

<h1>Cours</h1>	<h2>Cours ACQ 2</h2>	TS11 (Période 3)
	Conversion analogique – numérique Conversion numérique-analogique	1h
	Cycle 8 : Acquérir Conditionner Traiter	4 semaines

Docs ciné

ANALYSER Caractériser un constituant de la chaîne d'information.

MODELISER Identifier les phénomènes physiques à modéliser.

MODELISER Établir un modèle de comportement à partir d'une réponse temporelle ou fréquentielle.

EXPERIMENTER Justifier le choix d'un appareil de mesure ou d'un capteur vis-à-vis de la grandeur physique à mesurer.

EXPERIMENTER Identifier les erreurs de mesure et de méthode.

CONCEVOIR Choisir la technologie des composants de la chaîne d'information.

1 Utilité des signaux numériques

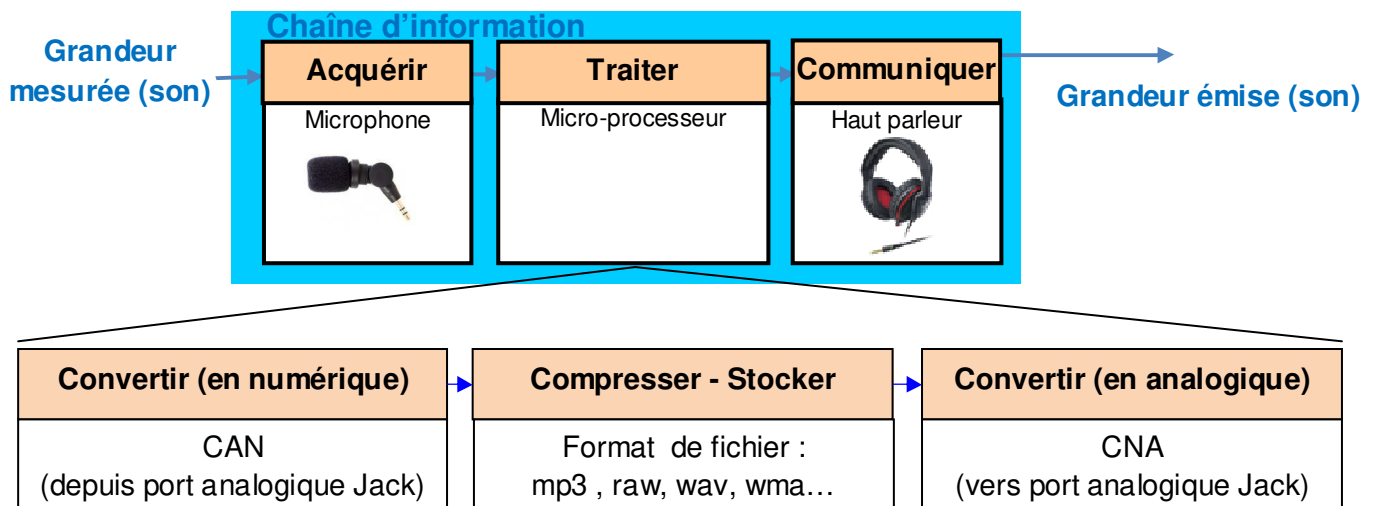
De nombreux systèmes électroniques utilisent la technique numérique, à base de microprocesseurs ou de microcontrôleurs pour les avantages qu'elle présente par rapport à la technique analogique :

- facilité de traitement de l'information (filtrage, compression...),
- mémorisation possible des informations,
- faible sensibilité au bruit...

Lorsque les informations issues de capteurs sont des grandeurs analogiques ou que les actionneurs sont commandés par des signaux analogiques, il est nécessaire de procéder à des conversions de données :

- le **convertisseur analogique numérique (CAN)** convertit le signal analogique du capteur en une suite de mots numériques qui pourront être compris et traités par le calculateur (microprocesseur).
- Le calculateur pourra générer en entrée du **convertisseur numérique analogique (CNA)** des mots numériques qui sont alors convertis en signaux analogiques.

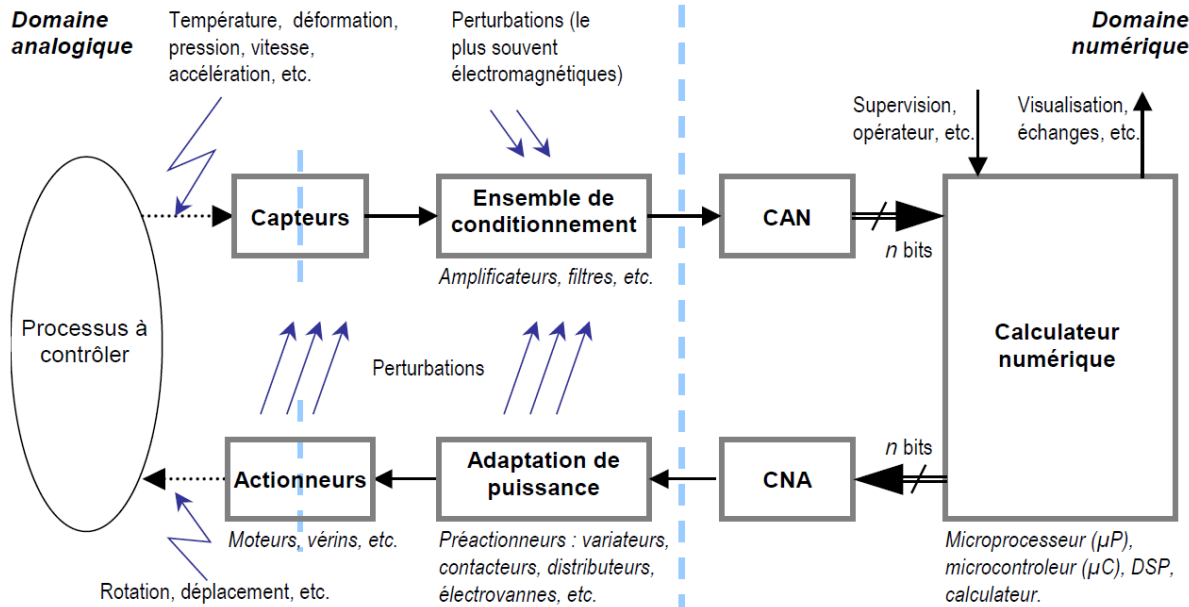
Exemple du traitement numérique d'un signal audio



• **CAN analogique numérique** : (vibrations acoustiques → micro → CAN → compression numérique → enregistrement en mémoire ou sur disque dur).

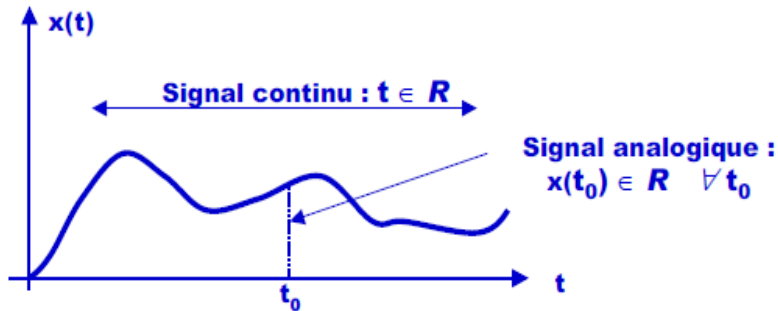
• **CNA numérique analogique** : (lecture des fichiers numériques stockés → filtrage numérique → CNA → amplification → écouteurs ou baffles vibrations acoustiques).

Exemple de convertisseurs intégrés dans un microcontrôleur de type DSP :

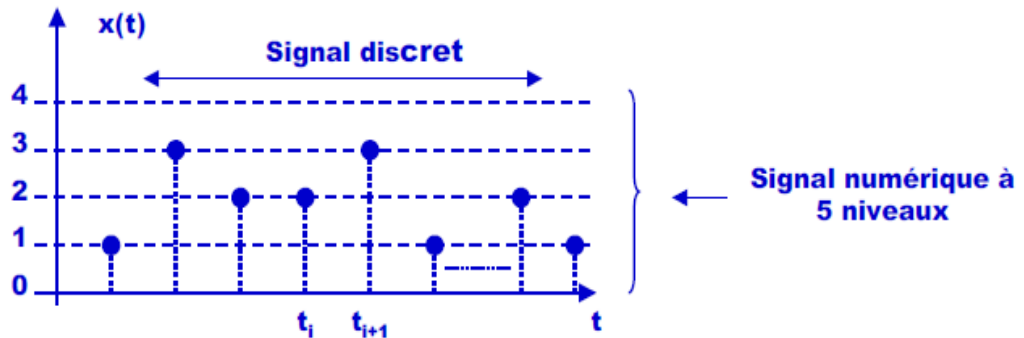


2 Nature des signaux

Le signal analogique



Le signal numérique échantillonné :



2.1 Codage des nombres en binaire

2.1.1 Conversion binaire → décimal

La conversion $N_{(2)} = (b_n b_{n-1} \dots b_i \dots b_1 b_0)_{(2)}$ en base $b=2$ pour obtenir $N_{(10)} = (d_n d_{n-1} \dots d_i \dots d_1 d_0)_{(10)}$ en base 10 s'obtient de la manière suivante :

Somme des chiffres affectés de leur poids respectifs : $N_{(10)} = \sum_{i=0}^n b_i \cdot b^i$

où $N_{(10)}$ est le nombre codé en décimal (base 10)

b est le nombre (en base 10) de caractères codables dans la base d'origine

b_i est le coefficient de rang i (en base 10) du nombre à convertir en base 10

Exemples : $63_{(10)}$ est l'écriture en base 10 de $6 \times 10^1 + 3 \times 10^0$.
 $10_{(2)}$ est l'écriture en base 2 de $1 \times 2^1 + 0 \times 2^0 = 2_{(10)}$.
 $A1_{(16)}$ est l'écriture en base 16 de $10 \times 16^1 + 1 \times 16^0 = 17_{(10)}$.

2.1.2 Codage décimal → binaire

Puissance de 2 et soustraction :

- on cherche la puissance de 2 supérieure ou égale au nombre décimal à convertir,
- on définit les valeurs des puissances de 2 par ordre décroissant,
- on procède par soustraction successive en ordre décroissant des puissances de 2 (si la soustraction est positive) jusqu'à atteindre 0.

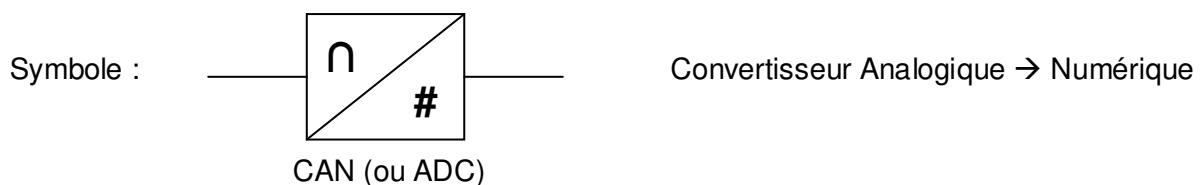
Exemple : conversion de $248_{(10)}$

Puissance de 2	$2^8 = 256$	$2^7 = 128$	$2^6 = 64$	$2^5 = 32$	$2^4 = 16$	$2^3 = 8$	$2^2 = 4$	$2^1 = 2$	$2^0 = 1$
Nombre binaire									
Reste à coder									

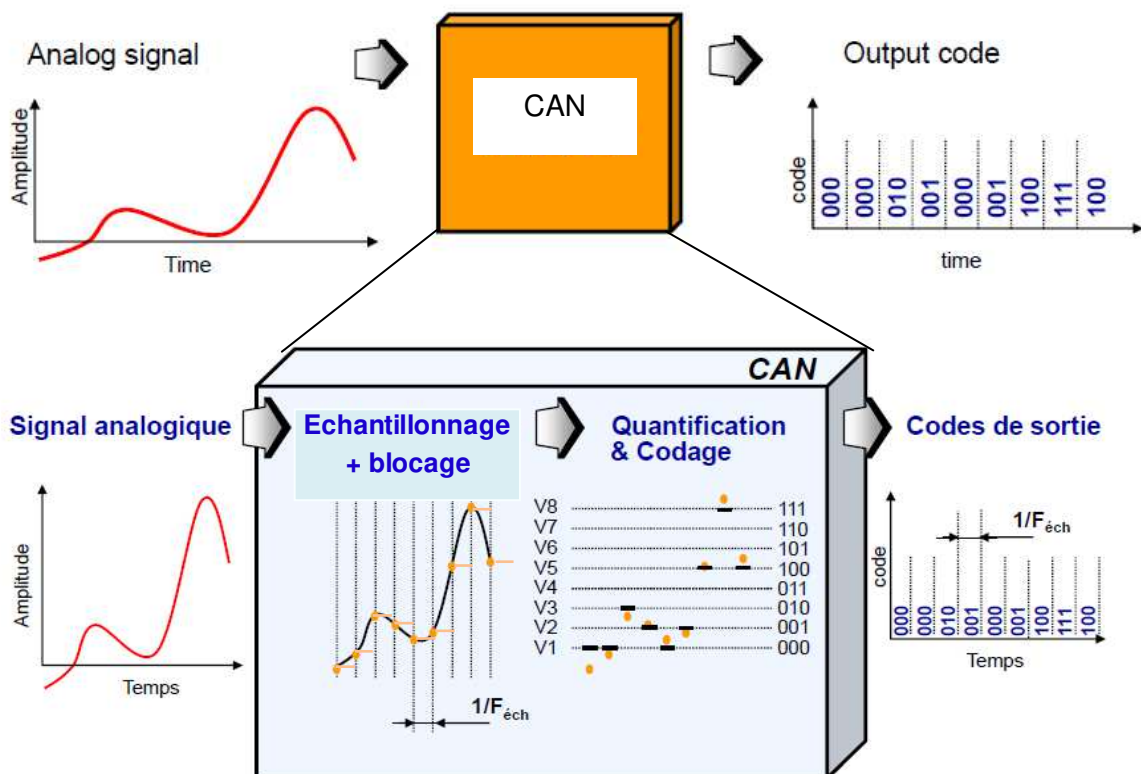
On en déduit que $248_{(10)} = 1001\ 0100_{(2)}$

Remarque : ce principe peut s'appliquer à la conversion vers n'importe quelle base (hexadécimale par exemple).

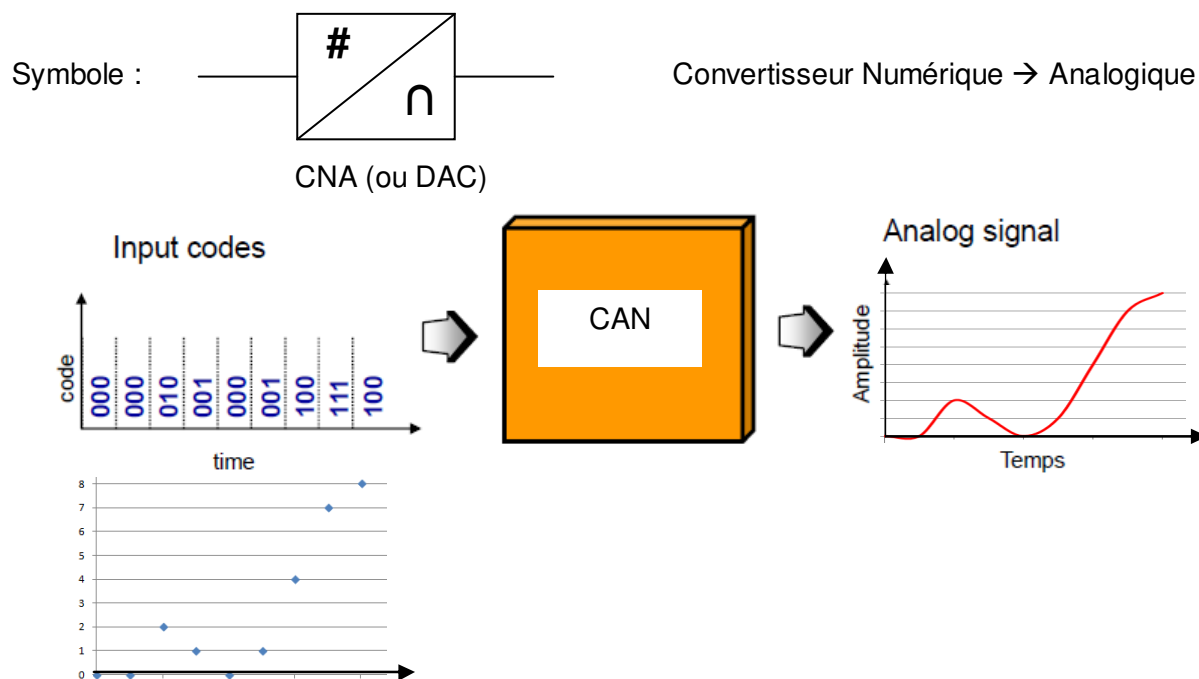
2.2 Conversion analogique-numérique



Le signal issu du capteur est mis en forme et amplifié par le conditionneur de signaux. Le convertisseur analogique-numérique effectue la numérisation du signal fourni par le conditionneur, après un échantillonneur bloqueur.



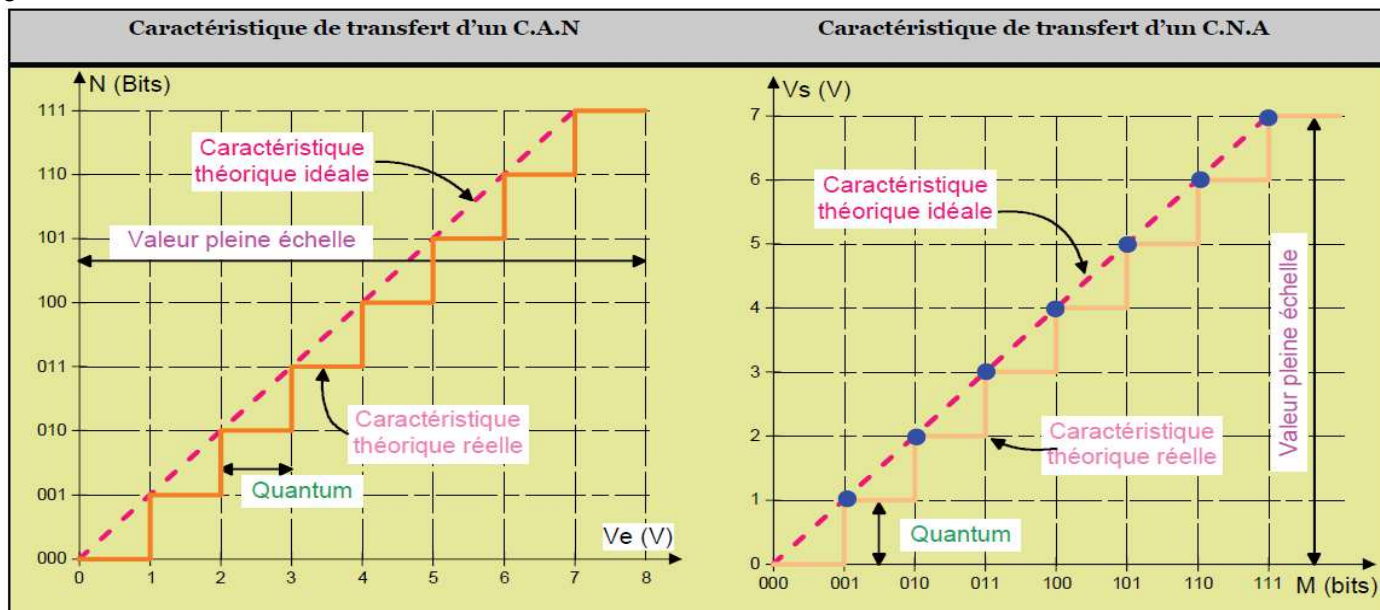
2.3 Conversion numérique-analogique



3 Caractéristiques des convertisseurs

3.1 Caractéristiques de transfert

La caractéristique d'un CNA ou CAN est la courbe représentant la grandeur de sortie en fonction de la grandeur d'entrée.



Le convertisseur est dit :

- **Unipolaire** s'il ne code que des grandeurs de même signe,
- **Bipolaire** s'il code des grandeurs de signe positif et négatif.

3.2 Quantum et résolution du convertisseur

Sur les courbes précédente, on remarque que plus le nombre de paliers est important plus la hauteur du palier devient petite et plus la courbe réelle du convertisseur se rapproche de la caractéristique idéale.

Cas du C.A.N		Cas du C.N.A	
$q = \frac{V_{PE}}{2^n}$	q : quantum en V $V_{PE} = V_{max} - V_{min}$: pleine échelle n : nombre chiffres binaires	$q = \frac{V_{PE}}{2^n - 1}$	q : quantum en V $V_{PE} = V_{max} - V_{min}$: pleine échelle n : nombre chiffres binaires

Exemple :

- CAN Arduino Uno : $V_{PE} = 5V$ sur $n = 10bits$ soit $2^n = 2^{10} = 1024$ et $q \approx 4,88 mV$
- CNA de quantum $q \approx 4,88 mV$ sur $n = 10bits \rightarrow V_{PE} = 5V$

La "résolution" du codeur CAN est liée à la sensibilité **s** du capteur associé :

$$\text{résolution} = \frac{q}{s}$$

où **s** est la sensibilité du capteur en $V. (unité\ mesurée)^{-1}$
résolution est la plus petite variation de la grandeur mesurée détectable
q est le quantum en **V** du codeur

Remarque : pour les codeurs de position, il arrive que l'on nomme "résolution" le nombre de paliers 2^n .

Le quantum **q** relie également la tension **Ve** à sont mot numérique **M** (aux erreurs d'arrondis près) :

$$q = V_e / M$$

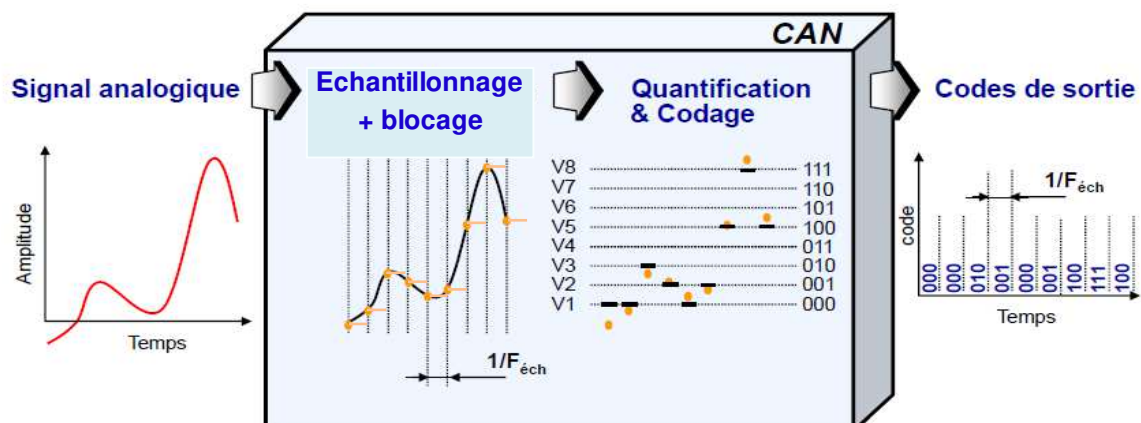
où **q** est le quantum en **V** du codeur
Ve est la tension analogique en **V** du signal
M est un entier (en base 10) :

- arrondi à l'entier inférieur ou égale pour un CAN
- arrondi à l'entier supérieur ou égale pour un CNA

3.3 Fréquence d'échantillonnage

Un convertisseur analogique / numérique (CAN) est un circuit hybride composé :

- d'un étage échantillonneur bloqueur
- d'un étage quantification codage qui transforme une grandeur analogique d'entrée V_e (souvent une tension) en une valeur numérique N exprimée sur n bits.



La fréquence d'échantillonnage doit respecter le théorème de Shannon pour une numérisation correcte :

Théorème de Shannon : $f_{éch} > 2.f_{max}$

où $f_{éch}$ est la fréquence d'échantillonnage (en Hz)
 f_{max} la fréquence maximale du signal analogique (en Hz)

3.4 Temps de conversion

Le **temps de conversion** ou temps d'établissement (**Setting time**) est le temps nécessaire pour convertir :

- Une valeur de tension en un nombre représentatif (pour un CAN)
- Un nombre numérique en une tension analogique proportionnelle (pour un CNA)

Il dépend de la technique employée pour la conversion. Il est donné par la documentation constructeur du composant.

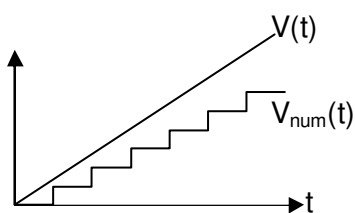
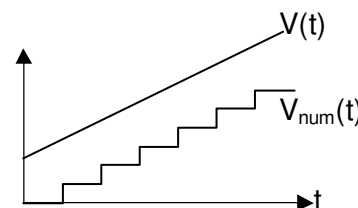
Exemple : carte Arduino Uno de l'ordre de 10ms ; carte NI6008 de l'ordre 0,1ms.

3.5 Précision et erreur

La précision est exprimée par l'écart entre la valeur réelle du signal et sa valeur discrétisée.

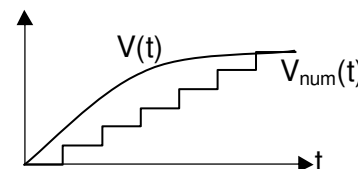
Les erreurs introduites dans la conversion proviennent des différents réglages à effectuer :

- **erreur de décalage** (la valeur numérisée a un offset par rapport à la valeur analogique) : se corrige par un offset au niveau de l'unité de traitement.



- **erreur de gain** : facteur d'échelle (ou pente) entre les signaux numériques et analogiques : se corrige par un facteur multiplicateur au niveau de l'unité de traitement.

- **erreur de linéarité** (la pente de conversion du convertisseur n'est pas linéaire) : plus difficile à corriger car l'unité de traitement doit appliquer un facteur multiplicateur variable adapté à la non linéarité constatée.



Pour limiter les erreurs de décalage, on peut donner $q/2$ comme tension au bit de poids faible (LSB).

Exemple : CAN codé sur $n=3$ bits, $q=1V$ et d'erreur $\epsilon_q = 1/2 q$ (LSB : tension du bit de poids faible)
Compléter ci-dessous le tableau.

Code	N_{10}	V_e (en V)

