed	Input(1)		double							
	up_th	Input(2)		double							
	down_th	Input(3)		double							
	gear	Output(1)		uint8		1	4	0			
	TWAIT	Constant		uint8							√

4. Opracowanie sprawozdania

Sprawozdanie ma zawierać dokumentację i analizę działania modeli SF z punktów 3.2 i 3.3.