

CAN Eurosmart : interface pour Scilab

1. Installation du module scilab

Ce document présente une interface des convertisseurs analogique/numérique (CAN) SysamSP5 (port USB) et SysamPCI (carte PCI) de [Eurosmart](#) pour le logiciel [Scilab](#)

Cette interface permet d'effectuer des acquisitions et des sorties directement depuis un programme Scilab.

L'interface fonctionne sous Windows (XP, Vista, Windows 7).

Elle est distribuée avec le code source sous licence [CeCILL](#).

Téléchargements :

▷ [sciCAN-3.1.zip](#) sources et binaires pour Scilab 5.5.1 32 bits

Décompresser le fichier `sciCAN-3.1.zip` dans `C:\Program files\scilab-5.5.1\contrib`. Redémarrer Scilab.

La compilation nécessite [pthread-win32](#).

Le nom du module est `can`. Il est accessible dans le menu Module ou par la commande `exec(SCI+"\contrib\can\loader.sce",-1)`.

2. Installation des pilotes

Pour installer la centrale SysamSP5, télécharger le pilote sur [Eurosmart](#). Décompresser le dossier contenant les pilotes. Brancher la centrale puis son câble USB. Attendre la reconnaissance par le système. Premier cas : il est demandé d'installer un pilote ; indiquer le chemin du dossier contenant les pilotes. Second cas : il y a seulement un message d'erreur qui indique que le pilote n'a pas été trouvé ; dans ce cas, aller dans Ordinateur/Gérer/Gestionnaire de périphériques, sélectionner SysamSP5 puis "mettre à jour le pilote" ; indiquer alors le chemin du dossier contenant les pilotes et valider.

La carte SysamPCI équipe certaines salles de TP dans les lycées. Elle est utilisée avec le boîtier de connexion BORA. Télécharger le dernier pilote sur [Eurosmart](#). Lors du démarrage de l'ordinateur après mise en place de la carte (ou désinstallation d'un ancien pilote), il est demandé d'installer le pilote. Indiquer le chemin du pilote téléchargé. Après installation du pilote, exécuter `Install_Sysam_PCI.exe` situé dans ce dossier. Pour finir copier la DLL `..contrib/can/sci_gateway/c/SysamPCI.dll` dans `C:\Windows\System32`.

3. Fonctions de l'interface

3.a. Ouverture et fermeture de la liaison avec le CAN

<code>CanOuvrir(nom)</code>

Ouverture de la liaison.

▷ `nom (string)` : nom du CAN : SysamSP5 ou SysamPCI.

```
CanFermer()
```

Fermeture de la liaison.

3.b. Configuration des entrées analogiques

```
CanConfigEntrees(voies,calibres [,differentielles])
```

Sélection des entrées à utiliser, de leur calibre et sélection éventuelle du mode différentiel. Les voies sont numérotées de 0 à 7.

- ▷ `voies` (`integer array`) : liste des voies sous la forme `[v1,v2,...]`.
- ▷ `calibres` (`float array`) : liste des tensions maximales des voies sous la forme `[u1max,u2max,...]`.
- ▷ `differentielles` (`integer array`) : argument optionnel. Liste des voies en mode différentiel.

Sur la carte SysamPCI, la fréquence d'échantillonnage maximale diminue lorsqu'on augmente le nombre de voies sélectionnées. Sur la centrale SysamSP5, il faut sélectionner les voies 0,1,2 et 3 pour bénéficier de la fréquence d'échantillonnage maximale (10 MHz).

Sur SysamPCI, la sélection d'une seule voie en mode différentiel les placent toutes en mode différentiel. Les tensions sont alors mesurées entre les entrées 0 et 4, 1 et 5, 2 et 6, 3 et 7. Sur SysamSP5, chaque canal (0-4,1-5,2-6,3-7) peut être placé en mode différentiel indépendamment des autres. Pour placer le premier et le second canal en mode différentiel, il faut affecter `[0,1]` au dernier argument.

3.c. Utilisation des entrées analogiques avec échantillonnage

```
CanConfigEchantillon(temps,nbpoints)
```

Choisir le temps d'échantillonnage et le nombre de points à acquérir.

- ▷ `temps` (`float`) : temps d'échantillonnage en microsecondes.
- ▷ `nbpoints` (`integer`) : nombre de points à acquérir.

```
CanConfigTrigger(voie,seuil [,montant [,pretrig [,ps [,hyst]]]])
```

Sélectionner et configurer le déclenchement sur une des voies.

- ▷ `voie` (`integer`) : voie utilisée pour le déclenchement.
- ▷ `seuil` (`float`) : tension seuil en volts.
- ▷ `montant` (`boolean`) : argument optionnel, déclenchement sur front montant (
- ▷ `pretrig` (`integer`) : argument optionnel, pour SysamSP5. Nombre de points enregistrés avant la condition de déclenchement.
- ▷ `ps` (`boolean`) : argument optionnel, pour SysamSP5.

- ▷ `hyst` (`integer`) : argument optionnel, pour SysamSP5. Hystérésis du déclenchement.

```
CanConfigTriggerExterne([pretrig [,ps]])
```

Sélectionner et configurer le déclenchement sur la voie EXT, par un front montant de type TTL (0 à 5V).

- ▷ `pretrig` (`integer`) : argument optionnel, pour SysamSP5. Nombre de points enregistrés avant la condition de déclenchement.
- ▷ `ps` (`boolean`) : argument optionnel, pour SysamSP5.

```
CanDesactiverTrigger()
```

Désactiver le déclenchement sur un signal de référence.

```
CanAcquerir()
```

Lancer l'acquisition. Si aucun déclenchement sur un signal de référence n'est configuré, l'acquisition démarre dès que l'ordre parvient au CAN.

```
tensions=CanEntrees()
```

Obtenir les tensions de la dernière acquisition.

- ▷ `tension` (`float array`) : matrice des tensions. Les tensions d'une voie sont rangées en ligne.

```
temps=CanTemps()
```

Obtenir les instants d'échantillonnage de la dernière acquisition. Sur SysamPCI, les instants des différentes voies sont légèrement décalés.

- ▷ `temps` (`float array`) : matrice des temps. Les temps d'une voie sont rangés en ligne.

3.d. Utilisation des sorties analogiques avec échantillonnage

La carte SysamPCI comporte une sortie analogique SA1. La centrale SysamSP5 comporte deux sorties SA1 et SA2.

```
CanConfigSortie(nsortie,temps,tensions,repetition)
```

Configuration d'une sortie.

- ▷ `nsortie` (`integer`) : sortie à configurer (1 ou 2)

- ▷ `temps` (`float`) : temps d'échantillonnage en microsecondes.
- ▷ `tensions` (`float array`) : liste des tensions (entre -10.0 V et +10.0 V), sous la forme d'une matrice ligne.
- ▷ `repetition` (`integer`) : -1 pour une répétition périodique, +1 pour une seule émission.

`CanDeclencherSorties(s1,s2)`

Déclenchement d'une ou deux sorties. Les deux sorties (SysamSP5) sont déclenchées simultanément. Pour qu'elles soient synchrones, il faut toutefois que les temps d'échantillonnage soient identiques. Sur SysamSP5, la fonction retourne immédiatement. Sur SysamPCI, elle retourne lorsque l'émission est terminée ou lorsque l'utilisateur presse la touche ESC.

- ▷ `s1` (`integer`) : 1 pour déclencher la sortie 1, 0 sinon.
- ▷ `s2` (`integer`) : 1 pour déclencher la sortie 2, 0 sinon.

3.e. Utilisation simultanée des entrées et sorties

Sur SysamSP5, il est possible de déclencher la sortie puis l'acquisition juste après (mais pas l'inverse). Sur SysamPCI, ceci est impossible avec les fonctions décrites plus haut. La fonction suivante permet d'effectuer simultanément une acquisition et une émission des sorties :

`CanAcquerirAvecSorties(tensions1,tensions2)`

Effectuer l'acquisition avec une utilisation simultanée des sorties. Le temps d'échantillonnage des sorties est le même que pour les entrées.

- ▷ `tensions1` (`float array`) : liste des tensions (entre -10V et 10V) à émettre sur la voie 1, sous forme d'une matrice ligne.
- ▷ `tensions2` (`float array`) : liste des tensions (entre -10V et 10V) à émettre sur la voie 2, sous forme d'une matrice ligne.

3.f. Lecture directe des entrées

La lecture directe des entrées peut être utilisée lorsque la cadence d'acquisition est très lente (moins de 1 points par minute). Dans ce cas, on utilisera la fonction `realtime` pour déclencher les lectures.

`CanActiverLecture(voies)`

Activer la lecture directe sur une ou plusieurs voies.

- ▷ `voies` (`integer array`) : liste des voies.

```
tensions=CanLire()
```

Lire les tensions sur les entrées.

▷ `tensions` (`float array`) : liste des tensions lues.

3.g. Écriture directe sur les sorties

```
CanEcrire(s1,tension1,s2,tension2)
```

Appliquer une tension constante sur une ou deux sorties.

▷ `s1` (`integer`) : 1 pour écrire sur la sortie 1, 0 sinon.

▷ `tension1` (`float`) : tension en volts à appliquer sur la sortie 1.

▷ `s2` (`integer`) : 1 pour écrire sur la sortie 2, 0 sinon.

▷ `tension2` (`float`) : tension en volts à appliquer sur la sortie 2.

3.h. Acquisition en mode parallèle (SysamSP5)

Lorsque le temps d'acquisition est long (supérieur à quelques secondes), il peut être utile de l'effectuer sur un fil d'exécution parallèle (thread) afin de faire simultanément un tracé ou un traitement des données. Les fonctions suivantes permettent d'effectuer une acquisition en mode parallèle, sur le CAN SysamSP5 seulement. Pour le CAN SysamPCI, le pilote ne permet pas ce mode d'acquisition.

```
CanLancer()
```

Lancer l'acquisition en mode parallèle. La fonction retourne dès que l'acquisition est lancée.

```
CanLancerAvecSorties(tensions1,tensions2)
```

Lancer l'acquisition en mode parallèle, avec utilisation simultanée des sorties.

▷ `tensions1` (`float array`) : liste fournissant le signal échantillonné à appliquer sur la sortie 1 (en volts).

▷ `tensions2` (`float array`) : liste fournissant le signal échantillonné à appliquer sur la sortie 2 (en volts).

```
A = CanPaquet(premier)
```

Lire le paquet de points déjà acquis lors d'une acquisition en mode parallèle.

▷ `premier` (`integer`) : indice du premier point à récupérer.

▷ `A` (`float array`) : matrice contenant les données. Si `N` est le nombre de voies acquises, les `N` premières lignes sont les instants de ces `N` voies et les `N` lignes suivantes sont les tensions.