

Utilisation du convertisseur SysamSP5

1. Ouverture de l'interface

Le module CAN peut être chargé manuellement à partir du menu boîtes à outil de la console Scilab. Pour le charger dans un script autonome, exécuter :

```
exec(SCI+'\\contrib\\can\\loader.sce',-1)
```

Pour ouvrir la liaison avec le convertisseur Sysam SP5 :

```
CanOuvrir('SysamSP5')
```

2. Utilisation des entrées avec échantillonnage sans trigger

Dans l'exemple qui suit, un GBF délivre une tension sinusoïdale de fréquence 1 kHz sur l'entrée EA0. La sortie TTL OUT du GBF est envoyée sur l'entrée EA1.

La fonction `CanConfigEntrees` permet de sélectionner les voies à utiliser et leur calibre. Le premier argument est la liste des voies (numérotées de 0 à 7), le second est la liste des tensions maximales. Le calibre est choisi de telle sorte que la tension maximale bénéficie du maximum d'amplification. On choisit ici les voies 0 et 1 :

```
CanConfigEntrees([0,1],[5.0,10.0])
```

Le troisième argument (optionnel) est la liste des canaux configurés en mode différentiel. Par exemple, pour configurer le canal EA0-AE4 en mode différentiel, on exécutera : `CanConfigEntrees([0,1],[5.0,10.0],[0])`.

La fonction `CanConfigEchantillon` configure l'échantillonnage des entrées. Le premier argument est la période d'échantillonnage en microsecondes. Si l'on utilise seulement les entrées 0,1,2,3, la période d'échantillonnage minimale est 0.1 microsecondes. Si l'une des entrées 4,5,6 ou 7 est utilisée, le minimum est 2 microsecondes. Le second argument est le nombre de points à acquérir (maximum 262144). On choisit ici une période de 10.0 microsecondes et 200 points :

```
CanConfigEchantillon(10.0,200)
```

Pour déclencher l'acquisition de manière logicielle, on doit désactiver le trigger :

```
CanDesactiverTrigger()
```

La fonction suivante déclenche l'acquisition dès que l'ordre parvient au CAN :

```
CanAcquerir()
```

La fonction `CanEntrees` permet d'obtenir les tensions lues lors de la dernière acquisition :

```
tensions=CanEntrees();
```

La fonction renvoie une matrice dont chaque ligne est la liste des tensions acquises sur une voie. Dans le cas présent, la première ligne correspond à la voie 0, la seconde à la voie 1. Pour obtenir des matrices lignes pour chaque voie :

```
u0=tensions(1,:);  
u1=tensions(2,:);
```

Voici par exemple la liste des 5 premières tensions lues sur la voie 0 :

```
y=u0(1:5)
```

```
-0.1806641 -0.1171875 -0.0585938 -0.0048828 0.0634766
```

La fonction `CanTemps` permet d'obtenir les instants d'échantillonnage.

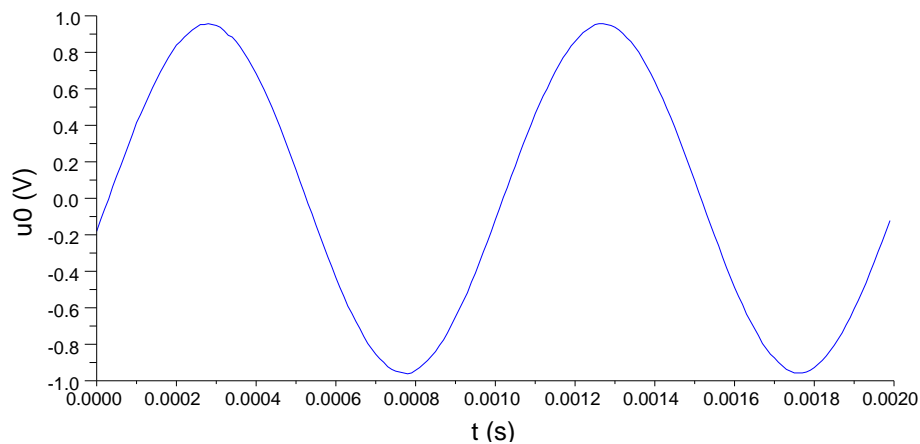
```
temps=CanTemps();
```

La matrice renvoyée a la même structure que la matrice des tensions. Pour le convertisseur SysamSP5, les différentes voies sont synchrones : les lignes de la matrice obtenue sont donc identiques. La matrice ligne des instants est donc obtenue par :

```
t=temps(1,:);
```

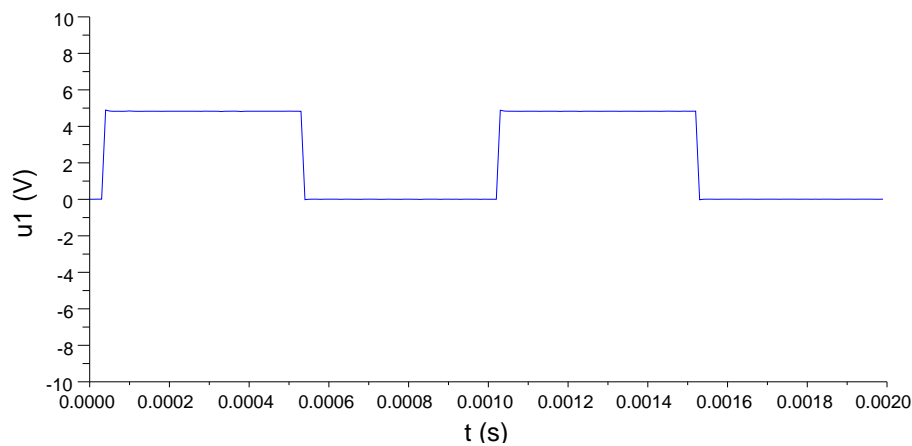
Tracé de `u0` en fonction du temps :

```
figA=scf();  
plot2d(t,u0,rect=[0,-1,0.002,1],style=2);  
xtitle('','t (s)','u0 (V)');
```



Tracé de `u1` en fonction du temps :

```
figB=scf();  
plot2d(t,u1,rect=[0,-10,0.002,10],style=2);  
xtitle('','t (s)','u1 (V)');
```



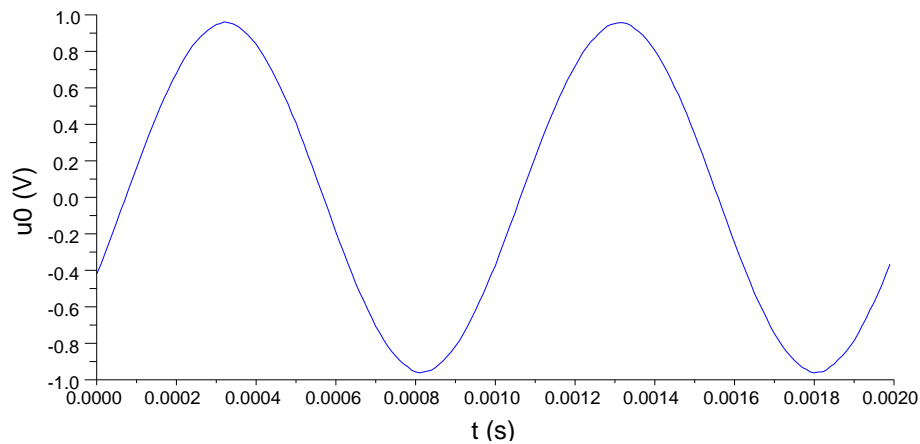
3. Utilisation des entrées avec échantillonnage et avec trigger

Le déclenchement de l'acquisition peut se faire à partir d'un signal de référence. Pour configurer une des voies configurées pour l'acquisition comme source de déclenchement, on utilise la fonction `CanConfigTrigger`. Dans le cas présent, le déclenchement se fait à partir de la voie 0 avec un seuil de 0.0 volts. Le déclenchement se fait sur un front montant avec 10 points mémorisés avant la condition de déclenchement (pretrigger) :

```
CanConfigTrigger(0,0.0,%T,10,%F)
```

La fonction `CanAcquerir` met le convertisseur en attente de la condition de déclenchement :

```
CanAcquerir()  
tensions=CanEntrees();  
temps=CanTemps();  
u0=tensions(1,:);  
t=temps(1,:);  
  
figC=scf();  
plot2d(t,u0,rect=[0,-1,0.002,1],style=2);  
xtitle('','t (s)','u0 (V)');
```

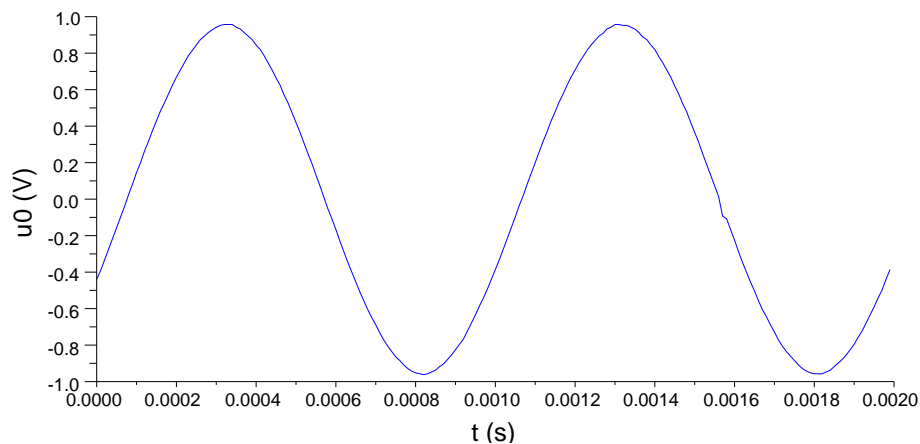


La source de déclenchement peut aussi être l'entrée SYNCHRO EXT du convertisseur. Dans ce cas, le signal doit être de type TTL et le déclenchement se fait sur un front montant :

```
CanConfigTriggerExterne(10,%T)

CanAcquerir()
tensions=CanEntrees();
temps=CanTemps();
u0=tensions(1,:);
t=temps(1,:);

figD=scf();
plot2d(t,u0,rect=[0,-1,0.002,1],style=2);
xtitle('','t (s)','u0 (V)');
```



4. Sauvegarde des tensions acquises

Pour utiliser ultérieurement les données acquises, il faut les sauvegarder sur fichier. Si les données doivent être traitées par Scilab, la sauvegarde sous fichier binaire est la solution la plus simple :

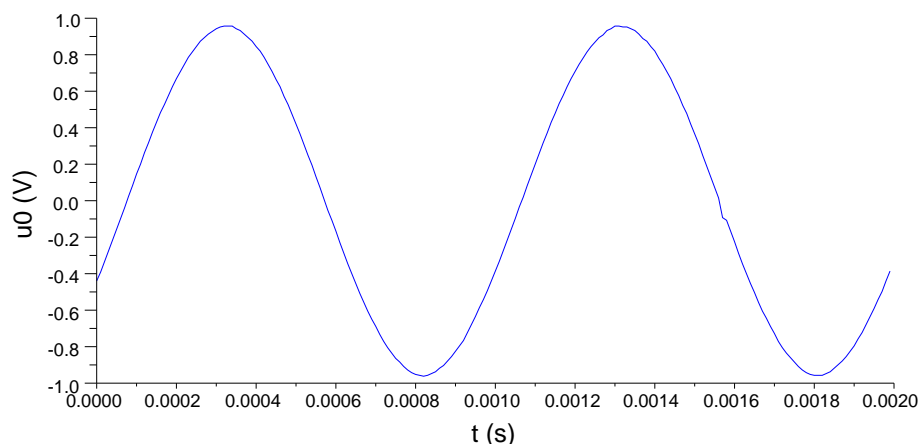
```
save('\users\fred\documents\scilab\electro\tensionsA.dat', tensions, temps)

clear tensions
clear temps
```

Voici comment les données sont récupérées :

```
load('\users\fred\documents\scilab\electro\tensionsA.dat', 'tensions', 'temps')

figE=scf();
plot2d(temps(1,:), tensions(1,:), rect=[0, -1, 0.002, 1], style=2);
xtitle('', 't (s)', 'u0 (V)');
```



5. Utilisation des sorties avec échantillonnage

Le convertisseur Sysam SP5 comporte deux sorties (2 convertisseurs numérique-analogique). Pour programmer les sorties, il faut d'abord calculer les tensions en volts (max +/- 10 volts) et les placer dans une matrice ligne. Par exemple, pour générer un signal périodique formé d'un fondamental et d'une harmonique de rang 2, il suffit de calculer les points sur une période :

```
n=1000;
t=linspace(0,10*%pi,n);
us1=2.0*sin(t)+1.0*sin(2*t);
us2=2.0*cos(t);
```

La fonction `CanConfigSortie` permet de configurer une sortie (numéro 1 ou 2) en précisant la période d'échantillonnage. Le dernier argument vaut +1 pour obtenir une seule émission de la liste fournie, -1 pour obtenir une répétition périodique. Le nombre maximal de points est 131071.

```
CanConfigSortie(1,10.0,us1,1)
CanConfigSortie(2,10.0,us2,1)
```

La fonction `CanDeclencherSorties` permet d'obtenir un déclenchement simultané des deux sorties :

```
CanDeclencherSorties(1,1)
```

Cette fonction retourne immédiatement. La dernière tension de la liste est maintenue en sortie après l'émission. Pour stopper les sorties, par exemple en cas d'émission périodique :

```
CanStopperSorties(1,1)
```

Cette dernière fonction doit également être utilisée avant toute reconfiguration des sorties.

Remarque importante : l'état des sorties est indéterminé avant l'ouverture de l'interface par la fonction `CanOuvrir`, après quoi les tensions sont nulles. Une saturation (-15 ou +15 V) est fréquemment observée.

6. Utilisation simultanée des entrées et sorties

Pour utiliser simultanément entrées et sorties, une première solution consiste à déclencher la ou les sorties comme précédemment puis à déclencher l'acquisition. L'inconvénient de cette méthode est le délai (non contrôlé) entre le début de l'émission des sorties et le début de l'acquisition.

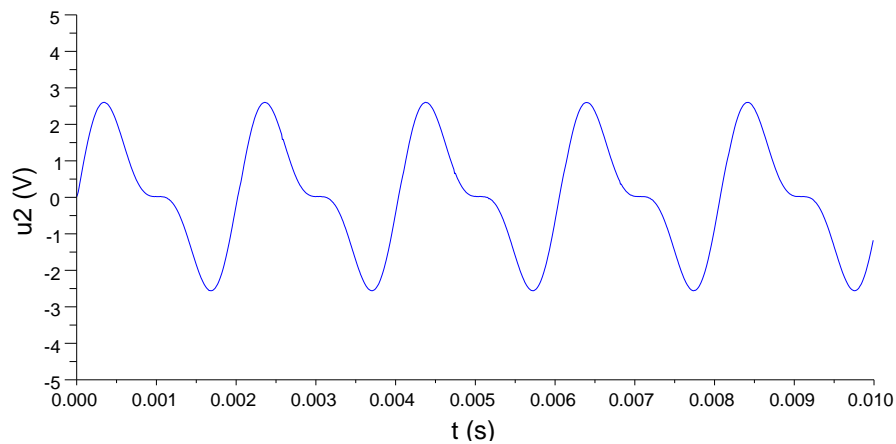
La seconde méthode consiste à utiliser la fonction `CanAcquerirAvecSorties`, qui permet de spécifier les valeurs à émettre sur les sorties, en simultané avec l'acquisition (même fréquence d'échantillonnage).

Dans l'exemple qui suit, la sortie SA1 est branchée sur l'entrée EA2. La sortie SA1 est programmée en simultané avec l'acquisition sur la voie 2 :

```
te=10.0
n=1000;
CanConfigEntrees([2],[5.0])
CanConfigEchantillon(te,n)
t=linspace(0,10*%pi,n);
us1=2.0*sin(t)+1.0*sin(2*t);
CanDesactiverTrigger()
CanAcquerirAvecSorties(us1,[])

tensions=CanEntrees();
u2=tensions(1,:);
temps=CanTemps();
t=temps(1,:);

figF=scf();
plot2d(t,u2,rect=[0,-5,n*te*1e-6,5],style=2);
xtitle('','t (s)','u2 (V)');
```



7. Utilisation des entrées sans échantillonnage

Le mode de lecture directe permet de lire les tensions à la volée. Dans ce cas, la lecture est déclenchée de manière logicielle. Ce mode est utile pour les périodes d'échantillonnages longues (supérieures à 1 secondes) ou pour les échantillonnages non réguliers. Il faut tout d'abord configurer les entrées comme précédemment puis activer le mode de lecture direct :

```
CanConfigEntrees([0,1],[5.0,10.0])
CanActiverLecture([0,1])
y=CanLire()
```

```
0.2978516
0
```

Pour exemple, voici un petit programme qui lit les deux entrées toutes les 5 secondes et place les valeurs dans une matrice :

```
realtimeinit(5.0)
realtime(0)
n=5
M=zeros(2,n);
for k=1:n,
    realtime(k);
    M(:,k)=CanLire();
end;
M
```

```
-0.2636719 -0.8544922 -0.8447266 -0.8837891 0.9521484
0.0097656 0.0097656 0.0097656 0.0048828 4.8339844
```

```
CanDesactiverLecture()
```

8. Utilisation des sorties sans échantillonnage

Il est possible d'appliquer une tension constante sur une des sorties (ou les deux). Ceci permet de commander un dispositif électrique. Pour appliquer 1.0 V sur la sortie 1 :

```
CanEcrire(1,2.0,1,0.0)
```

9. Fermeture de l'interface

```
CanFermer()
```