

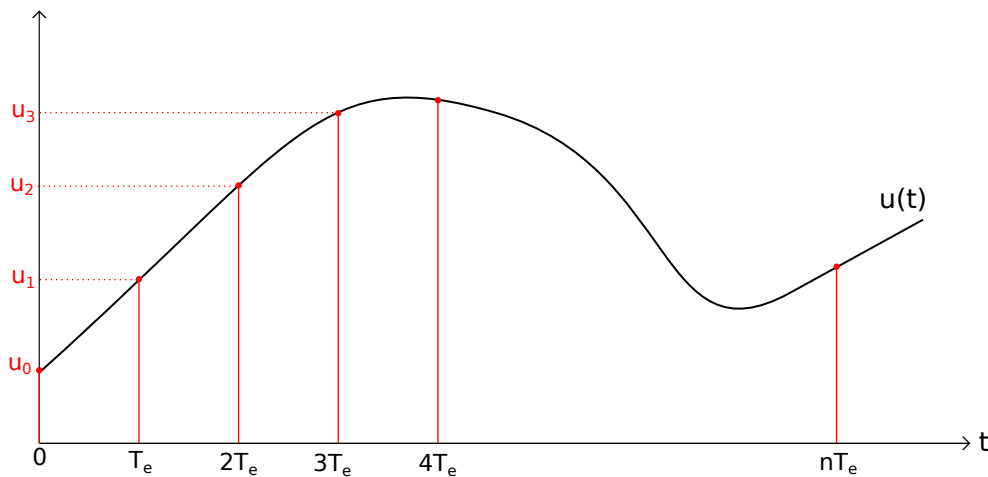
# TP 1 - Échantillonnage des signaux périodiques

## 1. Introduction

L'objectif est d'apprendre à utiliser le générateur de fonctions, l'oscilloscope numérique et la carte d'acquisition. L'utilisation de ces appareils permettra d'introduire la notion d'échantillonnage d'un signal analogique. Nous verrons aussi la condition de Nyquist-Shannon, qui précise la fréquence d'échantillonnage minimale à respecter pour échantillonner un signal sinusoïdal puis un signal périodique.

## 2. Échantillonnage

Un signal analogique  $u(t)$  est une grandeur physique, par exemple une tension électrique, qui varie au cours du temps. Lors de la numérisation d'un signal, on effectue un *échantillonnage*, consistant à prélever des valeurs du signal à intervalle de temps régulier. L'intervalle de temps entre deux échantillons est la *période d'échantillonnage*, qu'on notera  $T_e$ . Les échantillons obtenus forment une suite de nombres  $u_n = u(nT_e)$ , qui constitue un *signal numérique*, appelé aussi *signal échantillonné*. Alors que le signal analogique est défini pour toute valeur du temps (signal à temps continu), le signal numérique n'est défini que pour des instants multiples de la période d'échantillonnage (signal à temps discret). La figure suivante montre le principe de l'échantillonnage idéal :



Un signal numérique offre trois avantages par rapport au signal analogique :

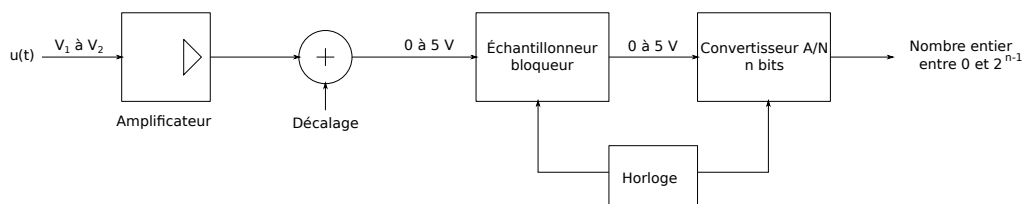
- ▷ Il peut être traité, c'est-à-dire transformé, par un ordinateur qui effectue des opérations sur la suite de nombres. On parle de *traitement numérique du signal*.
- ▷ Le stockage sous forme numérique (par exemple le son sur un CD audio) est plus fiable que le stockage analogique. D'autre part, un signal numérique peut être stocké dans un fichier informatique.
- ▷ La transmission (par exemple par onde radio) est plus fiable sous forme numérique.

La numérisation des signaux intervient dans de très nombreux appareils. On peut citer :

- ▷ Oscilloscope numérique.
- ▷ Carte d'acquisition pour ordinateur.
- ▷ Enregistrement et transmission du son, de l'image (téléphonie, télévision, etc.).
- ▷ Capteurs pour l'aviation, l'automobile, l'industrie, la robotique, etc.

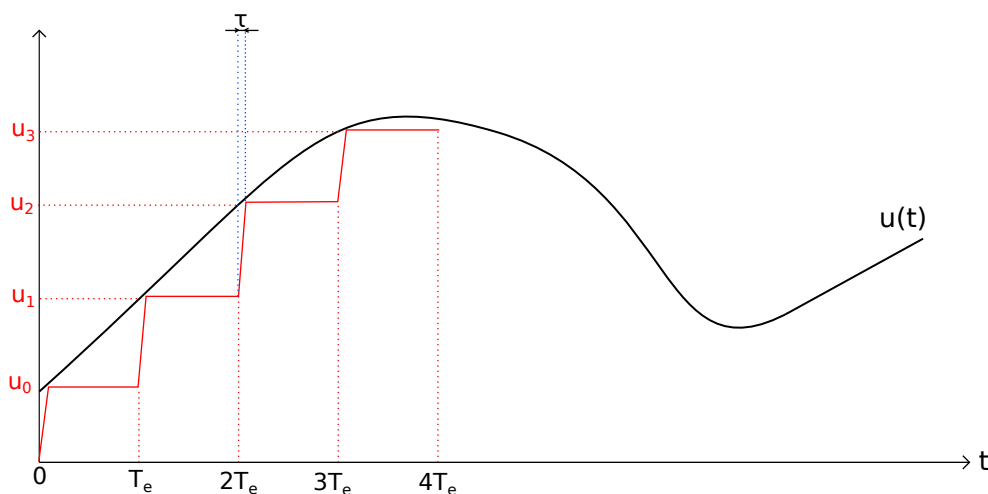
### 3. Échantillonnage-blocage et conversion analogique-numérique

La figure suivante montre le schéma d'un système électronique d'échantillonnage et de conversion analogique/numérique.



Le signal analogique est une tension  $u(t)$  comprise dans l'intervalle  $[V_1, V_2]$ . Cette tension est tout d'abord traitée par un amplificateur analogique, suivi de l'application d'une tension de décalage (offset) qui permet de ramener la tension dans un intervalle adapté au convertisseur A/N, typiquement 0 à 5 volts.

L'échantillonnage idéal défini précédemment ne peut être réalisé car la conversion analogique-numérique ne peut être instantanée. L'échantillonneur-blocage est un circuit chargé de bloquer la valeur de la tension pendant la durée qui permet au convertisseur A/N de faire la conversion, durée qui est inférieure à la période d'échantillonnage. La figure suivante montre l'allure de la tension en sortie de l'échantillonneur-blocage :



L'échantillonneur-blocage est piloté par l'horloge. À l'instant  $t_n = nT_e$ , la tension de sortie atteint une valeur proche de  $u_n = u(t_n)$  avec un temps de réponse ( $\tau$ ) qui doit être faible devant  $T_e$  puis elle reste bloquée à cette valeur jusqu'à l'instant  $t_{n+1}$ , ce qui laisse le temps d'effectuer la conversion analogique-numérique. En raison du temps de réponse de l'échantillonneur-blocage, la valeur convertie est légèrement différente de  $u_n = u(t_n)$  (plus grande si  $u(t)$  est

croissante), mais on peut en première approximation considérer que l'écart est négligeable, ce qui revient à considérer que l'échantillonnage est idéal.

Le convertisseur A/N à  $n$  bits convertit la tension comprise entre 0 et 5 volts en un nombre entier compris entre 0 et  $2^n - 1$ . La tension  $u(t)$ , comprise entre  $V_1$  et  $V_2$ , est donc convertie en un nombre entier compris entre 0 et  $2^n - 1$ . Dans la plupart des cas, on s'intéresse à la tension  $u(t)$  elle-même; le logiciel qui traite ces nombres les convertit donc en nombres réels (nombres à virgule flottante) qui représentent la tension  $u(t)$ .

La conversion d'une valeur réelle en nombre entier est une opération appelée *quantification*. Plus le nombre de bits utilisés pour la quantification est grand, plus celle-ci est précise. En effet, la résolution en tension pour un convertisseur à  $n$  bits est

$$\frac{V_2 - V_1}{2^n}$$

Considérons comme exemple un signal de faible amplitude, dont les valeurs sont comprises entre  $V_1 = -100$  mV et  $V_2 = 100$  mV. Le gain de l'amplificateur doit être 25, ce qui conduit à une tension comprise entre -2,5 et 2,5 volts, et un décalage de 2,5 volts doit être appliqué pour amener la tension dans l'intervalle [0,5] volts. Avec un convertisseur A/N 8 bits, la résolution est 0,78 mV. Avec un convertisseur 12 bits, elle est 0,048 mV.

Un convertisseur A/N a une fréquence d'échantillonnage maximale, égale à l'inverse de la durée de la conversion. Il est difficile d'avoir à la fois une numérisation rapide et très précise. Il existe des systèmes de numérisation précis mais lents et des systèmes rapides mais moins précis. Pour la numérisation du son, la fréquence d'échantillonnage n'a pas besoin d'être très élevée (maximum 96 kHz) mais la précision doit être très grande (au moins 16 bits). Il existe des systèmes de numérisation très rapides (par ex. l'oscilloscope), avec une fréquence d'échantillonnage maximale de l'ordre du GHz, mais leur précision est généralement faible (8 ou 10 bits).

L'horloge pilote l'échantillonneur et le convertisseur afin que la conversion se fasse avec une période égale à la période d'échantillonnage. Nous allons voir comment la fréquence d'échantillonnage doit être choisie, en fonction de la fréquence du signal à numériser.

## 4. Générateur de fonctions

Le générateur de signaux (SIGLENT SDG1025) permet de générer des signaux périodiques. Il s'agit d'un générateur numérique : les signaux sont générés sous forme numérique avant de subir une conversion numérique/analogique. Ce type de générateur permet un contrôle très précis de la fréquence des signaux.

Il comporte deux sorties programmables séparément mais délivrant deux signaux parfaitement synchrones. Dans ce TP, nous utiliserons seulement la sortie CH1.



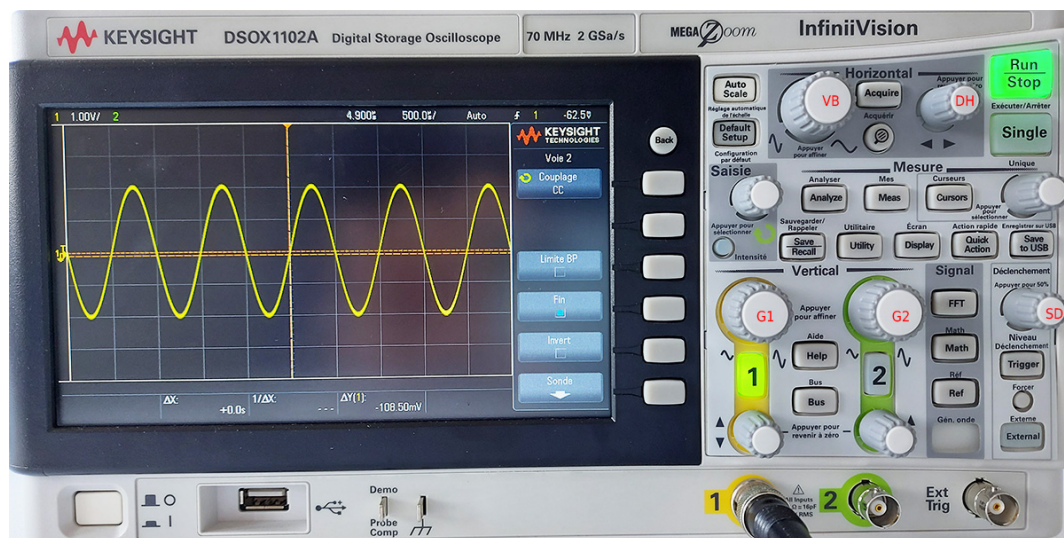
On doit sélectionner la sortie (bouton CH1/2), choisir la forme du signal, ajuster sa fréquence et son amplitude, puis éventuellement lui appliquer un décalage (offset). Pour modifier une valeur (par exemple la fréquence) on peut soit tourner la molette pour modifier le chiffre sélectionné, soit entrer la valeur souhaitée directement au clavier. Attention : une sortie n'est active qu'après appuie sur le bouton OUTPUT situé juste au dessus du connecteur BNC.

La fréquence affichée sur l'écran est exactement celle du signal délivré, avec une précision correspondant aux chiffres significatifs.

## 5. Oscilloscope numérique

### 5.a. Principe de fonctionnement

On utilise le modèle Keysight DSOX1102A à deux voies dont le panneau est reproduit ci-dessous (les légendes VB, G1, G2 et DH sont rajoutées sur la photo).



L'oscilloscope numérique est principalement constitué d'un amplificateur analogique, d'un convertisseur analogique/numérique (CAN 8 bits), d'une mémoire RAM, et d'un microprocesseur.

L'amplificateur sert à augmenter ou à réduire l'amplitude du signal analogique, afin de l'adapter à l'amplitude de fonctionnement du CAN. Chaque voie comporte son propre amplificateur. Le gain de l'amplificateur se modifie avec le bouton de réglage de l'échelle verticale (bouton

G1 pour la voie 1, G2 pour la voie 2). La bande passante de l'amplificateur est 70 MHz. À cette fréquence, un signal sinusoïdal visualisé est atténué de -3 dB par rapport au signal réel.

Le convertisseur analogique/numérique effectue les conversions à la fréquence d'échantillonnage, et fournit une suite de nombres stockée dans la mémoire RAM, l'intervalle de temps entre deux nombres étant la période d'échantillonnage. Ces nombres sont des entiers codés sur 8 bits, c'est-à-dire compris entre 0 et  $2^8 - 1 = 255$ . Si on note  $[V_1, V_2]$  la plage de tension obtenue par le réglage vertical, la valeur 0 correspond à  $V_1$ , la valeur 255 correspond à  $V_2$ . La résolution en tension est donc  $(V_2 - V_1)/256$ .

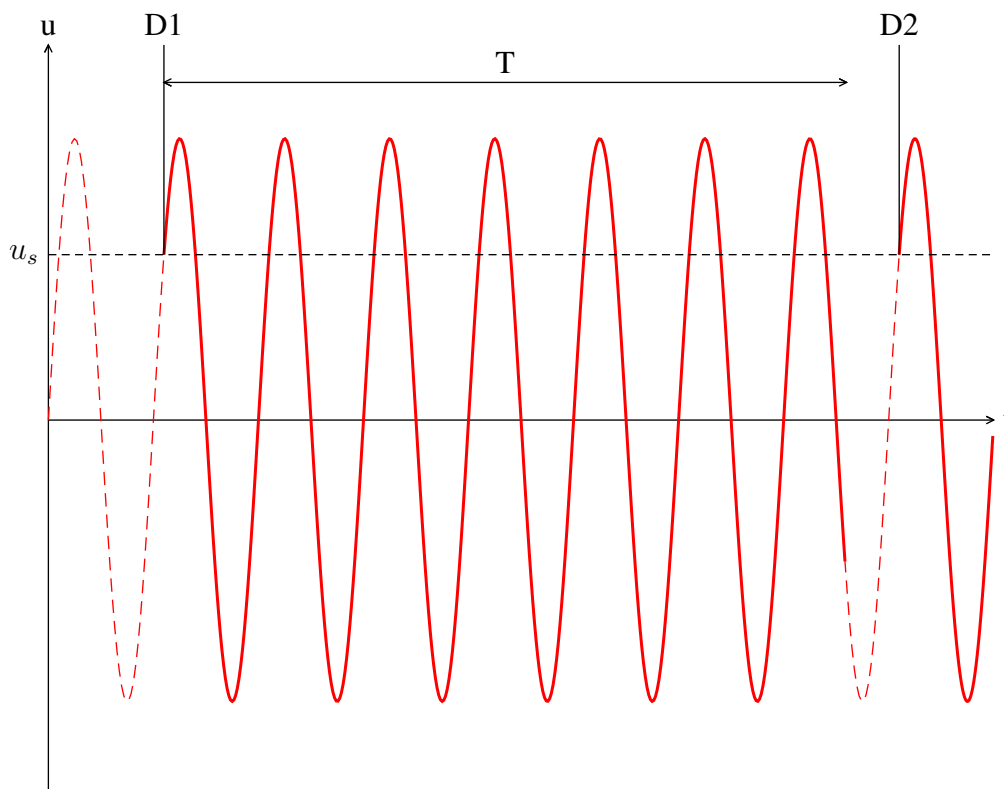
Le microprocesseur est chargé de traiter en temps réel le signal numérique enregistré en mémoire, et d'effectuer le tracé de la courbe. Les nombres entiers du signal numérique sont convertis en valeurs de tension, en utilisant l'échelle  $[V_1, V_2]$  du réglage vertical.

La fréquence d'échantillonnage maximale est 2 GSa/s, c'est-à-dire 2 giga échantillons par secondes, ou encore 2 GHz.

### 5.b. Acquisition

Pour tracer la courbe, le processeur effectue une acquisition du signal numérique (en lisant les données en mémoire) pendant une durée  $T = NT_e$ , comportant  $N$  échantillons. Le nombre d'échantillons est fixe, et la période d'échantillonnage se règle avec le bouton de vitesse de balayage (marqué VB sur la photo ci-dessous), qui ajuste l'échelle horizontale en s/div.

L'acquisition est répétée périodiquement afin qu'une modification du signal (fréquence ou amplitude) soit visible en temps réel sur l'écran. Pour obtenir une représentation stable d'un signal périodique, il faut que le déclenchement de l'acquisition soit synchronisé avec le signal. La figure suivante montre le principe du déclenchement de l'acquisition par seuil, dans le cas d'un signal sinusoïdal.



Considérons un déclenchement  $D1$ . La durée  $T$  de l'acquisition est un multiple de la période d'échantillonnage mais elle n'est pas en général multiple de la période du signal. On voit que le déclenchement de l'acquisition suivante ne doit pas se faire juste après la fin de cette durée. L'instant correct pour le déclenchement suivant est  $D2$  car la phase du signal en  $D2$  est la même qu'en  $D1$ . Pour obtenir cela, on effectue un déclenchement par seuil. La tension de seuil étant  $u_s$ , le déclenchement a lieu (une fois l'acquisition terminée) lorsque le signal franchit le seuil par valeur croissante. Le signal utilisé est appelé la source du déclenchement : ce peut être le signal de la voie 1, de la voie 2 ou un signal externe (entrée déclenchement externe).

[1] Utiliser le générateur de signaux pour générer un signal sinusoïdal d'environ 1 kHz et d'amplitude de crête à crête 10 V, à observer avec la voie 1 de l'oscilloscope.

[2] Activer la voie 1 en appuyant sur la touche 1. Sélectionner dans le menu le couplage CC. Ajuster l'échelle verticale et l'échelle horizontale pour les adapter à l'amplitude et à la fréquence du signal. Attention : il est important de toujours lire l'échelle de temps affichée en haut de l'écran et de vérifier qu'elle est cohérente avec la fréquence prévue du signal.


[3] Entrer dans le menu du déclenchement (Trigger). Choisir le mode Front (déclenchement par front), choisir la source du déclenchement, la pente (front montant ou descendant) et le mode Normal.


[4] Observer l'effet d'un changement du seuil de déclenchement, appelé aussi niveau de déclenchement (bouton SD). Repérer le point de déclenchement au centre de l'écran. Ce point peut être déplacé avec le bouton DH (décalage horizontal).

[5] Que se passe-t-il lorsque le seuil ne se trouve pas dans la plage de tension du signal ? Essayer avec le mode de déclenchement Auto (qui signifie déclenchement automatique). On préfère en général le mode Auto car il permet de voir une trace en temps réel même lorsque la synchronisation ne peut se faire.


### 5.c. Couplage d'entrée

Il y a deux modes de couplage pour l'entrée : le couplage CC (courant continu) et le couplage CA (courant alternatif). En couplage CC, le signal d'entrée est traité directement. En couplage CA, un filtre passe-haut est interposé entre l'entrée et l'amplificateur. Il s'agit d'un simple filtre RC passe-haut, dont la fréquence de coupure est égale à 10 Hz. Il sert à éliminer une éventuelle composante continue (la partie constante) du signal.

[6]  Faire le schéma du filtre RC passe-haut. Sachant que la résistance est de 1 MΩ (résistance d'entrée de l'amplificateur), calculer la capacité du condensateur.

[7]  Pour voir l'effet du couplage CA, modifier le réglage du générateur afin d'ajouter un décalage (Offset) au signal sinusoïdal et réduire l'amplitude de la sinusoïde à quelques dizaines de millivolts. Comparer les couplages CA et CC. Quel est l'intérêt du couplage CA dans ce cas ?

[8]  Observer la même tension sinusoïdale sur la voie 1 en couplage CC et sur la voie 2 en couplage CA. Mesurer l'atténuation du filtre RC à 10 Hz.

[9]  Essayer le couplage CA avec un signal de forme carrée (square) d'une fréquence d'environ 100 Hz. Que constate-t-on ? À partir de quelle fréquence la forme du signal carré n'est-elle pas modifiée par le couplage CA ?

### 5.d. Fréquence d'échantillonnage

La fréquence d'échantillonnage est affichée dans le menu principal à droite de l'écran (appuyer sur back pour la faire apparaître).

[10] Observer l'influence de la vitesse de balayage (bouton VB) sur la fréquence d'échantillonnage.

[11] Pour un signal de fréquence 1 kHz dont on visualise 5 périodes, calculer le nombre d'échantillons visibles sur l'écran. En déduire le nombre d'échantillons par période.

[12] Refaire la mesure pour un signal de fréquence 1 MHz.

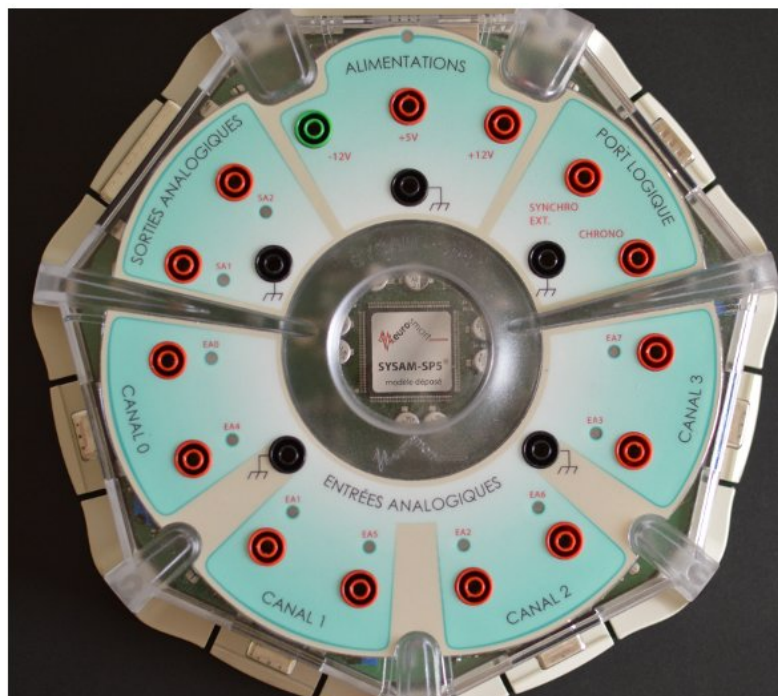
Le nombre d'échantillons par période est très grand. On parle alors de *sur-échantillonnage*. Pour tracer facilement la courbe représentant le signal, il est préférable, lorsque c'est possible, d'effectuer un sur-échantillonnage avec au moins 100 échantillons par période. Dans ce cas, une simple interpolation linéaire entre deux échantillons consécutifs permet d'obtenir une bonne reconstruction du signal analogique.

La fréquence d'échantillonnage maximale est de 2 GHz alors que la bande passante est de 70 MHz (ces valeurs sont écrites au dessus de l'écran). La bande passante est la fréquence à laquelle le signal subit une atténuation de  $-3$  décibel, à cause de la partie analogique du traitement, principalement l'amplificateur. Au delà de cette fréquence, l'atténuation est de plus en plus forte.

## 6. Carte d'acquisition

### 6.a. Fonctionnement et caractéristiques

On utilise la carte d'acquisition SYSAM-SP5, reliée à l'ordinateur par USB.



Elle comporte 4 canaux, chacun étant pourvu d'un amplificateur et d'un convertisseur analogique-numérique (CAN). Les amplificateurs ont 4 gains programmables, qui donnent les calibres suivants :  $\pm 10 V$ ,  $\pm 5 V$ ,  $\pm 1 V$  et  $\pm 0,2 V$ . Les CAN ont une résolution de 12 bits et fournissent donc des entiers compris entre 0 et  $2^{12} - 1 = 4095$ .

La fréquence d'échantillonnage maximale des CAN est de 10 MHz. Pour bénéficier de cette fréquence, il faut néanmoins n'utiliser qu'une entrée par canal. En utilisera donc les entrées de préférence dans l'ordre EA0, EA1, EA2, EA3.

La carte possède une mémoire RAM pouvant stocker  $2^{18} = 262142$  échantillons de 12 bits. Elle est remplie en temps réel pendant l'acquisition. Le signal numérique contenu dans cette mémoire est ensuite transféré à l'ordinateur via la liaison série USB.

La carte d'acquisition a une fréquence d'échantillonnage maximale beaucoup moins grande que l'oscilloscope (200 fois moins); elle n'est donc utilisable que pour des signaux de fréquence relativement basse. En revanche, sa précision de numérisation (12 bits au lieu de 8 bits) est  $2^4 = 16$  fois plus grande.

L'autre avantage de la carte est de fournir directement des données numériques, qui peuvent être facilement traitées, par exemple avec le langage Python. Elle offre aussi la possibilité d'effectuer des mesures de tension différentielle, ce que l'oscilloscope ne fait pas. Avec l'oscilloscope, on mesure toujours une tension entre un point et la masse. Avec la carte en mode différentiel, on peut mesurer une tension entre deux points quelconques du circuit.

### 6.b. Utilisation de Python

Nous allons effectuer l'acquisition en Python, ce qui nous permettra d'effectuer plus tard des traitements numériques du signal. Pour cela, on utilise le module `pycanum`, que l'on importe de la manière suivante :

```
import pycanum.main as pycan
```

On doit commencer par établir une liaison avec la carte d'acquisition, en déclarant un objet de la classe `Sysam` :

```
can = pycan.Sysam("SP5")
```

On doit ensuite configurer les entrées que l'on veut utiliser, en précisant la valeur de la tension maximale que l'on va appliquer sur chaque voie (celle-ci ne peut dépasser 10 V). Par exemple, pour configurer l'entrée EA0 avec un maximum de 10 V, on écrit :

```
can.config_entrees([0],[10.0])
```

Il faut choisir la période d'échantillonnage et le nombre de points à acquérir. On peut par exemple fixer d'abord la fréquence d'échantillonnage et la durée  $T$  de l'acquisition, puis calculer le nombre de points :

```
fe=5000.0
T=2.0
te=1.0/fe
N = int(T/te)
print(N)
```

Il faut vérifier que la période d'échantillonnage n'est pas inférieure à  $0,1 \mu\text{s}$  et que le nombre de points n'excède pas 262142, le maximum possible lorsqu'on utilise une voie (deux fois moins pour deux voies, etc.).

La configuration de l'échantillonnage se fait alors par la commande suivante, avec une période d'échantillonnage donnée en microsecondes :

```
can.config_echantillon(te*10**6,N)
```

On doit enfin déclencher l'acquisition. Il est possible d'effectuer un déclenchement par seuil, comme avec l'oscilloscope. On se contente pour l'instant d'un déclenchement logiciel :



```
can.acquerir()
```

Pour récupérer les instants et les tensions sous forme de tableaux de type `numpy.ndarray`, on écrit :

```
t=can.temps()[0]  
u0=can.entrees()[0]
```

Les valeurs de tensions obtenues sont exprimées en volts. Les nombres entiers fournis par le CAN ont donc été convertis en nombres à virgule flottante.

Pour finir, il est bon de fermer la liaison avec la carte :

```
can.fermer()
```

On peut sauvegarder les données obtenues dans un fichier, de la manière suivante :

```
numpy.save('nom_fichier.npy', numpy.array([t,u0]))
```

Pour les relire, il suffira d'écrire

```
[t,u0]=numpy.load('nom_fichier.npy')
```

ce qui a pour effet de reconstituer des tableaux `t` et `u0` identiques aux tableaux enregistrés.

Le format NPY est un format de fichier binaire, qui ne peut être lu que par `numpy`. Si on veut que les données soient lues par un autre logiciel (par exemple un tableur), il faudra utiliser la fonction `numpy.savetxt`, qui permet d'enregistrer le tableau au format texte (fichier TXT ou CSV). Le format binaire est beaucoup plus compact que le format texte, ce qui est un avantage pour le stockage des signaux et leur transfert vers le Jupyter Hub.

Il est possible de déclencher l'acquisition au moyen d'un seuil, en ajoutant les lignes suivantes juste avant `can.acquerir()` :

```
seuil = 0.0 # seuil de déclenchement  
voie = 0 # voie sur laquelle se fait le déclenchement  
sys.config_trigger(voie, seuil, montant=1, pretrigger=1)
```

### 6.c. Condition de Nyquist-Shannon

Condition de Nyquist-Shannon pour un signal sinusoïdal : pour échantillonner convenablement une sinusoïde de fréquence  $f$ , la fréquence d'échantillonnage doit être strictement supérieure au double de la fréquence de la sinusoïde :

$$f_e > 2f$$

Autrement dit, le nombre d'échantillons par période doit être strictement supérieur à 2.

Un nombre d'échantillons par période égal à 2 est insuffisant mais un nombre légèrement supérieur à 2 (par exemple 2,1) convient. Lorsque cette condition n'est pas vérifiée, il y a *sous-échantillonnage*. Lorsque la condition est très largement vérifiée, il y a *sur-échantillonnage*. Si la condition est remplie, il est théoriquement possible de reconstituer complètement la sinusoïde à partir des échantillons, une opération nommée *reconstruction du signal*.


### 6.d. Numérisation d'un signal sinusoïdal


Pour cette partie, on écrira un script Python au moyen de l'IDE IDLE Python 3.9, accessible depuis le dossier Math-Info situé sur le bureau.

On effectue avec la carte SysamSP5 la numérisation d'un signal sinusoïdal, que l'on contrôle en parallèle avec l'oscilloscope. On fixe la fréquence d'échantillonnage à  $f_e = 1000$  Hz et on fait varier la fréquence  $f$  du signal sinusoïdal. La durée de l'acquisition est environ  $10/f$ , c'est-à-dire 10 périodes du signal.


[13] Écrire un script Python qui effectue l'acquisition et trace le signal échantillonné avec la fonction `matplotlib.pyplot.plot`. Les points sont reliés par des segments, ce qui correspond à une reconstruction par interpolation linéaire.

[14] Faire une acquisition d'une sinusoïde d'environ 10 Hz (sur-échantillonnage) et tracer la tension en fonction du temps sous forme de points reliés par des segments. Remarque : ne pas régler la fréquence du signal à exactement 10 Hz, car il s'agit d'un cas trop particulier (fréquence sous-multiple de la fréquence d'échantillonnage).


[15]  Choisir une fréquence  $f$  telle que  $f_e$  soit à peine supérieure à  $2f$  (d'une dizaine de Hertz). Faire l'acquisition et tracer le signal échantillonné. Peut-on déterminer la fréquence en comptant le nombre de cycles sur une durée donnée ?

[16]  Choisir une fréquence  $f$  telle que la condition de Nyquist-Shannon ne soit pas respectée. Faire l'acquisition et tracer le signal échantillonné. Peut-on déterminer la fréquence en comptant le nombre de cycles sur une durée donnée ?

Comme on le voit sur l'exemple précédent, le respect de la condition de Nyquist-Shannon permet de conserver l'information de fréquence du signal mais ne suffit pas pour obtenir facilement la reconstruction du signal analogique. Pour faire une bonne reconstruction par interpolation linéaire, il faut que la fréquence d'échantillonnage soit grande devant la fréquence du signal (sur-échantillonnage). Lorsque le nombre d'échantillons par période est supérieur à 2 mais pas très grand, la reconstruction est théoriquement possible mais difficile à réaliser, car elle requiert un traitement numérique beaucoup plus lourd, avec un algorithme d'interpolation plus complexe ou un traitement par filtrage que nous verrons dans un prochain TP.

[17]  L'emploi d'une fréquence d'échantillonnage beaucoup trop faible peut conduire à un résultat étrange. Par exemple, faire l'acquisition d'une sinusoïde d'exactly 5010 Hz à une fréquence d'échantillonnage  $f_e = 1000$  Hz. Quelle fréquence mesure-t-on par observation du signal échantillonné ?

L'exemple précédent montre que, en cas de sous-échantillonnage, le signal reconstruit par interpolation linéaire peut avoir l'apparence d'une sinusoïde quasi parfaite mais dont la fréquence est beaucoup plus basse que celle du signal analogique. Pour détecter cette situation, il suffit de changer la fréquence d'échantillonnage.

[18]  Refaire l'acquisition avec  $f_e = 2000$  Hz et mesurer la fréquence sur le signal échantillonné. Que constate-t-on d'anormal ?

Ce type d'erreur dû à un sous-échantillonnage arrive parfois avec un oscilloscope ; il faut donc toujours vérifier que l'échelle de temps est cohérente avec l'ordre de grandeur de la fréquence attendue du signal.

### 6.e. Notebook sur le Jupyter Hub

Cette partie montre comment présenter un travail de traitement de données dans un notebook Python sur le [Jupyter Hub](#).

[19] Ouvrir une session sur le Jupyter Hub puis aller dans le dossier `Partage/TP/TP1` et créer un dossier intitulé `NOM1-NOM2` (nom des auteurs). Le notebook Python et les fichiers de données associés seront déposés dans `Partage/TP/TP1/NOM1-NOM2`.

Il est évidemment impossible d'utiliser la carte d'acquisition depuis un notebook dont le code Python s'exécute sur un serveur distant. En conséquence, les signaux numériques obtenus par la carte d'acquisition et le module `pycanum` devront être exportés sur le Jupyter Hub sous forme de fichiers binaires NPY.

[20] Faire la numérisation d'une tension sinusoïdale de fréquence environ 1000 Hz à une fréquence d'échantillonnage de 100 kHz.

[21] Enregistrer le signal (temps et tension) comme indiqué plus haut. Donner un nom significatif au fichier. Dans le dossier `Partage/TP/TP1/NOM1-NOM2`, télécharger le fichier de données (File Upload).

[22] Créer un notebook dans `Partage/TP/TP1/NOM1-NOM2`. Dans la première cellule de code, entrer :

```
import numpy as np
import matplotlib.pyplot as plt
plt.rcParams['figure.figsize'] = [13, 10]
```

[23] Dans une cellule de code, effectuer la lecture du fichier NPY et le tracé du signal. Les échelles doivent être choisies pour donner une bonne visibilité de la forme du signal. Ne pas oublier les légendes.

[24] Compléter par une cellule Markdown (cellule de texte) pour apporter toutes les informations utiles, par exemple la fréquence d'échantillonnage.

### 6.f. Numérisation d'un signal périodique

Un signal périodique non sinusoïdal peut être obtenu par synthèse harmonique. Cette méthode consiste à ajouter des sinusoïdes de différentes fréquences. Soit  $T$  la période du signal à synthétiser. La fréquence fondamentale est  $f = 1/T$ . On effectue la somme d'une sinusoïde de fréquence  $f$  et de sinusoïdes de fréquences multiples de  $f$ . Par exemple :

$$\begin{aligned}
 u(t) = & C_1 \cos(2\pi f t) \\
 & + C_2 \cos(4\pi f t + \phi_2) \\
 & + C_3 \cos(6\pi f t + \phi_3)
 \end{aligned}$$

Le premier terme est le *fondamental*. Les deux suivants sont respectivement les *harmoniques* de rang 2 et 3. On peut bien sûr ajouter des harmoniques de rang plus élevé. Chacun a sa propre amplitude, éventuellement nulle, et son propre déphasage par rapport au fondamental.

Pour effectuer une synthèse harmonique, nous allons utiliser [Pure Data](#), un logiciel de synthèse et de traitement du son. Le signal sera délivré par la sortie son de l'ordinateur.

[25] Télécharger le fichier [syntheseHarmonique.pd](#) et l'ouvrir avec Pure Data (Pd-Extended).

[26] Brancher la sortie son de l'ordinateur sur la voie 1 de l'oscilloscope, avec le câble fourni à cet effet.

[27] Cliquer sur DSP (en haut à droite) pour déclencher la synthèse. Ajuster la fréquence et les amplitudes du fondamental et des harmoniques.

[28] Observer la forme du signal sur l'oscilloscope. Faire varier les amplitudes des harmoniques et leur phase.

Pour choisir la fréquence d'échantillonnage, il faut considérer la version complète de la condition de Nyquist-Shannon appliquée aux signaux périodiques à bande de fréquences limitée, c'est-à-dire dont la série de Fourier s'arrête à un certain rang :

Condition de Nyquist-Shannon pour un signal périodique : pour échantillonner convenablement un signal périodique à bande de fréquences limitée, la fréquence d'échantillonnage doit être strictement supérieure au double de la fréquence de l'harmonique de rang le plus élevé.

Dans l'exemple ci-dessus, la plus grande fréquence est celle de l'harmonique de rang 3, c'est-à-dire  $3f$ . La condition s'écrit donc  $f_e > 6f$ .

[29] Noter la valeur de  $f$  choisie. En déduire la fréquence d'échantillonnage minimale.

[30] Faire une acquisition avec la carte Sysam-SP5 en respectant le critère largement (sur-échantillonnage).

[31] Comparer la courbe obtenue à celle donnée par l'oscilloscope.

[32] Faire une acquisition pour laquelle le dernier harmonique (celui de plus grande fréquence) est sous-échantillonné alors que les autres sont correctement échantillonnés.