

## Conversion Analogique-Numérique (CAN)

### THÉMATIQUE

- Signal analogique et signal numérique

### CONCEPTS OU NOTIONS ABORDÉS

- Résolution et pas d'un Convertisseur Analogique-Numérique (CAN).
- Période d'échantillonnage d'une conversion analogique-numérique.

### OBJECTIFS DE FORMATION

- Pratiquer une démarche expérimentale permettant de mesurer le pas et la résolution du CAN d'une carte à microcontrôleur.
- Pratiquer une démarche expérimentale permettant de réaliser un CAN à l'aide d'une carte à microcontrôleur, afin de mettre en évidence l'influence de la période d'échantillonnage et le pas du convertisseur sur l'acquisition d'un signal analogique.

## Introduction

Consulter la page éducol associée au thème « [Programmer en physique-chimie](#) ».

### Présentation des activités

À l'aide d'une carte à microcontrôleur, les élèves numérisent le signal analogique délivré par un générateur de tension basse fréquence. Ils étudient ensuite ce signal numérisé afin de déterminer certaines caractéristiques du CAN d'une carte à microcontrôleur (pas et résolution) de type Arduino™.

Dans un deuxième temps, les élèves utilisent un programme permettant de simuler le fonctionnement d'un convertisseur analogique-numérique dont on peut faire varier le pas et la période d'échantillonnage, afin d'acquérir un signal analogique. Ils peuvent ainsi étudier l'influence de la période et du pas d'échantillonnage sur la qualité du signal numérisé.

### Activité 1 : acquisition et visualisation d'un signal à l'aide d'une carte à microcontrôleur

À l'aide d'une carte à microcontrôleur, les élèves numérisent le signal analogique délivré par un générateur de tension basse fréquence. Les mesures du signal numérisé permettent de déterminer les valeurs du pas et de la résolution du CAN.

## Activité 2 : Réaliser un CAN avec une carte à microcontrôleur afin d'étudier l'influence de la période et du pas d'échantillonnage sur la qualité d'un signal numérisé

Les élèves numérisent le signal analogique délivré par un générateur de tension basse fréquence, en faisant varier le pas et la période d'échantillonnage. Les mesures du signal numérisé sont visualisées dans un tableur, permettant d'étudier l'influence du choix du pas et de la période d'échantillonnage.

Pistes de validation et d'expérimentation

- Comparer le signal numérique au signal analogique vu sur un oscilloscope analogique.
- Mettre en évidence expérimentalement l'influence du pas d'échantillonnage sur la qualité de la conversion.

### LOGICIELS UTILISÉS

- Logiciel de programmation Arduino.
- Tableur

### COMPÉTENCES INFORMATIQUES TRAVAILLÉES

- Comprendre un programme simple en langage Arduino.
- Modifier le codage d'un programme Arduino afin de faire une conversion.
- Afficher une valeur à l'aide du moniteur série de l'interface Arduino.
- Afficher un graphique à l'aide du traceur série de l'interface Arduino.

### TABLEURS

- Afficher un graphique.

## Exemple de contextualisation

Une contextualisation possible est d'expliquer le fonctionnement de l'enregistrement d'un son sur un ordinateur, un CD ou un téléphone.

## De la situation physique au traitement numérique

Les signaux électriques issus de capteurs sont des signaux analogiques. Leur traitement numérique, par exemple un ordinateur, nécessite une conversion qui, forcément, dégrade en partie ces signaux, notamment du fait de la résolution limitée du traitement de l'information. L'idée est ici d'approcher expérimentalement certaines de ces limitations.

## Ce que les élèves doivent retenir

En mesurant le signal obtenu, les élèves s'approprient :

- la notion de pas et de résolution d'un convertisseur analogique-numérique ;
- le codage d'une tension analogique par un nombre entier et la prise en compte de l'unité de mesure ;
- les notions de période d'échantillonnage et de transmission de l'information ;
- les limitations en fréquence associées à l'échantillonnage.

# Activité 1 : acquisition et visualisation d'un signal à l'aide d'une carte à microcontrôleur

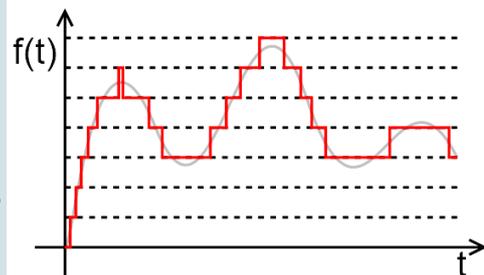
Les propositions suivantes ne sont pas prescriptives. Il s'agit de bases pour illustrer la situation d'apprentissage qu'il convient d'ajuster en fonction de sa place dans la progression et dans la séquence pédagogique, du niveau de maîtrise par les élèves de l'outil informatique choisi, etc.

## Exemple de consignes pour les élèves

La carte à microcontrôleur est une carte peu onéreuse, programmable, et répandue. On souhaite étudier certaines caractéristiques de ses « entrées analogiques » (Conversion Analogique-Numérique CAN) et la programmer afin de comprendre le principe de fonctionnement d'un CAN.

### DOCUMENT MIS À DISPOSITION

L'échantillonnage correspond à la prise de mesures, d'une tension le plus souvent, à intervalle de temps donné et constant. Cet intervalle de temps est la période d'échantillonnage, souvent notée  $T_e$ . Les mesures prises, appelées « échantillons », sont alors « bloquées » à des valeurs constantes pendant la durée  $T_e$ . La valeur d'un échantillon reste donc constante pendant la durée  $T_e$  jusqu'à la mesure de l'échantillon suivant.



Source : [wikimedia](https://commons.wikimedia.org/wiki/File:Sampling_theorem_1d.png)

Le signal numérisé ne peut prendre que des valeurs discrètes. La quantification s'accompagne d'une perte d'information sur la tension d'origine. Le pas de numérisation correspond au plus petit écart de tension, en volt V, entre deux valeurs du signal numérisé.

Nous appelons résolution du CAN le nombre de bits que le convertisseur utilise pour coder le signal analogique en signal numérique. Enfin, le calibre correspond à la tension maximale mesurable par le CAN.

### MATERIEL À DISPOSITION

- Un ordinateur relié à une carte à microcontrôleur ;
- Un générateur de tension variable ;
- Un voltmètre ;
- Des fils de connexion ;
- Le programme suivant permet d'afficher la valeur mesurée à l'entrée analogique d'une carte à microcontrôleur :

```
1 float mesure = 0; // On définit une variable appelée "mesure" à laquelle on attribue la valeur 0
2
3 void setup() {
4   Serial.begin(115200); // Ouvre une liaison avec le moniteur série afin de visualiser le résultat
5 } // sur l'écran de l'ordinateur
6
7 void loop() {
8   mesure = analogRead(A0); //Met dans "mesure" la valeur lue par le CAN sur l'entrée A0 de la carte
9   Serial.println(mesure); //Affiche la valeur lue par le CAN dans le moniteur série
10 }
```

Les étapes ci-dessous peuvent se conclure par un appel au professeur pour aide ou validation.

### Proposition d'un protocole expérimental (compétences S'approprier et Analyser)

Proposer un protocole expérimental permettant de déterminer expérimentalement le calibre de la carte à microcontrôleur.

### Mise en œuvre du protocole expérimental (compétence Réaliser)

Mettre en œuvre le protocole expérimental proposé.

### Détermination de caractéristiques du CAN de la carte à microcontrôleur (compétence Analyser)

Déduire des résultats de l'expérience la résolution et le pas du CAN de la carte à microcontrôleur.

### Amélioration du programme (compétence Analyser)

Proposer une modification du programme fourni afin d'afficher la valeur de la tension électrique en lieu et place du nombre affiché par le CAN.

### Mise en œuvre expérimentale (compétence Réaliser)

Mettre en œuvre une modification du programme fourni de façon à visualiser sur le traceur série une tension délivrée par un générateur de tension basse fréquence en fonction du temps. Il convient à ce stade de choisir un signal qui soit fidèlement reproduit (fréquence d'une dizaine de Hz et amplitude de quelques volts, ne prenant que des valeurs positives).

## Exemples d'aides et de réalisation

### Solution du protocole expérimental

Le CAN de la carte à microcontrôleur est dit monopolaire : sa gamme de tension mesurable varie de 0 V à une tension  $U_P$  à déterminer.

On fait varier, tout en la mesurant, la tension à l'entrée analogique de la carte à microcontrôleur et on regarde les nombres affichés par le programme. Les nombres varient entre 0 et 1023 au maximal. On mesure la tension limite  $U_P$  pour laquelle le nombre 1023 est affiché : c'est le calibre de la carte à microcontrôleur.

### Solution des caractéristiques du CAN

Le nombre affiché par la carte à microcontrôleur varie entre 0 et 1023, ce qui correspond à un nombre total de 1024 possibilités : la résolution est donc de 10 bits car  $2^{10} = 1024$ .

Connaissant le calibre du CAN (5,00 V), on peut en déduire le pas  $p$  du CAN :

$$p = \frac{5,00 \text{ V}}{1023} = 4,89 \text{ mV}$$

Note : on utilise 1023 et non 1024 car entre 0 et 1023 il y a bien 1023 « pas », autrement dit 1023 intervalles.

### Amélioration du programme

Il faut modifier la ligne numéro 9 afin qu'elle calcule la valeur de la tension. Le nouveau programme est le suivant :

```
1 float mesure = 0; // On définit une variable appelée "mesure" à laquelle on attribue la valeur 0
2
3 void setup() {
4     Serial.begin(115200); // Ouvre une liaison avec le moniteur série afin de visualiser le résultat
5 }                               // sur l'écran de l'ordinateur
6
7 void loop() {
8     mesure = analogRead(A0); //Met dans "mesure" la valeur lue par le CAN sur l'entrée A0 de la carte
9     Serial.println(mesure/1023*5.0); //Affiche la valeur de la tension (en V) dans le moniteur série
10 }
```

## Activité 2 : Réaliser un CAN avec une carte à microcontrôleur afin d'étudier l'influence de la période et du pas d'échantillonnage sur la qualité d'un signal numérisé

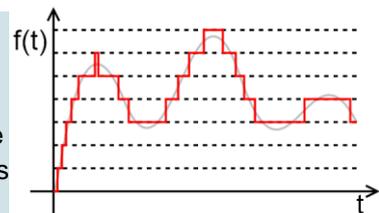
Les propositions suivantes ne sont pas prescriptives. Il s'agit de bases pour illustrer la situation d'apprentissage qu'il convient d'ajuster en fonction de sa place dans la progression et dans la séquence pédagogique, du niveau de maîtrise par les élèves de l'outil informatique choisi, etc.

### Exemple de consignes pour les élèves

Les signaux issus de capteurs sont analogiques, car ils traduisent des phénomènes physiques qui varient continûment dans le temps. Le domaine numérique est maintenant prédominant, et la numérisation de signaux analogiques permet aisément leur traitement, leur stockage, leur transformation, leur restitution, copie... On souhaite montrer l'influence du pas et de la période d'échantillonnage d'un Convertisseur Analogique-Numérique (CAN) sur la qualité de numérisation d'un signal analogique. Ces paramètres ne sont pas directement modifiables par le langage de programmation Arduino : il s'agit de proposer, avec cette carte, une opération de numérisation à paramètres réglables.

#### DOCUMENT MIS À DISPOSITION

L'échantillonnage correspond à la prise de mesures, d'une tension le plus souvent, à intervalle de temps donné et constant. Cet intervalle de temps est la période d'échantillonnage, souvent notée  $T_e$ . Les mesures prises, appelées « échantillons », sont alors « bloquées » à des valeurs constantes pendant la durée  $T_e$ . La valeur d'un échantillon reste donc constante pendant la durée  $T_e$  jusqu'à la mesure de l'échantillon suivant.



Source : [wikimedia](#)

Le signal numérisé ne peut prendre que des valeurs discrètes. La quantification s'accompagne d'une perte d'information sur la tension d'origine. Le pas de numérisation, en V, correspond au plus petit écart de tension entre deux valeurs du signal numérisé.

#### MATÉRIEL À DISPOSITION :

- un ordinateur, disposant du tableur Calc (LibreOffice) relié à une carte à microcontrôleur ;
- un générateur de tension basse fréquence avec des fils de connexion ;
- le fichier pré rempli disponible dans le dossier « [conversion analogique numerique.zip](#) » ;
- le programme suivant permet de numériser, en faisant varier le pas et la période d'échantillonnage, un signal électrique appliqué à l'entrée analogique d'une carte à microcontrôleur :

```

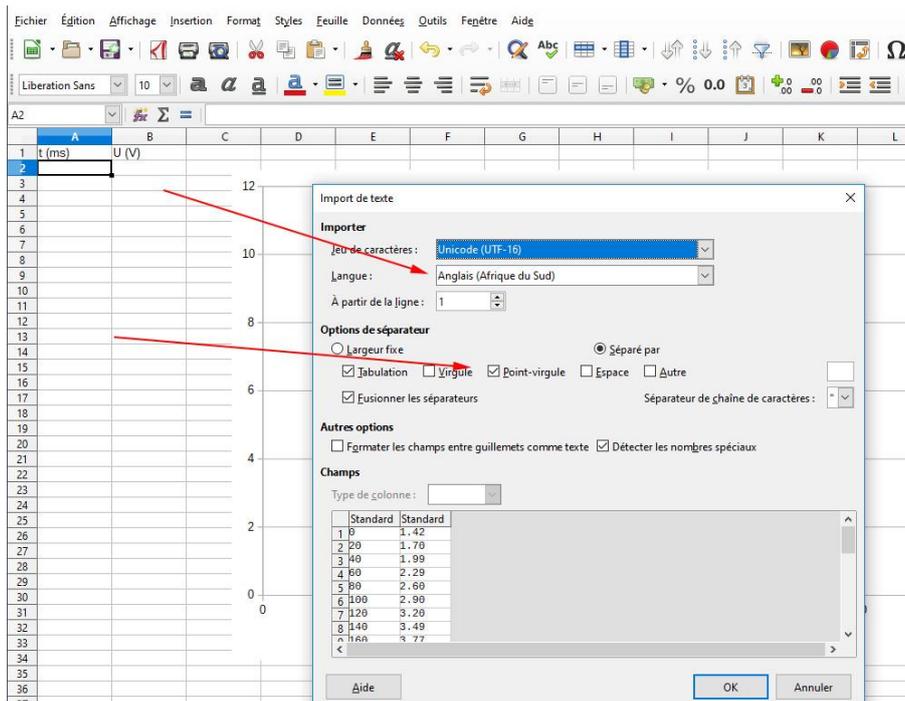
1 float mesure[200] = {}; // On définit un tableau de variables appelée "mesure" pouvant contenir jusque 201 données et laissé vide pour l'instant
2 int instant[200] = {}; // On définit un tableau de variables appelée "instant" pouvant contenir jusque 201 données et laissé vide pour l'instant
3 int Te = 10 ; // Définit la valeur de la période d'échantillonnage Te en ms (valeur à prendre entre 10 ms et 200 ms)
4 float pas = 0.01; // Définit le pas en V (valeur à prendre entre 0,00489 V et 5.00 V)
5 float tension = 0; // Définit une variable tension = 0
6 float arrondi = 0; // Définit une variable arrondi = 0
7
8
9 void setup() {
10     Serial.begin(57600); // Ouvre une liaison avec le moniteur série afin de récupérer les données sur l'écran de l'ordinateur
11 }
12
13
14 void loop() {
15     for (int i=0;i<2000/Te; i++) { //Boucle d'acquisition de mesures durant 2 s
16         tension = analogRead(A0)/1023.0000*5.0000; //Met dans tension la valeur lue par le CAN sur l'entrée A0 de la carte, convertie en V
17         arrondi=int(tension/pas); // la valeur de "arrondi" est égale à la partie entière de (tension/pas)
18         mesure[i] = arrondi*pas; // met dans la mesure n° i la valeur de (arrondi*pas)
19         instant[i] = i*Te; // met dans instant n° i la valeur de l'instant d'échantillonnage
20         delay(Te); //Attend la durée de la période d'échantillonnage Te
21     }
22
23     for (int i=0;i<2000/Te; i++) { //Boucle d'envoi des données vers le moniteur série
24         Serial.print(instant[i]); //Affiche l'instant n°i de prise de la mesure dans le moniteur série
25         Serial.print(" "); //affiche un séparateur entre instant et mesure
26         Serial.println(mesure[i]); //affiche la mesure n°i dans le moniteur série
27     }
28     delay(10000); //Attend 10 s avant de refaire une série de mesures
29 }

```

## Réalisation d'un protocole expérimental et traitement de données à l'aide d'un tableur (compétences S'approprier et Réaliser)

Les données envoyées dans le moniteur série du programme Arduino sont facilement importables dans un tableur type LibreOffice. Pour cela il suffit de :

- relier le générateur de tension basse fréquence à l'entrée A0 de la carte à microcontrôleur et générer un signal sinusoïdal de 1,00 Hz, variant entre 0,00 V et 5,00 V ;
- charger le programme Arduino fourni, en l'adaptant pour avoir une fréquence d'échantillonnage de 100 Hz et un pas d'échantillonnage de 0,005 V ;
- compiler et lancer le programme ;
- démarrer le moniteur série de l'interface Arduino ;
- dès que le programme démarre, il lance une acquisition pour une durée de 2 s, puis affiche les valeurs acquises, attend 10 secondes et recommence l'opération ;
- dès que la série de données acquises apparaît dans le moniteur série, sélectionner les valeurs récupérées (CTRL+A) puis les copier (CTRL+C) dans le presse-papier ;
- coller (CTRL+V), dans la case A2 du fichier tableur LibreOffice fourni, les données récupérées.



Astuce : cocher « Point-virgule » en options de séparateurs et de choisir la langue « Anglais » (de façon à ce que les points décimaux délivrés par le moniteur série soient transformés en virgules pour que le tableur les reconnaisse). On obtient les données dans deux colonnes la première avec l'instant d'échantillonnage et la deuxième la tension mesurée par le convertisseur analogique-numérique de la carte à microcontrôle (en volt V) correspondant à cet instant.

- Visualiser le signal échantillonné par la carte, vérifier que la période d'échantillonnage du signal correspond bien à la fréquence d'échantillonnage demandée (100 Hz).
- Mesurer sur le graphe la période du signal échantillonné, et en déduire sa fréquence.

### Mise en évidence de l'influence de la fréquence d'échantillonnage (compétence Analyser)

Proposer un protocole pour mettre en évidence l'influence de la fréquence d'échantillonnage sur l'acquisition du signal par le CAN.

### Mise en œuvre expérimentale (compétence Réaliser)

Mettre en œuvre le protocole proposé afin de mettre en évidence l'influence de la fréquence d'échantillonnage sur l'acquisition du signal.

Observer et comparer les signaux échantillonnés, et conclure quant au choix de la fréquence d'échantillonnage.

### Sous-échantillonnage (compétence Analyser)

Le théorème de Shannon indique que, pour numériser convenablement un signal, il faut que la fréquence d'échantillonnage soit au moins deux fois supérieure à la fréquence du signal à numériser.

En utilisant une fréquence d'échantillonnage de 100 Hz pour le programme Arduino, proposer un protocole pour mettre en évidence les problèmes posés lorsqu'on sous-échantillonne le signal.

### Mise en œuvre expérimentale (compétence Réaliser)

Mettre en œuvre le protocole proposé.

## Influence du pas de numérisation (compétence Analyser)

Proposer un protocole pour :

- mesurer le pas de numérisation sur le signal numérisé ;
- mettre en évidence l'influence du pas de numérisation sur l'acquisition du signal par le CAN.

## Mise en œuvre expérimentale (compétence Réaliser)

Mettre en œuvre le protocole proposé.

## Exemples d'aides et de réalisation

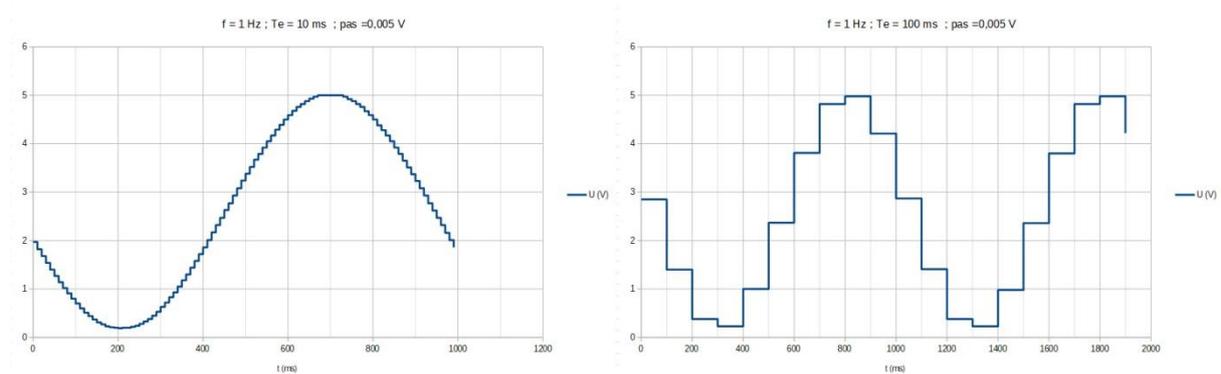
### Solution du protocole expérimental

Il faut modifier les lignes suivantes :

- ligne 3 : `int Te = 10 ;`
- ligne 4 : `int pas = 0,005.`

### Mise en évidence de l'influence de la fréquence d'échantillonnage

On peut comparer un même signal échantillonné avec un même pas mais sur deux fréquences d'échantillonnage différentes : 100 Hz ( $T_e = 10$  ms) et 10 Hz ( $T_e = 100$  ms) par exemple, et ainsi mettre en évidence la dégradation du signal numérique par rapport au signal analogique :

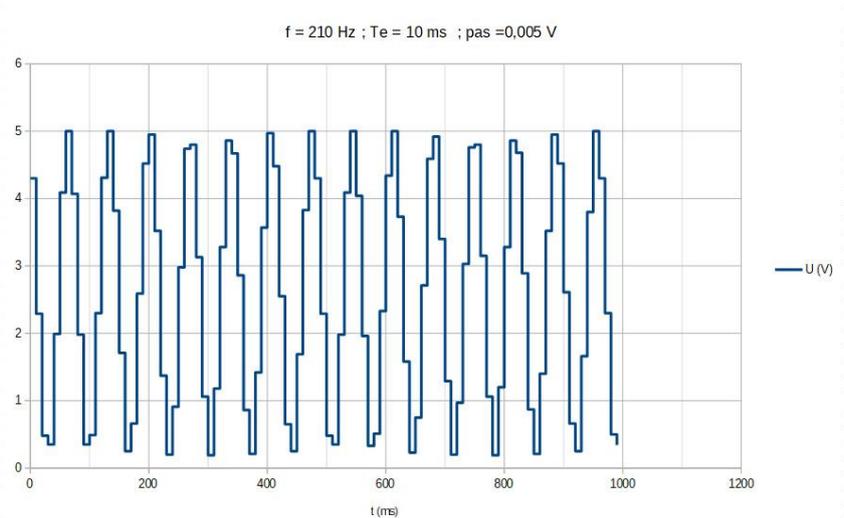


### Sous-échantillonnage

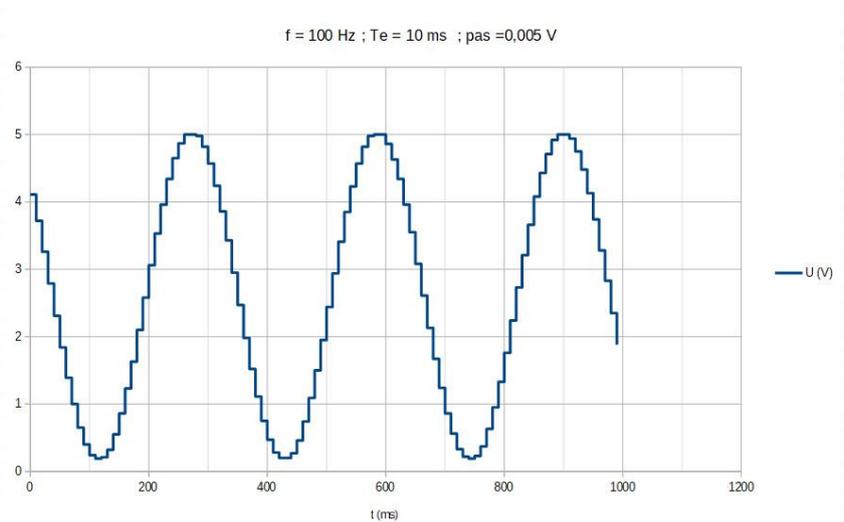
On peut comparer observer un signal analogique de fréquence supérieure à 200 Hz et vérifier que la période du signal échantillonné-bloqué ne correspond absolument pas à la période du signal d'origine (qu'on peut valider en visualisant en parallèle le signal réel sur un oscilloscope).

Exemples :

- La fréquence du signal analogique est de 210 Hz, la fréquence du signal échantillonné de 15 Hz :

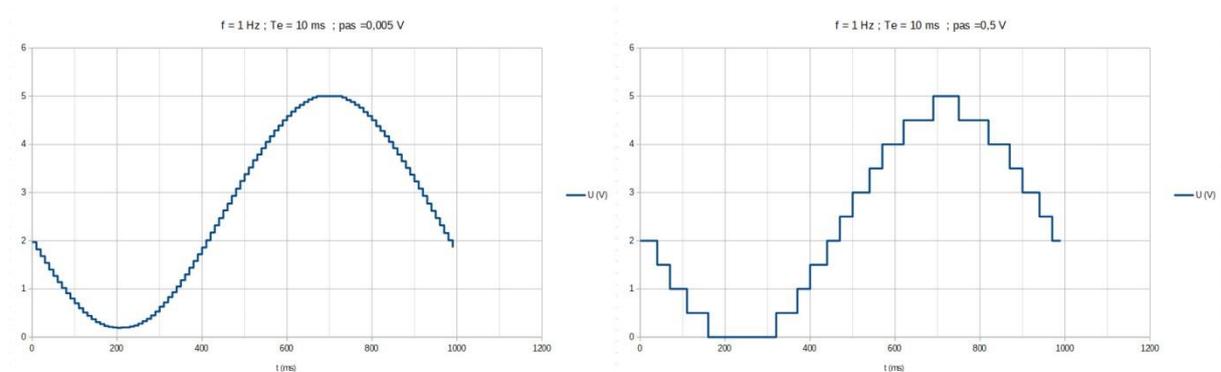


- La fréquence du signal analogique est de 100 Hz, la fréquence du signal échantillonné de 3 Hz :



### Mise en évidence de l'influence du pas de numérisation

On peut changer la valeur du pas dans le programme et mesurer ainsi ce dernier sur le signal mesuré. On peut aussi comparer un même signal analogique, échantillonné avec deux valeurs de pas différents :

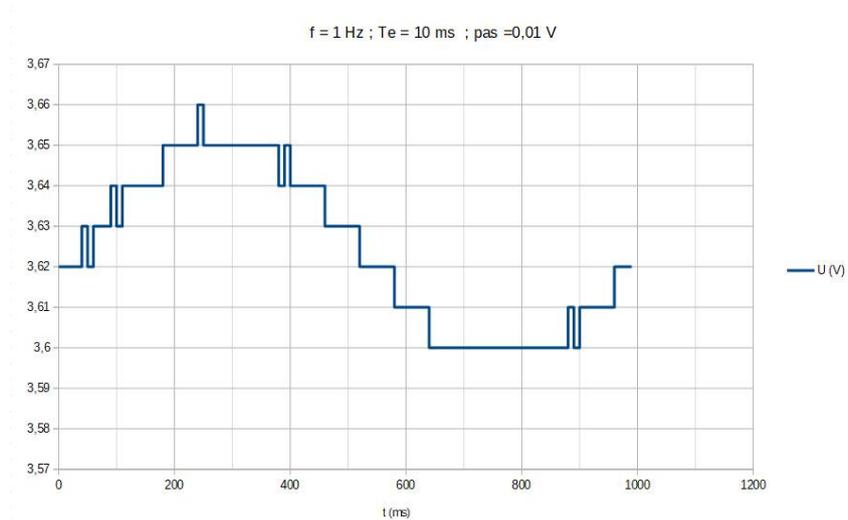


On mesure ici le pas de 0,5 V sur le signal numérisé.

### Pistes de validation et d'expérimentation possibles

Il est intéressant de comparer le signal analogique observé à l'oscilloscope, au signal échantillonné qui lui est plus ou moins fidèle selon les conditions de numérisation.

Il peut être également intéressant de visualiser l'amplification du bruit d'un signal analogique bruité. Pour cela on peut par exemple générer un signal sinusoïdal de très faible amplitude avec offset. Voici page suivante une mesure obtenue avec un signal variant faiblement autour de 3,62 V.



### Note sur la réalisation

L'échantillonnage se fait en réalité à une période supérieure à la période  $T_e$  car le temps de traitement des données n'a pas été pris en compte. Le programme a ainsi volontairement été borné à 100 Hz maximum de fréquence d'échantillonnage afin de limiter l'influence du temps traitement des données sur la période d'échantillonnage.

Dans le dossier « [conversion analogique numerique.zip](#) », vous trouverez les fichiers relatifs à ce document et des exemples de réalisations.