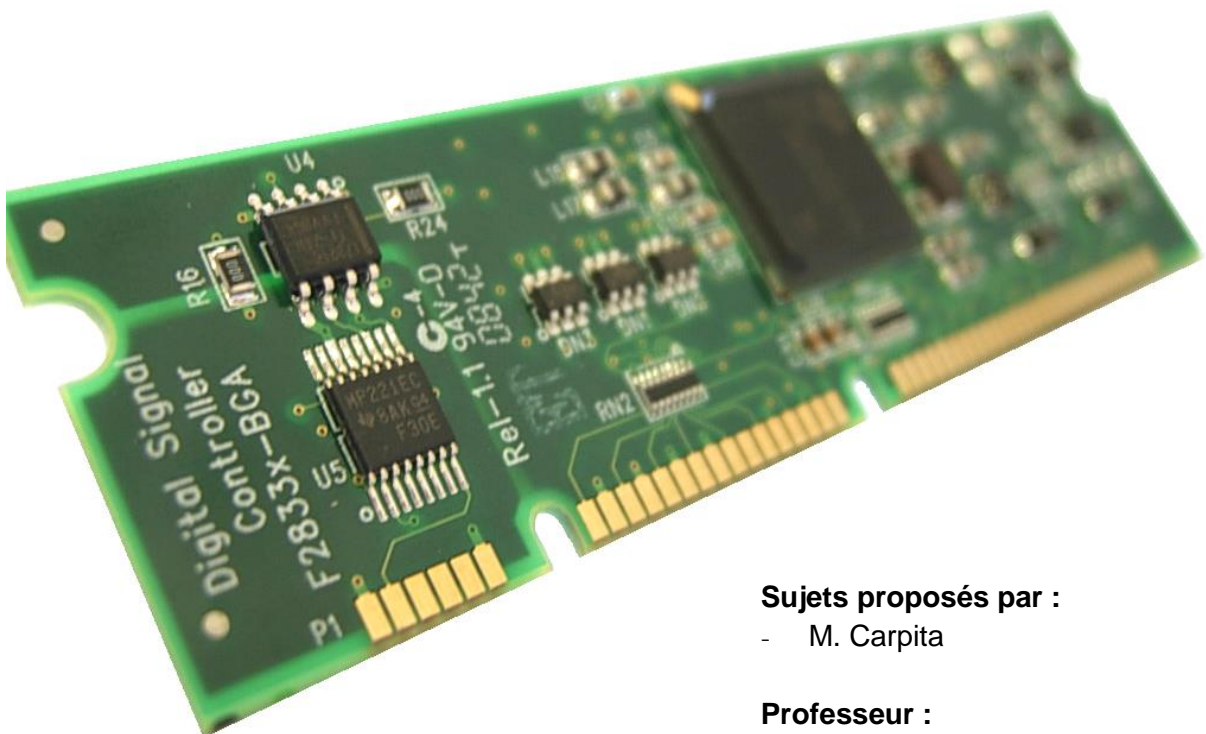

MA_PowELSys

Mesures via Mat2DSP



Sujets proposés par :

- M. Carpita

Professeur :

- Mauro Carpita

Utilisation de Mat2DSP

Mat2DSP est un code Matlab servant à accéder aux variables du DSP pour les afficher et les enregistrer dans Matlab.

Pour pouvoir utiliser Mat2DSP, il faut vérifier que les fichiers adéquats soient présents dans votre projet dans Code Composer. Si besoin voir avec l'assistant.

Ensuite pour récupérer les données il faut procéder comme suit :

- 1- Démarrer Matlab et rentrer le chemin d'accès qui correspond au répertoire de votre projet Code Composer.
- 2- Ouvrir le gestionnaire de périphérique et identifier sur quel port COM le DSP est branché (voir avec l'assistant).
- 3- Taper xlist dans la fenêtre de commande Matlab.
- 4- Configurer la boîte de dialogue avec le bon port COM et le bon chemin d'accès (voir avec l'assistant).
- 5- Utiliser les différentes fonctions décrites ci-dessous dans Matlab pour récupérer les variables nécessaires à la vérification du bon fonctionnement de la PLL et de la commande en BO.

Commande Matlab	Description et exemples
xlist	Affiche les variables globales du DSP dans l'ordre alphabétique avec les propriétés des différentes variables. Sont exclus les variables globales de type struct et les variables globales cachées. <i>xlist(1)</i> inclut dans l'affichage les variables cachées.
tmsvar	<i>tmsvar('toto')</i> : lit et affiche le contenu de la variable globale toto du DSP <i>x = tmsvar('toto')</i> : lit et affecte à x le contenu de toto Attention : tmsvar est insensible aux minuscules/majuscules => on peut par commodité tout mettre en minuscule. Un warning apparaît s'il y a deux variables DSP 'toto' et 'ToTo'. <i>tmsvar('toto', 12345)</i> : écrit la valeur 12345 dans la variable toto <i>tmsvar('Ptr', 'totovar')</i> : affecte le pointeur Ptr avec l'adresse de la variable totovar <i>tmsvar('Tab', [11.5, 13.7, -99.123])</i> : affecte un tableau virgule flottante <i>tmsvar({hex2dec('E000'), 2, 'long'})</i> : lecture sauvage de 2 words à l'adresse 0xE000 interprétés comme type 'long'
varinfo	Retrouve les informations relatives à une variable <i>[adr, type, WSize, Idx, success] = varinfo('toto')</i> <i>[adr, type, WSize, Idx, success] = varinfo({'toto1', 'toto2', 'toto3'})</i>

scope	<p><code>scope('Var1Int16 Var2Int16 Varfloat32', 500, 200) :</code></p> <p>Affiche 500 échantillons des variables 'Var1Int16' 'Var2Int16' et 'Varfloat32', échantillonnés 1 fois sur 200. Le facteur de sous-échantillonnage est optionnel (default : 1). Les noms de variable sont espacés d'un ou de plusieurs espaces (blanks).</p> <p><code>MatX = scope('Var1Int16 Var2Int16 Varfloat32', 500, 200) :</code></p> <p>La matrice MatX comprend sur une même ligne les différentes variables. MatX(m,n) correspond à l'échantillon numéro m de la variable numéro n.</p> <p><code>scope('Uabc Uabc Uabc Ud',500,',',[0 1 2]) :</code></p> <p>Affiche 500 échantillons des variables Uabc[0] Uabc[1] Uabc[2] et Ud pris à chaque échantillonnage. Les variables de type tableau 1D sont à mettre en premier dans la liste et les index correspondants dans le même ordre.</p>
--------------	--

trigger	<p><i>trigger('Var1Int16', 10000, '=', 100)</i> :</p> <p>Trigger si le signal 'Var1Int16' vaut exactement 10000 avec 100 échantillons de prétrigger</p> <p><i>trigger('Var1Int16', 10000, '+', 100)</i> : trigger sur flanc montant</p> <p><i>trigger('Var1Int16', 10000, '-', 100)</i> : trigger sur flanc descendant</p> <p><i>trigger('Var1Int16', 10000, 'x', 100)</i> : trigger sur flanc montant ou descendant</p> <p>Attention : fonctionne pour l'instant uniquement avec des signaux int16.</p> <p>Pour utiliser la fonction trigger il faut que la variable sur laquelle on veut trigger soit de type int16, si ce n'est pas le cas il faut créer une copie en int16 (en multipliant éventuellement pour que ce soit significatif en 'int16').</p> <p>La fonction trigger est similaire à la configuration du trigger sur l'oscilloscope, il faut ensuite appeler scope pour afficher les données voulues.</p> <p>Il est possible d'utiliser trigger sur une variable et d'en afficher une autre avec scope.</p>
----------------	--