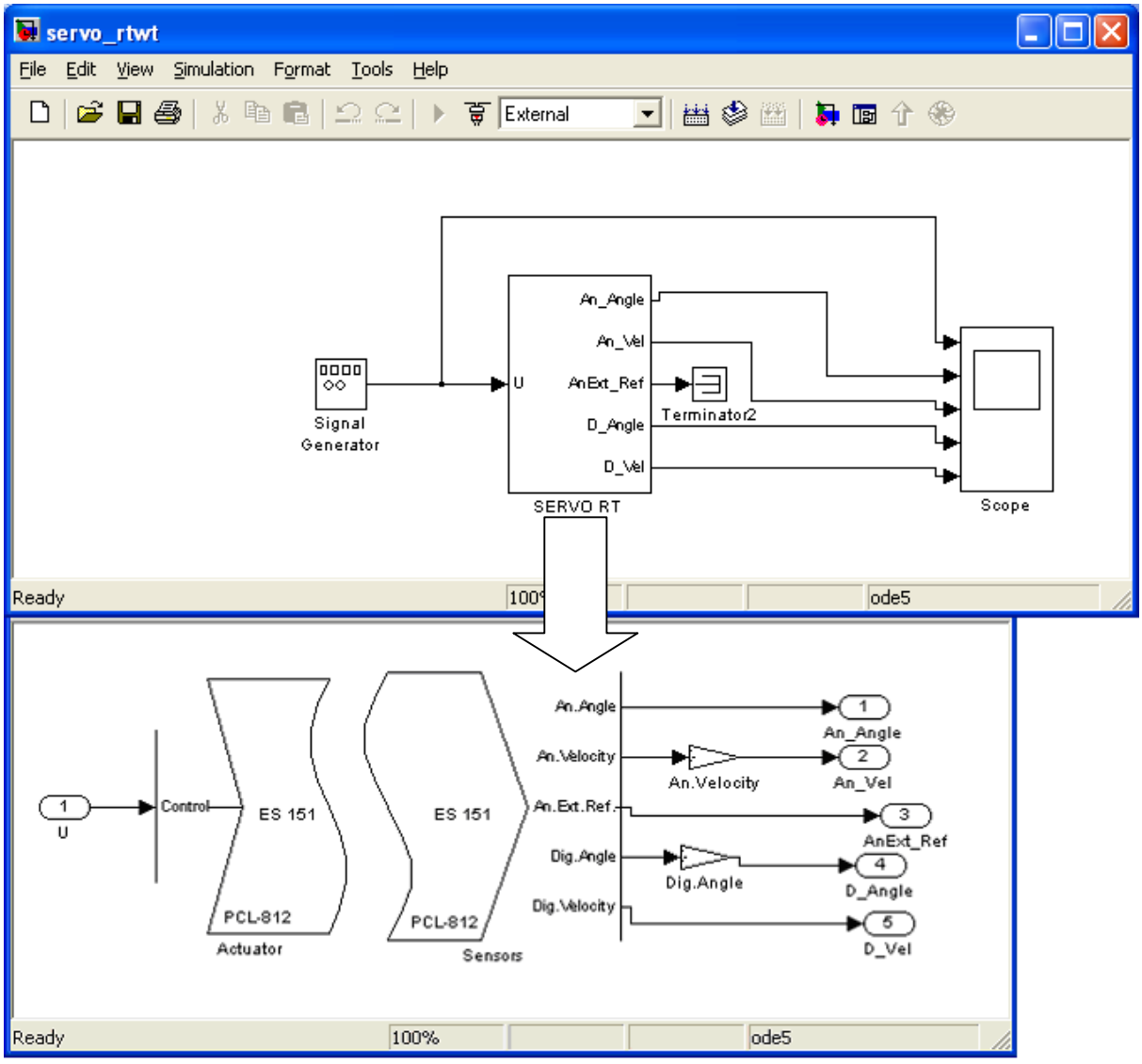
Model czasu rzeczywistego serwomechanizmu

Na Rys. 1 pokazany jest model *servo_rtwt.mdl*, który umożliwia pomiary i sterowanie w czasie rzeczywistym serwomechanizmu z silnikiem prądu stałego. Model jest skonfigurowany do generacji kodu wykonywalnego czasu rzeczywistego z wykorzystaniem przyborników RTW i RTWT. Jest on bazą, na której można skonstruować dowolny układ regulacji dla serwomechanizmu podczas zajęć laboratoryjnych.
W niniejszej instrukcji opisano szczegóły budowy i obsługi tego modelu czasu rzeczywistego.



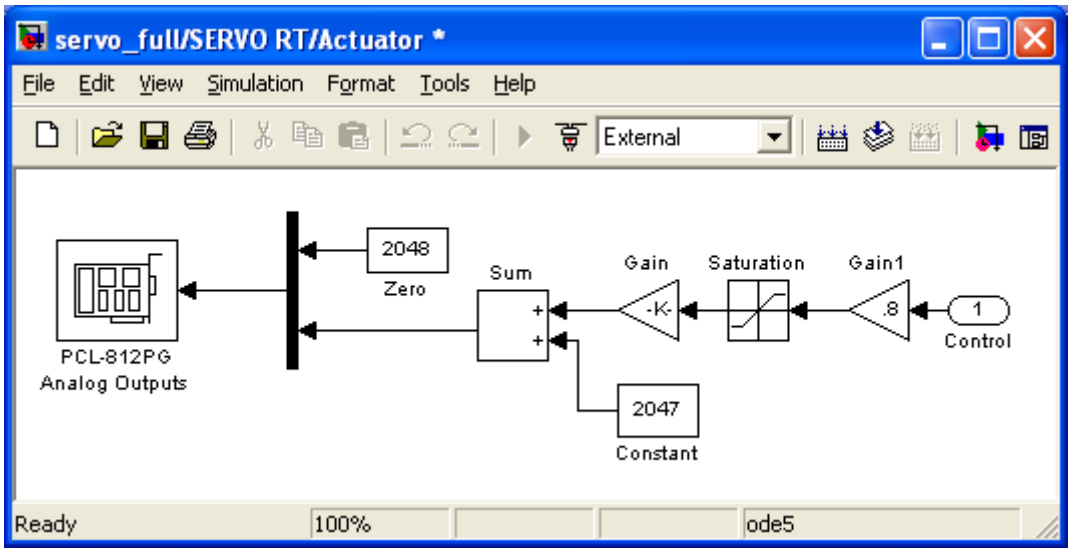
Rys. 1. Model czasu rzeczywistego serwomechanizmu. Na dolnym rysunku przedstawione jest wnętrze bloku *SERVO_RT*

Zwróćmy uwagę, że przedstawiony model nie różni się, na pierwszy rzut oka, od typowych, znanych nam modeli Simulinka. Niemniej różnice istnieją:

- Model zawiera bloki *Actuator* i *Sensors*. Są to tzw. device drivery czyli bloki wejść i wyjść komunikujące model Simulinka z obiektem rzeczywistym. Ich zawartość

przedstawiono na Rys. 2 i Rys. 3. Bloki te używa się w ten sam sposób jak inne bloki biblioteczne Simulinka.

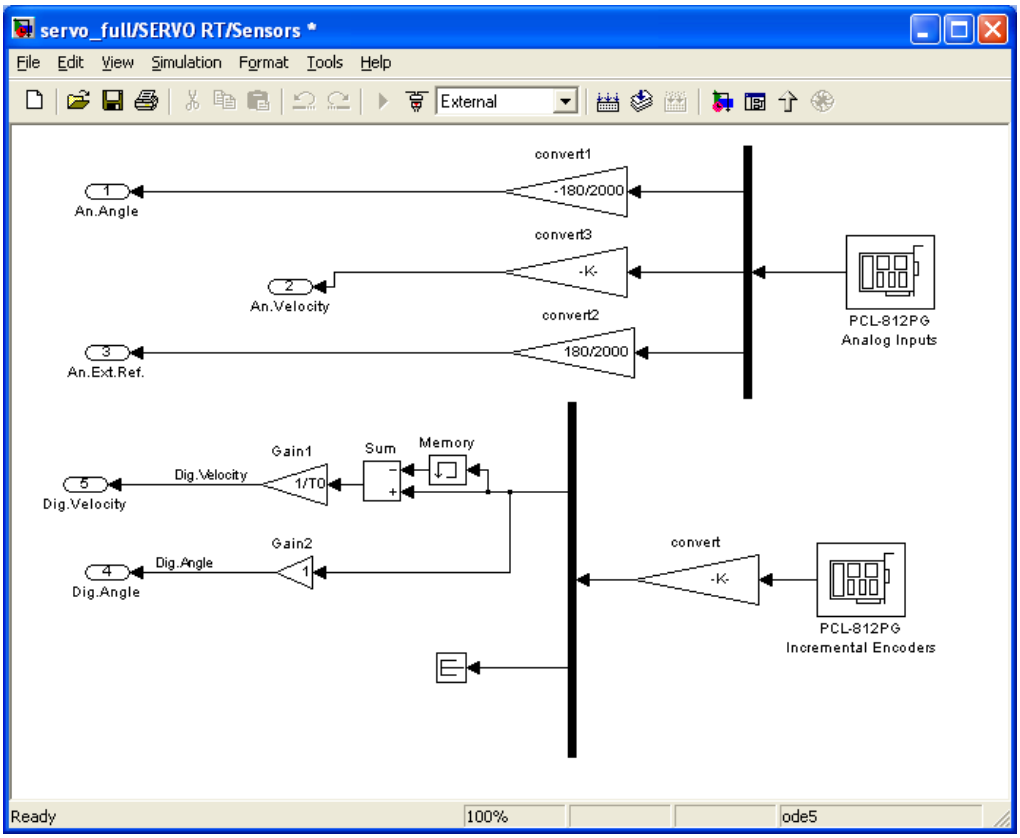
- Model ten nie jest wykonywany w trybie symulacji *Normal,* jak zwykły model symulacyjny, ale musi być przed użyciem skompilowany do kodu czasu rzeczywistego za pomocą RTW i RTWT, i dopiero potem wykonany w trybie *External.*
- Model jest specjalnie skonfigurowany. Konfiguracja ta jest ukryta w opcjach okna modelu i będzie wyjaśniona w następnym punkcie.



Rys. 2 Szczegóły bloku sterowania (Actuator)

Blok *Actuator* (Rys. 2) przekazuje sterowanie z modelu Simulinka do silnika serwomechanizmu. Sygnał *Control* jest bezwymiarowy i ograniczony do przedziału +/-1. Przetwornik na karcie jest 12 bitowy, więc zakres sygnału wyjściowego wynosi [0 – 4096]. Żeby uzyskać sygnał wyjściowy z karty z zakresu +/- 10V sygnał *Control* jest przesuwany do środka zakresu karty. Ze względu na fakt, że karta PCL-812PG posiada dwa kanały C/A na kartę podawane są dwa sygnały: jeden zerowy (stała 2048) i właściwy sygnał sterowania. Z odpowiedniego pinu wyjściowego karty sygnał sterowania analogowy, małej mocy, z zakresu +/-10V podawany jest na wzmacniacz mocy, który podaje napięcie bezpośrednio na silnik.

Na Rys. 3 pokazane są szczegóły bloku *Sensors,* który przekazuje pomiary z obiektu do modelu Simulinka. Pomiar kąta jest realizowany z wykorzystaniem enkodera inkrementalnego o rozdzielczości 1000 działek na obrót. Sygnał z enkodera jest przemnożony przez odpowiednią stałą i pojawia się jako *Dig_Angle* wyrażony w stopniach. Stała uwzględnia przekładnię a więc *D_Angle* jest kątem tarczy wyjściowej serwomechanizmu. Po przejściu przez wzmacniacz kąt *D_Angle* wyrażony jest w radianach. Na podstawie pomiaru kąta jest obliczana chwilowa prędkość kątowa *D_vel*. Analogowy pomiar kąta nie jest wykorzystywany.
Prędkość kątowa jest mierzona tachoprądnicą, której napięcie wyjściowe jest podawane na przetwornik A/C na karcie I/O. Napięcie to jest przeliczane na stopnie/s (Rys. 3) a potem na na rad/s (*An_velocity na* Rys. 1). W tym przypadku również jest uwzględniona przekładnia.



Rys. 3 Schemat bloku pomiarów (*Sensors*)

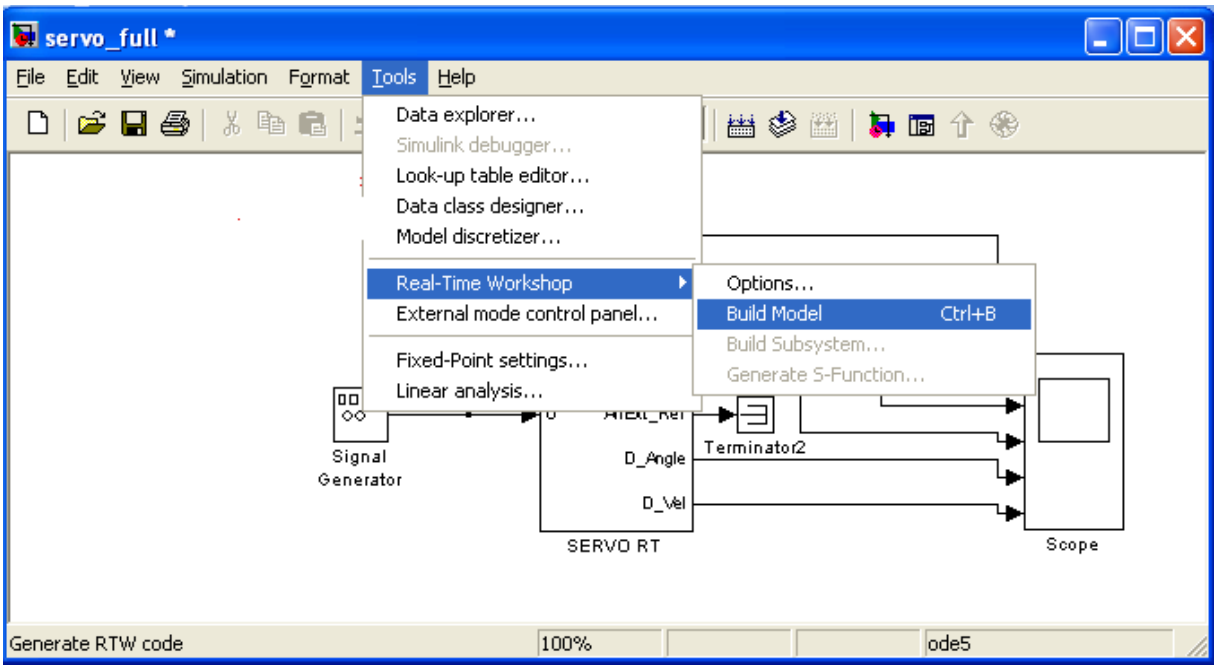
Podsumowując device driver serwomechanizmu realizuje dwa główne zadania:

- Udostępnia pomiary: z enkodera i tachoprądnicy i przekazuje je do komputera:
  Pozycję wyjściowej tarczy serwomechanizmu *D_Angle* [rad/s],
  Prędkość kątową tarczy wyjściowej [rad/s],
- Generuje sygnał sterujący silnikiem. Sygnał sterujący są bezwymiarowy i należy do przedziału [-1 , +1].

Projektowanie własnego modelu czasu rzeczywistego w środowisku Matlaba/Simulinka

Żeby zbudować system, który będzie działał w czasie rzeczywistym należy kolejno:

- zbudować w Simulinku model systemu sterowania używając dedykowanego dla serwomechanizmu device drivera oraz potrzebnych bloków bibliotecznych,
- wygenerować kod czasu rzeczywistego wybierając odpowiednią opcję w menu modelu (patrz Rys. 4) lub stosując skrót klawiaturowy *Ctrl+B*,
- uruchomić kod poprzez kliknięcie kolejno opcji w menu modelu: *Simulation/Connect to target* i *Simulation/Start real-time code*.



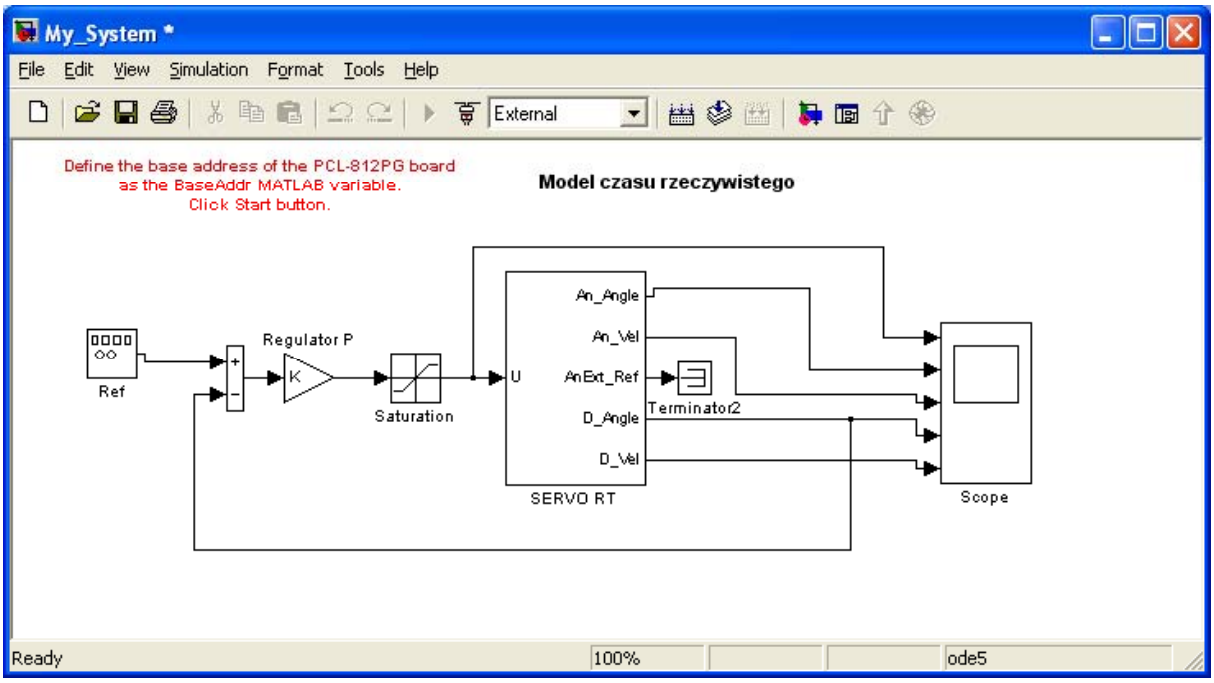
Rys. 4. Generacja kodu czasu rzeczywistego

Budowa modelu

Najprostszym sposobem zbudowania własnego modelu czasu rzeczywistego jest wykorzystanie jako wzorca dostępnego modelu *Servo_rtwt.mdl.* Należy otworzyć model, zapisać go pod inną nazwą (np. *My_System)* i zmodyfikować. W prezentowanym na Rys. 5 modelu zaimplementowano regulator P śledzący zadany (*Ref)* kąt tarczy

Budowa własnego modelu na bazie już istniejącego zapewnia, że wszystkie wewnętrzne opcje modelu zostaną właściwie ustawione. Te opcje są konieczne do poprawnego przeprowadzenia procesów generacji, kompilacji i linkowania kodu czasu rzeczywistego.

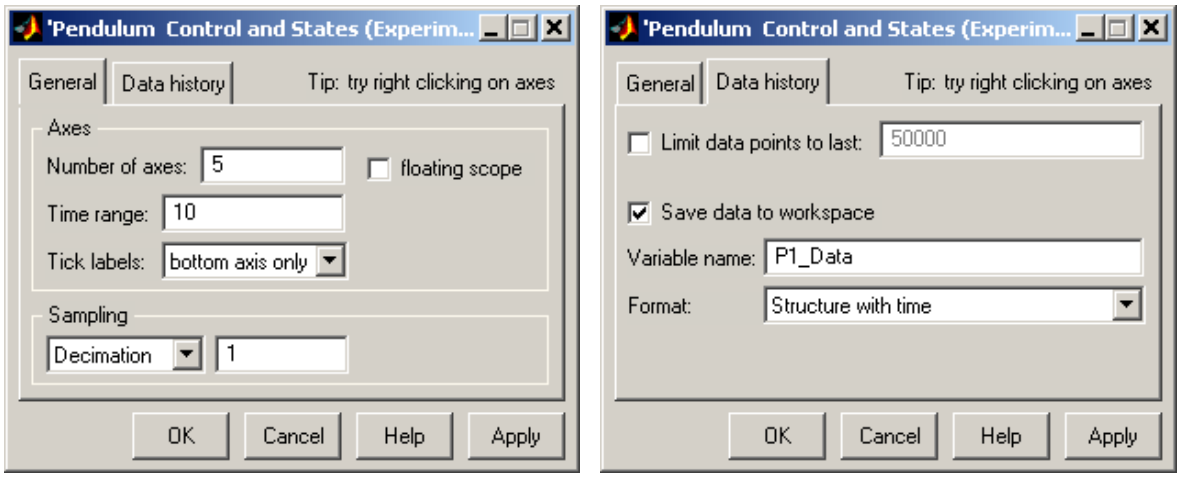
Użytkownik ma całkowitą dowolność przy projektowaniu własnego układu regulacji. Może użyć większości bibliotecznych bloków dostępnych w bibliotece Simulinka. Nie wolno mu jedynie usunąć device drivera, ponieważ model utraci połączenie z obiektem rzeczywistym



Rys. 5. Model układu regulacji w Simulinku

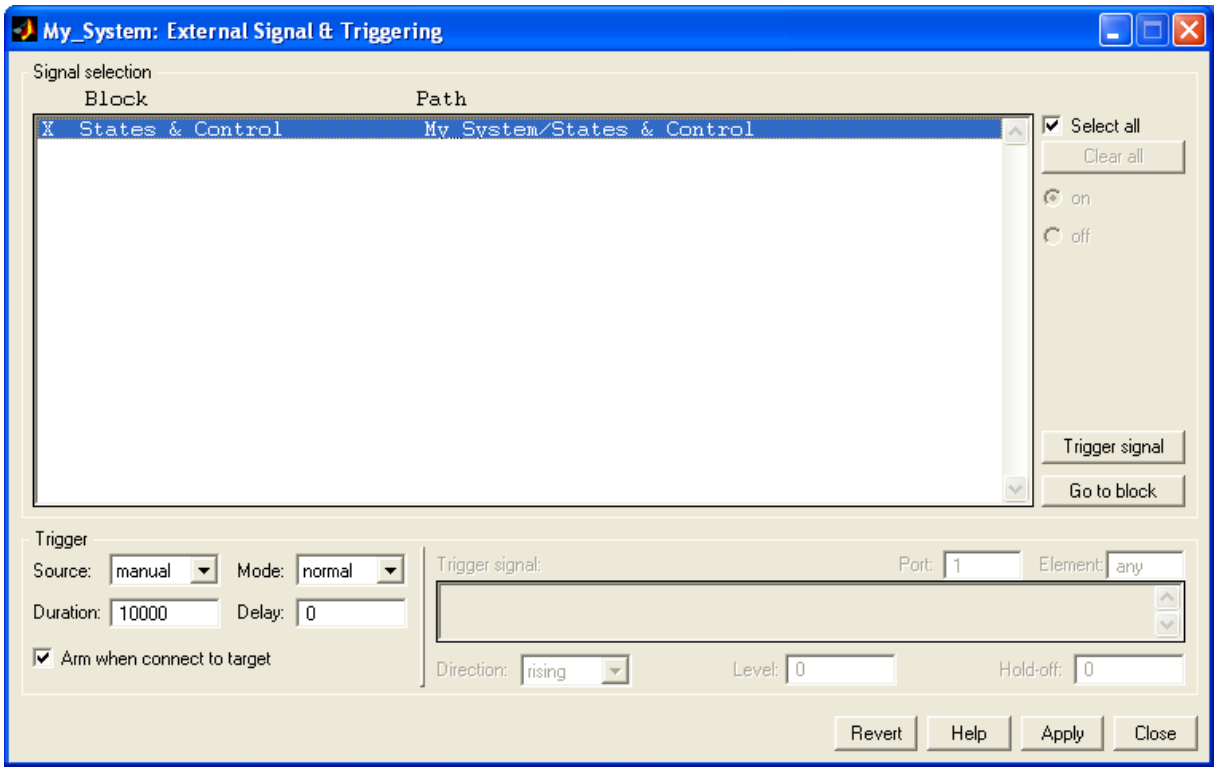
Chociaż nie jest to konieczne zalecane jest również pozostawienie oscyloskopu. Będzie on potrzebny do obserwacji zachowania się modelu. Właściwości oscyloskopu są dostępne w zakładce *Scope/Properties* (patrz Rys. 6). Zapamiętanie danych pomiarowych, w celu późniejszej obróbki off line, można wymusić zaznaczając opcję *save data to workspace*. Dane zostaną zapisane w zmiennej o nazwie wpisanej do okienka edycyjnego *Variable name*. Ten sposób jest jedyny dla pomiarów w czasie rzeczywistym, ponieważ normalnie używany blok *To Workspace* nie działa w RTWT.

Należy jeszcze zwrócić uwagę na ustawienie próbkowania. Jeżeli *Decimation* jest równe 1 to znaczy, że każda próbka jest rysowana na wykresie i równocześnie jest zapamiętywana w zmiennej. Ustawienie *Decimation* równe 10 oznacza, że jedynie co dziesiąta próbka jest zapamiętywana i wyświetlana.



Rys. 6. Parametry bloku oscyloskopu

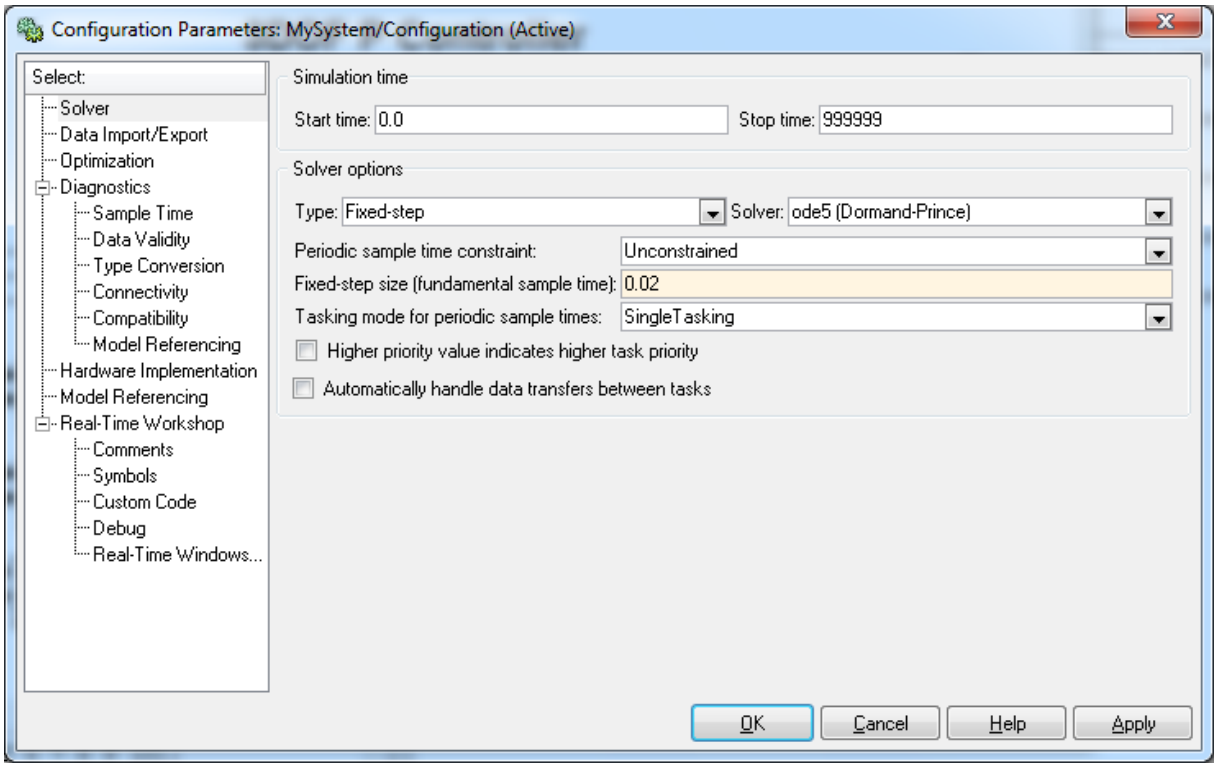
Sposób zbierania danych pomiarowych w czasie rzeczywistym określa się również w opcji *Tools/External Mode Control Panel*. Po kliknięciu klawisza *Signal Triggering* otworzy nam się okno pokazane na Rys. 7. Należy zaznaczyć blok oscyloskopu czyli *States & Control* (tzn., że tam będą zbierane dane pomiarowe), ustawić *Source* jako manual, a *Duration* równe liczbie próbek, którą będziemy chcieli zapamiętać na wykresach w oscyloskopie. Wielkość ta nie powinna być mniejsza niż zadeklarowana długość bufora w bloku *Scope*. Należy także zaznaczyć opcję *Arm when connect to target.*

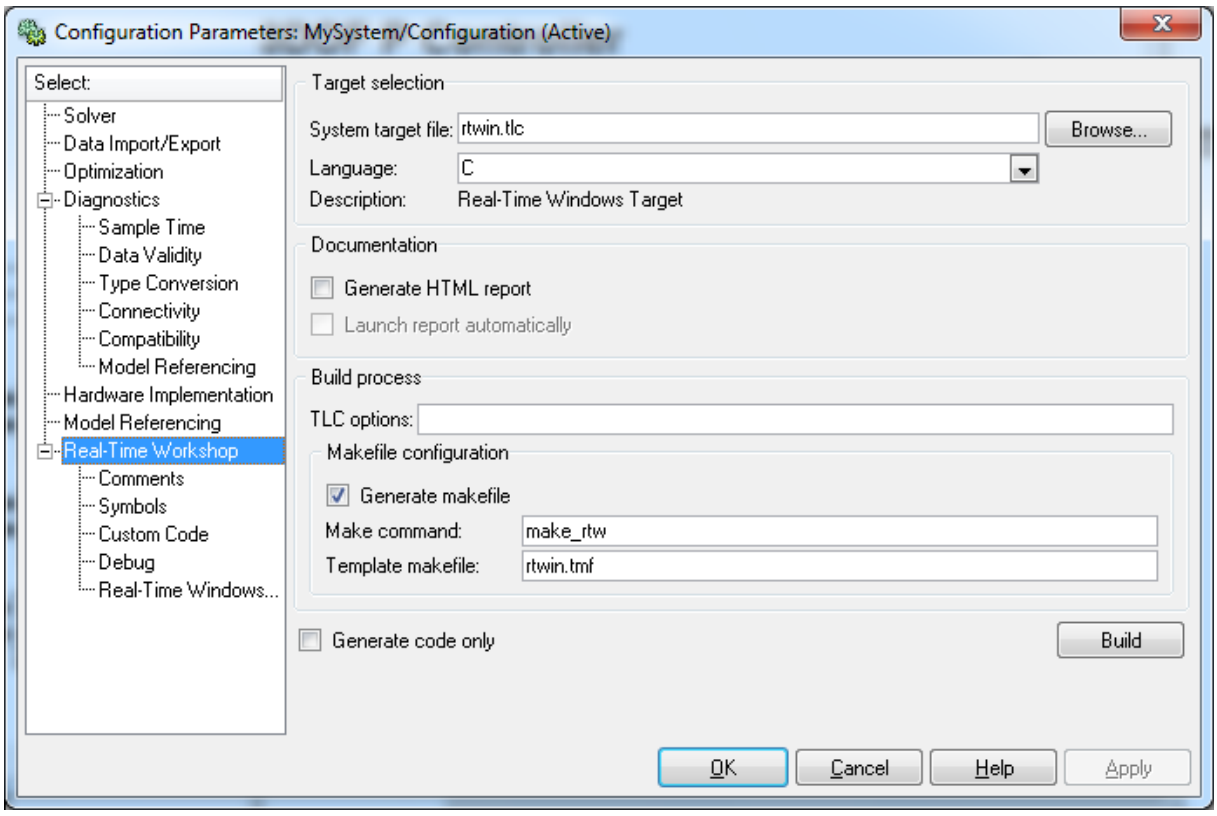


Rys. 7. Okno *External Signal & Triggering*

Proces generacji kodu czasu rzeczywistego

Kod jest generowany z użyciem Target Language Compiler (TLC) (patrz opis *Simulink Target Language*). Plik makefile jest używany do automatycznego budowania i załadowania zbiorów **.obj* związanych z zastosowanymi driverami systemu rzeczywistego
Rys. 8 przedstawia jak muszą być ustawione opcje dotyczące procesu tworzenia kodu czasu rzeczywistego, żeby proces ten nie generował błędów.





Rys. 8 Poprawnie ustawione parametry w opcji *Simulation parameters*

Plik o nazwie *make_rtw* zarządza procesem generacji. Plik *rtwin.tmf* to tzw. *template makefile*. Plik ten jest odpowiedzialny za generację kodu w C z użyciem wbudowanego kompilatora Open Watcom.

Zakładka *Solver* pozwala ustawić parametry symulacji. Konieczne jest ustawienie opcji *Fixed Step* czyli stałego kroku próbkowania. Wartość kroku próbkowania należy ustawić równą 0.01 [s].

Uwaga – jeżeli w modelu używane są bloki dyskretne należy pamiętać, że bloki te i ustawiony krok próbkowania muszą mieć wspólny podzielnik.

Po ustawieniu wszystkich parametrów możemy wygenerować kod czasu rzeczywistego. W tym celu naciskamy klawisz *Build* w zakładce *Real Time Workshop* opcji *Simualtion/Simulation Parameters* lub przy aktywnym oknie modelu naciśniemy kombinację klawiszy Ctrl+B. Pomyślna generacja kodu kończy się informacją w oknie Matlaba:

Model My_System.rtd successfully created

Successful completion of Real-Time Workshop build procedure for model: My_System

Uruchomienie modelu czasu rzeczywistego

W celu uruchomienia kodu czasu rzeczywistego musimy kolejno kliknąć klawisz *Simulation/Connect to target* – kod zostanie załadowany do pamięci. Następnie należy kliknąć opcję *Run real-time code* co uruchamia działanie modelu w czasie rzeczywistym.

Klikając opcję *Stop* w menu okna modelu zatrzymujemy uruchomiony model w dowolnej chwili.