



Initiation au système dSPACE,
de l'interface RTI dans Simulink
et de ControlDesk Next Generation 5.4
version 2.0

Abdallah DARKAWI

POLYTECH NANTES
Département Génie Électrique
Site de Gavy Océanis - BP 152
44603 SAINT NAZAIRE CEDEX
<http://web.polytech.univ-nantes.fr/>

1. Configuration minimale requise

La configuration minimale requise pour utiliser le système dSPACE est :

Processeur : Pentium 4 at 2 GHz (ou équivalent)

Mémoire vive : 2 GB RAM

Espace disque : 5.5 GB

Pour la licence (Dongle License) : un port USB pour installer le Dongle

Un slot : pour installer la DS1104, il faut un slot de libre dans la carte mère du PC (33 MHz/32-bit 5 V PCI slot).

2. Carte dSPACE

Le pilotage en temps réel des systèmes continus se fait à l'aide d'un PC connecté à la carte dSPACE DS1104. La programmation se fait à l'aide de l'outil de modélisation SIMULINK, qui aide à poser le problème d'une manière graphique en utilisant les blocs interconnectés. En fait, beaucoup de systèmes de développement en temps réel à base de DSP viennent maintenant avec une interface à Simulink par laquelle ils peuvent convertir les blocs de Simulink en un code machine qui peut être exécuté sur un système à base de DSP. Ceci réduit considérablement le temps de développement et de prototypage pour le control des systèmes. Le prototypage passe alors par trois étapes :

1. Construction du système de commande en utilisant les blocs de Simulink
2. Simulation du système pour voir les résultats dans différents scénarios.
3. Exécution du modèle en temps réel à travers la carte DS1104.

Le processeur principal est un **MPC8240**, avec une horloge de 250MHz. Les caractéristiques de la carte sont données dans le tableau de la figure 1 (document constructeur).

Parameter		Specification
Processor		<ul style="list-style-type: none"> ■ MPC8240 processor with PPC 603e core and on-chip peripherals ■ 64-bit floating-point processor ■ CPU clock: 250 MHz ■ 2 x 16 KB cache, on-chip ■ On-chip PCI bridge (33 MHz)
Memory	Global memory	■ 32 MB SDRAM
	Flash memory	■ 8 MB
Timer	4 general-purpose timers	<ul style="list-style-type: none"> ■ 32-bit down counter ■ Reload by hardware ■ 80-ns resolution
	1 sampling rate timer (decrementer)	<ul style="list-style-type: none"> ■ 32-bit down counter ■ Reload by software ■ 40-ns resolution
	1 time base counter	<ul style="list-style-type: none"> ■ 64-bit up counter ■ 40-ns resolution

Figure 1. Principales caractéristiques du DS1104

La carte DS1104 (Master PPC) dispose de 8 convertisseurs analogiques numériques ADC dont la tension d'entrée est comprise entre -10V et +10V; et 8 convertisseurs numériques analogiques DAC dont la tension délivrée est comprise entre -10V et +10V. Il dispose également de plusieurs interfaces notamment des entrées sorties numériques, des codeurs incrémentaux, etc...(voir figure 2). La carte DS1104 dispose également d'un DSP esclave, le **TMS320F240 DSP** qui sera utilisé pour générer les signaux PWM. Pour plus de détails sur la carte DS1104 vous pouvez consulter la documentation en ligne en cliquant sur le lien suivant : http://www.sport-transfer.eu/media/blfa_files/wymiary_boiska.pdf

Remarque : les signaux PWM générés par le DSPACE sont des signaux TTL de niveau de tension compris entre 0 et 5V. Les onduleurs pédagogiques SEMIKRON sont dotés des cartes drivers permettant de piloter les IGBT des bras de pont.

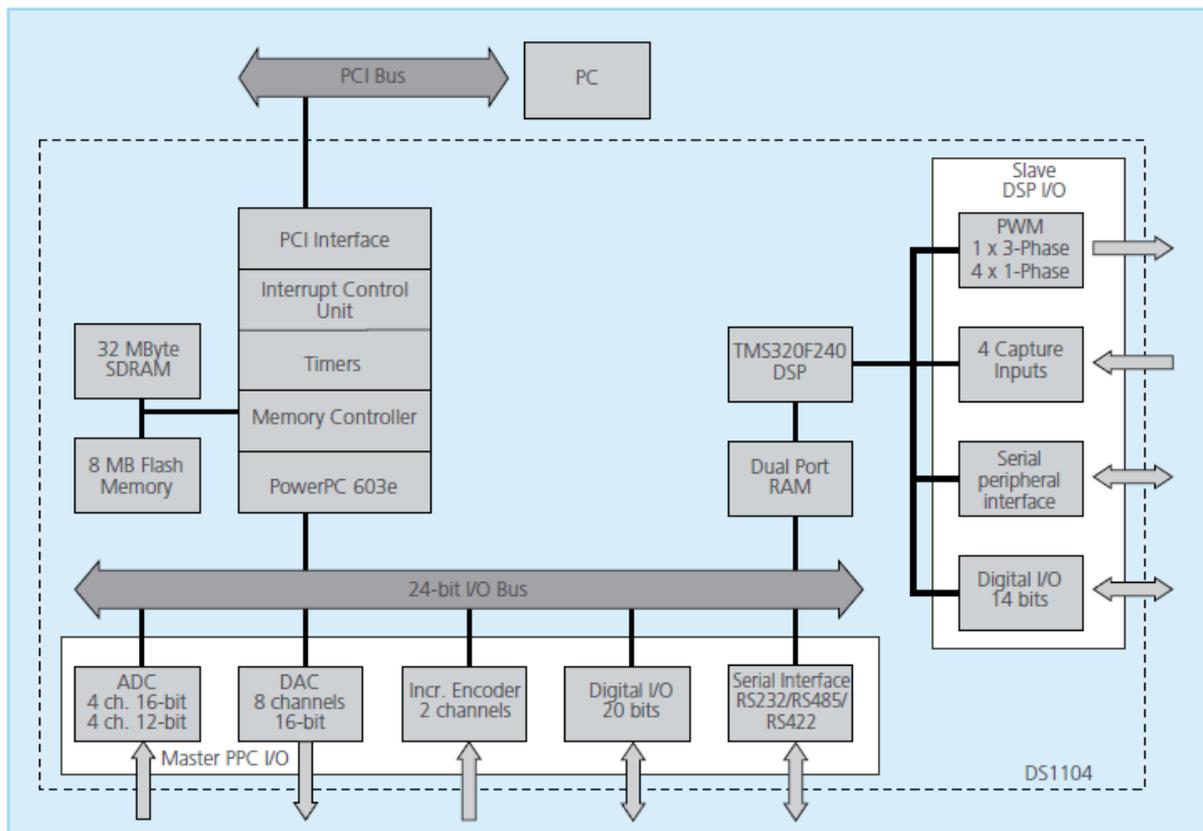


Figure 2. Architecture du DS1104

Le master PPC contrôle deux types de convertisseurs Analogic to Digital Converter :

- Un convertisseur A/D (ADC1) multiplexé pour les signaux (ADCH1 à ADCH4) :
 - Résolution 16-bit
 - Plage de tension ± 10 V
 - Marge d'erreur Offset ± 5 mV
 - Marge d'erreur sur le gain $\pm 0.25\%$
 - Rapport signal sur bruit SNR > 80 dB (at 10 kHz)
- Quatre convertisseurs A/D (ADC2 ... ADC5) pour les signaux (ADCH5 à ADCH8). Les caractéristiques de ces convertisseurs sont :
 - Résolution 12-bit,
 - Plage de tension d'entrée ± 10 V,
 - Marge d'erreur Offset ± 5 mV,
 - Marge d'erreur sur le gain $\pm 0.5\%$,
 - Rapport signal sur bruit > 70 dB

Lorsqu'on place un bloc ADC dans le programme Simulink, on doit tenir compte du fait que toutes les entrées analogiques à l'entrée du dSPACE sont multipliées par un gain de 0.1 pour avoir des signaux compris en -1V et +1V. Il faut donc multiplier par un gain de 10 toutes les entrées analogiques à chaque fois que l'on utilise les blocs ADC. Après avoir placé le bloc ADC dans le programme, il faut choisir le channel sur lequel on veut connecter le signal en double-cliquant sur le bloc, on obtient la fenêtre de la figure 3.

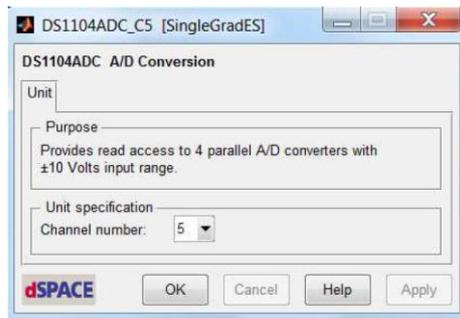


Figure 3. Choix du channel ADC

Les caractéristiques des canaux « Digital to Analogic » (pour les signaux DACH1 ... DACH8) sont :

- Résolution 16-bit
- Plage de tension des sorties analogiques ± 10 V
- Plage d'erreur Offset ± 1 mV,
- Plage d'erreur sur le gain $\pm 0.1\%$
- Rapport signal sur bruit >80 dB (à 10 kHz)

De même, lorsqu'on place un bloc DAC dans le programme Simulink, on doit diviser par un gain de 10 pour avoir des signaux compris en -1V et +1V. Il faut également choisir le channel sur lequel on veut avoir le signal en double cliquant sur le bloc. On peut également régler les paramètres d'initialisation et de terminaison (figure4).

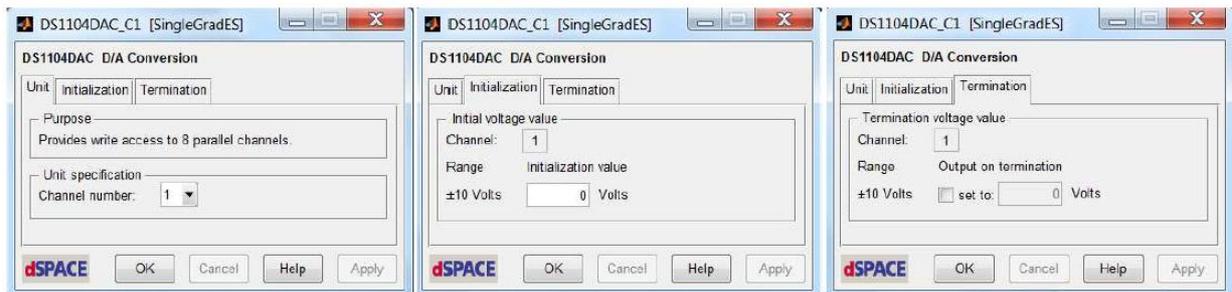


Figure 4. Propriétés des bloc DAC

Pour implémenter la commande temps réel en utilisant la carte « dSPACE DS1104 R&D Controller Board » et le logiciel Matlab/Simulink, on a besoin de :

- 1) La carte de contrôle dSPACE DS1104



Figure 5. Illustration de la carte DS1104

- 2) La licence sous forme de clé USB (Dongle)



Figure 6. Dongle

- 3) La licence software (fichier Licence.dsp)
- 4) Le fichier Keys.dsp
- 5) Et enfin le panneau de connexion CP1104

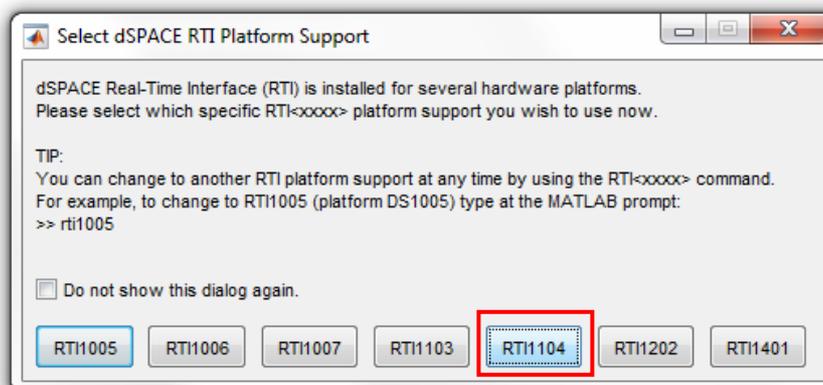


Figure 7. Illustration du Panneau de contrôle CP1104 (Les panneaux que nous disposons dans la salle de projet sont exactement les mêmes mais avec des Leds)

3. Construction du Model SIMULINK

Avant de créer un model sous Simulink, démarrez le programme Matlab puis choisissez le model de votre carte DSPACE à l'ouverture de Matlab.

```
-----  
Configuring dSPACE(R) Software for MATLAB(R) 8.5.0.197613 (R2015a) ...
```



Une fois la carte choisie et activée vous aurez le message suivant :

```

documents ▸ MATLAB ▸
Command Window

=====
Configuring dSPACE(R) Software for MATLAB(R) 8.5.0.197613 (R2015a) ...

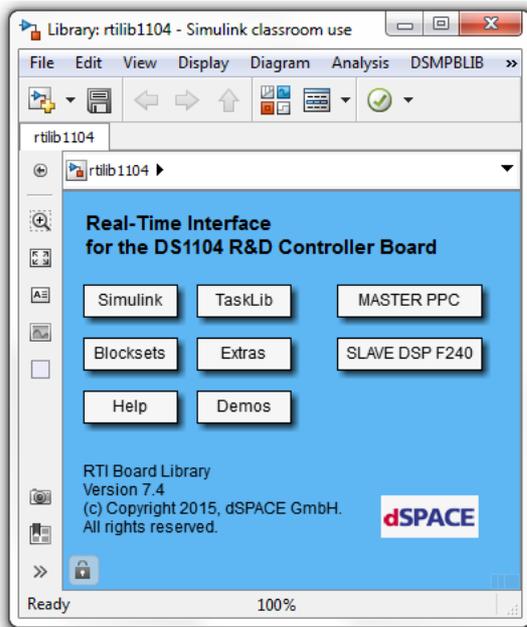
RTI                Real-Time Interface to Simulink (RTI1104)  7.4      02-May-2015 okay
RTIFPGA            RTI FPGA Programming Blockset                        2.9      02-May-2015 okay
DSMPBLIB           Model Port Block Library                             3.0      02-May-2015 okay
DSRT               dSPACE Run-Time Target (DSRT)                       3.0      02-May-2015 okay
DSMSBLIB           Model Separation Block Library                       3.0      02-May-2015 okay
DSMLCON24          dSPACE MATLAB Connection 2.4 (win64)                2.4      01-Jun-2012 okay
=====

*** RTI Platform Support RTI1104 activated.

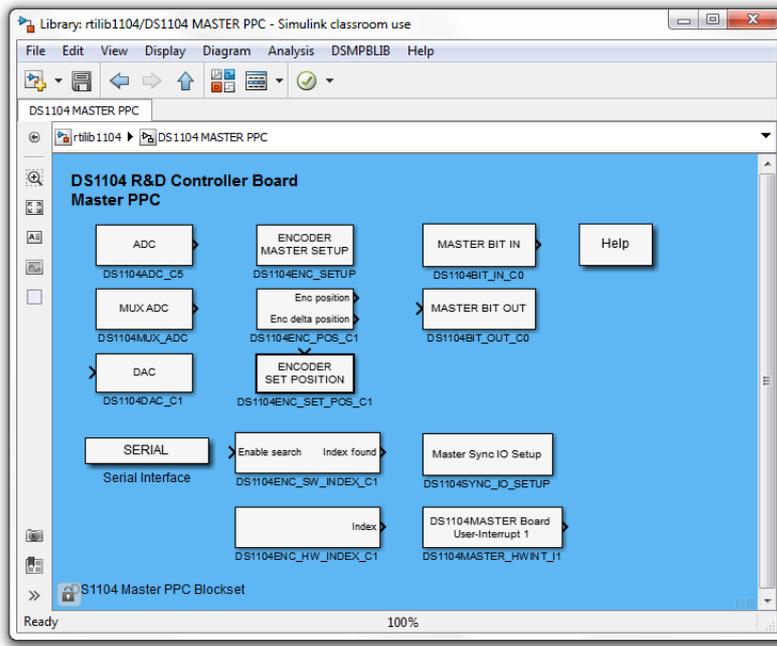
This is a Classroom License for instructional use only.
Research and commercial use is prohibited.

fx >> |
    
```

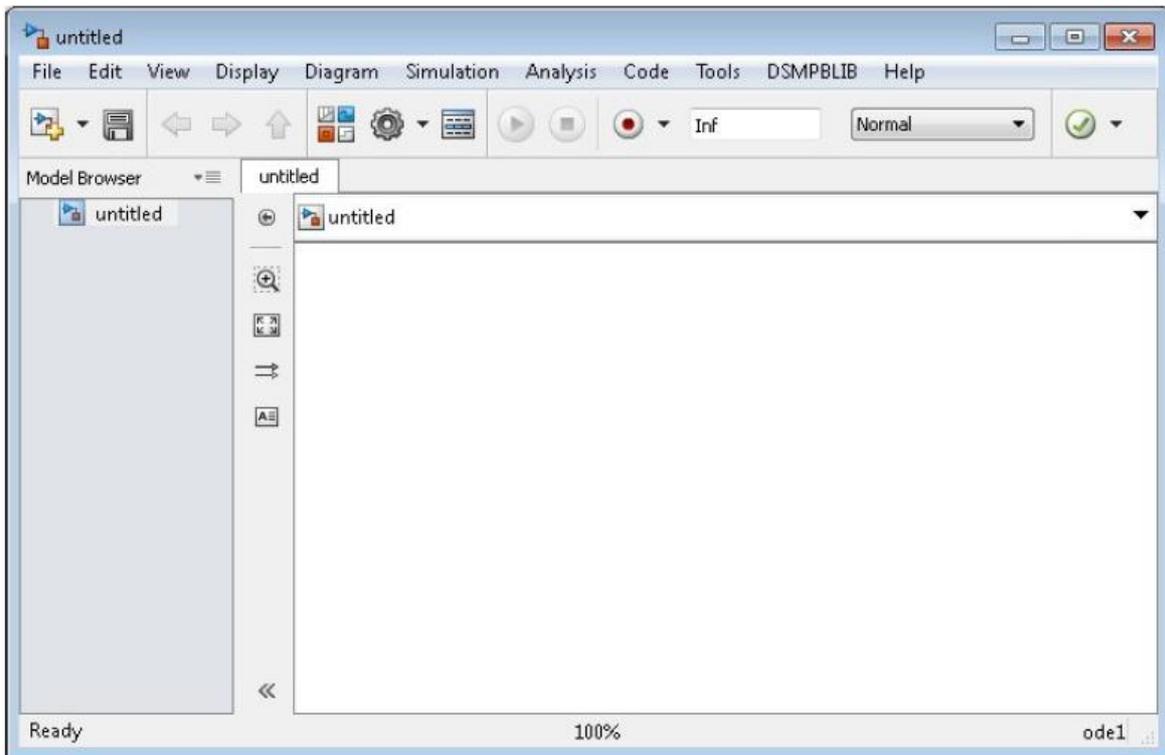
Démarrez ensuite le programme Simulink dans Matlab. Pour ajouter des blocs de la librairie RTI1104, ouvrez la librairie en tapant la commande RTI1104, ou à partir de Simulink Library Brower.



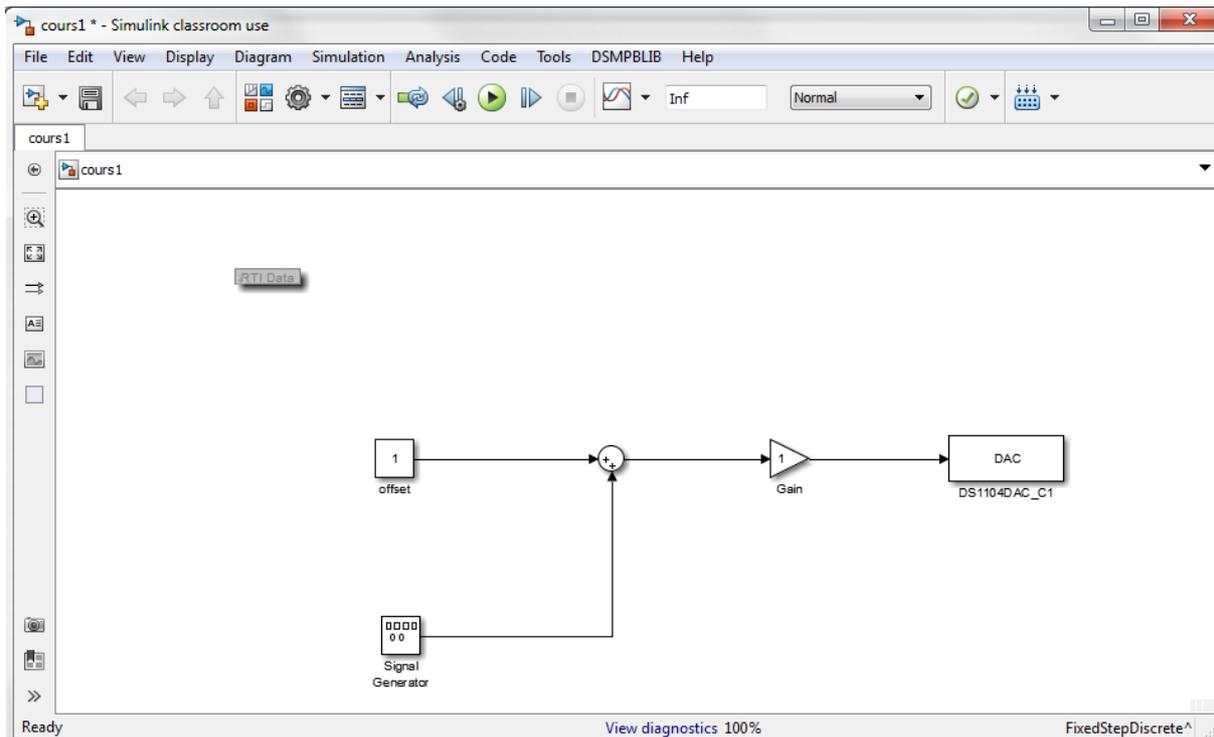
Cliquez par exemple sur MASTER PPC, la fenêtre suivante s'ouvre :



Choisissez new file

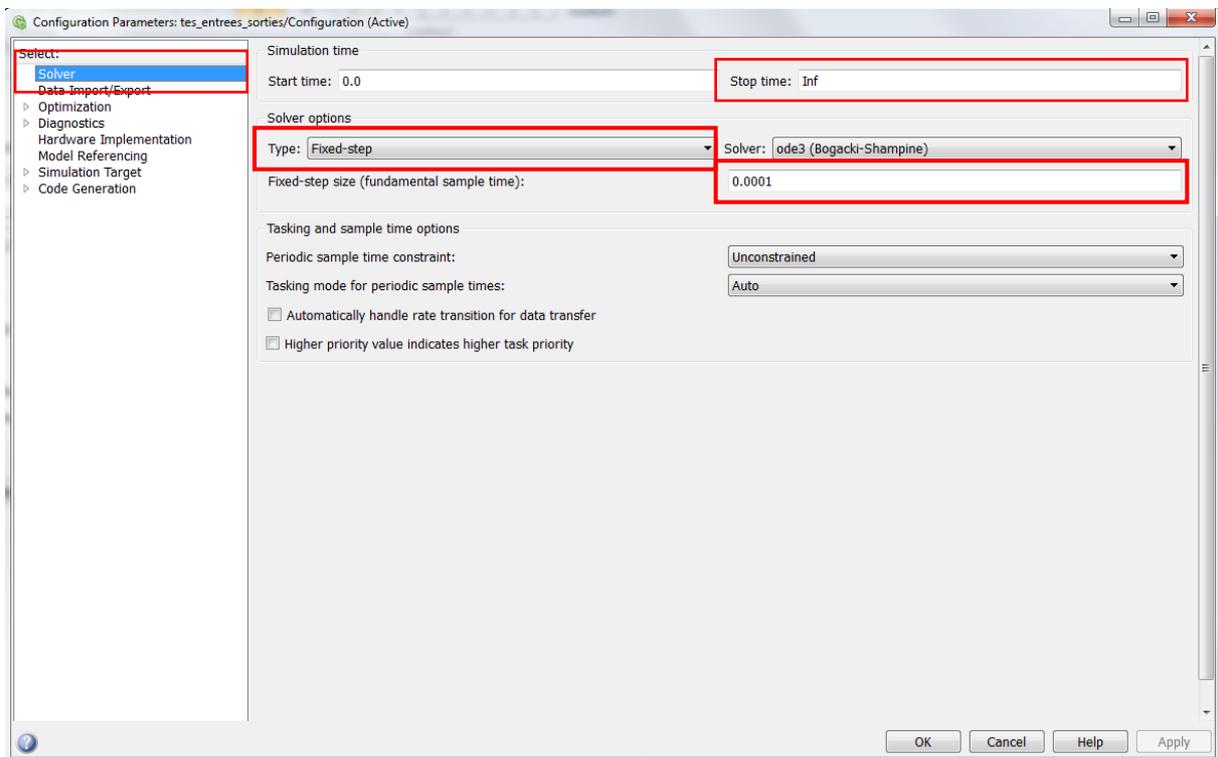


Ajoutez des blocs Simulink et des blocs ADC et DAC à partir de la librairie Master PPC et ajoutez les connexions suivantes

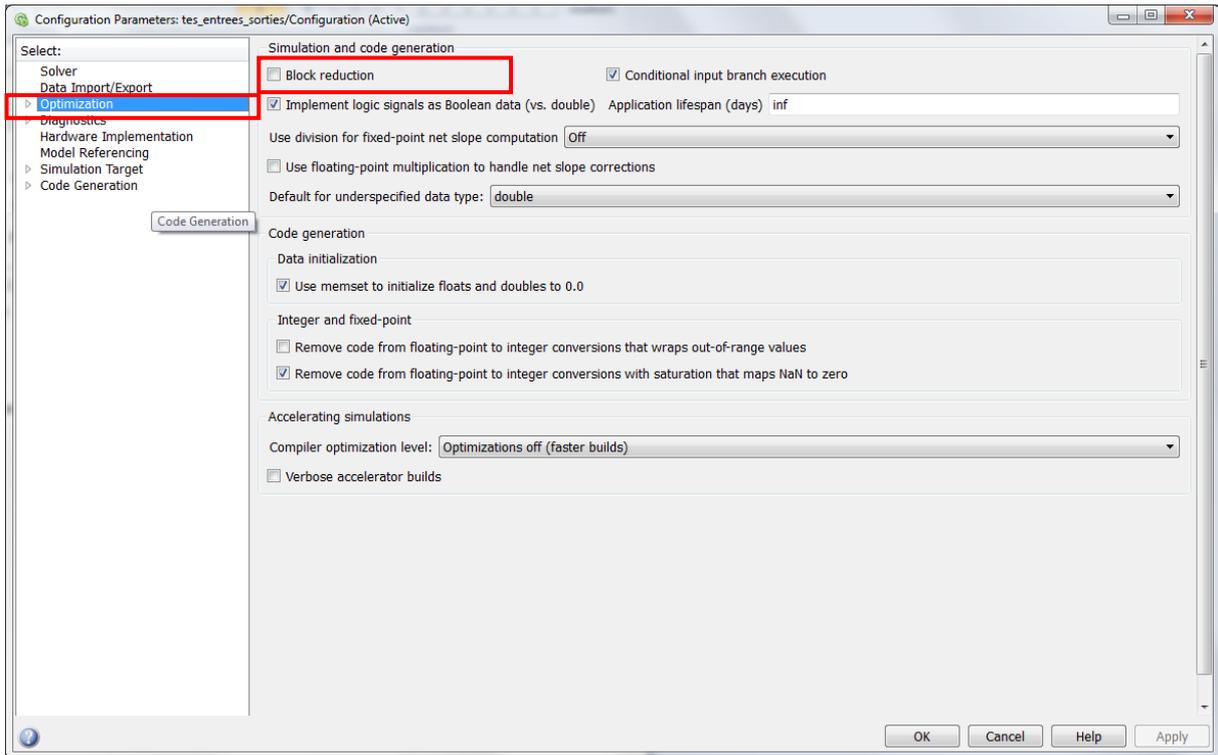


Enregistrez le fichier et ensuite réglez les paramètres de simulation.

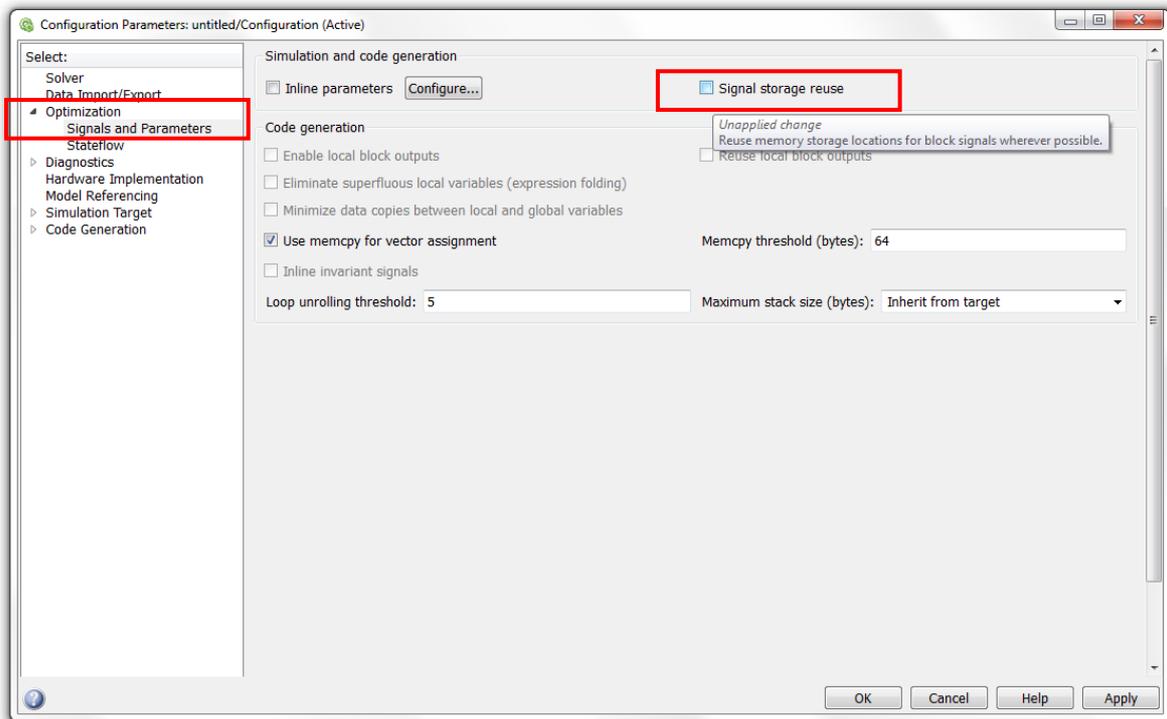
Dans l'onglet Simulation, puis Model Configuration Parameters. Dans l'onglet **Solver**, modifiez le Stop time et mettez Inf (infini). Dans **Solver Options, Type**, choisissez **Fixed-step**, et ensuite choisissez ensuite une période d'échantillonnage, choisissez par exemple 1e-3 ou 1e-4.



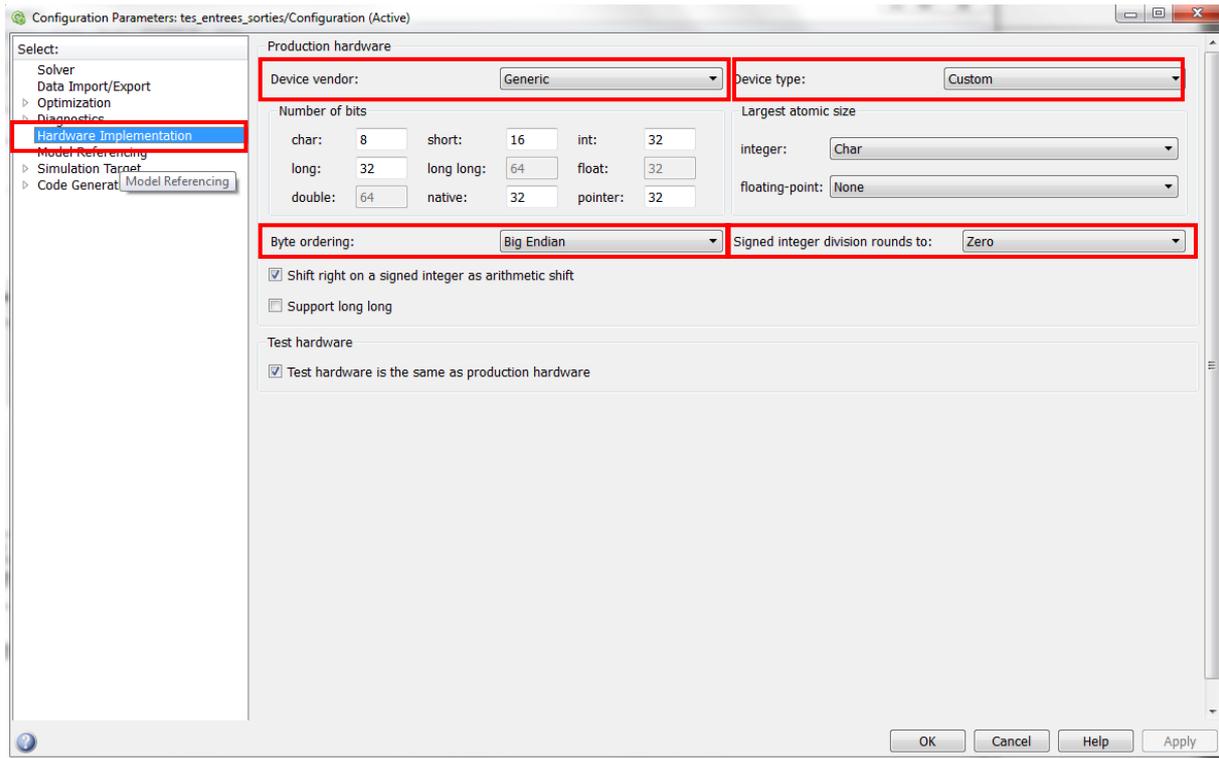
Dans **Optimization**, décochez la case **Bloc reduction**



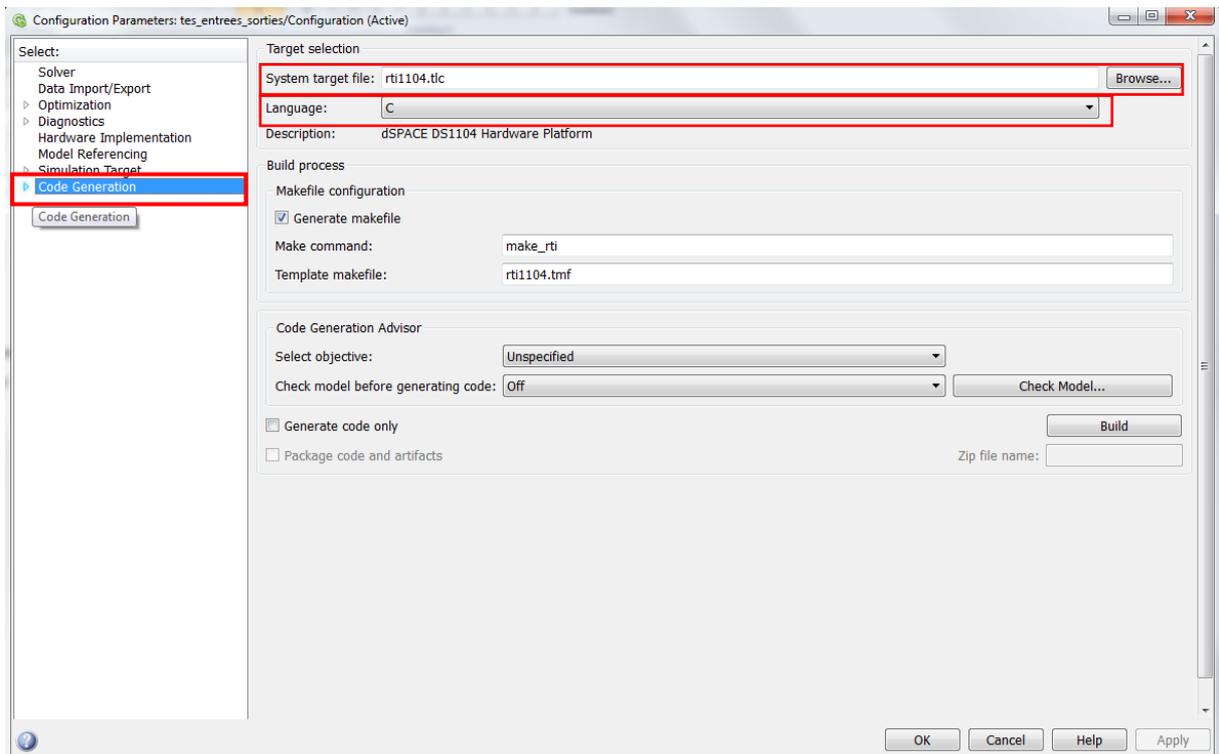
Toujours dans **Optimization** puis **Signals and Parameters** puis décochez la case **Signal storage reuse**



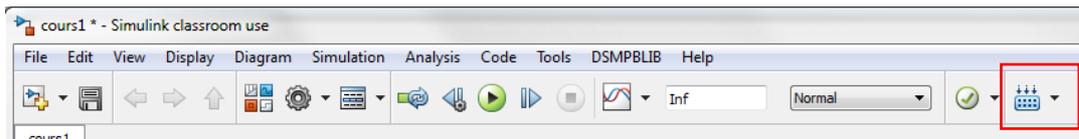
Allez dans **Hardware implementation** puis dans Device Vendor, choisissez Generic, Device Type Custum, Byte ordering : Big Endian et pour le paramètre Signed Integer division rounds to : Zero.



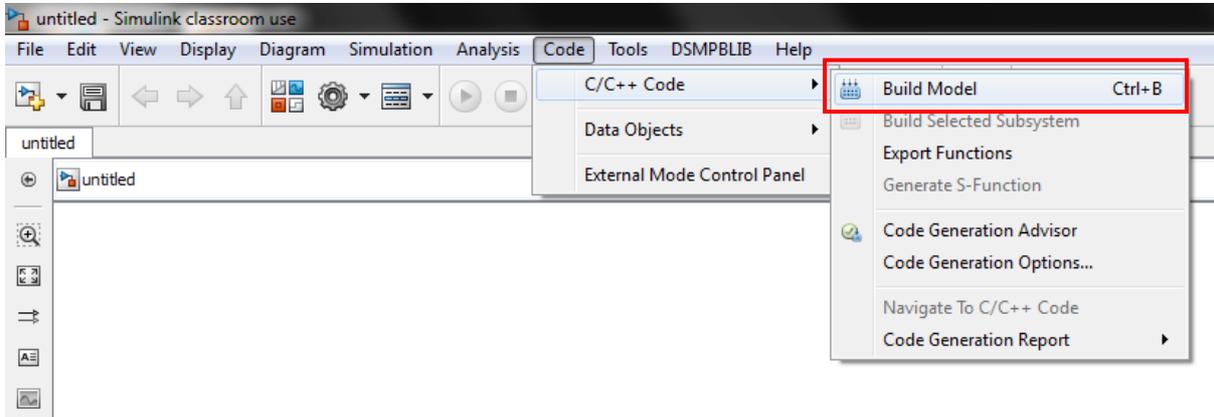
Ensuite allez dans **Code Generation** puis choisissez dans **System target file**, le fichier rti1104.tlc, et pour le **Language** choisissez C (pour générer un code C) puis cliquez sur **Apply**.



Vous pouvez ensuite exécuter le programme et générer le code C en cliquant sur control+B (B comme build) ou en cliquant sur l'icone .



Vous pouvez également cliquer dans le menu **Code** puis **C/C++ Code** et **Build Model**



Le code C sera créé, compilé et envoyé à la carte DSPACE. Un fichier *.sdf sera ensuite généré dans le répertoire de travail. Ce fichier contient toutes les variables utilisées dans votre programme Simulink.

4. Implémentation d'une application temps réel avec ControlDesk Next Generation

ControlDesk est une interface qui permet de visualiser en temps réel différentes variables du programme développé sous Simulink et de modifier également des paramètres définissant le mode de fonctionnement des blocs constituant le schéma Simulink. La visualisation de variables ou de signaux et la modification de paramètres sont possibles par l'intermédiaire d'instruments graphiques que l'on peut sélectionner dans la fenêtre **instruments**.

L'espace de travail sous ControlDesk est composé :

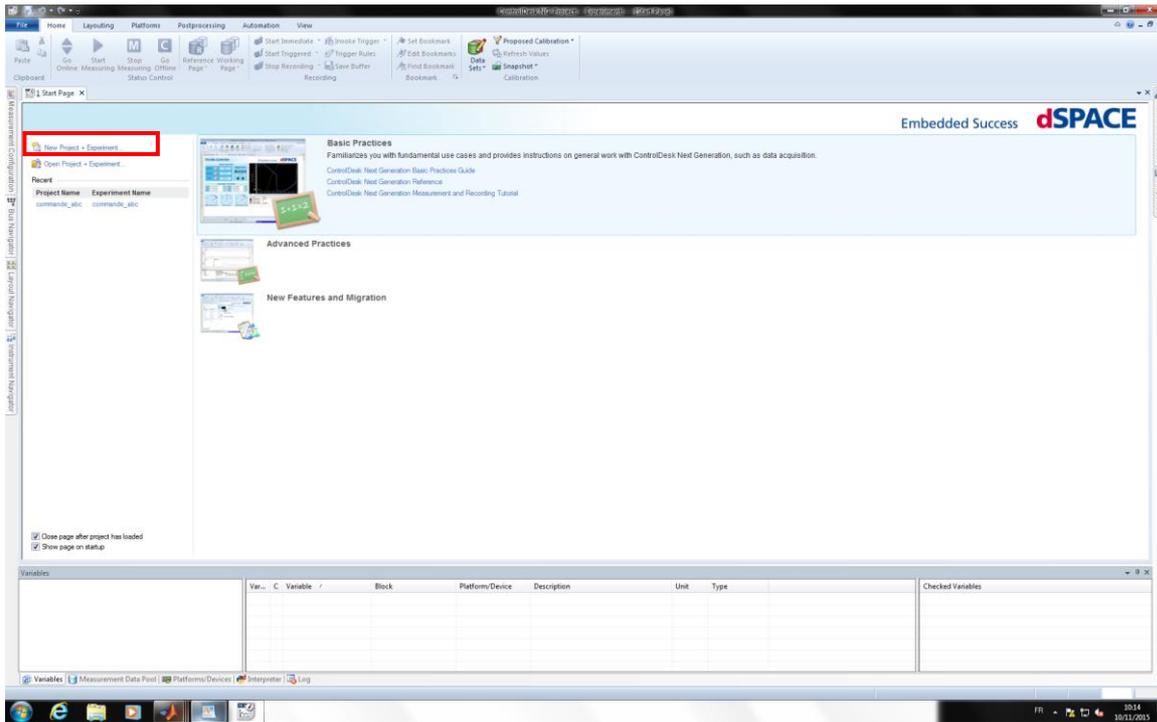
- d'une fenêtre de navigation (**Navigator**), elle liste les Layouts, la carte connectée et le programme en cours.
- de l'aire de travail contenant le Layout,
- d'une fenêtre contenant les instruments pour la création d'un Layout (**instrument selector**),
- et d'une fenêtre d'outils (**tools window**), la fenêtre des variables Simulink (fichier .sdf). les variables sont représentées sous forme d'arbre en correspondance avec les blocs et sous blocs du programme Simulink.

Il existe trois modes de fonctionnement sous ControlDesk :

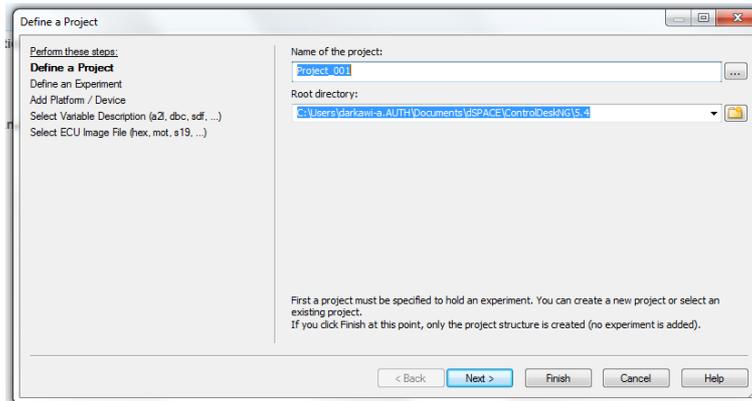
- Le mode **Layouting** : mode de construction de Layout
- Le mode **Standard** : mode permettant de tester la validité des connexions entre le Layout et les variables Simulink.
- Le mode **Measuring** : mode d'exécution du programme contenu dans le DSP avec le Layout comme IHM.

Comment créer un nouveau projet sous ControlDesk ?

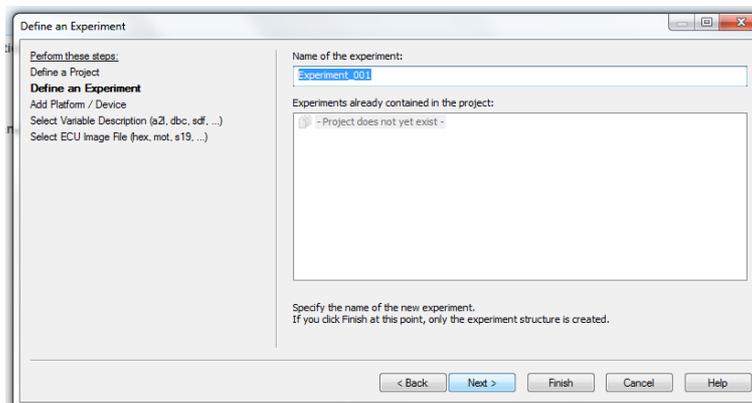
En double cliquant sur l'icône ControlDesk Next Generation qui se trouve dans votre bureau la fenêtre suivante s'ouvre :



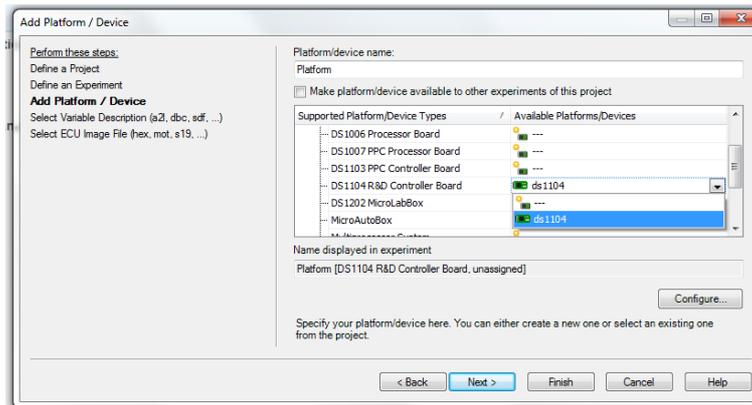
Dans ControlDesk un projet doit contenir au moins un **experiment**. Dans le menu File, cliquer sur **New Project+Experiment** puis saisir un nom en choisissant le répertoire dans lequel sera enregistré le projet.



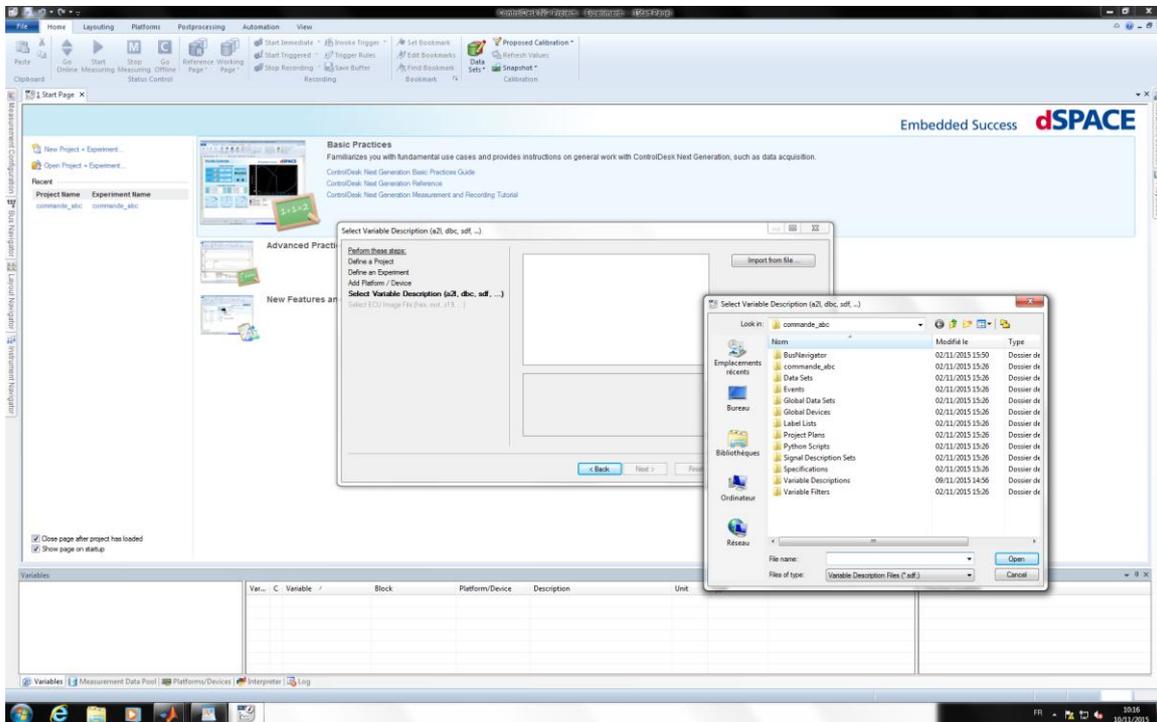
La fenêtre de définition du projet s'ouvre. Saisissez le nom du projet et ensuite cliquer sur suivant (Next). Choisissez



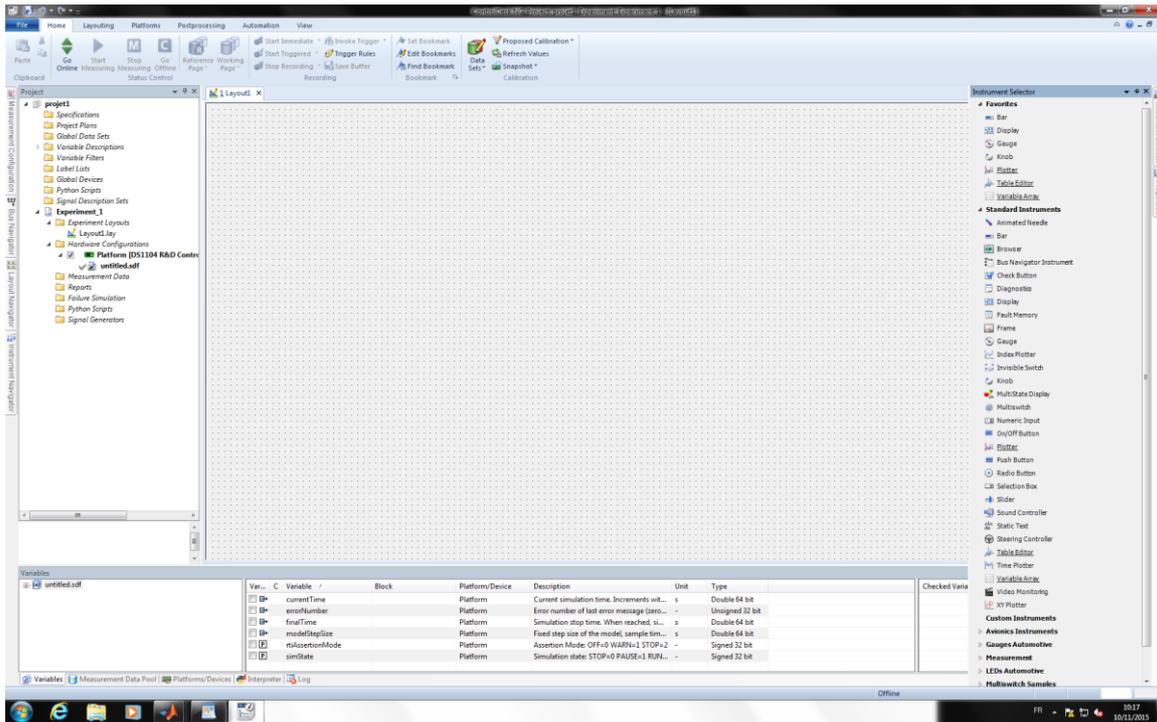
Sélectionnez ensuite la plateforme (DS1104), puis cliquez sur Next,



Choisissez le fichier **.sdf** correspondant (fichier généré lors de l'exécution de votre programme Simulink) et cliquez sur Finish.



Vous venez de créer un projet. Ensuite il faut créer un ou plusieurs **Layout** associé à ce projet. Un **Layout** est une interface graphique à laquelle on peut ajouter divers instruments de contrôle et/ou de visualisation, dans le but de visualiser et/ou modifier en temps réel les différentes variables du projet. Exemple, on peut modifier en temps réel les paramètres d'un correcteur, modifier une consigne, etc...



Le Layout est créé dans l'espace de travail. Ensuite il faut y ajouter des instruments. Il existe deux types d'instruments :

- les **virtual instruments** : ce sont les instruments de visualisation et/ou de modification de variable,
- les **data instruments** : ce sont les instruments qui permettent l'acquisition de variables (à l'écran ou dans un fichier).

Toutefois, ControlDesk offre la possibilité d'éditer de nouveaux instruments et de les sauvegarder dans une librairie grâce à l'onglet '**Custom instruments**'.

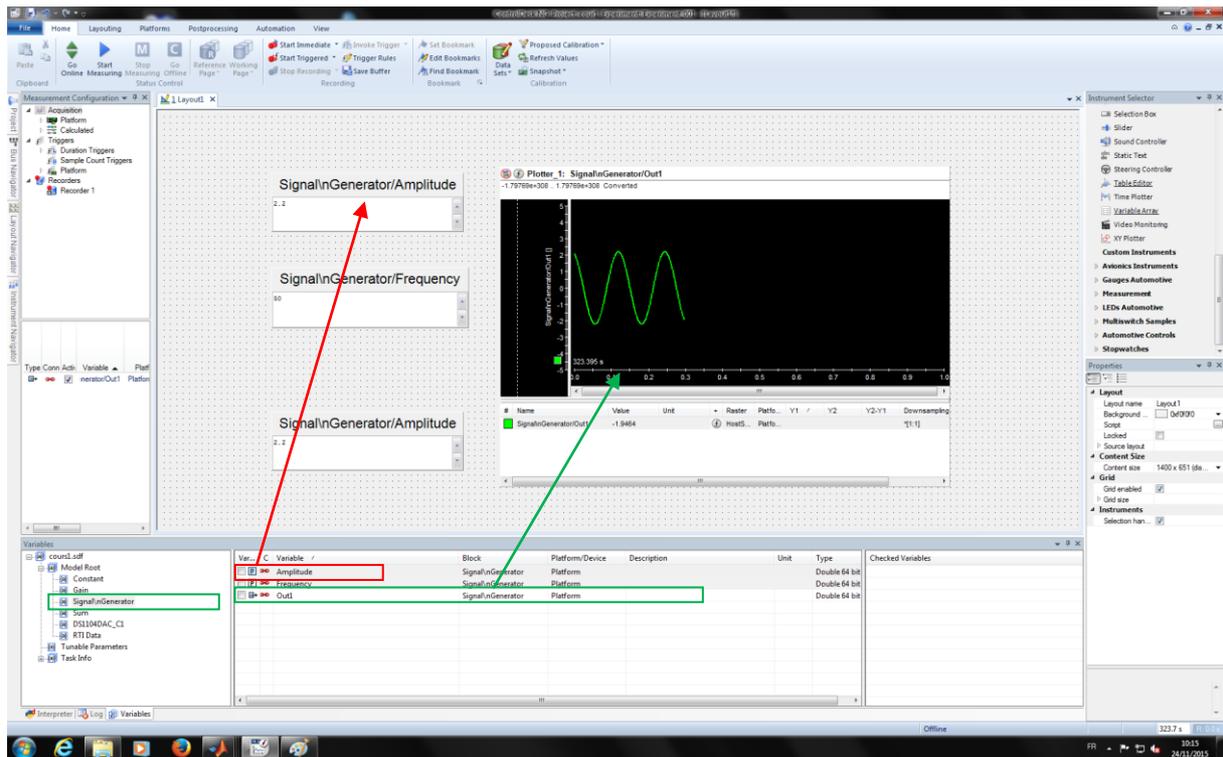
Après avoir rajouté des instruments, il faut affecter à chaque bloc une variable et modifier les propriétés de l'instrument si besoin en cliquant sur le bouton droit et ensuite sur **properties**.

Pour initialiser un objet graphique, il suffit de faire glisser la variable à l'instrument désiré. Le symbole  Out1 signifie que l'on ne peut que visualiser la variable, tandis que le symbole  Value signifie que l'on peut visualiser et modifier la variable.

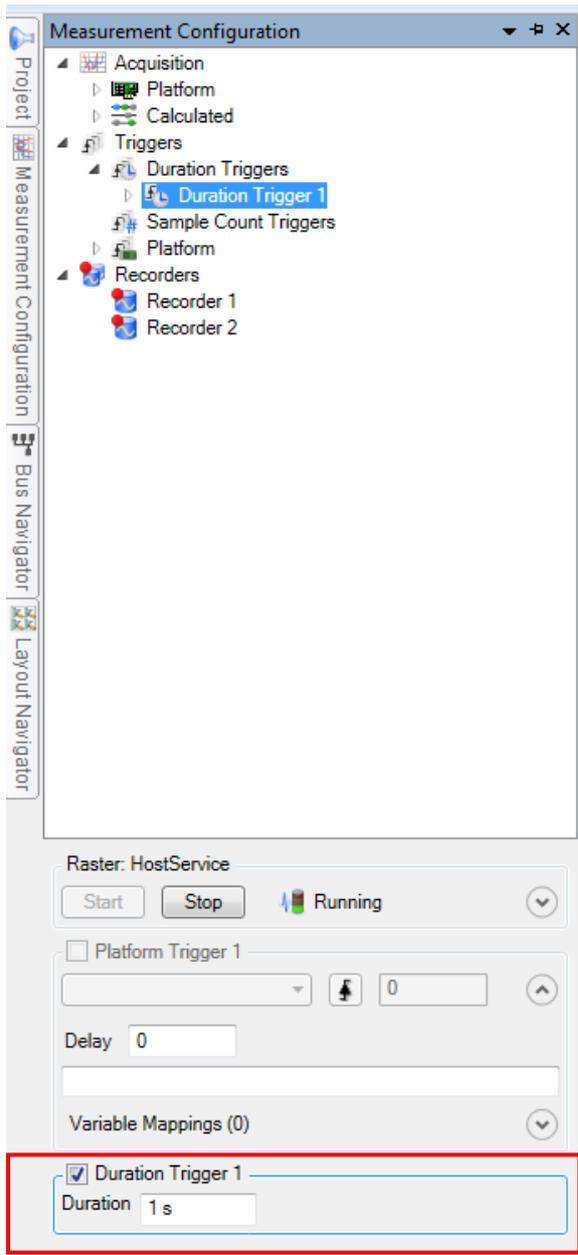
5. Comment faire une mesure

Pour faire la mesure d'une variable, il faut connecter la variable à un **Virtual Instrument**. Le fait d'avoir une mesure correcte confirme que votre application ControlDesk Next Generation fonctionne correctement.

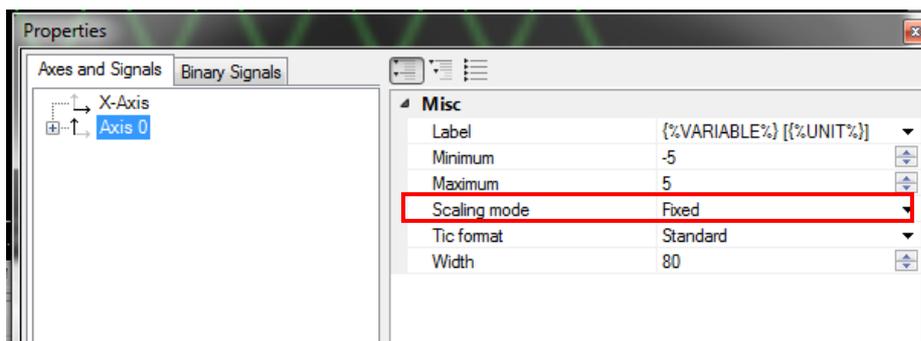
Insérez dans le Layout un « **plotter** » et des « **numerical input** », ensuite glissez les variables correspondantes à chaque instrument virtuel.



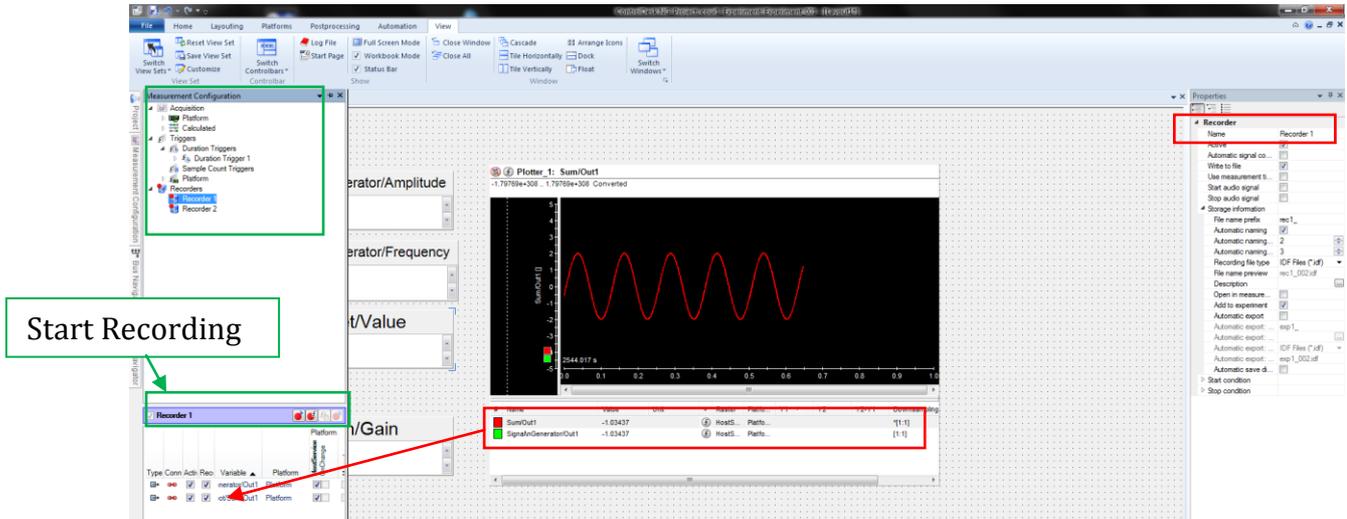
Pour augmenter la durée de visualisation des mesures, allez dans **"Measurement Configuration > Triggers > Duration Trigger1"** puis changer **"Duration"** to 1s.



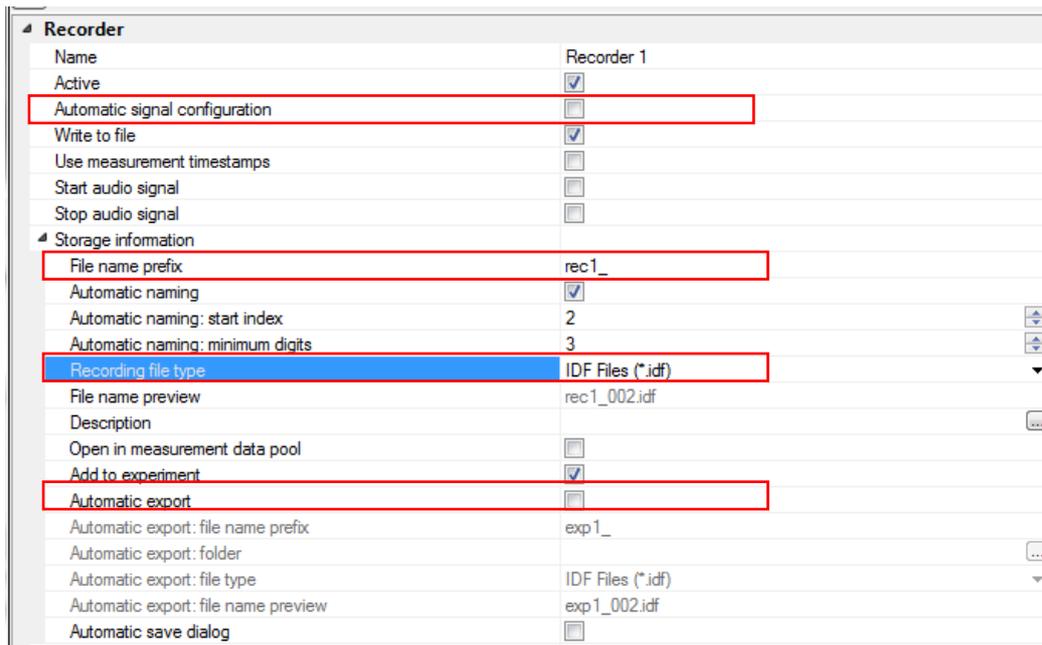
Vous pouvez à tout moment modifier les propriétés des axes X et Y. Pour se faire, cliquez le bouton droit et puis sur **Properties**. Allez dans la fenêtre **Properties**, Axis, dans **scaling mode** choisissez **fixed**, puis ensuite modifier le max et le min.



Pour enregistrer les données (mesures) dans votre ordinateur allez sur **Measurement** puis **View Measurement Configuration**, puis dans **Recorders** puis choisissez **Create New Recorder**. Nommer l'enregistrement puis glisser les variables que vous voulez mesurer dans le tableau **Recorder**.

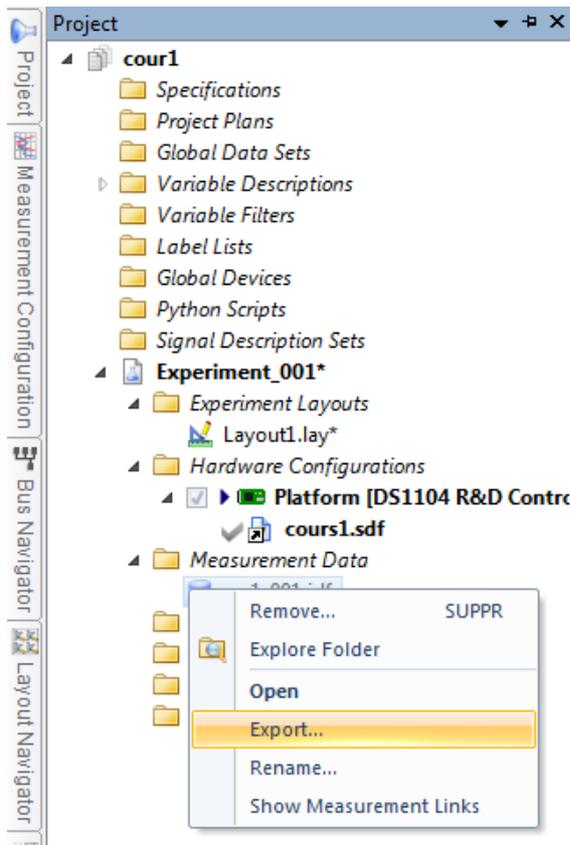


Réglez les paramètres d'enregistrement dans la fenêtre Propriétés (propriétés) et commencer l'enregistrement en cliquant sur le bouton Start Immediate Recording:



Le nom du fichier d'enregistrement par défaut est rec1.idf.

Pour exporter les données vers un fichier Matlab, allez dans **Project, Experiment1, Measurement Data** et cliquez droit sur le fichier .idf créé puis export en choisissant bien le type de fichier .mat dans l'onglet **Save as Type**.



Une fois vous avez exporté le fichier .mat, vous pouvez le charger et extraire les variables en utilisant le programme de récupération des données suivant.

```
%Programme de récupération de données sous Matlab
%nom du fichier test.mat
%le fichier doit être placé dans le répertoire de travail de Matlab
load test.mat; % pour charger les données
t = test.X.Data; % création du vecteur temps
variable_1=test.Y(1).Data; %récupération des variables
variable_2=test.Y(2).Data;
%ainsi de suite
%*****%
%****tracé des courbes****%
figure;
plot(t, variable_1); grid;
figure;
plot(t, variable_2); grid;
%fin%
```

Vous pouvez aussi exporter automatiquement les données en cochant la case **Automatic export**. Il est également possible d'exporter les données sous forme de tableau en **.csv**.