

EI20n – CONVERSION ANALOGIQUE / NUMÉRIQUE

Auteur : Frédéric Marquet

TI-Nspire™ CAS

Mots-clés : échantillonnage, quantification, convertisseur analogique-numérique.

Fichiers associés : CAN_3bits_8V_CAS.tns,
CAN_4bits_8V_CAS.tns,
CAN_Echantillonnage_CAS.tns,
CAN_Temperature_CAS.tns,
CAN_EchantillonnageQuantification_CAS.tns,
EI8nElev_ConversionAnalogiqueNumerique.pdf.

1. Objectifs

- Utiliser la calculatrice **TI-Nspire** pour comprendre le principe de l'**échantillonneur bloqueur**,
- Comprendre le principe de la **numérisation** d'un signal,
- Utiliser un convertisseur **analogique / numérique**.

2. Introduction

► De nos jours, avec l'avènement des **ordinateurs** et des **microprocesseurs**, la technologie est entrée dans l'**ère numérique** : la plupart des appareils que nous utilisons dans la vie de tous les jours (électroménager, téléviseurs, lecteurs mp3, calculatrice, etc.) sont numériques. Ils sont capables d'interagir avec leur environnement, d'échanger des informations sous la forme de nombres composés de **0** et de **1**.

Comment le passage du « **monde analogique** » (monde qui nous entoure) au « **monde numérique** » est-il possible ?

C'est la question à laquelle nous allons tenter de répondre en étudiant le problème de la **conversion analogique-numérique**.



3. L'échantillonneur bloqueur

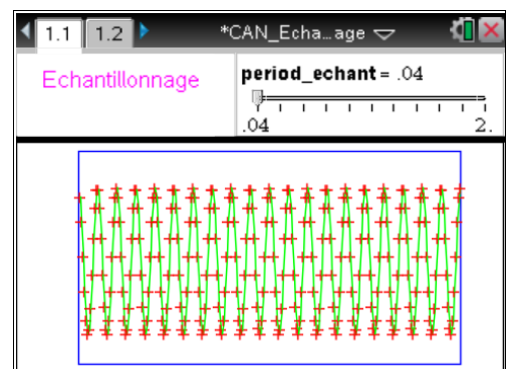
Lorsqu'un ordinateur doit **stocker** des **informations**, sa mémoire n'étant pas infinie, il ne peut pas le faire de façon continue : il récupère les informations à intervalles de temps donnés : c'est ce que l'on appelle l'**échantillonnage**.

- Ouvrir le fichier *CAN_Echantillonnage_CAS.tns*.

La courbe verte représente un **signal sinusoïdal** de période 0,5 s. Avec le curseur (associé à la variable **period_echant**), on peut régler la période d'échantillonnage de 0,04 s à 2 s.

Le signal échantillonné apparaît sur la page **1.1** et le signal échantillonné et bloqué apparaît sur la page **1.2**.

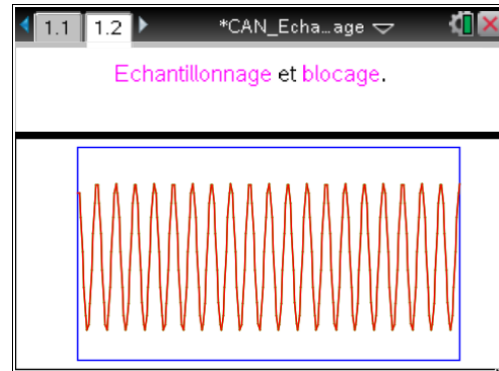
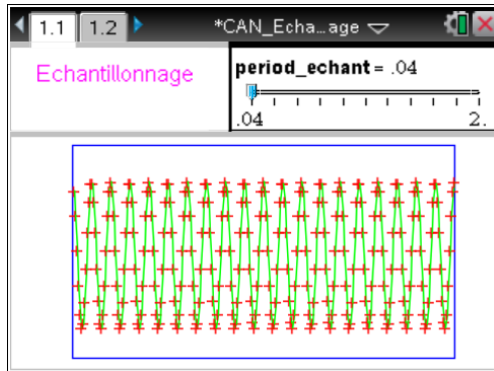
Montrer que dans certaines situations, le signal échantillonné n'est plus du tout représentatif du signal de départ. Expliquer pourquoi.



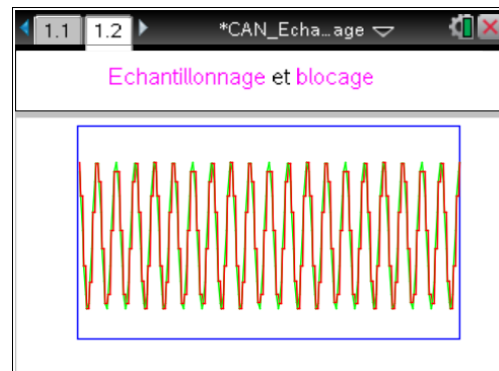
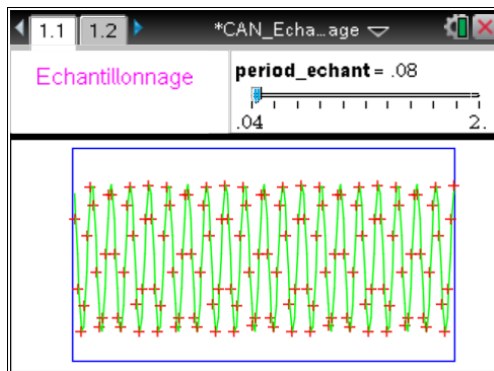
Notes pour le professeur

Pour certaines valeurs bien particulières de la **période d'échantillonnage**, on constate en effet qu'il apparaît des « **artefacts** » : le signal échantillonné n'a plus rien à voir avec le signal d'origine.

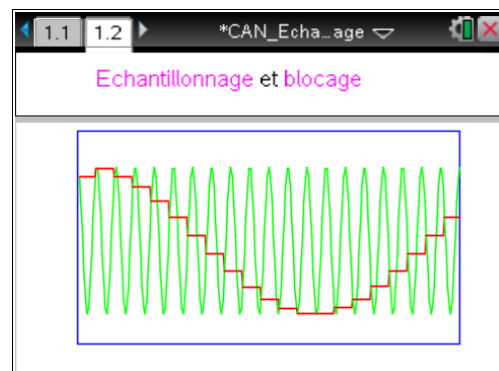
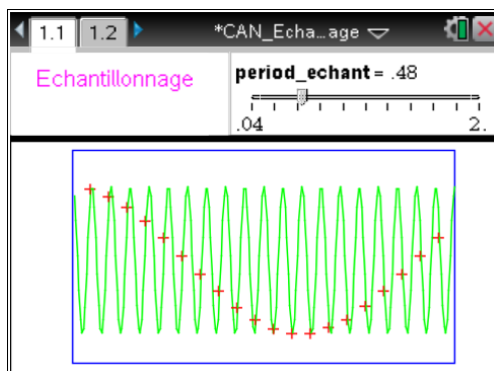
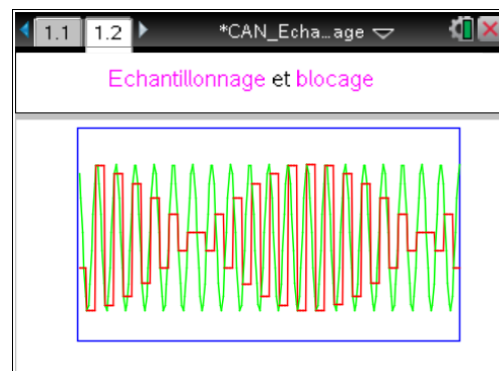
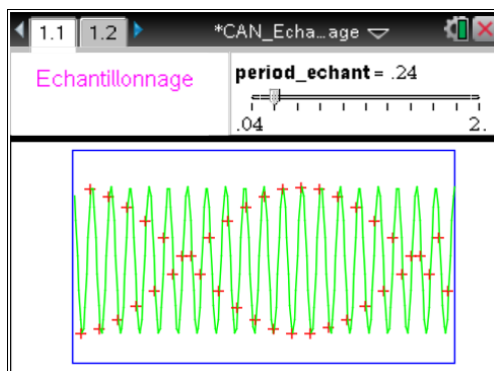
Signal correctement échantillonné



Échantillonnage « limite »



Signal « mal échantillonné » : apparition « d'artefacts »



4. La numérisation

- Pourquoi les ordinateurs travaillent-ils avec des **0** et des **1** ? Expliquer brièvement le principe du système de numération en **base 2**.

→ Les ordinateurs n'ont pas comme nous 10 doigts ! Ils ne peuvent distinguer que 2 états. Ils travaillent avec des niveaux électriques : « **niveau bas** (0 V) » = 0 et « **niveau haut** (5 V) » = 1. En base 2, on pourra écrire :

$$N = \sum_{i=0}^n b_i \times 2^i, \text{ avec } b_i = 0 \text{ ou } 1.$$

- Ouvrir le fichier *CAN_3bits_8V_CAS.tns*.

Un **convertisseur 3 bits** de tension d'entrée **u** comprise entre 0 et 8 V apparaît. Sa sortie est constituée des 3 bits s'affichant en bleu. Faire varier la valeur de la tension d'entrée **u** grâce au curseur, tracer sa fonction de transfert et expliquer son principe de fonctionnement.

→ Pas du convertisseur : $\frac{8}{2^3} = 1 \text{ V}$.

Si $0 \leq u < 1$, alors « Sortie = 0 » soit en binaire « **000** »,

Si $1 \leq u < 2$, alors « Sortie = 1 » soit en binaire « **001** » etc.

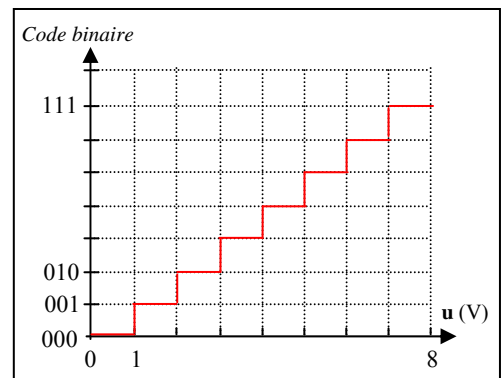
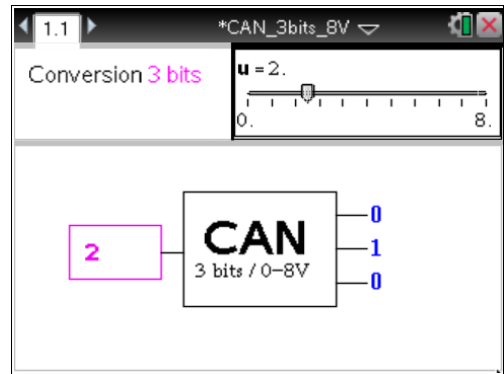
Fonction de transfert d'un convertisseur **3 bits** ; 0-8 V.

Définir :

- La **plage de fonctionnement** du convertisseur : → **0-8 V**.

- Le **nombre d'états** de sortie possibles : → $2^3 = 8$ états.

- Le **pas** du convertisseur : → $\frac{8}{2^3} = 1 \text{ V}$.

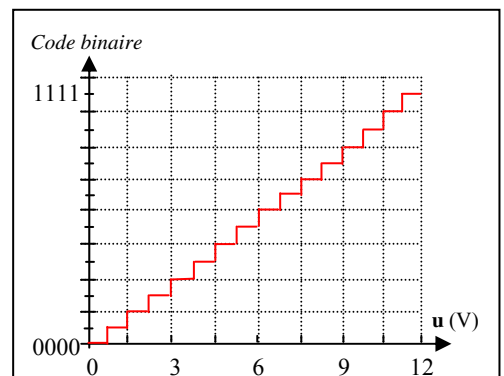
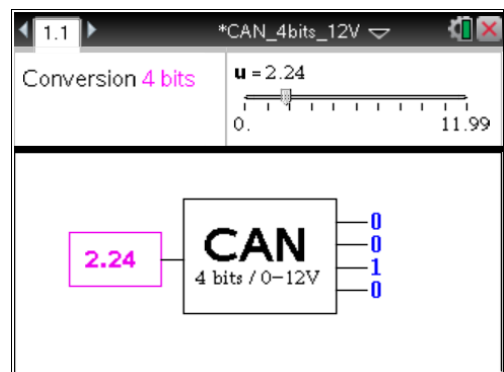


- Ouvrir le fichier *CAN_4bits_12V_CAS.tns*.

Un **convertisseur 4 bits** de tension d'entrée **u** comprise entre 0 et 12 V apparaît.

Sa sortie est constituée des 4 bits s'affichant en bleu.

Faire varier la valeur de la tension d'entrée **u** grâce au curseur, tracer sa fonction de transfert.



Fonction de transfert d'un convertisseur **4 bits** ; 0-12 V.

Définir :

- La **plage de fonctionnement** du convertisseur : → **0-12 V**.

- Le **nombre d'états** de sortie possibles : → $2^4 = 16$ états.

- Le **pas** du convertisseur : → $\frac{12}{2^4} = 0,75 \text{ V}$.

Application

- On souhaite mesurer les **températures** sur la plage **0-32 °C** et avoir une précision égale à **0,5 °C**.
Combien de bits doit comporter le convertisseur ?

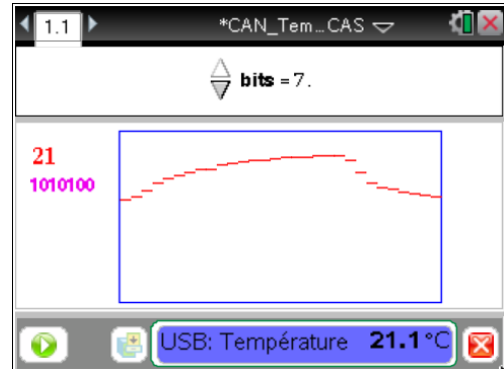
- Si la température est égale à **29,7 °C**, quelle la **valeur binaire** à la sortie du convertisseur ?

- Ouvrir le fichier *CAN_Temperature_CAS.tns*.

Connecter la **sonde de température** au port mini-USB de la calculatrice.

Le graphe représentant l'évolution de la température au cours du temps apparaît. Régler le **nombre de bits** du convertisseur avec la valeur déterminée précédemment. Faire varier la température en tenant par exemple la sonde entre les mains et vérifier :

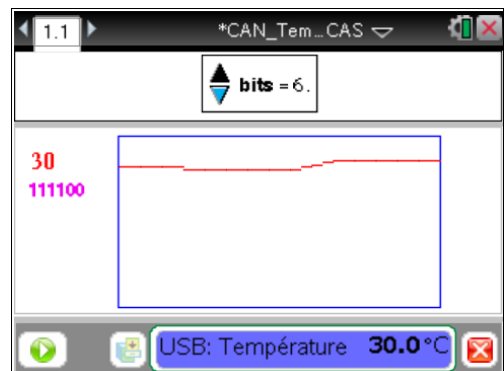
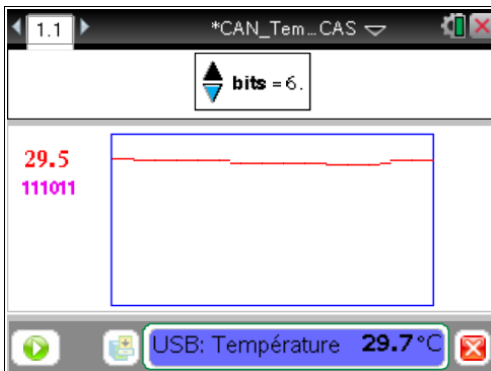
- que le **pas du convertisseur** est bien celui attendu (**0,5 °C**),
- que le **nombre binaire** affiché pour $T = 29,7\text{ °C}$ est correct.



Notes pour le professeur

- Il faut un convertisseur **6 bits** pour pouvoir obtenir une précision égale à $\frac{32}{2^6} = 0,5\text{ °C}$.

On règle donc le convertisseur sur « 6 bits » et on vérifie, en faisant varier la température de la sonde, que le pas est bien égal à **0,5 °C** :



- Une température (en °C) $29,5 \leq T < 30,0$ correspondra à la valeur $29,5\text{ °C}$. Pour une température égale à **29,7 °C** le nombre binaire en sortie est donc **111011** (59 en décimal) car $59 \times 0,5\text{ °C} = 29,5\text{ °C}$. Lorsque la température atteint $30,0\text{ °C}$, il bascule à 111100 (60 en décimal), correspondant à un affichage de $30,0\text{ °C}$.

5. La chaîne de conversion analogique-numérique complète

Pour numériser un signal, par exemple les signaux électriques issus d'un microphone, il faut :

- **Échantillonner** le signal avec une fréquence d'échantillonnage appropriée,
- **Numériser** le signal échantillonné, c'est-à-dire le transformer en un mot binaire d'une longueur donnée.

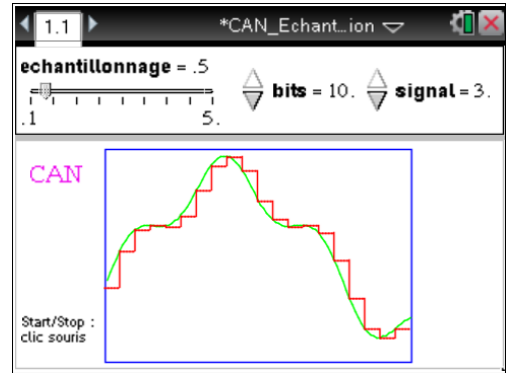
Ces deux étapes induisent, bien sûr, une **perte d'information**. Un **nombre** trop élevé de **bits** ou une **fréquence d'échantillonnage** trop grande conduiront à un stockage d'informations trop important. À l'inverse si ces deux valeurs sont trop faibles, on aura une perte d'information trop importante. Il faudra donc choisir une fréquence d'échantillonnage appropriée et un nombre de bits adapté.

- Ouvrir le fichier :

CAN_EchantillonnageQuantification_CAS.tns.

Le simulateur permet de définir :

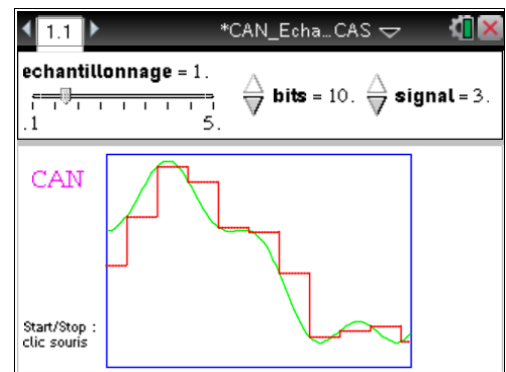
- La **période d'échantillonnage (echantillonnage)** : valeur allant de 0, 1 s à 5 s avec un pas de 0,1 s,
- Le **nombre de bits** pour la conversion (**bits**) : valeur entière comprise entre 1 et 10,
- Le **type de signal (signal)** : 1 pour un signal triangulaire, 2 pour un signal sinusoïdal et enfin 3 pour le signal représenté ci-contre.



- Est-il cohérent d'utiliser une période d'échantillonnage égale à 5 secondes ? Que penser d'une période d'échantillonnage de 1 seconde ?
- On choisit une période d'échantillonnage égale à 0,2 s. A l'aide du simulateur, déterminer un choix cohérent pour le nombre de bits du convertisseur.

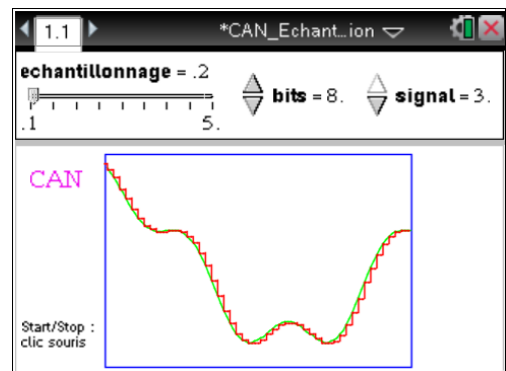
Notes pour le professeur

- Une **période d'échantillonnage** égale à **5 s** est clairement insuffisante.
- Avec une période d'échantillonnage égale à **1 s**, on a un cas limite (voir figure ci-contre).

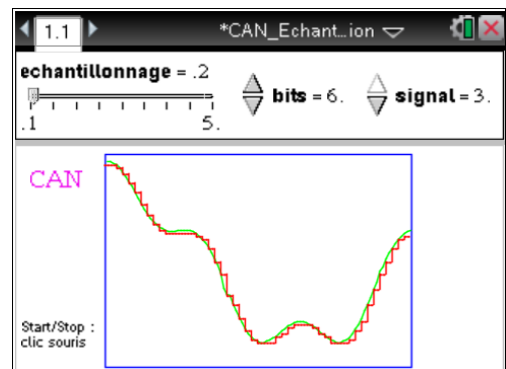


Si on choisit :

- une **période d'échantillonnage** égale à **0,2 s**,
 - une conversion sur **8 bits**,
- on obtient le signal présenté ci-contre.

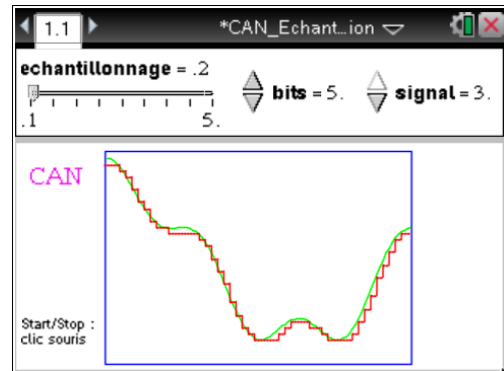


En gardant une **période d'échantillonnage** égale à **0,2 s**, mais en optant pour une conversion sur **6 bits** seulement on obtient le signal présenté ci-contre.



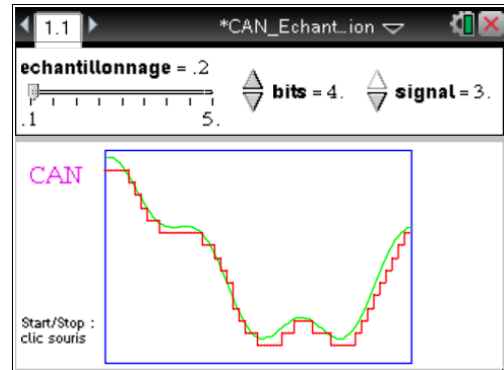
On constate que, pour la période d'échantillonnage choisie, il n'y a pratiquement aucune différence entre une quantification sur **8 bits** et une quantification sur **6 bits**.

En conservant une **période d'échantillonnage** égale à **0,2 s**, et en optant pour une conversion sur **5 bits** seulement on obtient le signal présenté ci-contre.



Enfin, en réduisant encore à 4 bits pour la conversion, on obtient le signal présenté ci-contre.

Pour une quantification sur **5 bits**, le signal reste acceptable. Par contre, à partir de **4 bits** on peut considérer que le pas de quantification est trop faible.



Remarque : Dans le cas de la numérisation / compression d'un morceau de musique, on doit prendre en compte la plage de fréquence de la voix humaine et des instruments de musique (de l'ordre de 100 Hz à quelques kHz). Généralement, on pourra choisir une fréquence d'échantillonnage égale à quelques dizaines de kHz (par exemple la fréquence d'échantillonnage par défaut du logiciel enregistreur et éditeur audio libre Audacity est égale à 44,1 kHz). On prend aussi en compte les imperfections du récepteur qu'est l'oreille humaine : au-delà d'un certain nombre de bits, l'oreille n'est plus capable de déceler la différence. En tout état de cause, il existe une règle fondamentale, connue sous le nom de « **théorème de Shannon** » qui stipule qu'il faut au moins **échantillonner à 2 fois la fréquence maximale** contenue dans le signal enregistré (si ce théorème n'est pas respecté, il apparaît des artéfacts, comme illustré dans la partie 3).