



# Politechnika Wroclawska

---

Katedra Inżynierii Biomedycznej, Mechatroniki i Teorii Mechanizmów

**LABORATORIUM**

**Podstawy mechatroniki**

Sterowanie ruchem modułu przesuwu liniowego

WROCLAW 2015

# Laboratorium układów mechatronicznych

## Sterowanie ruchem modułu przesuwu liniowego

Widok stanowiska:



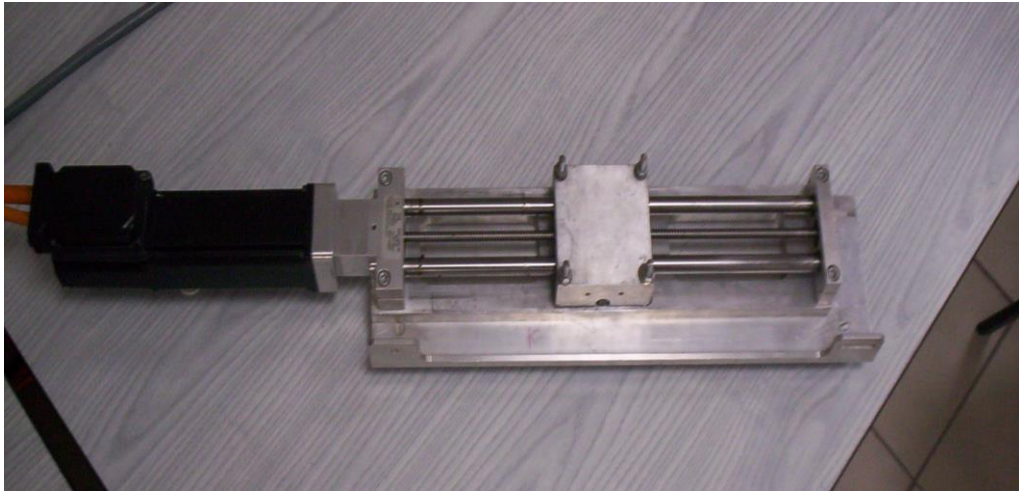
Opis stanowiska:

Zadaniem modułu przesuwu liniowego jest zamiana ruchu obrotowego serwonapędu na ruch prostoliniowy sań według dowolnej charakterystyki. Ruch postępowy sań zostaje wymuszony obrotem śruby pociągowej. Elektronika sterująca i zasilająca umieszczona jest w szafie sterowniczej. Sterowanie napędem odbywa się poprzez kartę kontrolera umieszczonej w szynie ISA komputera, sterownika dSpace oraz modułu we/wy dSpace, przy pomocy odpowiedniego oprogramowania zainstalowanego na komputerze PC.

Cel zajęć:

1. Napisanie programu sterującego serwonapędem w układzie otwartym.
2. Napisanie programu sterującego serwonapędem w układzie zamkniętym, ze sprzężeniem zwrotnym.
3. Identyfikacja modelu, dobranie wartości parametrów sterowania.
4. Realizacja zaplanowanego ruchu.
5. Wizualizacja zmiennych i kontrola programu sterującego.
6. Analiza przebiegu zarejestrowanych sygnałów

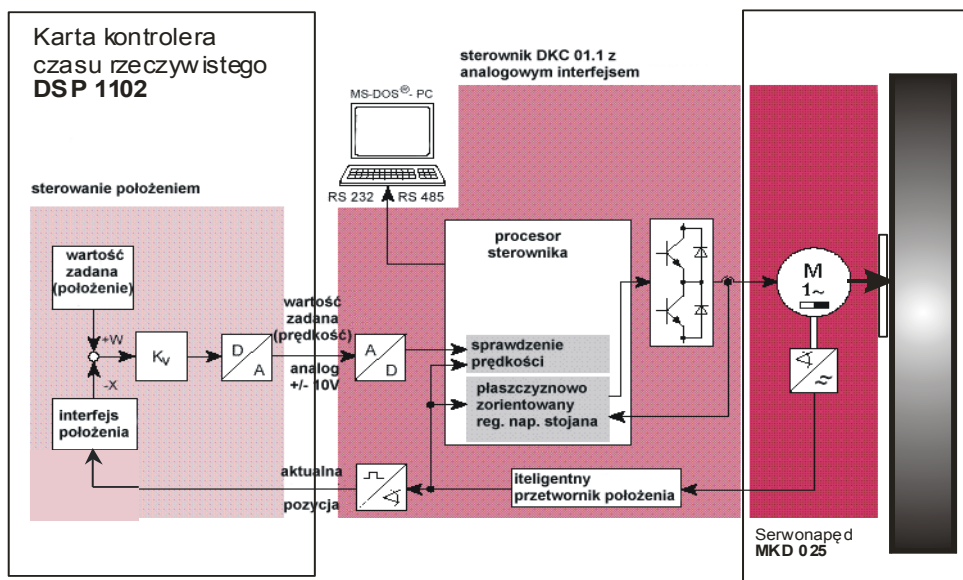
## 1. Opis stanowiska



Podstawowymi elementami wchodzącymi w skład stanowiska laboratoryjnego są sianie liniowe, serwonapęd oraz układ sterowania. Zadaniem modułu przesuwu liniowego jest zamiana ruchu obrotowego serwonapędu na ruch prostoliniowy sań według dowolnej charakterystyki. Ruch postępowy sań zostaje wymuszony obrotem śruby pociągowej (kulowo-tocznej).

## 2. Układ sterowania

W module przesuwu liniowego zastosowano klasyczne rozwiązanie sterowania napędem. Silnik wraz ze sterownikiem pracuje w trybie analogowego sterowania prędkością i „zarządzany” jest przez zewnętrzny kontroler czasu rzeczywistego. Ogólny schemat tego rozwiązania został przedstawiony na rys. 1.



Rys.1. Schemat układu sterowania

Do sterownika serwonapędu jest przekazywany analogowy sygnał sterujący  $\pm 10V$ , który jest interpretowany przez sterownik pracujący jako regulator prędkości jako  $\pm \omega_{max}$

silnika (10V - 2500 obr/min). Pętla sprzężenia zwrotnego jest zamknięta w kontrolerze czasu rzeczywistego (pracującego jako regulator położenia), który zbiera sygnał położenia z enkodera silnika i porównuje z wartością zadaną położenia. Wyliczany uchyb położenia służy do wyliczenia i wygenerowania sygnału sterującego. Układ regulacji w kontrolerze pracuje jako regulator proporcjonalny ze stałą wzmocnienia  $K_v$ .

Układ sterowania jest rozbudowany o obsługę dodatkowych sygnałów cyfrowych, które służą do realizacji czynności pomocniczych służących do synchronizacji i kontroli napędów. Są to sygnały służące do określania położenia bazowego napędu oraz sygnały od czujników położenia krańcowych.

## 2.1 Karta kontrolera czasu rzeczywistego

Kontroler DS 1103 firmy dSPACE jest kompletnym programowalnym system sterowania w czasie rzeczywistym. Kontroler jest zbudowany jako jednopłytkowy układ zawierający wszystkie niezbędne układy wejścia i wyjścia zarówno analogowe jak i cyfrowe. Na rys. 2 przedstawiono widok kontrolera. Kontroler znajduje się w osobnej obudowie. Z komputerem PC komunikuje się przez specjalną kartę montowaną w szynie ISA. Kontroler jest wyposażony we własny procesor PowerPC PPC 750GX pracujący z częstotliwością 1 GHz, procesor DSP Texas Instruments TMS320F240 pracujący z częstotliwością 20 MHz oraz w 32 MB pamięci SDRAM zarezerwowane na program i 96 MB pamięci SDRAM zarezerwowane na dane i komunikację z komputerem. Układ jest wyposażony w m.in. w:

- ADC (16bit): 16 kanałów (multipleksowane po 4 kanały do jednego przetwornika ADC), oraz 4 niezależne kanały, napięcie wejściowe +/-10V,
- DAC (16bit): 8 kanałów, napięcie wyjściowe +/-10V,
- Liczniki: 2 liczniki ogólne 32bit,
  - 1 licznik próbkujący 32bit,
  - 1 licznik czasu 64bit,
- Przerwania: 20 przerwania, w tym 4 zewnętrzne,
- Wejścia/Wyjścia: 4x8bit, napięcia zgodne z TTL, max prąd 10mA na wyjście
- Enkodery: 6 cyfrowych 24bity, max prędkość 1,65MHz;
  - 1 analogowy 32bity, 1Vpp lub 11μApp, max prędkość 600kHz
- DSP: Texas Instruments TMS320F240 DSP, zasoby: 16 kanałów ADC, 10 kanałów PWM, 2 porty szeregowo



Rys. 2. Karta kontrolera czasu rzeczywistego dSPACE DS 1103

## 2.2 Teoria sterowania

Kontroler jest w pełni programowalny. Napisany program po załadowaniu do kontrolera jest autonomicznie wykonywany w czasie rzeczywistym. Wszystkie sygnały wychodzące z kontrolera są wyprowadzone do zewnętrznego terminala do którego podłącza się wszystkie układy zewnętrzne (rys. 3).



Rys. 3. Widok terminala

Karta DS 1103 pracuje w czasie rzeczywistym z zadaniem krokiem  $\Delta t$ . W każdej chwili czasowej  $t_i$  wylicza sygnał sterujący  $\omega_i$ , który jest zadawany w czasie  $\Delta t$ . W chwili  $t_i$  należy wygenerować sygnał sterujący  $\omega_i$  stały dla czasu  $\Delta t$ , który jest określony związkiem:

$$e_i = \frac{1}{\Delta t} (q_{i+1}^z - q_i^R)$$
$$\omega_i = K_v e_i$$
(1)

gdzie:  $q_i^R$  - rzeczywiste położenie kątowne napędu w chwili  $t_i$ ,  
 $q_{i+1}^z$  - zadane położenie kątowne napędu w chwili  $t_{i+1}$ ,  
 $K_v$  - stała regulacji,  
 $e_i$  - uchyb regulacji dla czasu  $t_i$ .

Tak wyliczany przez kontroler DS 1103 sygnał sterujący  $\omega_i$  powinien zapewnić realizację ruchu napędu po zadanych punktach  $q^z$ . Ze względu na błędy realizacji analogowego sygnału  $\omega_i$  oraz bezwładność sterownika napędu (pracującego jako regulator prędkości) i dynamikę układu napęd porusza się z błędem  $\Delta q$ . Układ sterowania wylicza sygnał sterujący  $\omega_i$  w każdym kroku bazując na rzeczywistym uchybie  $e_i$ . Błąd  $\Delta q$  jest uwzględniany w wyliczeniu tego sygnału i jest korygowany na bieżąco. W chwili  $t_i$  błąd realizacji trajektorii  $\Delta q_i$  można określić wzorem:

$$\Delta q = q^R - q^z \quad (2)$$

gdzie:  $q_i^R$  - rzeczywiste położenie kątowe napędu w chwili  $t_i$ ,

$q_i^z$  - zadane położenie kątowe napędu w chwili  $t_i$ .

Rzeczywisty sterownik napędu jest sterowany analogowym sygnałem  $-10V \leq U_i \leq +10V$ , który jest interpretowany jako sygnał  $-\omega_{\max} \leq \omega_i \leq +\omega_{\max}$ . Enkoder mierzący położenie wału jest zamontowany na silniku, a wartości  $q^z$  odnoszą się do wału za przekładnią. W związku z tym należy dokonać przeliczenia wartości  $\omega_i$  na sygnał  $U_i$  [V] uwzględniający przełożenie przekładni:

$$U_i = \frac{10V \cdot i_p}{\Delta t \cdot \omega_{\max}} (q_{i+1}^z - q_i^r)$$

$$U_i = K_v e_i \quad (3)$$

$$K_v = \frac{10V \cdot i_p}{\Delta t \cdot \omega_{\max}}$$

gdzie: 10V – maksymalna wartość sygnału w woltach,

$i_p$  – przełożenie przekładni,

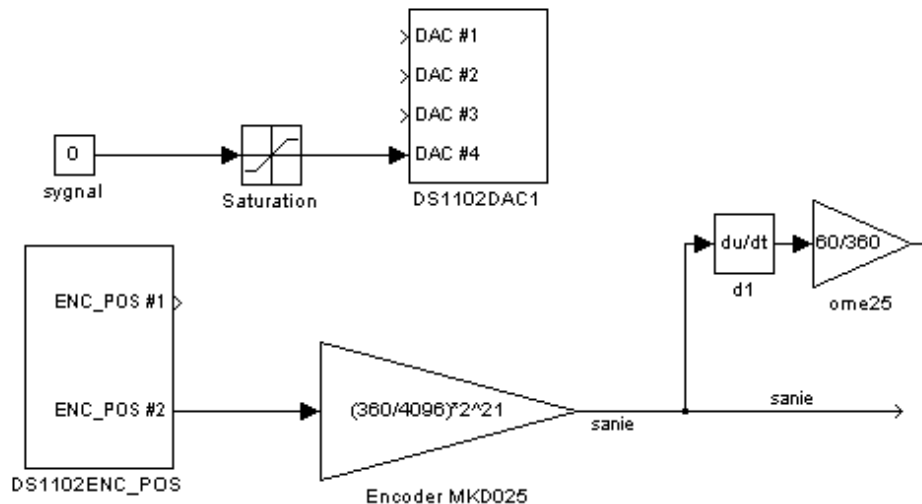
$\Delta t$  – czas próbkowania,

$\omega_{\max}$  – prędkość maksymalna napędu.

## 2.3 Program sterujący

Karta DSP 1102 jest programowana przy pomocy oprogramowania SIMULINK® systemu analizy matematycznej MATLAB®. Dostęp do zasobów kontrolera DSP 1102 (wyjścia/wejścia analogowe i cyfrowe, wyjścia enkoderowe, przerwania itp.) jest realizowany przy pomocy modułów biblioteki RTI (Real Time Interface) firmy dSPACE.

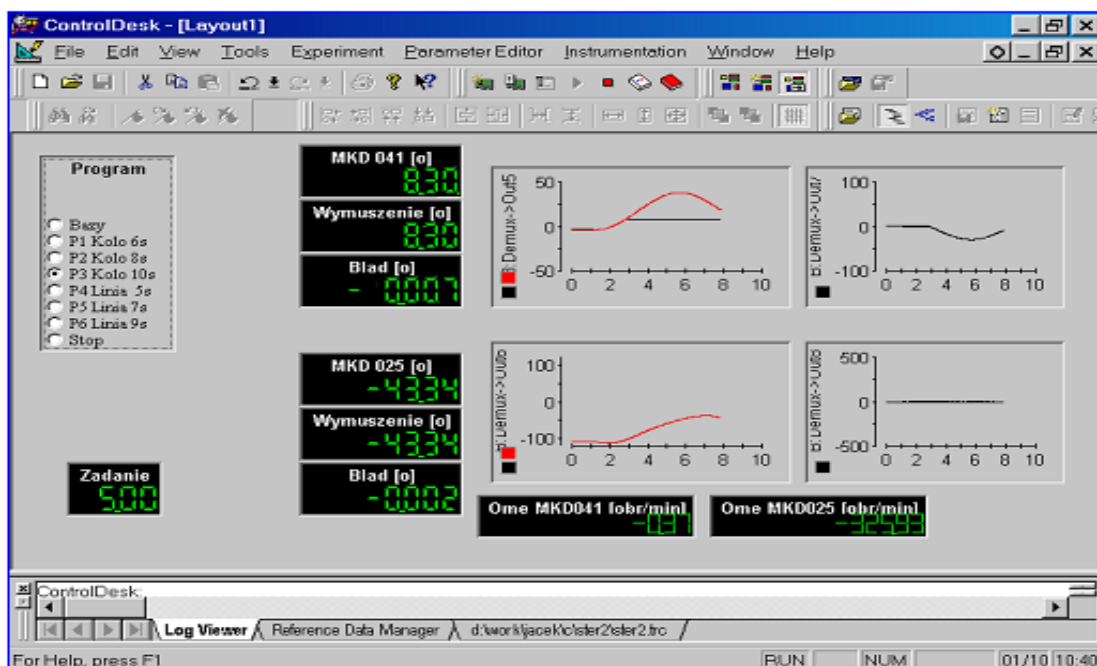
Utworzony logiczny układ sterowania w postaci bloków i sygnałów SIMULINK'a jest następnie kompilowany do postaci kodu maszynowego dla procesora TI (Texas Instrument) TMS320F240 karty DS 1103. Tak skompilowany kod jest ładowany do kontrolera, w którym jest wykonywany w czasie rzeczywistym dla zadanego stałego kroku obliczeniowego. Przykładowy program sterujący realizujący zadanie regulatora położenia w postaci bloków i sygnałów SIMULINKA został przedstawiony na rys. 4.



Rys.4. Przykładowy program sterujący w SIMULINK'u

Załadowany do kontrolera program jest wykonywany autonomicznie. Dostęp do kontrolera umożliwiający wykonywanie funkcji systemowych (uruchamianie, zatrzymywanie programu, resetowanie kontrolera, dostęp do sygnałów sterujących) jest możliwy za pomocą oprogramowania CONTROL DESK® uruchamianego w komputerze PC, w którym jest osadzona karta kontrolera. W środowisku CONTROL DESK-a można utworzyć wygodne panele sterujące kontrolerem przy pomocy myszy i klawiatury PC.

Na rys.5 został pokazany przykładowy panel sterujący dla układu sterowania dwoma silnikami.. Panel ten umożliwia wykonywanie następujących zadań: synchronizacja mechanizmu (najazd silników na mikroprzełączniki bazowe), startowanie i zatrzymywanie mechanizmu, startowanie programów realizacji ruchu po zaprogramowanych trajektoriach. Panel umożliwia wizualizację sygnałów w czasie rzeczywistym w postaci wyświetlaczy cyfrowych i wykresów. Dzięki panelowi jest możliwy zapis sygnałów w czasie rzeczywistym na dysk PC.

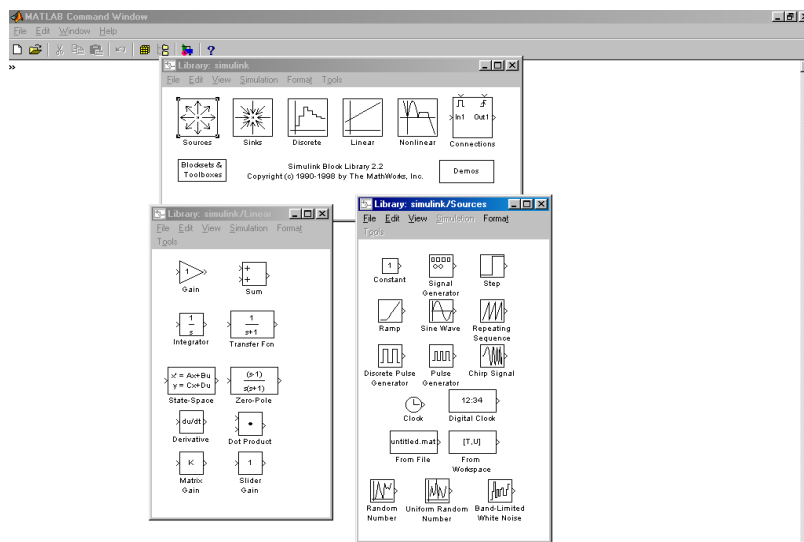


Rys.5. Widok panelu sterowania układu utworzonego w programie CONTRL DESK®

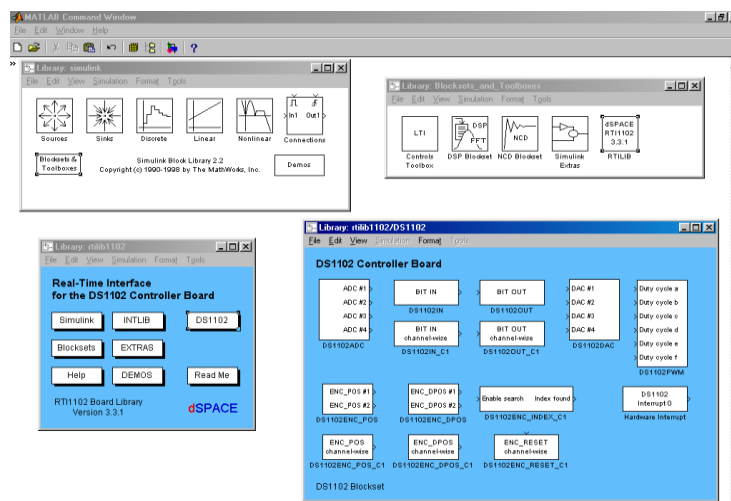
### 3. Przebieg ćwiczenia

#### 3.1. Opis programu SIMULINK

Uruchomić program MATLAB, ustalić bieżący katalog roboczy (np. D:\student), a następnie należy przejść do modułu SIMULINK. Zestaw bibliotek programowych (bloków) przedstawiono na rys. 6. Jedną z bibliotek jest *Blocksets&Toolboxes*, w której znajduje się blok RTI1103 (Real Time Interface) umożliwiający sterowanie sygnałami karty DS 1103 (rys. 7).



Rys.6. Zestawienie podstawowych bibliotek modułu Simulink



Rys. 7. Zestawienie bibliotek Simulink'a z RTI DS1102

#### 3.2. Sterowanie w układzie otwartym:

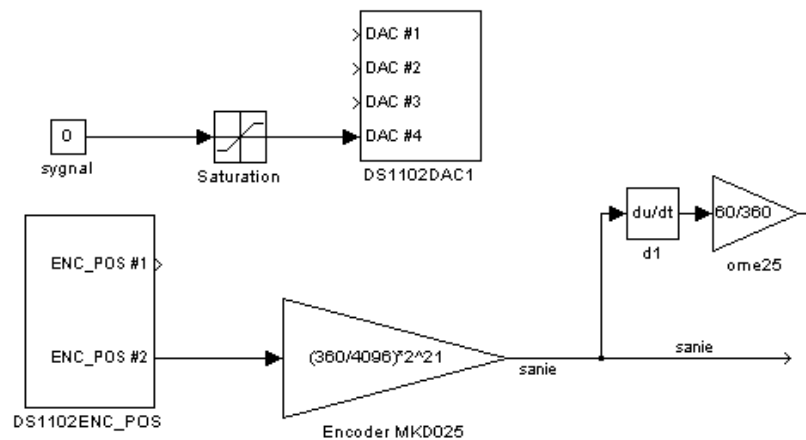
Bazując na blokach modułu SIMULINK należy napisać program sterujący, którego zadaniem jest:

1. Wystawienie stałego sygnału sterującego prędkością obrotową silnika w układzie analogowego sterowania prędkością,



2. Odczyt prędkości obrotowej wału silnika na podstawie sygnałów generowanych przez encoder.

Sterowanie odbywa się w układzie otwartym, czyli bez sprzężenia zwrotnego. Program ten w postaci bloków Simulinka przedstawiono na rys. 8.



Rys. 8. Program realizujący sterowanie silnikiem w układzie otwartym

Do realizacji powyższego programu wykorzystywane będą następujące bloki biblioteki *RTI*:

- *DS1103DAC* – blok 4 wyjść analogowych (sprawdzić, do którego kanału jest podłączony sterownik silnika)
- *DS1103ENC\_POS* – blok wejść enkoderowych (sprawdzić, do którego kanału jest podłączony enkoder silnika)

Pozostałe bloki to:

- *signal* – generowanie stałego co do wartości sygnału (wpisana wartość - 0)
- *saturation* – blok normalizujący wartość zadawanego sygnału (min=-1, max=1)
- *gain (Encoder MKD025)*- wzmacniacz pełniący funkcję przelicznika, należy przeliczyć impulsy enkoder na jednostki, w których podajemy wartość zadaną (np. mm), potrzebne dane: liczba impulsów/ obrót: 4096, skok śruby: sprawdzić
- *d1 (du/dt)* – człon różniczkujący
- *gain (ome25)* – wzmacniacz którego zadaniem jest przeliczanie jednostek prędkości (opcjonalny)

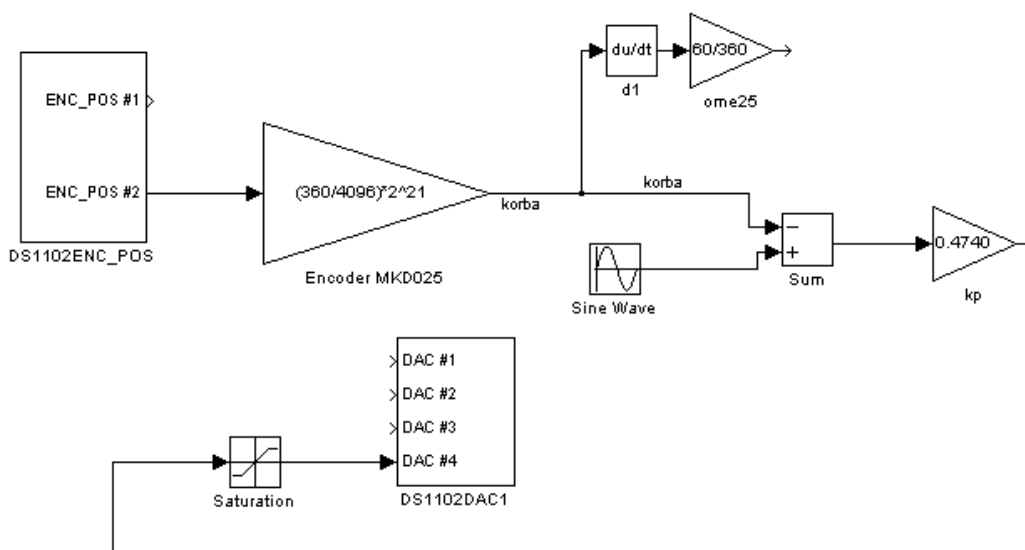
Po napisaniu zadanego programu należy go zapisać w katalogu roboczym. Następnie z menu „*Tools*” należy wybrać opcję „*RTW build*”. Komenda ta powoduje skompilowanie programu do postaci maszynowej, po czym następuje przesłanie kodu do karty DSP1102 oraz automatyczne wykonanie zapisanego programu sterującego.

### 3.3. Sterowanie w układzie zamkniętym

Bazując na blokach modułu SIMULINK należy napisać program sterujący, którego zadaniem jest:

1. Zadawanie sygnału sinusoidalnie zmiennego sterującego prędkością obrotową silnika w układzie analogowego sterowania prędkością ze sprzężeniem zwrotnym,
2. Odczyt wartości zadawanego sygnału oraz odczyt prędkości obrotowej wału silnika na podstawie sygnałów generowanych przez encoder silnika.

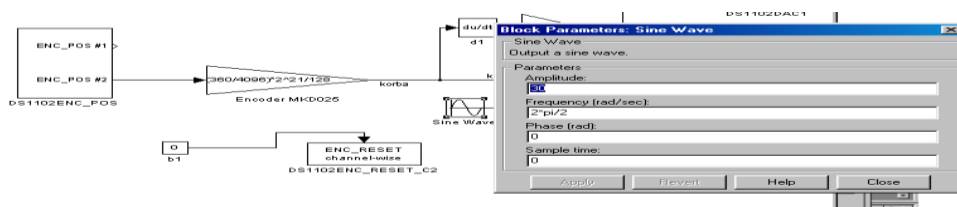
Sterowanie odbywa się w układzie otwartym, czyli bez sprzężenia zwrotnego. Program ten w postaci bloków Simulinka przedstawiono na rys. 8.



Rys. 9. Program realizujący sterowanie silnikiem ze sprzężeniem zwrotnym

Program realizujący sterowanie w układzie zamkniętym w porównaniu do poprzedniego (sterowanie w układzie otwartym) musi być uzupełniony o następujące dodatkowe bloki:

- *Sine Wave* – generowanie sygnału sinusoidalnego o parametrach jak na rys. 10
- *Sum* – blok sumatora, wyliczający uchyb regulacji
- *kp* – wzmacnienie (stała regulacji  $k_p=0.474$ )

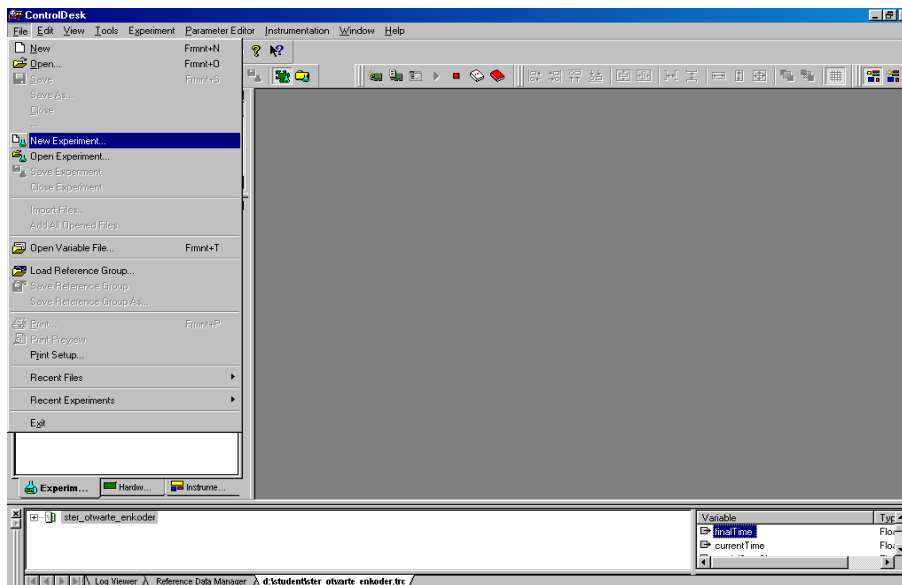


Rys. 10. Parametry bloku *Sine Wave* jakie należy wpisać.

Po napisaniu zadanego programu należy go zapisać w katalogu roboczym. Następnie z menu „Tools” należy wybrać opcję „RTW build”. Komenda ta powoduje skompilowanie programu do postaci maszynowej, po czym następuje przesłanie kodu do karty DSP1102 oraz automatyczne wykonanie zapisanego programu sterującego.

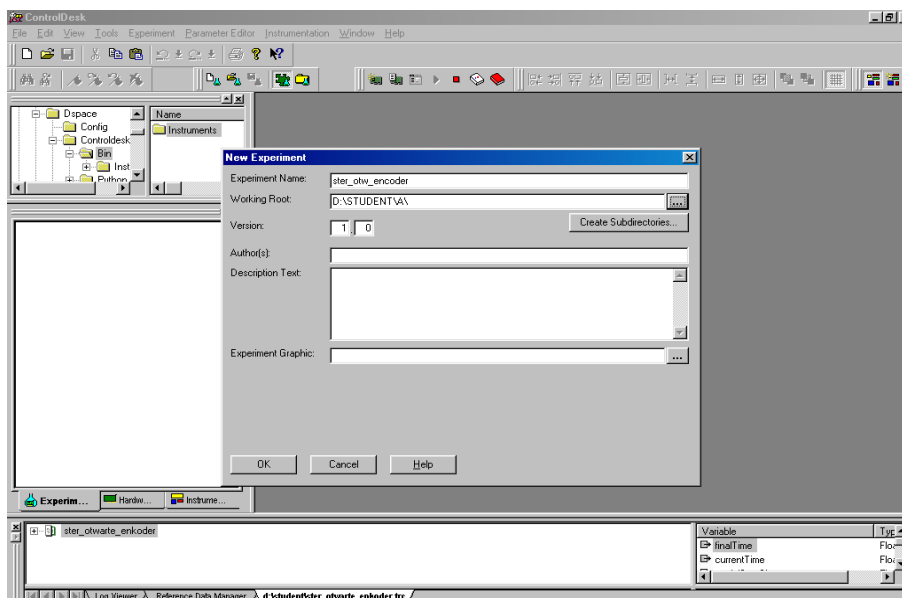
### 3.4. Panel sterujący CONTROL DESK®

W środowisku CONTROL DESK-a można utworzyć wygodne panele sterujące kontrolerem DSP1102 przy pomocy myszy i klawiatury PC. Uruchamianie, zatrzymywanie układu oraz pomiary odbywają się przy pomocy opracowanego w systemie CONTROL DESK® panelu sterowania. Widok panelu po uruchomieniu programu przedstawiono na rys.11.



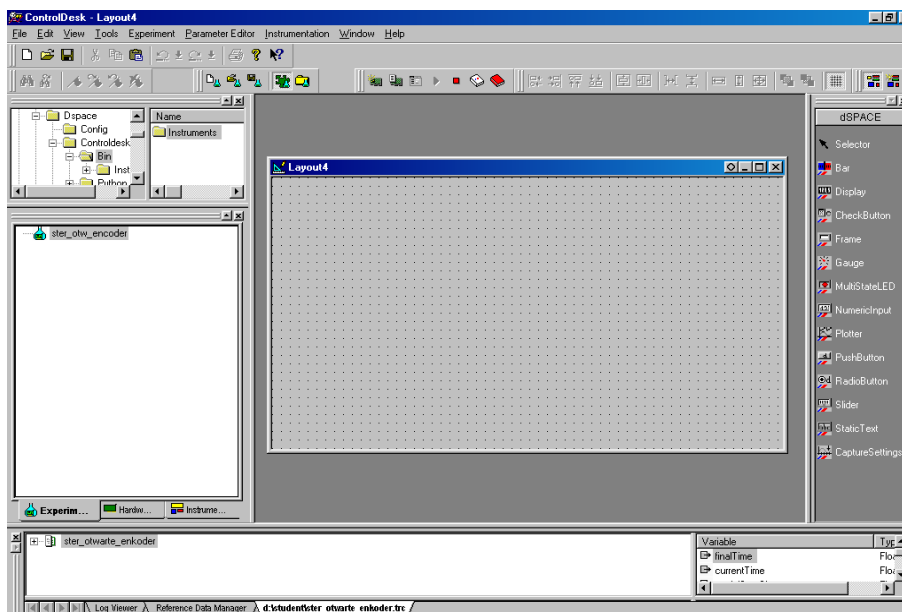
Rys. 11. Widok CONTROL DESK-a po jego uruchomieniu.

Z menu „File” należy wybrać opcję „New experiment”, po czym należy nadać mu nazwę oraz określić katalog roboczy (rys.12).



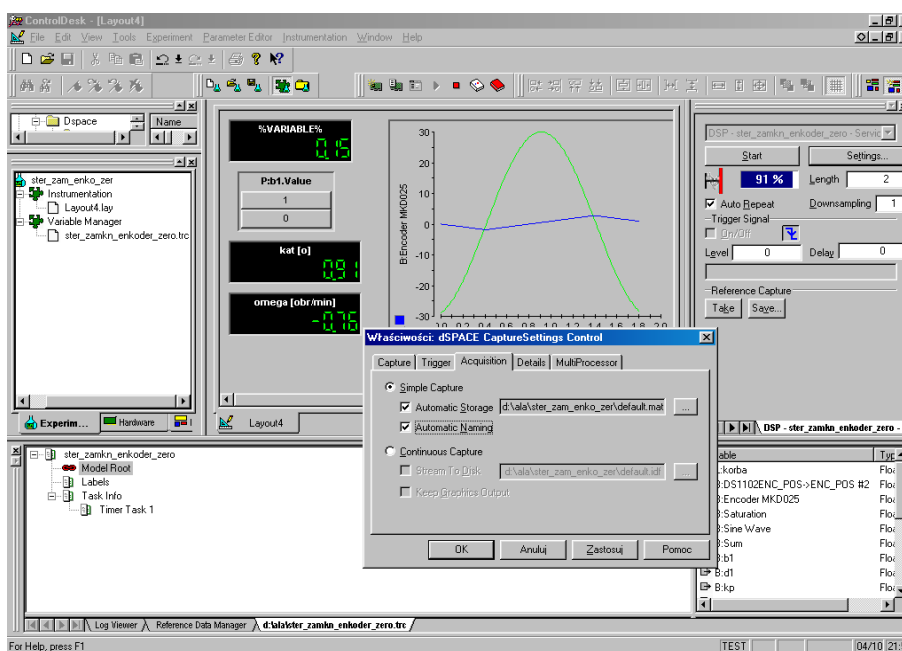
Rys.12. Tworzenie nowego eksperymentu

W następnej kolejności z menu „File” należy wybrać opcję „New”. Powoduje to pojawienie się podmenu z którego należy wybrać opcję „Layout”. Widok ekranu po wykonaniu powyższych czynności przedstawiono na rys. 13. Obecnie program jest w trybie edycji *Edition mode* W celu uruchomienia utworzonego programu należy z menu górnego *Instrumentation* wybrać opcję *Animation mode*.



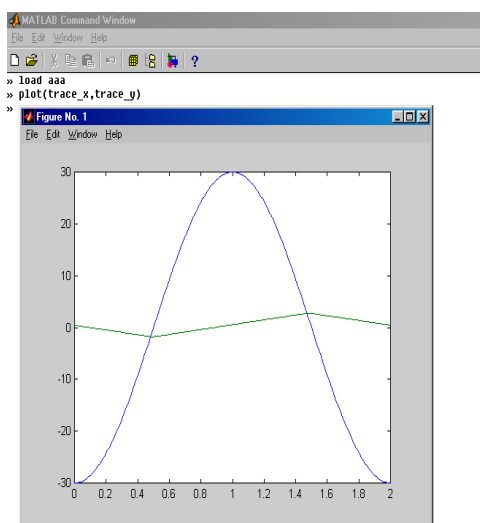
Rys.13. Widok ekranu CONTROL DESK-a przy tworzeniu nowego eksperymentu

Panel ten umożliwia startowanie i zatrzymywanie programu sterującego ruchem silnika (ikony: *start* i *stop*). Panel umożliwia również wizualizację sygnałów w czasie rzeczywistym w postaci wyświetlaczy cyfrowych i wykresów jak również zmianę wartości zadawanego sygnału (*Slider*). Tworzenie tych elementów odbywa się poprzez ich wybór z bocznego (prawego) menu, a następnie narysowaniu w oknie *Layout*. W celu obserwacji zmian wybranej zmiennej np. na wyświetlaczu (wykresie) należy ją wybrać myszką i „przenieść” w pole narysowanego wcześniej w oknie graficznym wyświetlacza (wykresu). Dla każdego elementu „pod prawym klawiszem myszy” znajdują się opcje szczegółowych ustawień parametrów tego elementu. Dzięki panelowi jest możliwy zapis wykresów w czasie rzeczywistym na dysk PC, a sposób dokonania zapisu przedstawiono na rys. 14. Przycisk *start* z prawego okna ekranu powoduje zapis zmiennych na dysk, przycisk *stop* przerywa ten zapis.



Rys. 14. Zapis przebiegu zmiennych na dysk do pliku default.mat

W celu dokonania analizy przebiegu zmiennych wygodnie jest użyć do tego celu funkcji Matlab'a *plot*, ale uprzednio należy wczytać zapisany plik ze zmiennymi (\*.mat) poleceniem *load* (rys. 15)



Rys. 15. Wykres zmiennych zapisanych z pliku utworzonego w CONTROL DESK-u

### 3. Procedura uruchamiania silnika

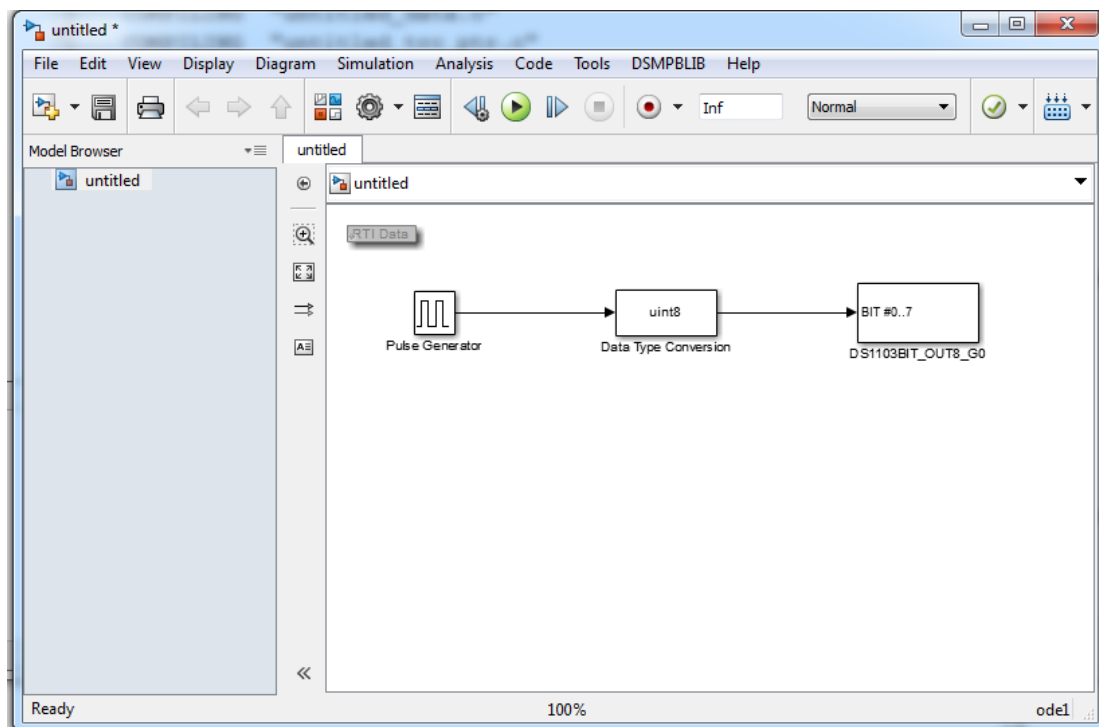
1. wczytać napisany w SIMULINK'u program sterujący należy, następnie z menu „Tools” należy wybrać opcję „RTW build”.
2. wczytać zapisany panel sterujący CONTROL DESK (*Open, New experiment*)
3. włączyć zasilanie główne
4. włączyć zasilanie 24V
5. włączyć zasilanie 230V
6. włączyć *RF* i *START*
7. w utworzonym panelu CONTROL DESKa przejść do trybu *Animation mode* i kliknąć ikonkę *start*
8. w przypadku źle działającego programu szybko przełączyć dźwignię *start* – napęd zatrzyma się



Rys. 16. Widok panelu zasilania

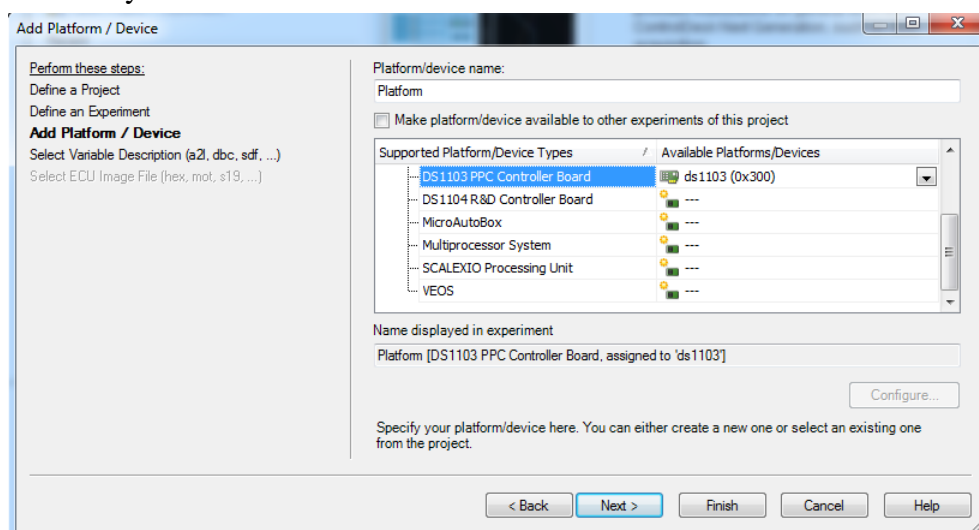
## 4. Przykładowy program z wykorzystaniem dSpace.

1. Utworzyć nowy model w SIMULINKU, taki jak na rysunku 17.:



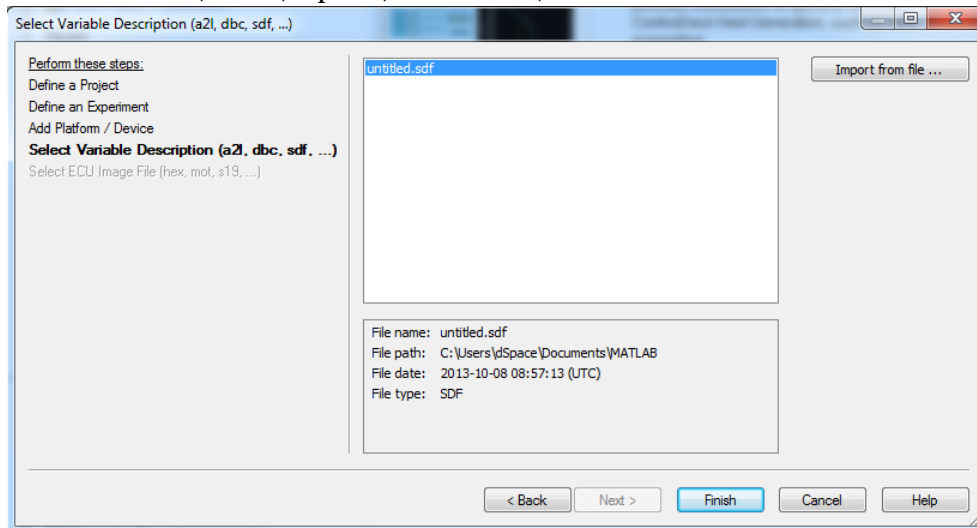
Rys. 17. Okno programu Simulink

2. Zapisać projekt.
3. Skompilować model, Code->C/C++->Build
4. Sprawdzić w oknie głównym MATLABa czy model jest poprawnie skompilowany.
5. W programie ControlDesk tworzymy nowy projekt
  - 5.1. W oknie Add Platform/Device wybrać odpowiednią kartę, jak na rys. 18.



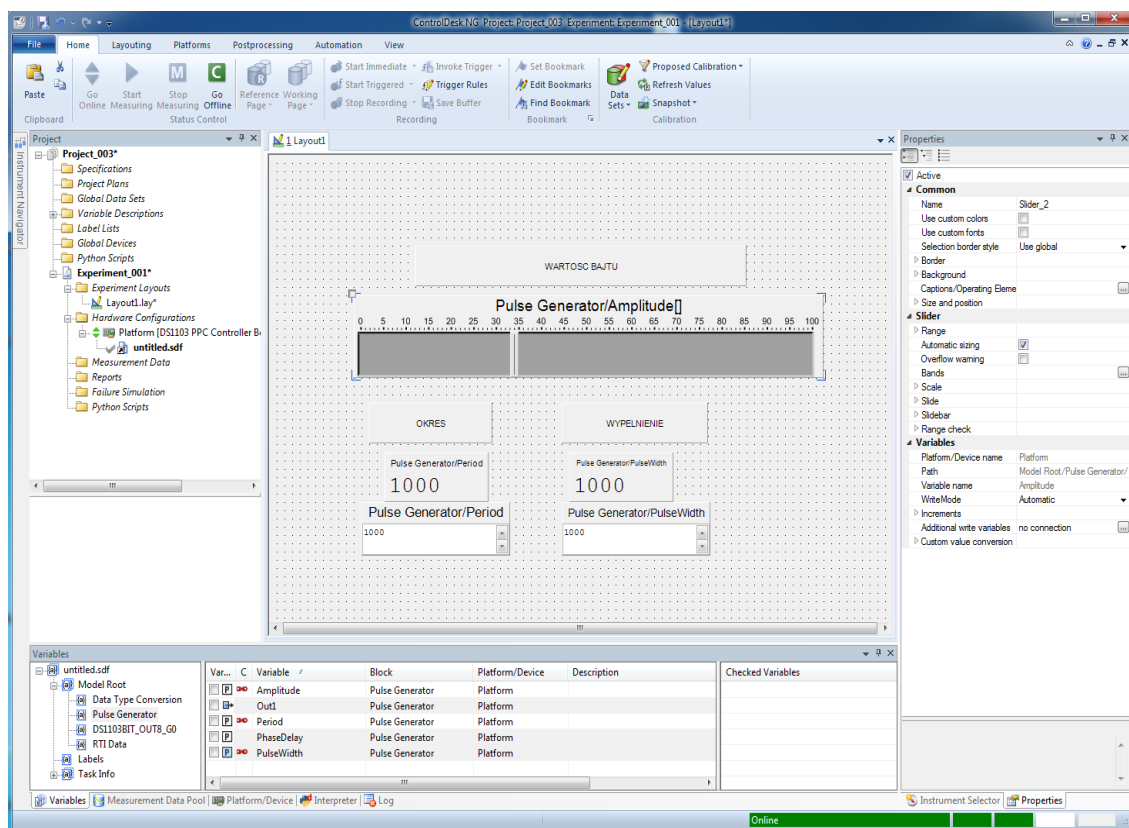
Rys. 18. Okno wyboru urządzenia

5.2. W oknie Select Variable Description zaimportować plik z SIMULINKa o rozszerzeniu \*.sdf, \* - nazwa zapisanego projektu w SIMULINKu. Standardowa ścieżka zapisu plików to:  
C:\Users\dSpace\Documents\MATLAB



Rys. 19. Okno wyboru opisu zmiennych

6. Utworzyć layout jak na rys. 20 z elementów dostępnych w rubryce Instrument Selector po prawej stronie okna.



Rys. 20. Okno projektu w ControlDesk



7. Aby powiązać elementy layoutu ze zmiennymi modelu należy w dolnym oknie Variables odnaleźć zmienną, którą chcemy przypisać do elementu, a następnie przenieść metodą drag and drop na wybrany element layoutu.
8. Przypisać wszystkie zmienne tak jak na rysunku powyżej.
9. Nacisnąć przycisk Go Online na górze po lewej.

## 5. Błąd przy kompilacji w SIMULINKu.

W przypadku wystąpienia błędu „RTI Build Error” przy kompilacji modelu w SIMULINKu należy sprawdzić czy są ustawione właściwe parametry, otwieramy:

w oknie projektu simulink --> Code --> C/C++ Code --> Code Generation Options --> Hardware Implementation

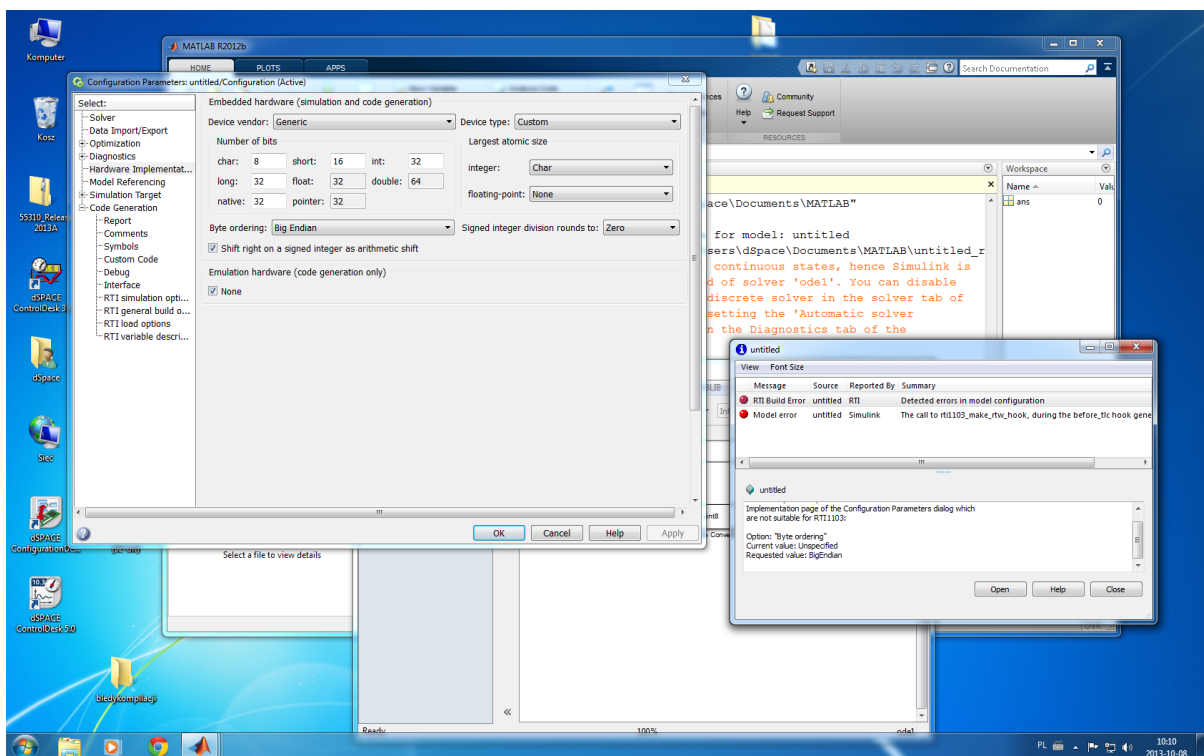
w nim zmieniamy:

Device vendor: Generic

Device type: Custom

Byte ordering: Big Endian

Signed integer division rounds to: Zero



Następnie wciskamy OK i skompilować projekt ponownie.