

## Szybkie prototypowanie z wykorzystaniem systemu modułowego dSPACE DS1005 oraz oprogramowania RTI i ControlDesk

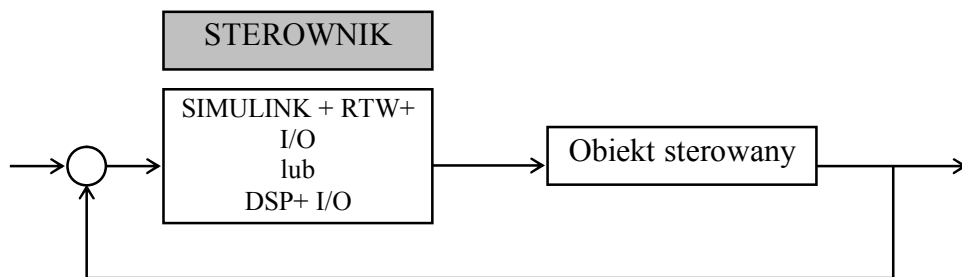
Wiele zalet pakietu MATLAB-SIMULINK sprawia, że prototypowanie sterowników jak i każde inne modelowanie systemów statycznych i dynamicznych jest dużo łatwiejsze dla użytkownika. Należą do nich:

- prostota obsługi pakietu połączona z możliwością przejrzystej graficznej reprezentacji symulowanych procesów
- bezpośrednia interakcja z użytkownikiem realizowana poprzez wybór opcji z menu lub specjalizowany język poleceń
- integracja oprogramowania numerycznego, graficznego i systemowego
- szeroka biblioteka procedur i funkcji pozwalająca projektantowi koncentrować uwagę na istocie modelowanego systemu a nie na metodach numerycznych, których dokładne poznanie wymaga dodatkowego nakładu czasu i pracy
- obiektowa technologia pakietu.

Bardzo istotnym elementem pakietu jest kilka rozszerzeń SIMULINK'a zorientowanych na zastosowanie w projektowaniu systemów sterowania w czasie rzeczywistym. Bardzo przydatnym elementem należącym do rozszerzeń SIMULINK'a jest *C-code Generator*, za pośrednictwem którego dokonywana jest automatyczna translacja S-funkcji SIMULINK'a na równoważne konstrukcje języka C.

Poprzez C-code Generator możliwe jest wykorzystanie prototypów (utworzonych w trybie off-line) w oprogramowaniu bezpośrednio sprzężonym (za pomocą interfejsu pomiarowo-sterującego) ze sterowanym procesem.

Podczas prototypowania sterowników, obiekt połączony jest za pomocą urządzeń I/O z komputerem. W komputerze tym jest symulowane i testowane zachowanie sterownika oraz sprawdzana jest poprawność działających algorytmów (Rys.1).



Rys.1 Obrazowy schemat sterownika prototypowanego w systemie dSPACE.

Model systemu utworzony w SIMULINK'u można przetworzyć na odpowiadający mu kod źródłowy C. Następnie jak każdy inny kod programu w języku C można go skompilować

i zlinkować wraz z dołączonymi przez użytkownika modułami (np. sterownikami kart pomiarowych) w postaci S-funkcji oraz wykonać w czasie rzeczywistym z zadeklarowanym czasem próbkowania.

Istnieje również inna metoda prototypowania sterowników wykorzystująca środowisko MATLAB-SIMULINK. Polega ona na zastosowaniu specjalistycznych kart wykorzystujących procesory sygnałowe DSP, PowerPC i inne oraz kart układów wejścia/wyjścia. Zarówno dla prototypowania poprzez RTW (Real-Time Workshop) jak i poprzez DSP proces projektowania i testowania układu sterowania jest podobny:

- utworzenie schematu obiektu w SIMULINKU
- generacja kodu źródłowego w C, przy wykorzystaniu RTW
- kompilacja i linkowanie oraz utworzenie kodu wynikowego na wybrany procesor.

Różnica pomiędzy tymi dwoma sposobami polega na innym środowisku sprzętowym, w którym wykonywany jest wygenerowany w oparciu o schemat blokowy SIMULINK'a program.

Dla RTW i kart pomiarowych program wykonywany jest w samym komputerze, dla specjalizowanych kart natomiast cała aplikacja jest ładowana i wykonywana na karcie procesora, co daje istotne zwiększenie możliwości systemu. Ponadto zdecydowanie zmniejsza to obciążenie procesora w PC, co z kolei umożliwia generowanie lepszej reprezentacji graficznej procesu.

W tym przypadku komputer jest jedynie platformą do komunikacji z użytkownikiem, komunikacji pomiędzy interfejsami. **Nie pełni zaś roli komputera, na którym realizowane jest sterowanie**. Proces testowania wykonywany jest w całości poprzez kartę procesorową.

W prototypowaniu sterowników karta procesora, połączona za pomocą układów wejścia/wyjścia ze sterowanym obiektem, wykorzystywana jest do symulowania zachowania się sterownika w celu sprawdzenia koncepcji algorytmu sterowania lub doboru nastaw sterownika.

Karty firmy dSPACE mogą być połączone z komputerem poprzez złącze Ethernet lub też montowane bezpośrednio na magistrale AT-bus. Zarówno magistrala jak i Ethernet służą do załadowania programu na kartę oraz do komunikacji między komputerem a kartą procesora. Wszystkie karty komunikują się również pomiędzy sobą poprzez wspólną pamięć.

Firma dSPACE oferuje bogaty zestaw kart z procesorami, pełniącymi rolę "master", jak również kart do obsługi wejścia-wyjścia. Są to karty z przetwornikami AD i DA (o rozdzielczości 12, 14, 16 bitów), karty do obsługi enkoderów obrotowych, karty we-wy cyfrowych, specjalizowane karty do generowania szybkich sygnałów analogowych, zintegrowana karta sterownika (zawiera procesor sygnałowy, cztery wejścia analogowe, cztery wyjścia analogowe, dwa wejścia z enkoderów, we-wy cyfrowe) i wiele innych.

Komunikacja pomiędzy kartą "master" a kartami we-wy i innymi kartami systemu jest realizowana poprzez szybką specjalizowaną 32-bitową magistralę PHS.

Do kart z procesorami jest dołączane oprogramowanie narzędziowe umożliwiające ich łatwą obsługę w czasie prototypowania i w czasie testowania.

Dla środowiska MATLAB-SIMULINK takimi interfejsami programowymi pozwalającymi na łatwe wykorzystanie kart systemu dSPACE są:

- moduł **RTI (Real Time Interface)**.

RTI umożliwia automatyczną implementację w dSPACE aplikacji przygotowanych w SIMULINK'u;

Model SIMULINKOWy, z dołączonymi blokami obsługującymi wykorzystywane karty pomiarowe (z biblioteki RTILIB) jest kompilowany a następnie poprzez RTI „przeładowywany” i uruchamiany na karcie procesora.

RTI umożliwia dokonywanie zmian w symulowanym modelu z poziomu interfejsu graficznego poprzez automatyczne ładowanie wygenerowanego kodu na kartę z procesorem sygnałowym. Umożliwia to dokonywanie zmian nastaw w czasie rzeczywistym.

- moduł **ControlDesk**, pozwalający na tworzenie pulpitu operatorskiego umożliwiającego dostęp i zmianę zmiennych sterownika lub zmiennych symulowanego obiektu w czasie realizacji procesu bez konieczności ponownej generacji kodu oraz monitorowanie i rejestrowanie zmiennych modelu w czasie realizacji procesu. Graficzna prezentacja procesu umożliwia lepsze poznanie jego dynamiki poprzez możliwość obserwacji i modyfikacji on-line.

## I. Opis systemu dSPACE DS1005.

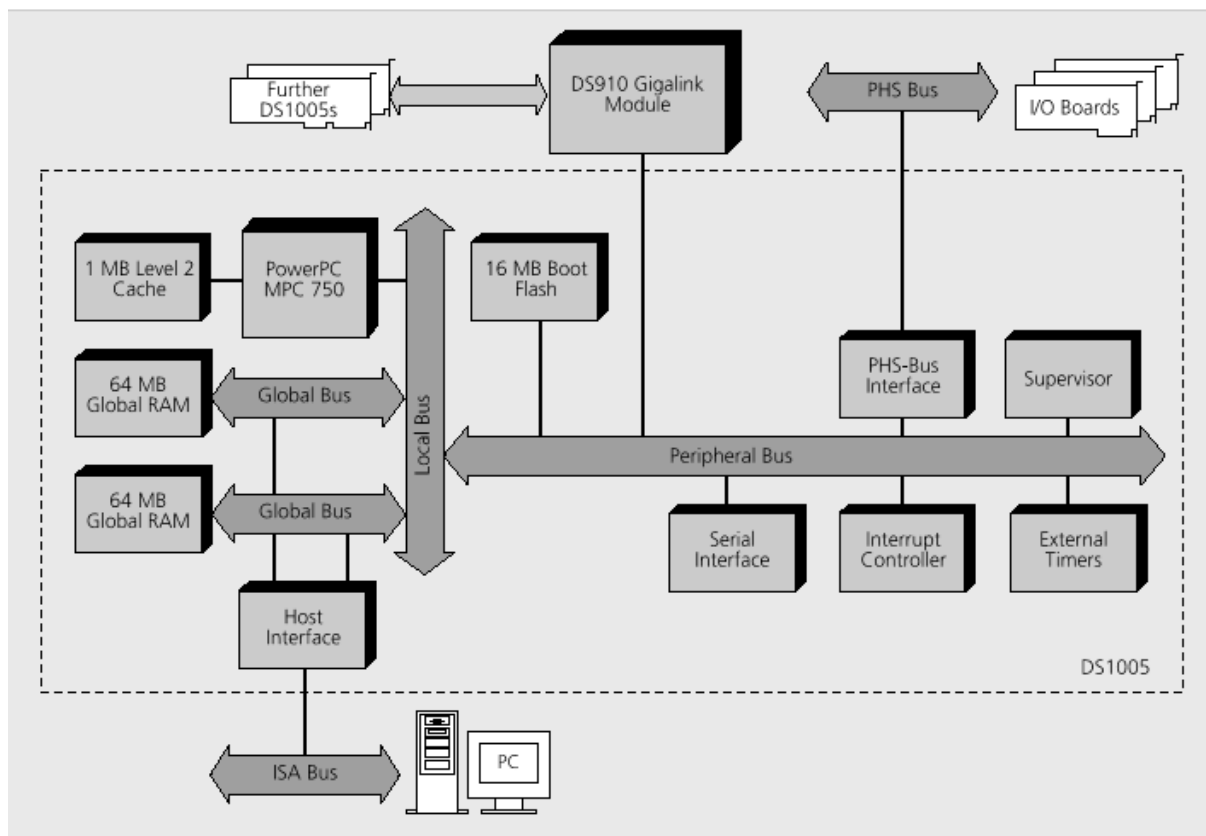
System modułowy *dSPACE* zawiera kartę master *DS1005* (z procesorem PowerPC) oraz różnego rodzaju specjalizowane karty wejścia/wyjścia będące elementem rodziny kart *DSP-CITpro*, zaprojektowanych specjalnie do prototypowania szybkich sterowników cyfrowych i symulatorów pracujących w czasie rzeczywistym, mających zastosowanie w takich dziedzinach techniki jak robotyka, automatyka, przemysł lotniczy, elektronika samochodowa, peryferia komputerowe i wiele innych.

Kompatybilność została zapewniona przez użycie szybkiej, 32-bitowej szyny (*PHS-bus*) wspólnej dla wszystkich kart, umożliwiającej transmisję równoległą z szybkością 26.7 MB/s. Wszystkie karty pomiarowo-sterujące komunikują się z kartą 'master' za pośrednictwem szyny PHS.

Karta DS1005 bazuje na procesorze PowerPC, który stanowi główną jednostkę obliczeniową systemu *dSPACE* zapewniając wystarczającą szybkość przetwarzania nawet dla wymagających aplikacji.



Karta DS1005



Rys.2 Schemat blokowy karty DS1005

Główną jednostką obliczeniową karty jest procesor PowerPC 750.

Procesor umożliwia korzystanie z dużej przestrzeni adresowej z różnymi modelami adresowania. Pozwala też na użycie języków wysokiego poziomu do tworzenia aplikacji.

Oto kilka podstawowych cech tej karty:

Procesor główny:

- IBM PowerPC 750 / 480 MHz
- 32KB cache dla instrukcji i 32KB cache dla danych

Pamięć:

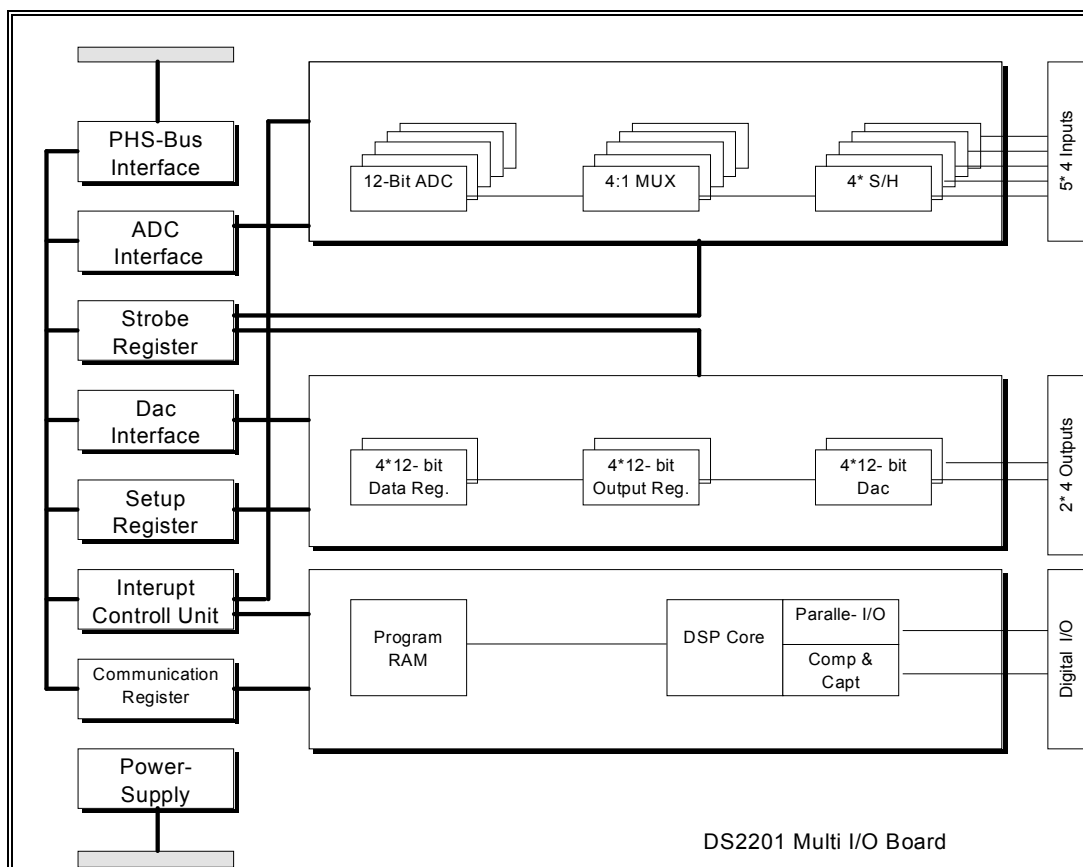
- 16 MB SDRAM (15 MB -aplikacje użytkownika, 1MB –firmware)
- 128 MB SDRAM (dwa bloki po 64MB pamięci)

Możliwość wieloprocesorowej pracy kilku kart DS1005.

## 2. Karta Multi I/O DS2201

Wielofunkcyjna karta pomiarowa DS2201 posiada:

- 20 wejść analogowych
- 8 wyjść analogowych
- 16-to bitowy port równoległy I/O z bitowym wyborem kierunku sterowania
- obwody zabezpieczające na wejściach i wyjściach.



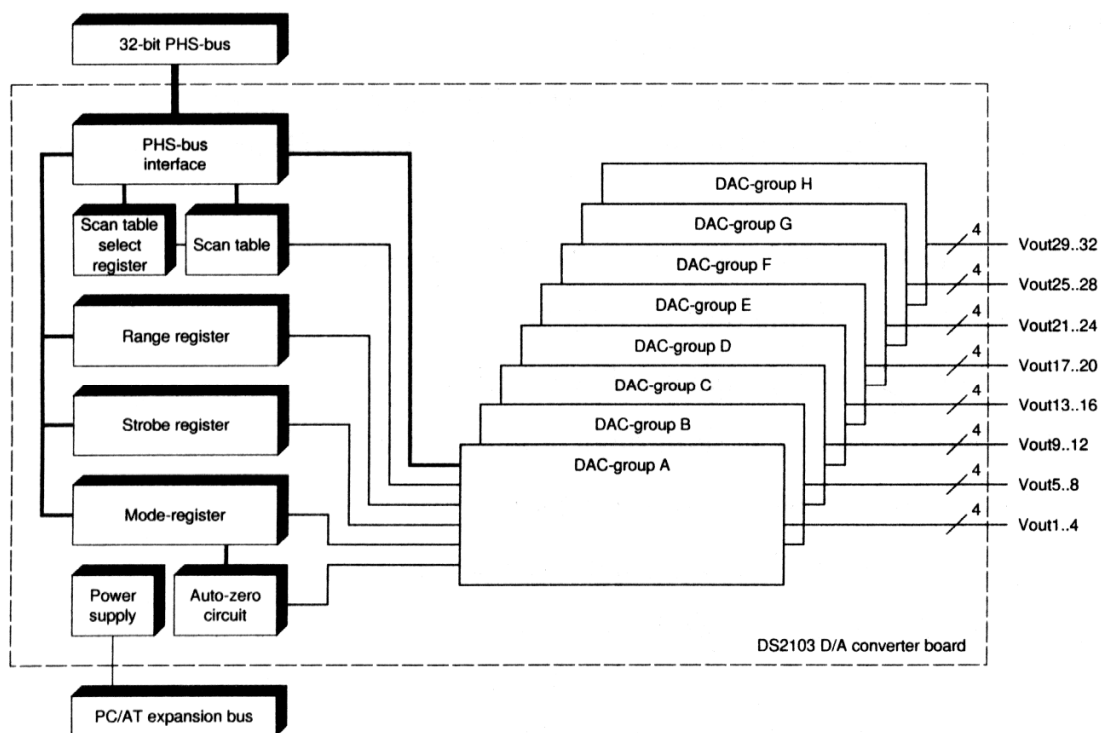
Rys.3 Schemat karty DS2201.

W skład karty wchodzi:

- pięć 12-to bitowych przetworników analogowo- cyfrowych, z których każdy obsługuje (poprzez multiplexer) 4 z 20 wejść analogowych;
- zakres napięć wejściowych  $\pm 10V$ ;
- czas przetwarzania dla każdej liczby kanałów (do 20) wynosi 32.5  $\mu s$
- osiem dwunasto- bitowych przetworników cyfrowo- analogowych obsługujących wyjścia analogowe;
- zakres napięć wyjściowych  $\pm 10V$ ; czas przetwarzania 4  $\mu s$
- podsystem wejść/ wyjść cyfrowych realizowany przez procesor TMS320E14 zapewniający:
- 16 niezależnych linii I/O (każda linia jest zabezpieczona przed pojawieniem się napięcia powyżej 5V),
- 6 kanałów PWM (10-cio bitowa rozdzielczość dla 25 kHz),
- 2 programowalne liczniki.

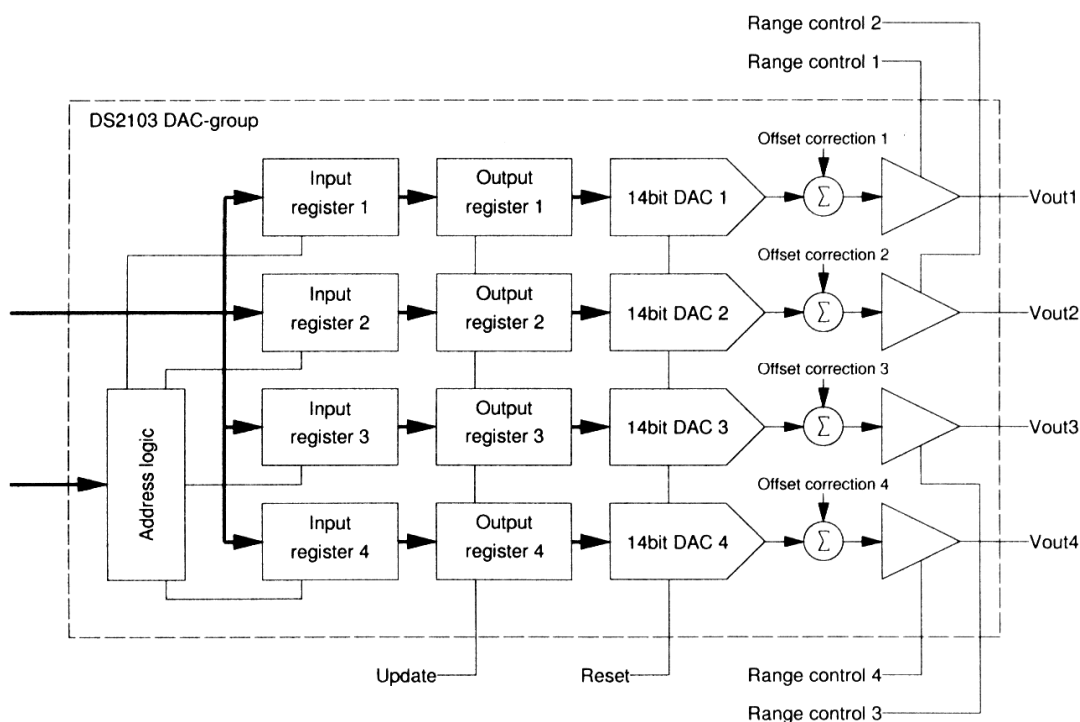
### 3. Karta wyjść analogowych DS2103.

Karta *DS2103* posiada (Rys.4) 32 równoległe kanały wyjściowe z 14 – bitowymi przetwornikami cyfrowo – analogowymi. Umożliwia w sposób programowy zmianę zakresu wyjść analogowych.



Rys.4 Schemat karty *DS2103*.

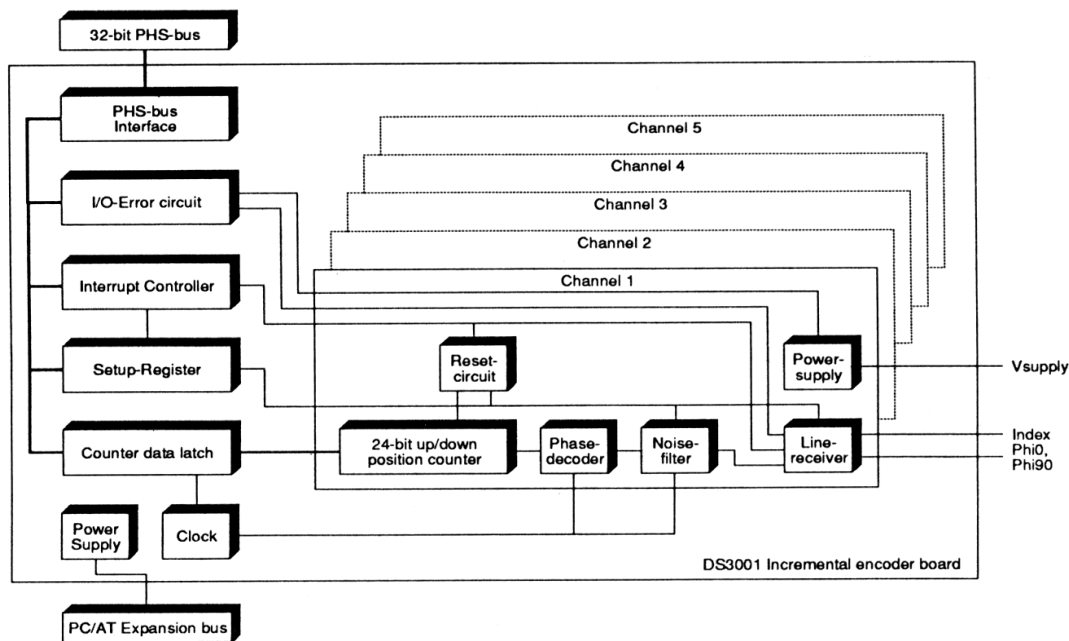
Wyjścia analogowe karty zgrupowane są w 8 grup DAC, z których każda obsługuje 4 kanały (Rys.5). Każdy kanał wyposażony jest w niezależny 14-bitowy przetwornik cyfrowo-analogowy, korekcję przesunięcia oraz wzmacniacz sygnału z możliwością wyboru zakresu wyjściowego  $\pm 5V$  lub  $\pm 10V$ .



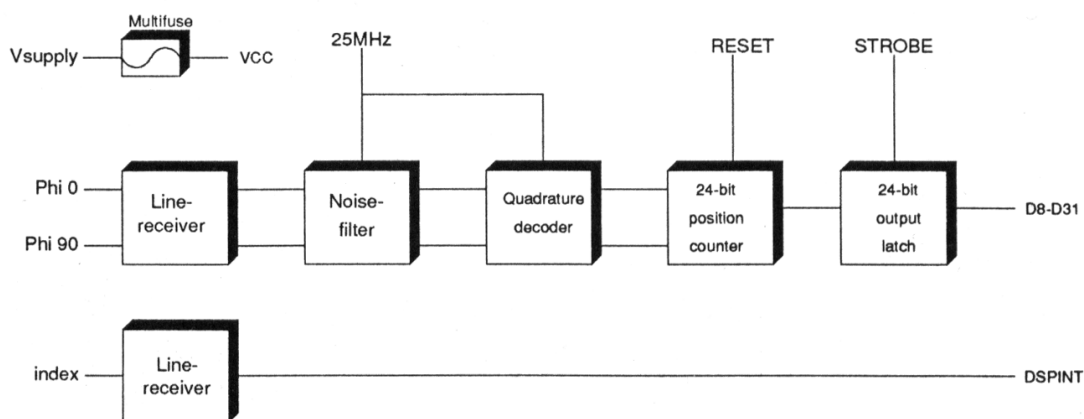
Rys.5 Schemat bloku *DAC – Group A...H* karty *DS2103*.

#### 4. Karta DS3001.

Karta *DS3001 Incremental Encoder* (Rys.6) realizuje zliczanie impulsów z enkoderów. Składa się z pięciu kanałów pomiarowych zawierających: możliwość wyboru rodzaju wejść (wejście różnicowe – RS422 lub poziom TTL), zasilanie enkoderów o napięciu 5V, cyfrowe filtry eliminujące zakłócenia.



Rys.6 Schemat karty *DS3001*.



Rys.7 Schemat kanału pomiarowego karty *DS3001*.

Każdy z pięciu kanałów posiada wejścia z linii enkoderów (*Index, Phi0, Phi90*) oraz wyjście zasilające oznaczone *Vsupply* (Rys.7).

Sygnały pomiarowe z enkoderów wprowadzone poprzez linie odbiorcze są przepuszczane przez filtr cyfrowy eliminujący zakłócenia. Kierunek obrotu wału na którym

umieszczono enkoder rozpoznawany jest w bloku *Quadrature Decoder*. Impulsy zlicza 24-bitowy licznik pozycji. Zliczone impulsy zapisane są w 24-bitowym zatrasku wyjściowym.

Razem z implementacją kart firmy dSPACE i narzędziami generującymi kod, użytkownik systemu otrzymuje pakiet, którego właściwości przyspieszają proces tworzenia i uruchamiania nowych aplikacji. Program użytkownika może być załadowany, monitorowany lub zmieniany w dowolnym momencie podczas pracy DSP.

## II. Programy narzędziowe systemu dSPACE.

Karty z procesorami sygnałowymi firmy dSPACE są dostarczane wraz z oprogramowaniem narzędziowym umożliwiającym ich łatwą obsługę w czasie przygotowywania i przeprowadzania eksperymentu.

Dla środowiska MATLAB/SIMULINK takimi interfejsami programowymi pozwalającymi na łatwe wykorzystanie kart systemu dSPACE są:

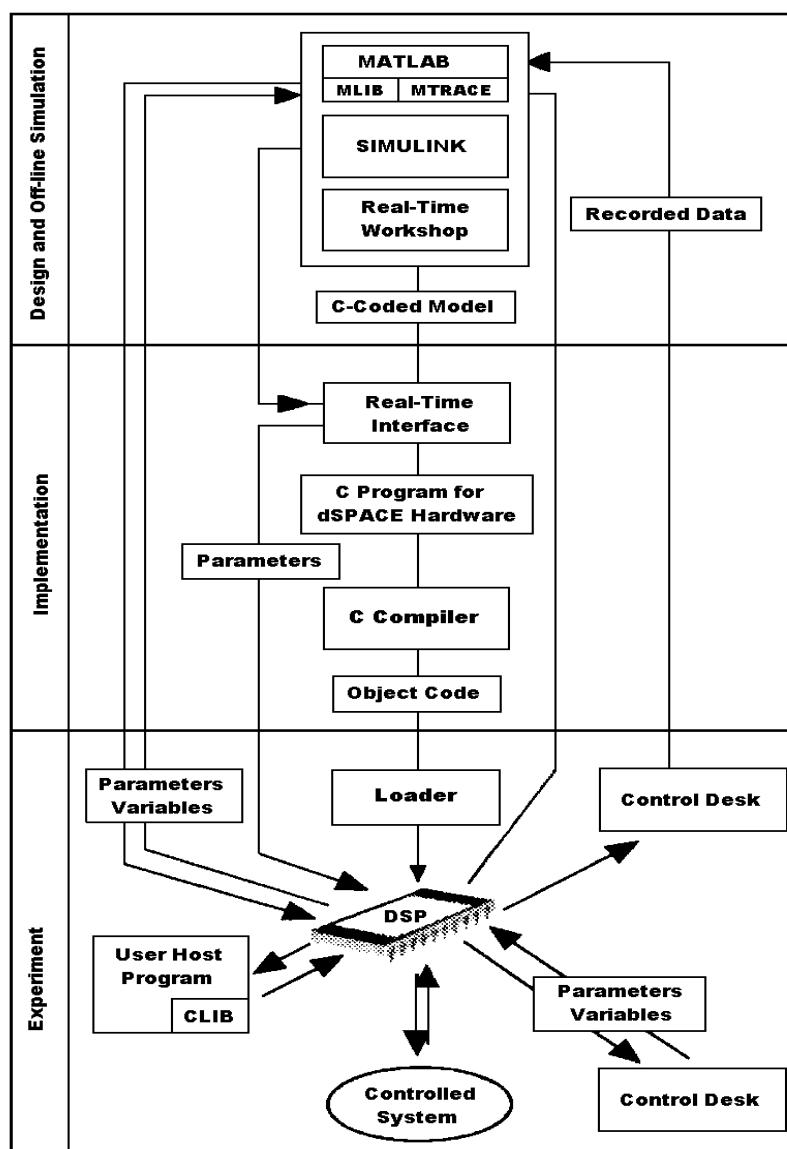
- moduł RTI (Real-Time Interface), który umożliwia automatyczną implementację aplikacji przygotowanych w Simulink-u. Zastosowanie tego oprogramowania polega na przygotowaniu aplikacji za pomocą Simulink-a, z uwzględnieniem modeli stosowanych kart, skompilowaniu wygenerowanego kodu oraz uruchomieniu na karcie procesorowej. Ładowanie wygenerowanego kodu na kartę z procesorem w celu wykonania w czasie rzeczywistym odbywa się automatycznie. Umożliwia z poziomu interfejsu graficznego Simulink-a zmianę parametrów symulowanego modelu. Moduł ten zawiera bibliotekę RTILIB dla Simulink-a zawierającą bloki odpowiadające poszczególnym kartom.
- moduł ControlDesk, zapewniający graficzny interfejs użytkownika, pozwalający w trybie on-line na dostęp i zmianę zmiennych sterownika lub zmiennych symulowanego obiektu w czasie realizacji procesu na karcie, bez konieczności ponownej generacji kodu oraz na monitorowanie i rejestrowanie zmiennych w czasie realizacji procesu

Typowy proces tworzenia i testowania aplikacji w systemie dSPACE składa się z następujących kroków:

1. Przy użyciu Matlab'a, Simulink'a i biblioteki I/O systemu dSPACE dodanej do Simulink'a, projektujemy aplikację i określamy sposób podłączenia wejść i wyjść.
2. Przy użyciu RTW generowany jest z aplikacji kod w języku C i aktywowany RTI, który wykonuje wszystkie niezbędne kroki aby przygotować aplikację do testów w czasie rzeczywistym a następnie ładuje ją do karty procesora .
3. Używając programu ControlDesk można oglądać przebieg zmiennych użytych w aplikacji oraz zmieniać i monitorować parametry aplikacji.

W procesie tworzenia aplikacji wyżej wymienione kroki wykonywane są zazwyczaj wielokrotnie. Na Rys.8 pokazana jest struktura systemu z podziałem na warstwy. Zamieszczone są tam wszystkie narzędzia wykorzystywane w procesie tworzenia aplikacji, oraz widać jak współpracują ze sobą. Dzięki RTI graficzny model aplikacji utworzony w Simulink-u jest automatycznie przenoszony do systemu dSPACE przez co użytkownik nie musi się troszczyć o szczegóły implementacji.



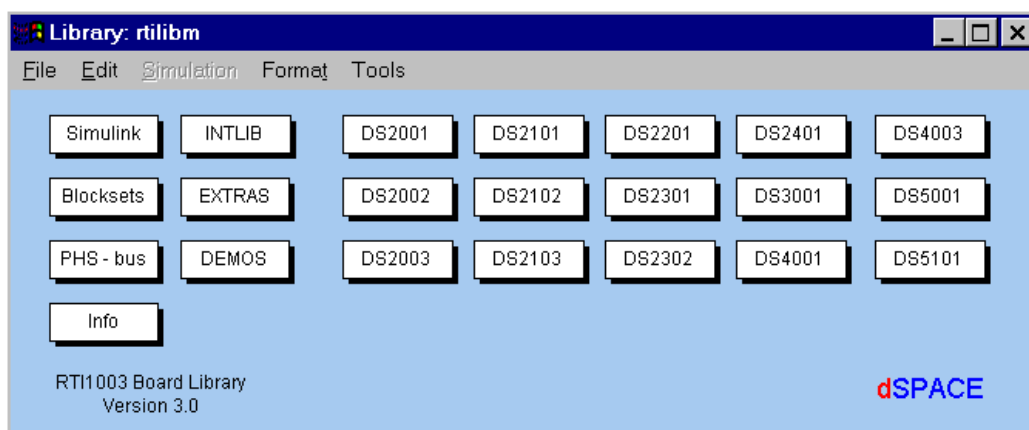


Rys.8 Proces tworzenia i testowania aplikacji.

## 1. Real-Time Interface (RTI).

Interface czasu rzeczywistego (RTI) łączy pakiety Matlab, Simulink i Real Time WorkShop (RTW) z systemem czasu rzeczywistego dSPACE, tworząc zintegrowane i gotowe do użycia środowisko przeznaczone do tworzenia aplikacji pracujących w czasie rzeczywistym. Przeprowadza także automatyczną i jednolitą implementację graficznego modelu sterownika w Simulink'u na procesorze sygnałowym systemu dSPACE.

Ponadto, RTI dodaje do Simulink'a bibliotekę RTILIB zawierającą bloki sterowników do kart pomiarowych I/O. Sterowniki te, tak jak wszystkie bloki w Simulink'u, mają postać ikon łączonych z innymi blokami za pomocą myszy.



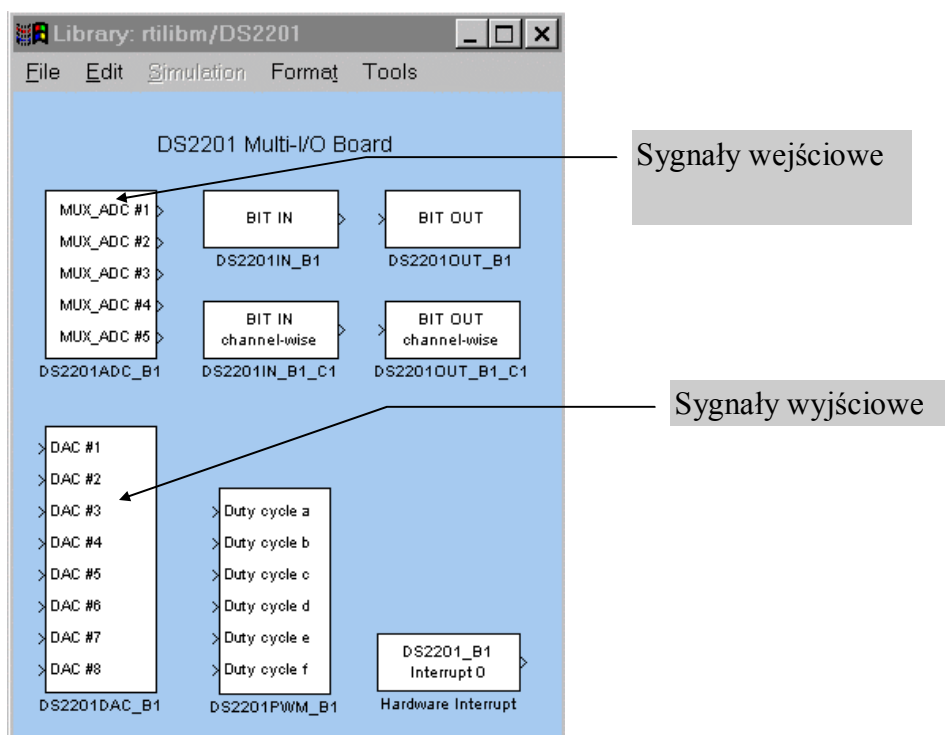
Rys.9 Biblioteka *rtlib* z podbibliotekami kart pomiarowych.

Wszystkie bloki dSPACE, które czytają sygnały z urządzeń wejściowych, automatycznie skalują swoje sygnały wyjściowe na liczby z zakresu  $\pm 1.0$ . Analogicznie, wszystkie bloki dSPACE, które wysyłają sygnały do urządzeń wyjściowych, oczekują na swych wejściach sygnałów z zakresu  $\pm 1.0$ .

Z powyższego wynika, że wartości wszystkich sygnałów aplikacji pobieranych z ikon reprezentujących urządzenia wejściowe dSPACE (np. ADC) będą leżeć w podanym zakresie.

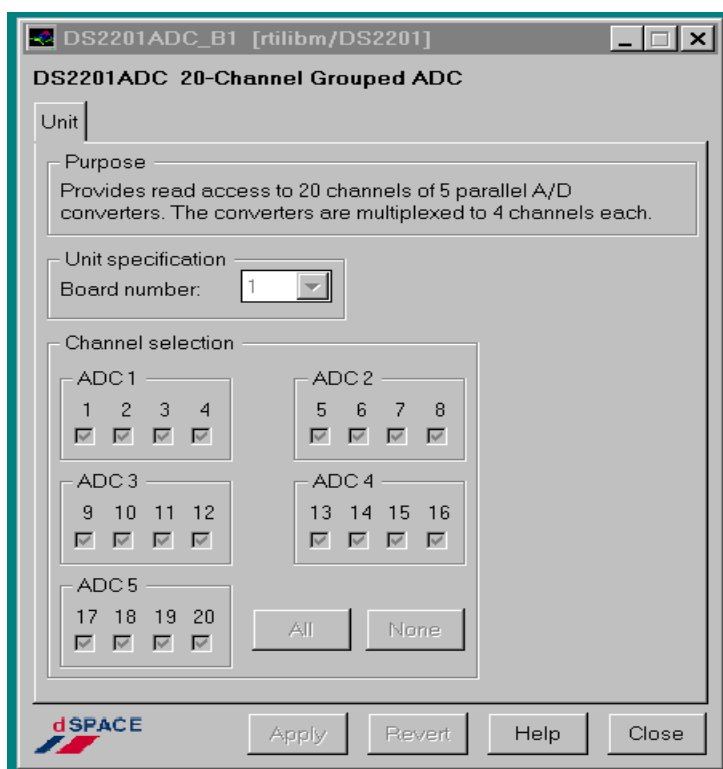
Pociąga to jednak za sobą konieczność wyskalowania wszystkich sygnałów aplikacji, które są podłączone do ikon reprezentujących urządzenia wyjściowe dSPACE (np. blok DAC), w taki sposób, aby maksymalny zakres wartości był odwzorowany w zakresie  $\pm 1.0$ . W przypadku takiego wyskalowania rozdzielczość przetwornika C/A jest w pełni wykorzystana.

Każda z podbibliotek zawiera ikony reprezentujące funkcje pomiarowe i sterujące poszczególnych kart.

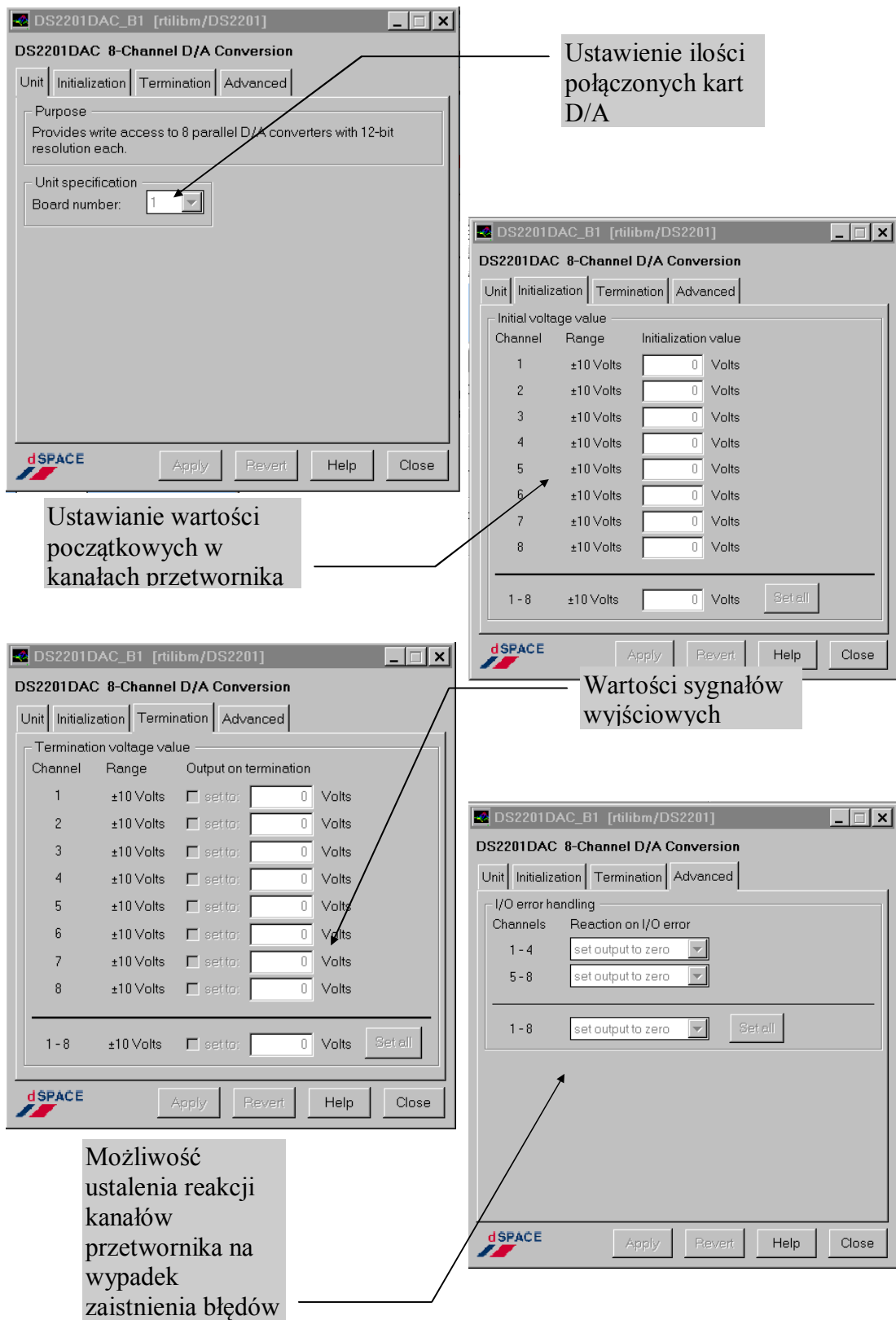


Rys.10 Podbiblioteka dla karty DS2201

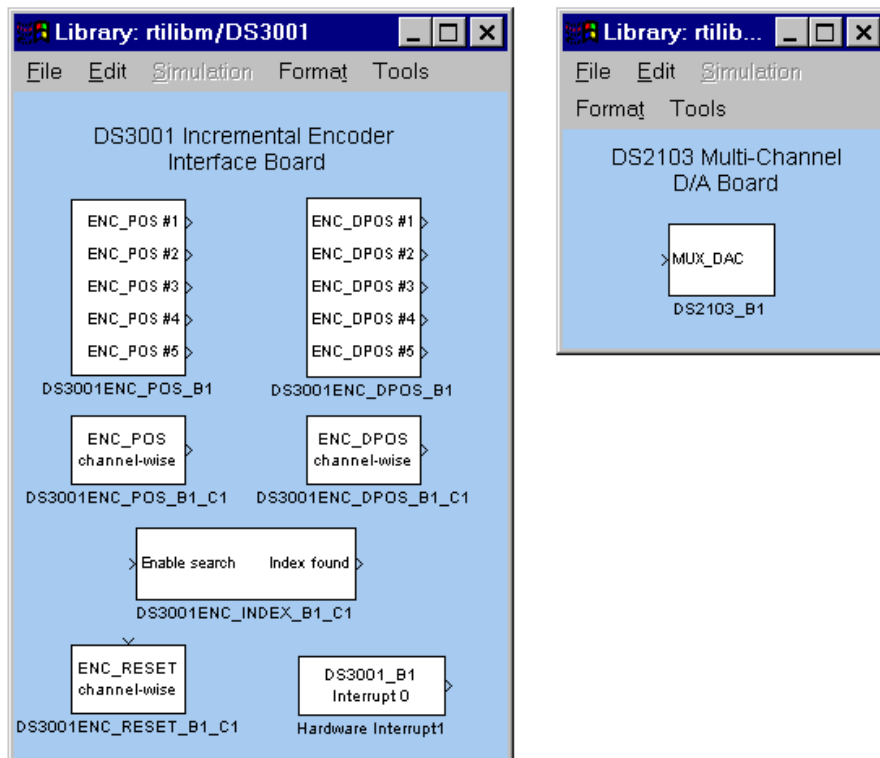
Po umieszczeniu w tworzonym schemacie ikony reprezentującej wybraną funkcję karty możliwe jest ustawienie parametrów w polu dialogowym związanym z danym blokiem.



Rys.11 Pole dialogowe dla operacji AD karty DS2201



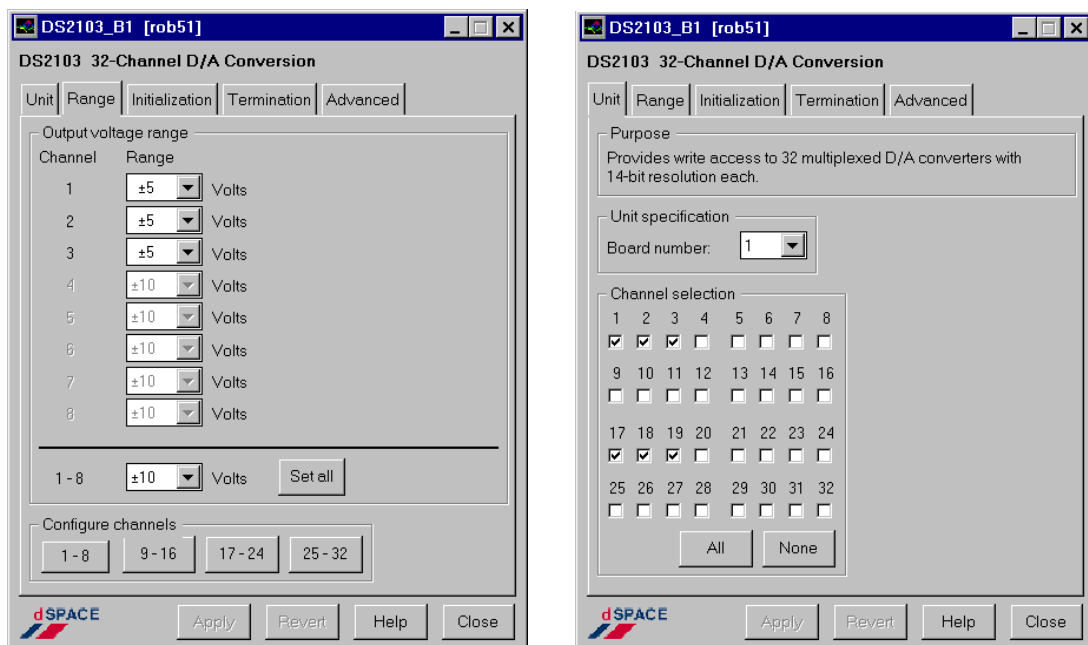
Rys.12 Pole dialogowe operacji DA karty DS2201-poszczególne strony

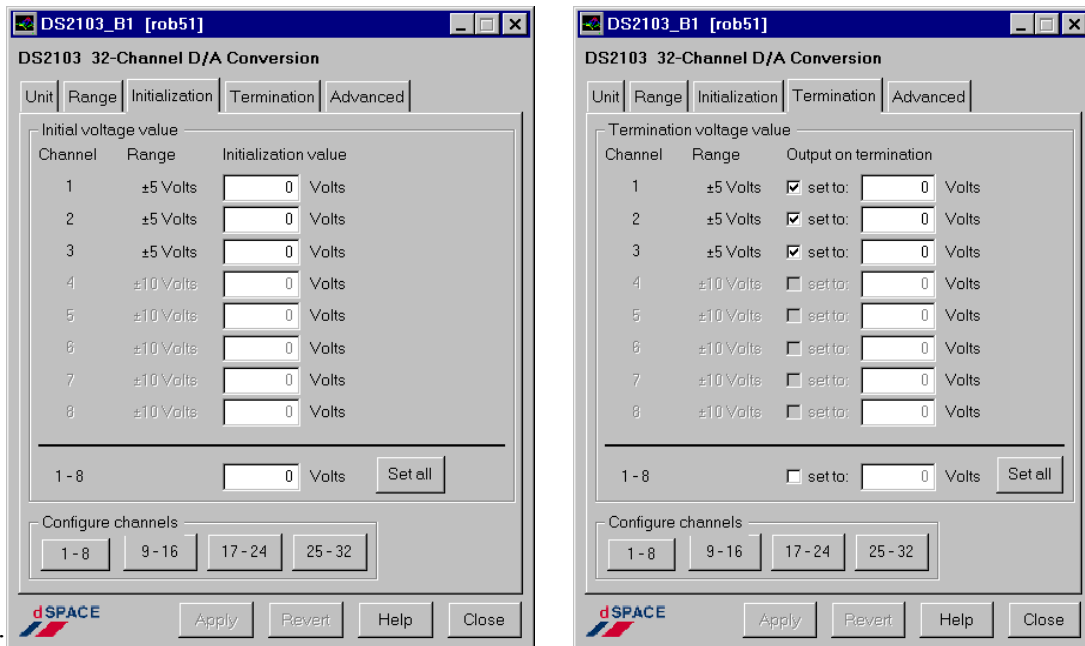


Rys.13 Podbiblioteki dla kart DS2103 i DS3001

Po wstawieniu w *Simulink*'u bloku reprezentującego kartę *DS2103* należy ustawić parametry karty.

Okna dialogowe bloku dla karty *DS2103*.





Rys.14 Okno dialogowe karty *DS2103* – poszczególne strony.

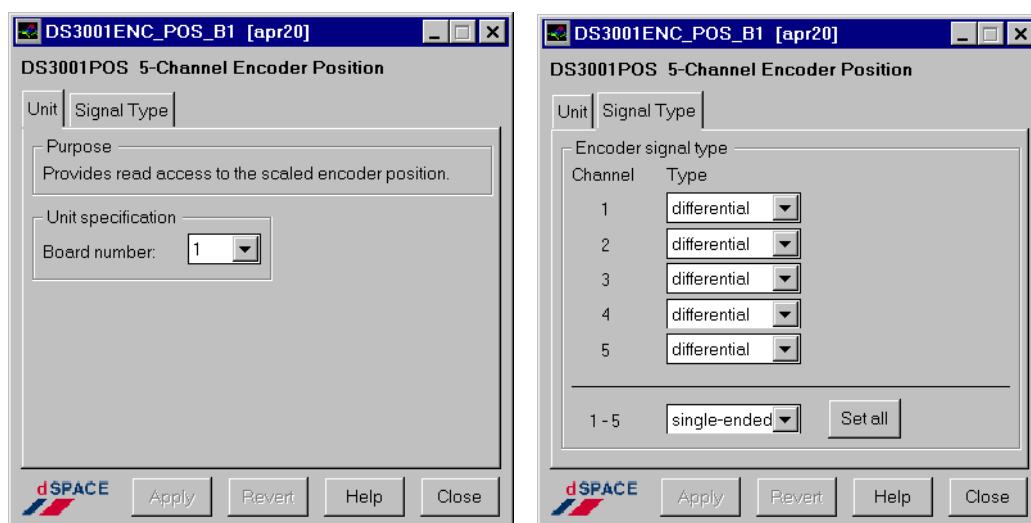
W oknie tym możliwy jest

-wybór kanałów przetwornika C/A strona *Unit*,

-zakres napięciowy wyjść karty ( $\pm 5$  [V] lub  $\pm 10$  [V]) – strona *Range*,

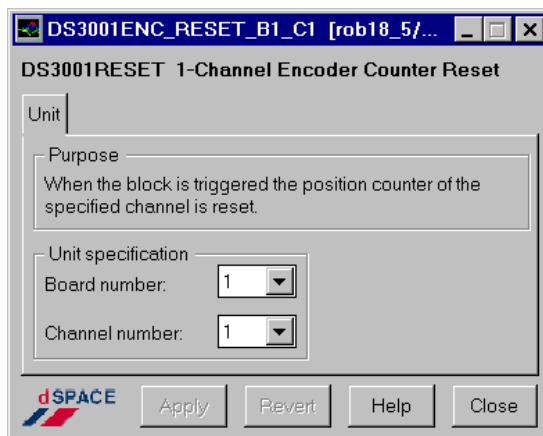
oraz reakcje na zaistniałe zdarzenia w systemie – strony *Initialization*, *Termination* i *Advanced*.

Podobnie dla bloku reprezentującego kartę *DS3001* należy przeprowadzić ustawienia dotyczące numeru zainstalowanej karty oraz typu sygnału odczytywanego z enkodera, klikając podwójnie na ikoncie. Po wykonaniu tej czynności pojawi się okno dialogowe przedstawione na Rys.15.



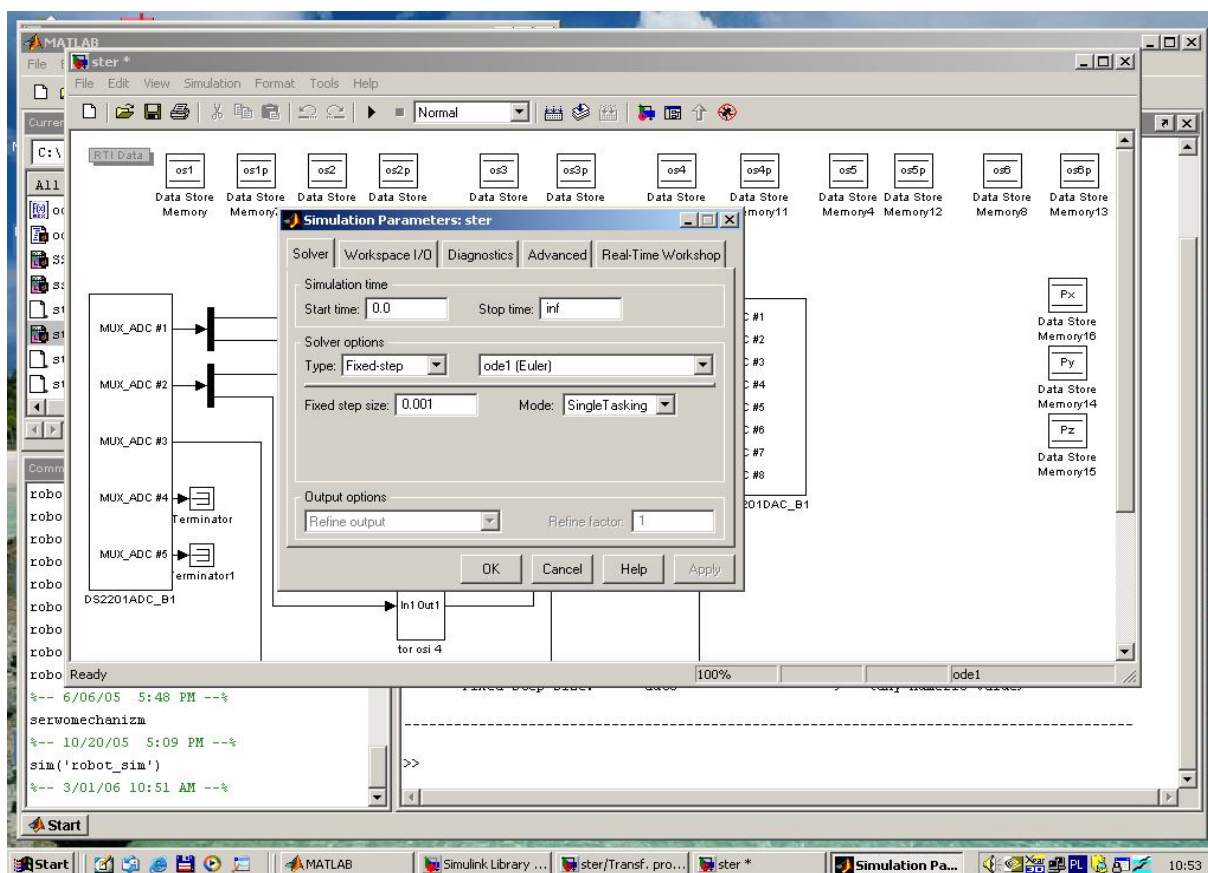
Rys.15 Okno dialogowe karty *DS3001*.

Możliwość zerowania licznika enkodera daje blok *DS3001ENC\_RESET*. W oknie tym wybiera się kanał enkodera (kanały 1-5), którego licznik pozycji ma być zerowany (Rys.16).



Rys.16 Okno dialogowe bloku resetowania licznika enkodera.

Aby graficzny model stworzony w Simulink'u został załadowany do karty DS1005, należy otworzyć okno dialogowe RTW Options z menu Tools (Rys.17), które umożliwia wprowadzanie parametrów niezbędnych do utworzenia wykonywalnego programu. Ukaże się okno dialogowe składające się ze stron opisanych poniżej.



Rys.17 Okno dialogowe RTW Options – strona Solver.

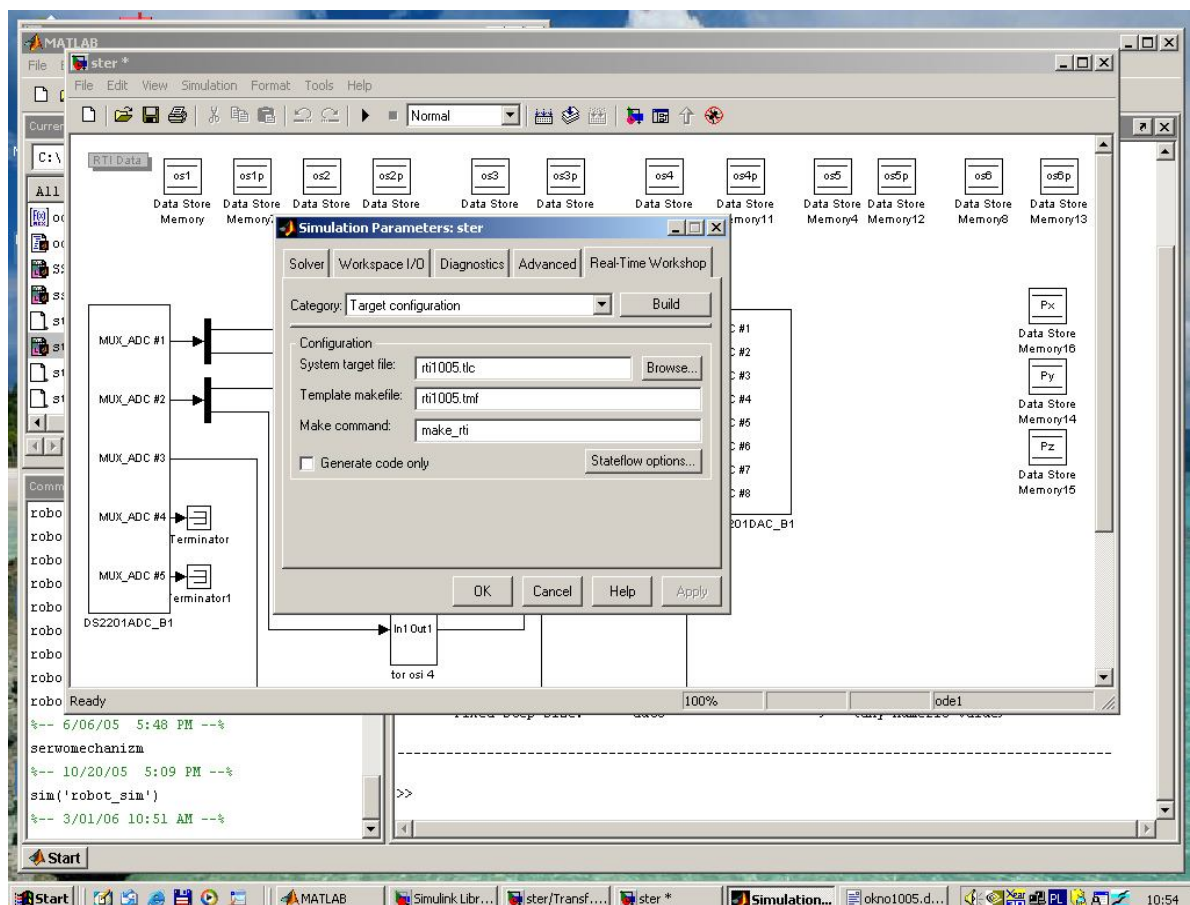
## 1. Strona Solver (Rys.17).

Strona ta zawiera opcje wyboru dotyczące czasu symulacji, metody całkowania i kroku całkowania.

Pole *Simulation Time* umożliwia wprowadzenie czasu rozpoczęcia symulacji – okno *Start Time* (Czas ten odnosi się tylko do symulacji w trybie off – line) jak i czas zatrzymania symulacji – okno *Stop Time*. W polu *Solver options* wybrać można zmienny lub stały krok całkowania oraz metodę całkowania numerycznego w oknie *Type*. W polu *Fixed Step Size* podaje się krok całkowania.

Należy pamiętać, aby dla aplikacji pracujących w czasie rzeczywistym wszystkie metody całkowania pracowały ze stałym krokiem. Rozwiązanie takie gwarantuje ukończenie wykonywania każdego kroku w ustalonym przedziale czasu.

## 2. Strona RTW – parametry Real Time Workshop.



Rys.18 Okno dialogowe RTW Options – strona RTW.

Pole *System target file* (plik *rti1005.tlc*) określa docelowe środowisko dla którego wykonywana jest generacja kodu (w przypadku systemu dSPACE jest to karta DS1005).

W polu *Template Makefile* należy podać nazwę wzorcowego pliku RTI - *rti1005.tmf*, według którego model zakodowany w języku C jest automatycznie implementowany w systemie dSPACE (nadzoruje proces kompilacji i linkowania programu).

Pole *Make Command* zawiera wywołanie Matlab'owskiego programu *make\_rti* sprawującego całkowitą kontrolę nad procesem budowania i ładowania programu.



Proces budowania i ładowania programu inicjowany jest przez naciśnięcie przycisku *Build* na stronie RTW okna RTW Options i przebiega następująco:

1. dla graficznego modelu stworzonego w Simulink'u, Real-Time WorkShop generuje kod w języku *C* w postaci plików *model.c* i *model.h*

2. wykonywany jest program *make\_rti*, który bazując na wzorcowym pliku RTI tworzy plik roboczy aplikacji zawierający wszystkie komendy niezbędne do zbudowania i załadowania programu.

3. wykonywany jest plik roboczy aplikacji, który wywołuje kolejno:

- preprocesor RTI, który modyfikuje kod *C* oraz generuje wywołania funkcji I/O dla implementacji w systemie dSPACE.
- kompilator/Linker dla procesora PowerPC, który kompiluje model i środowisko czasu rzeczywistego dSPACE, a następnie łączy pliki obiektowe i biblioteki w program wykonywalny.
- program ładujący dSPACE, który ładuje program do karty procesora i rozpoczyna jego wykonywanie
- program kontrolny, który sprawdza czy aplikacja pracuje poprawnie.

## 2. ControlDesk

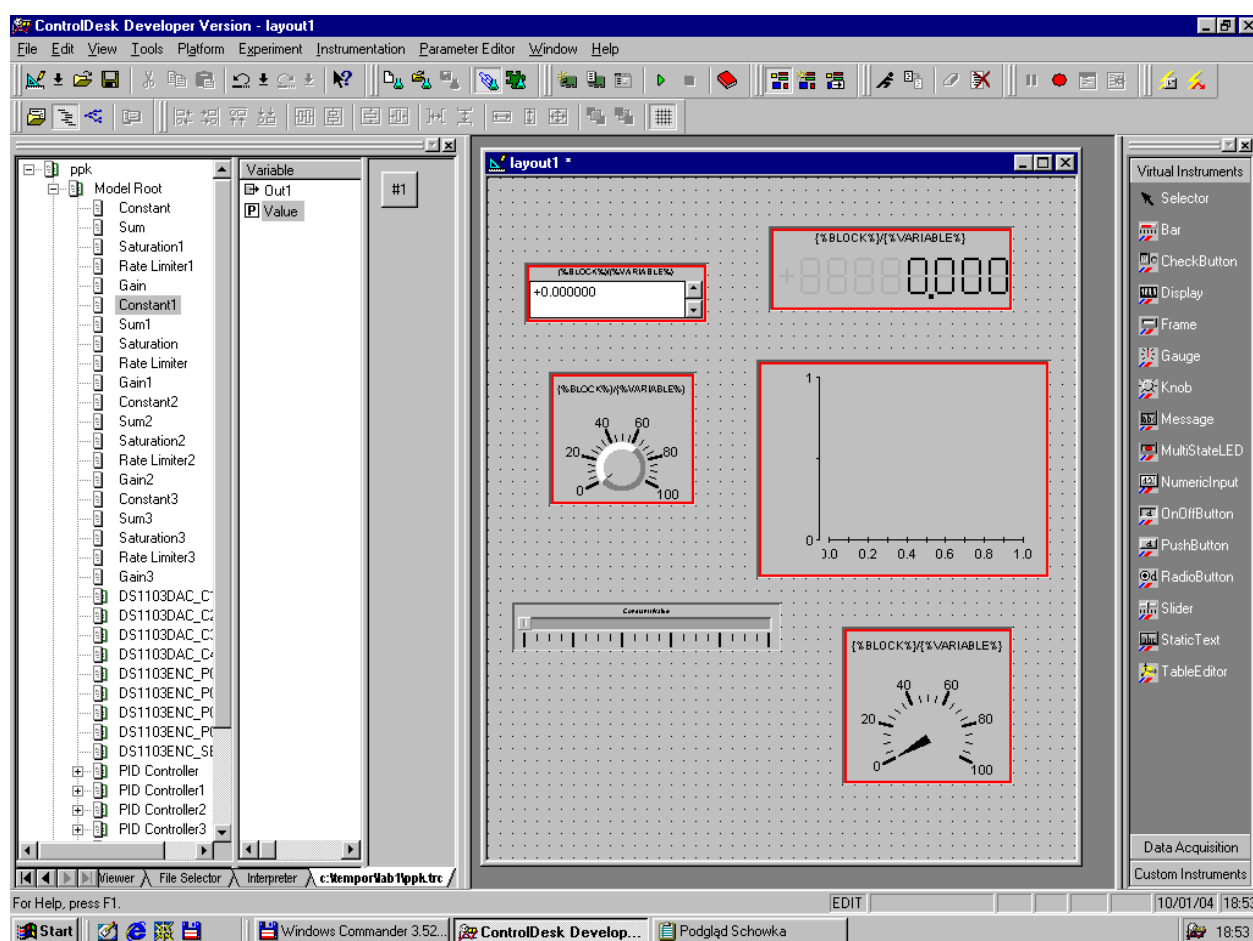
Oprogramowanie ControlDesk umożliwia zmiany i monitorowanie parametrów dzięki tworzeniu wirtualnych paneli sterowniczych. Możliwa jest także rejestracja przebiegów zmiennych. Panele mogą składać się z różnorodnych instrumentów wyświetlanych na ekranie monitora komputera, imitujących swym wyglądem tradycyjne instrumenty. Są to różnego rodzaju przyciski, wyświetlacze, wskaźniki, pola wczytowe, pola komunikatów, pola tekstowe, kontrolki, suwaki, edytor zmiennych tablicowych (macierzowych). Przebieg zmian parametrów i zmiennych modelu można obserwować dzięki ploterom, przedstawiającym przebieg w czasie lub w zależności od innej zmiennej. Dostępny jest również analizator stanów logicznych. Zmienne obserwowane na ploterach można rejestrować w postaci plików danych MATLAB'a „.mat”. W celu ograniczenia ilości danych rejestrować można próbki co pewną zadaną ilość okresów próbkowania. Rejestracja może być uruchamiana przez użytkownika bądź wyzwana poprzez wartość wybranego sygnału występującego w aplikacji.

Tworzenie pulpitu polega na wybraniu odpowiedniego instrumentu poprzez kliknięcie na ikonie symbolizującej dany instrument i zaznaczeniu na tworzonym pulpicie obszaru, które powinien zajmować. Utworzony w ten sposób instrument posiada standardowe właściwości, które można zmieniać poprzez wywołanie okienka właściwości dwukrotnym kliknięciem myszką w obszarze elementu. Okno to zawiera informacje zależne od rodzaju instrumentu. Standardowo są to informacje o wyglądzie (opisy, tło, ramki) oraz wartościach zmiennych odpowiadających poszczególnym stanom elementu (np. przycisk wciśnięty) lub dopuszczalnych zakresach zmiennych (np. dla pól wczytowych). Inne są właściwości dla ploterów (np. rodzaj linii) czy kontrollek (przypisanie kolorów wartościom sygnału). Możliwe jest utworzenie kilku pulpituów odpowiadających np. różnym trybom pracy sterownika. Kilka pulpituów można używać tworząc tzw. eksperyment łączący graficzny interfejs oraz przypisaną mu aplikację. Wówczas aktywny jest tylko jeden pulpit co ogranicza ilość instrumentów, których stan jest śledzony przez operatora. Ułatwia to pracę operatora i zmniejsza ryzyko pomyłki (np. naciśnięcia nieodpowiedniego przycisku). Po utworzeniu pulpitu sterującego należy wczytać plik zawierający informacje o zmiennych i parametrach występujących w modelu. Są one zawarte w pliku „nazwa\_aplikacji.trc” wygenerowanym podczas tworzenia aplikacji. Nazwy zmiennych i parametrów są widoczne w odpowiedniej

zakładce okienka narzędzi programu ControlDesk (pogrupowane według podsystemów tworzonych w SIMULINK'u), skąd stosując technikę przeciągania możemy je przypisywać odpowiednim instrumentom pulpitu. Istnieje podział na zmienne będące parametrami bloków SIMULINK'a i zmienne reprezentujące wartości sygnału na wyjściu bloku. Parametry bloków posiadają nazwy rozpoczynające się od „P:” natomiast sygnały wyjściowe bloków oznaczone są jako „Out:”.

Po zakończeniu tworzenia pulpitu można załadować aplikację do karty procesora DSP i przejść do trybu pracy aplikacji w celu przetestowania działania aplikacji i pulpitu sterującego. Dostępna jest opcja zatrzymywania i uruchamiania aplikacji w procesorze DSP. Podczas pracy aplikacji można przejść w tryb edycji w celu dokonania korekt w działaniu pulpitu.

Wygląd okna programu ControlDesk przedstawiono na rysunku poniżej.



Bardzo przydatnym uzupełnieniem interfejsu programowego dla systemu dSPACE są oferowane przez firmę moduły *MLIB* i *MTRACE*.

Moduł *MLIB* (MATLAB-DSP Interface Library) umożliwia bezpośredni dostęp z poziomu *Matlab*-a do pamięci karty DSP, a tym samym pozwala zmieniać parametry jak również wpływać na operacje wykonywane przez kartę. Daje on użytkownikowi możliwość automatyzacji procesu prototypowania. Poprzez napisany M-file użytkownik może przysyłać do karty np. sekwencje testów dla określonych odcinków czasu, odczytywać dane, przetwarzać je, obliczać nowe parametry sterownika (wykorzystując dostępne procedury) i ponownie przysyłać do karty. Jest więc bardzo pomocny w procesie automatycznego strojenia parametrów sterownika.

Moduł *MTRACE* (MATLAB -TRACE), stanowiący alternatywę lub uzupełnienie dla programu, jest narzędziem do zbierania danych w czasie rzeczywistym i ich graficznego przedstawiania. Podobnie do *MLIB*-a pozwala na automatyzację procesu zbierania danych bezpośrednio z poziomu *Matlab*-a. Poprzez napisany M-file można zaprogramować jakie zmienne rejestrować, jak często, w jakim odcinku czasu itp. Jest on szczególnie przydatny przy zbieraniu danych z procesu prototypowania na długim odcinku czasu. Oba moduły uzupełniają się. *ControlDesk* jest gotowym, łatwym do użycia interfejsem graficznym, natomiast *MTRACE* jest bardziej uniwersalnym w procesie automatyzacji eksperymentu, ale wymaga napisania m-fila.

*Kraków 2013*