

Utilisation du convertisseur SysamSP5

1. Ouverture de l'interface

Pour charger le module :

```
Needs["CAN"]
```

Pour ouvrir la liaison avec le convertisseur Sysam SP5 :

```
CanOuvrir["SysamSP5"]
```

Une icône sysam apparaît dans la barre des tâches de Windows. En cas d'erreur, il est possible de fermer directement la liaison par cette icône, ou avec le gestionnaire des tâches.

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 CAN Sysam SP5 comporte 4 convertisseurs analogique/numérique (canal 0,1,2,3). Chaque canal peut être utilisé en mode différentiel. Par exemple le canal 0 utilisé en mode différentiel fournit la différence de potentiel entre les bornes EA0 et EA4. Pour programmer un mode différentiel sur un canal, il faut utiliser l'option `VoiesDifferentielles`. Par exemple, pour configurer le canal 0 en mode différentiel : `VoiesDifferentielles->{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[];
```

Cette fonction renvoie une liste de la forme $\{U1, U2\}$ où $U1$ est la liste des tensions acquises sur la première voie sélectionnée (ici EA0), $U2$ la liste des tensions acquises sur la seconde voie (ici EA1). Voici par exemple la liste des 5 premières tensions acquises sur EA0 :

```
u=Table[tensions[[1]][[k]],{k,5}]
```

```
{0.5078125, 0.5615234375, 0.6103515625, 0.654296875, 0.693359375}
```

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

```
temps=CanTemps[];
```

La liste renvoyée a exactement la même structure que la liste précédente. Pour la centrale SysamSP5, les listes de temps correspondant aux différentes voies sont identiques car les voies fonctionnent en parallèle, de manière synchrone. La liste des 5 premiers temps (en seconde) est :

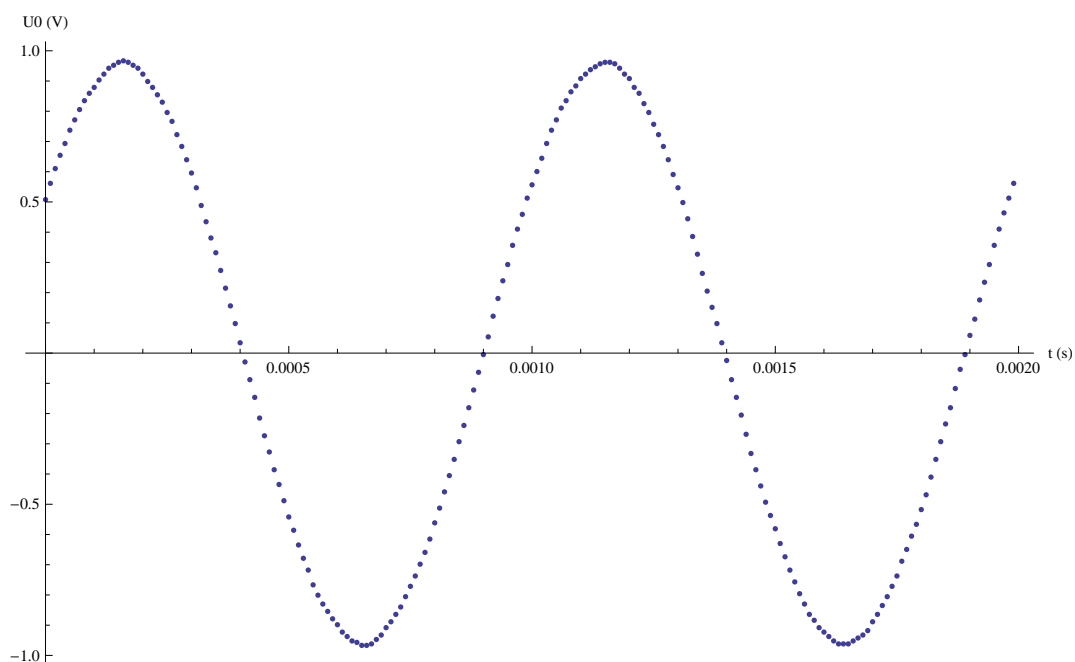
```
t=Table[temps[[1]][[k]],{k,5}]
```

```
{0., 9.999999999999999*^-6, 0.00001999999999999998, 0.00002999999999999997, 0.00003999999999999996}
```

Pour tracer la courbe d'une tension en fonction du temps, il faut tout d'abord générer une liste de paires x,y. Par exemple, la tensions EA0 en fonction de t est obtenue par :

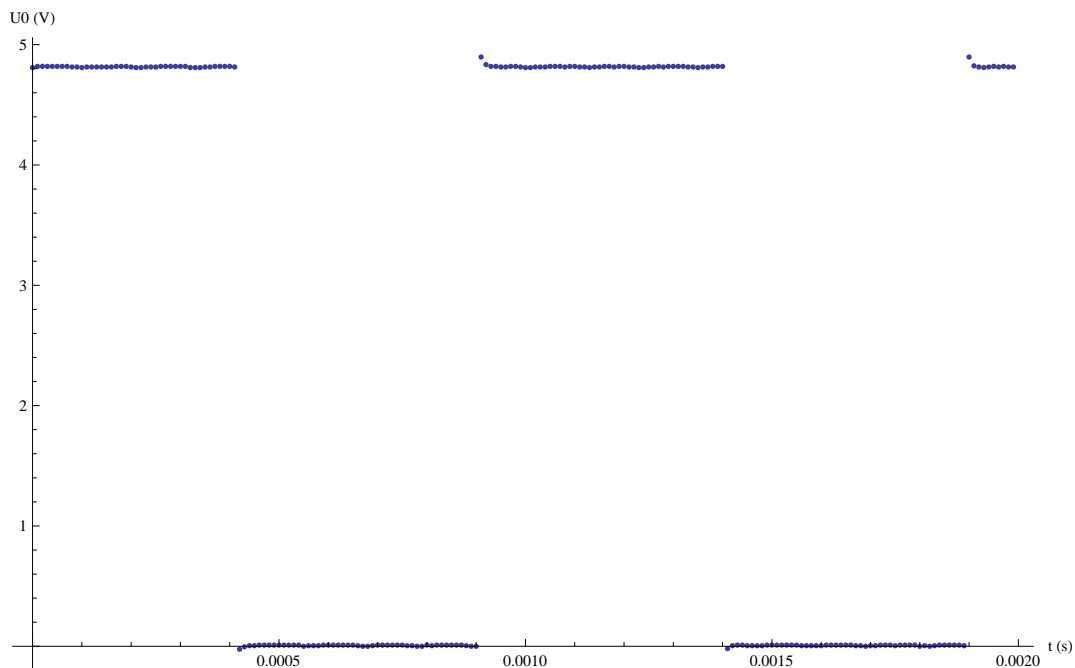
```
listeEA0=Table[{temps[[1]][[k]], tensions[[1]][[k]]}, {k, 1, Length[temps[[1]]]}];
```

```
ListPlot[listeEA0, AxesLabel -> {"t (s)", "U0 (V)"}, Joined -> False]
```



De même pour la seconde voie :

```
listeEA1=Table[{temps[[2]][[k]],tensions[[2]][[k]]},{k,1,Length[temps[[1]]]}];
ListPlot[listeEA1,AxesLabel->{"t (s)","U0 (V)"},Joined->False]
```



3. Utilisation des entrées avec échantillonnage et avec trigger

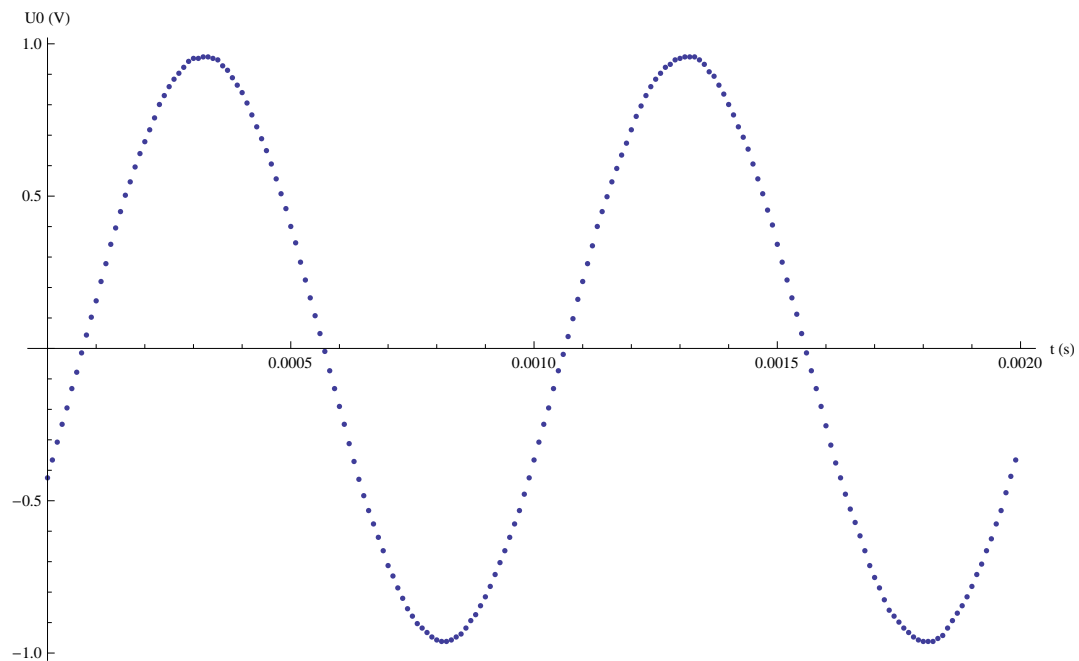
Le déclenchement de l'acquisition peut se faire à partir d'un signal de référence. Pour configurer une des voies configurées pour l'acquisition comme source de déclenchement, on utilise la fonction `CanConfigTrigger`. Dans le cas présent, le déclenchement se fait à partir de la voie 0 avec un seuil de 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,Montant->1,Pretrigger->10,PretriggerSouple->0]
```

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

```
CanAcquerir[]
tensions=CanEntrees[];
temps=CanTemps[];
listeEA0=Table[{temps[[1]][[k]],tensions[[1]][[k]]},{k,1,Length[temps[[1]]]}];

ListPlot[listeEA0,AxesLabel->{"t (s)","U0 (V)"},Joined->False]
```



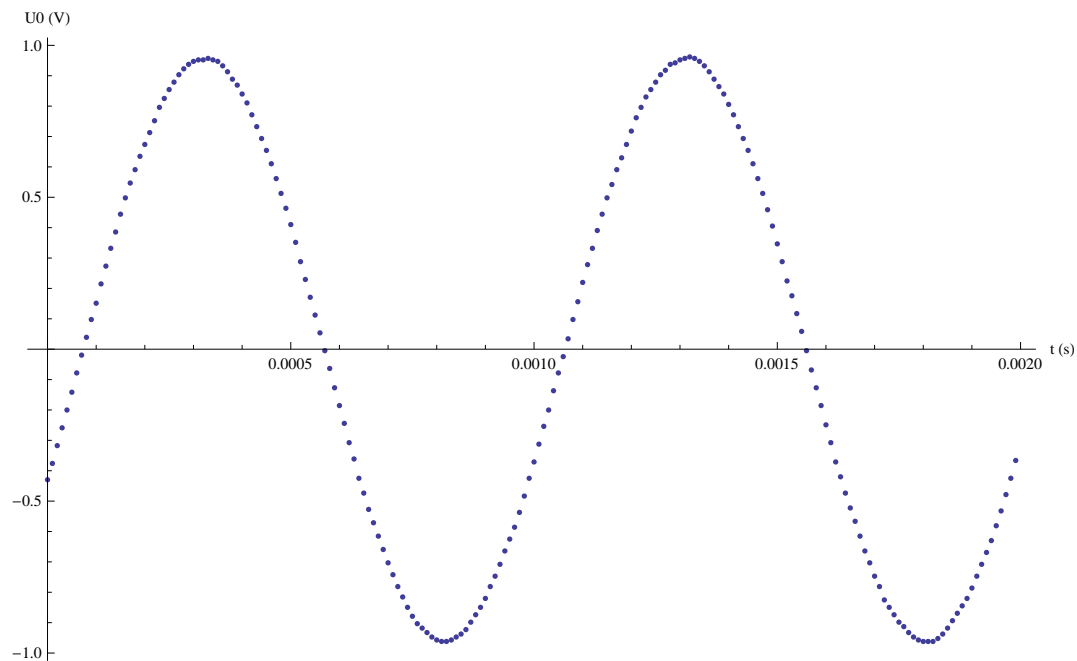
La source de déclenchement peut aussi être l'entrée SYNCHRO EXT du convertisseur. Dans ce cas, le signal doit être de type TTL et le déclenchement se fait sur un front montant.

```
CanConfigTriggerExterne[Pretrigger->10,PretriggerSouple->0]
```

Dans le cas présent, la sortie TTL OUT du GBF est utilisée comme source de déclenchement.

```
CanAcquerir[]  
tensions=CanEntrees[];  
temps=CanTemps[];  
listeEA0=Table[{temps[[1]][[k]],tensions[[1]][[k]]},{k,1,Length[temps[[1]]]}];
```

```
ListPlot[listeEA0,AxesLabel->{"t (s)","U0 (V)"},Joined->False]
```



4. Sauvegarde des tensions acquises

Pour utiliser les données acquises dans un autre notebook ou dans une autre session, il faut les sauvegarder sur fichier :

```
Export["tensions1.dat", tensions, "Table"]  
Export["temps1.dat", temps, "Table"]
```

Pour récupérer les données :

```
tensions=Import["tensions1.dat", "Table"];  
temps=Import["temps1.dat", "Table"];
```

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 liste. 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;  
s1 = Table[N[Sin[2*Pi*k/n] + 0.5*Sin[4*Pi*k/n]], {k, 0, n}];
```

Pour la seconde voie, on choisit de générer une rampe :

```
s2 = Table[N[1.0*k/n], {k, 0, n}];
```

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.

Par exemple, pour programmer une seule émission de la liste `s1` sur la voie 1 et de la liste `s2` sur la voie 2 avec un temps d'échantillonnage de 10 microsecondes :

```
CanConfigSortie[1, 10.0, s1, 1]  
CanConfigSortie[2, 10.0, s2, 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 (ramener leur tension à 0), 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

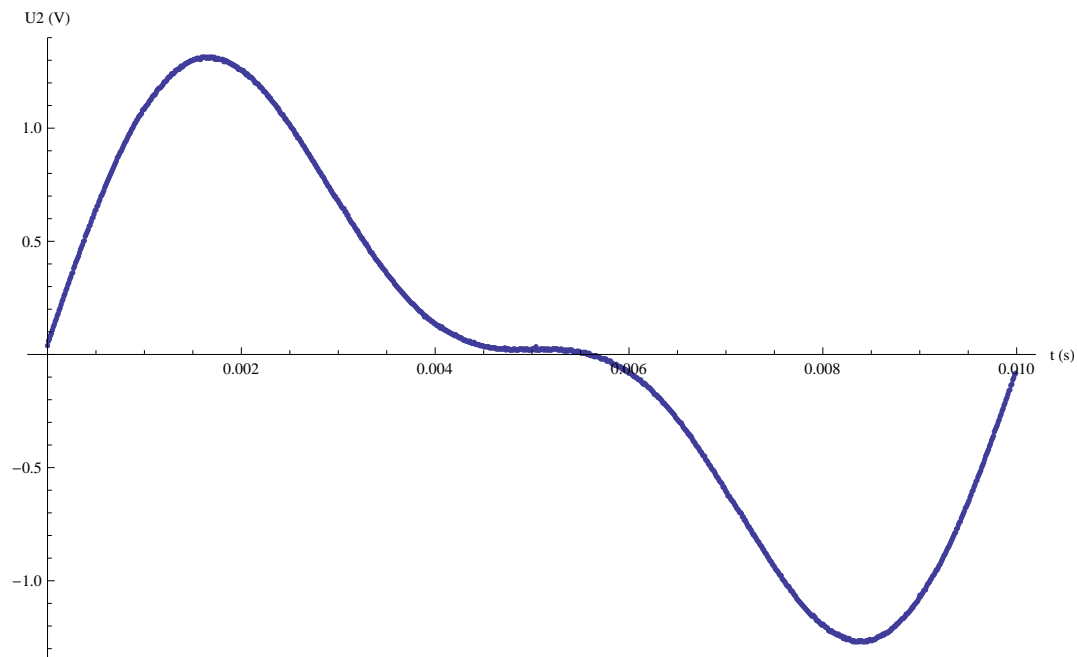
Pour utiliser simultanément entrées et sorties, une première solution consiste à déclencher la ou les sorties comme précédemment puis à déclencher l'acquisition. L'inconvénient de cette méthode est le délai (non contrôlé) entre le début de l'émission des sorties et le début de l'acquisition.

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;  
s1 = Table[N[Sin[2*Pi*k/n]+0.5*Sin[4*Pi*k/n]], {k, 1, n}];  
CanConfigEntrees[{2}, {10.0}]  
CanConfigEchantillon[te, n]  
CanDesactiverTrigger[]
```

```
CanAcquerirAvecSorties[s1, {}]  
tensions = CanEntrees[];  
temps = CanTemps[];  
listeEA2 = Table[{temps[[1]][[k]], tensions[[1]][[k]]}, {k, 1, Length[temps[[1]]]}];  
  
ListPlot[listeEA2, AxesLabel -> {"t (s)", "U2 (V)"}, Joined -> False]
```



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'échantillonnage 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}]
```

Pour effectuer la lecture :

```
y=CanLire[]
```

```
{0.8251953125, 4.814453125}
```

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

```
liste = {};  
For[i = 0, i < 5, i++,  
    liste = Append[liste, CanLire[]];  
    Pause[5];  
]  
liste
```

```
{{-0.8056640625, 0.009765625}, {0.83984375, 4.8193359375}, {0.1416015625, 4.814453125}}
```

```
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 2.0 V sur la sortie 1 et 0.0 V sur la sortie 2 :

```
CanEcrire[1,2.0,1,0.0]
```

9. Fermeture de l'interface

```
CanFermer[]
```