

## Numérisation d'un signal

### I Signaux analogiques et numériques

#### Définition

On appelle **signal**, toute grandeur physique mesurée au cours du temps.

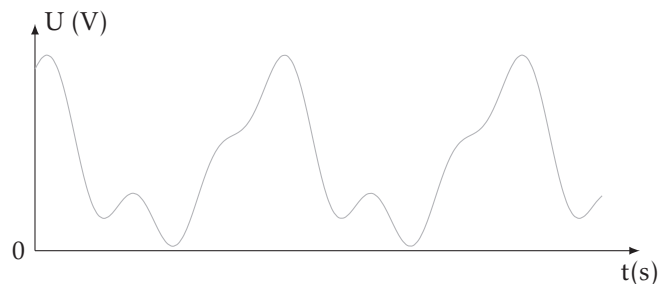
Exemples : tension électrique (en V), température (en en °C), niveau sonore (dB), pH (sans), cours d'une action ...

a. Signal analogique.

**Analogique** : contraire de logique !

La logique étant ici "l'art de manipuler" des 0 et des 1 ! (Mais que sont donc ces 0 et ces 1 ?)

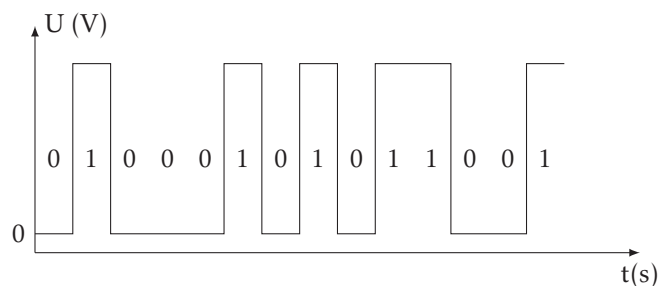
Un signal **analogique** est un signal **continu** au cours du temps.



b. Signal numérique.

#### Définition

Un signal **numérique** est une suite de 0 et de 1 logiques.



### II Numérisation d'un signal

a. Intérêt.

**C'est une question de qualité!**

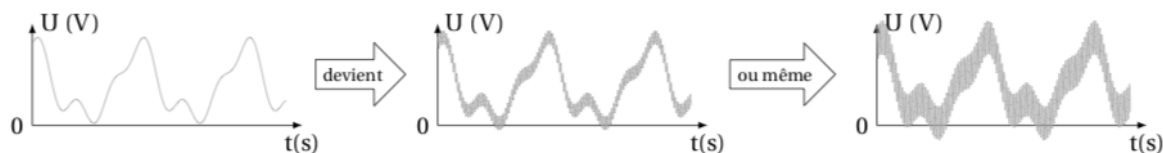
Parmi les bambins de terminale S lisant ce cours, peu connaissent le "doux" grésillement d'un disque vinyle !!

(Il paraît même qu'il y a des nostalgiques !)

C'est ce qu'on appelle le **bruit de fond**.

Il a pour origines, l'agitation thermique des électrons libres (transmetteurs du signal) dans les circuits électriques et surtout les perturbations dues aux ondes électromagnétiques dans lesquelles nous baignons littéralement !

Alors, un joli signal peut devenir :

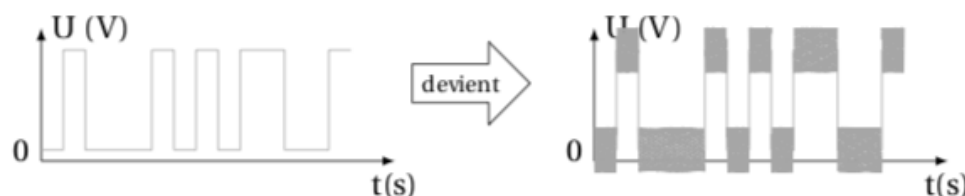


Si le signal est trop faible, on peut même ne plus rien reconnaître du tout !



Il est donc impossible de transmettre un tel signal !

Alors qu'en numérique :

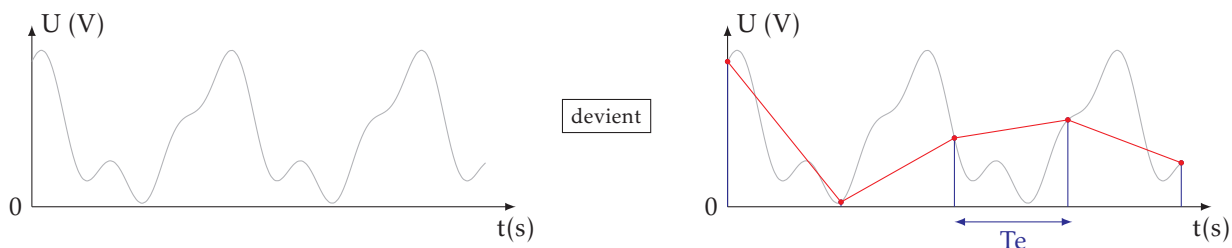


Malgré le bruit, on reconnaît quand même les 0 et les 1 !

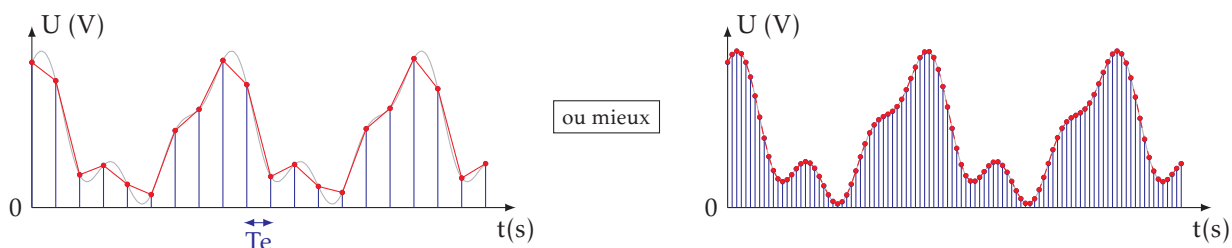
### III Échantillonnage

#### Définition

Comme son nom l'indique, l'échantillonnage consiste à prendre un échantillon du signal à intervalle de temps régulier. Les échantillons, mis bout à bout doivent restituer le signal le plus fidèlement possible.



Ici, la courbe rouge ne représente pas du tout le signal original. Il faut échantillonner plus **fréquemment** !



## Propriété

Le "rendu" d'un signal est meilleur si la **fréquence d'échantillonnage** est élevée.  
Le signal est "saucissonné" dans le temps le plus finement possible !

## IV Quantification

Il faut également saucissonner le signal en amplitude et attribuer un nombre entier aux différentes valeurs de la tension.

Comme l'ordinateur "travaille" en binaire, on divise la tension par une puissance de 2.

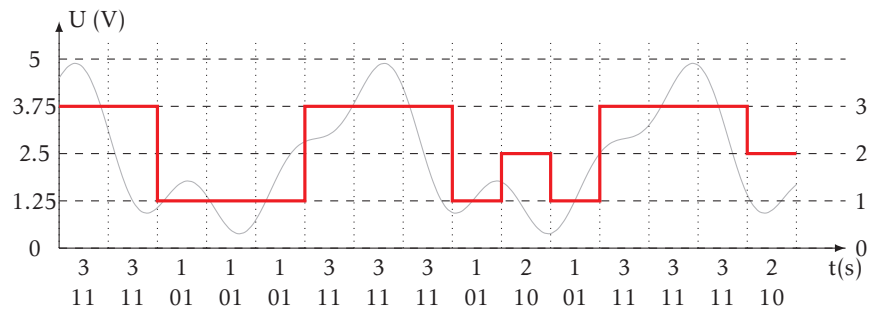
Dans les exemples suivants, on prendra toujours la même fréquence d'échantillonnage.

La tension varie entre 0V et 5V et la valeur numérique attribuée est immédiatement supérieure à la valeur réelle.

Codé sur 2 bits :

On obtient les correspondances :

U (V)	Décimal	Binaire
0	0	00
1,25	1	01
2,5	2	10
3,75	3	11



## Définition

On appelle **pas**, la différence de tension entre deux valeurs successives.

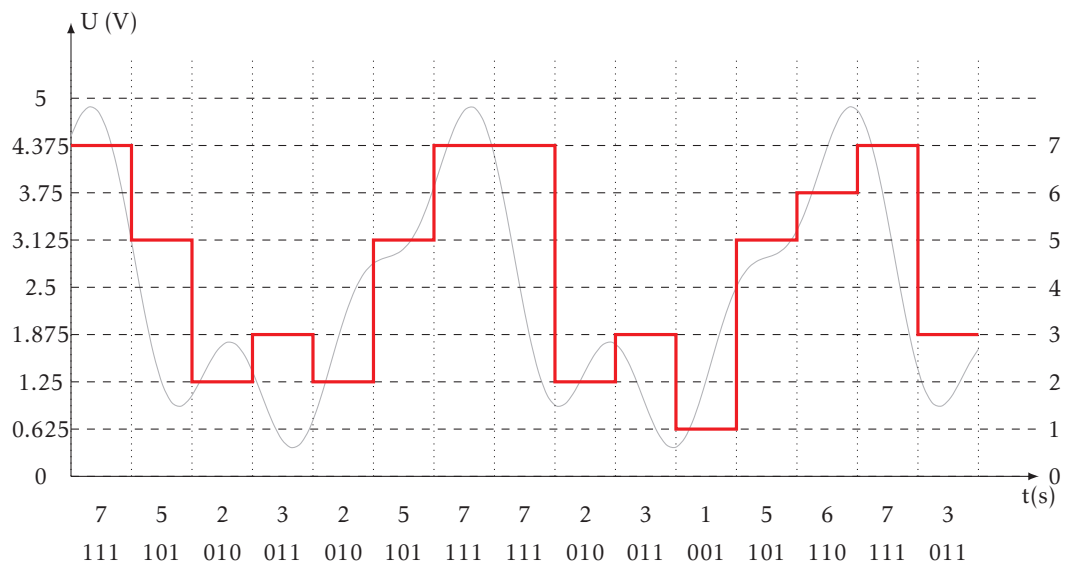
Il dépend du nombre  $n$  de bits utilisés pour le codage ainsi que de la valeur de la tension maximale  $U_{max}$  :

$$p = \frac{U_{max}}{2^n}$$

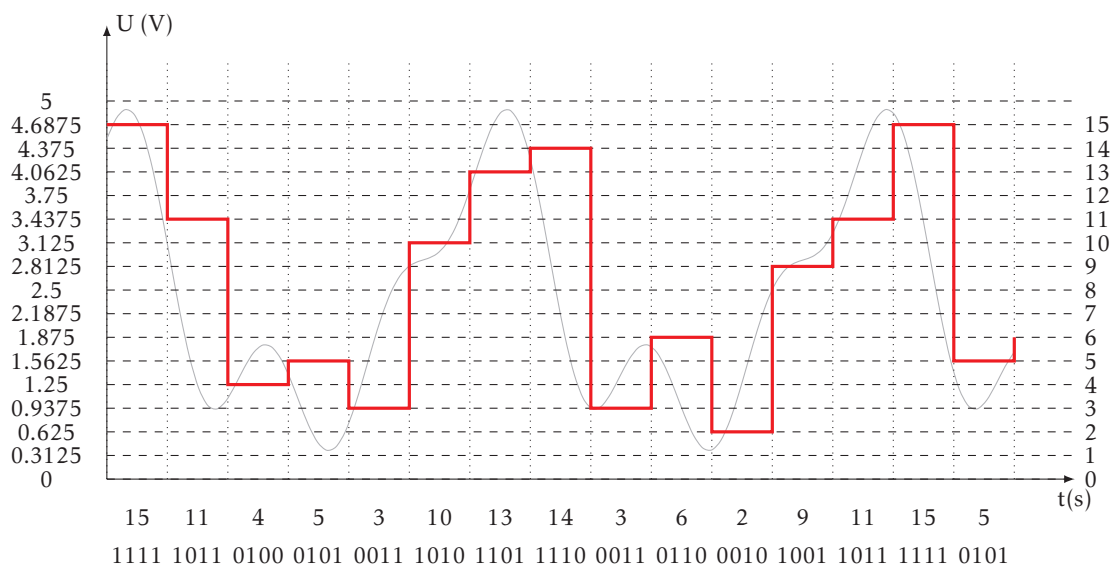
Le signal rouge est la restitution en tension du signal numérique. Il n'est pas très ressemblant au signal analogique !

Il faut donc "saucissonner" davantage.

Sur 3 bits

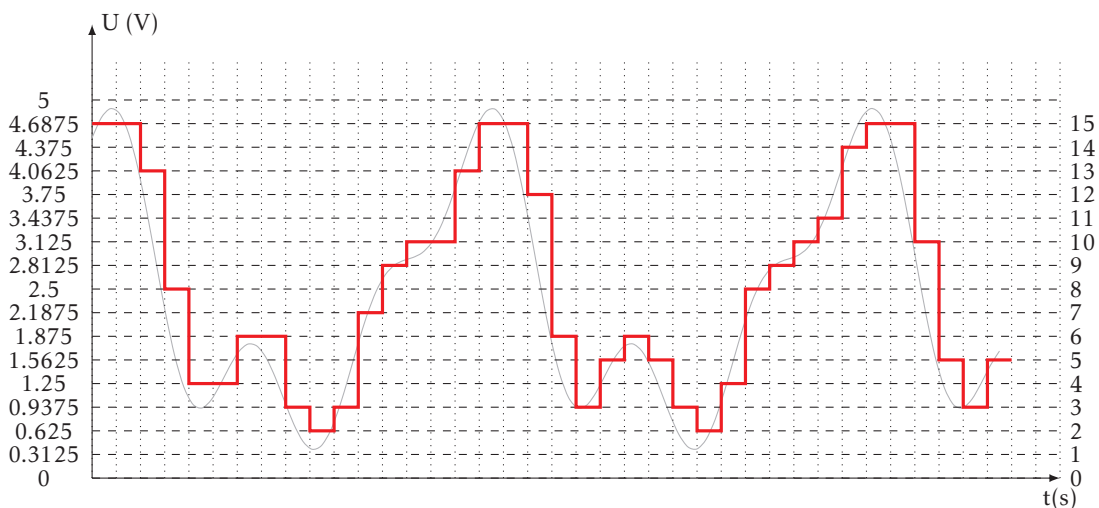


### Sur 4 bits



Même si le signal rouge s'approche un peu de signal réel, la ressemblance n'est tout de même pas frappante!

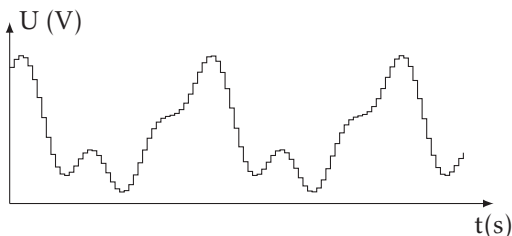
Pour progresser encore, il faut augmenter la fréquence d'échantillonnage et diminuer le pas.



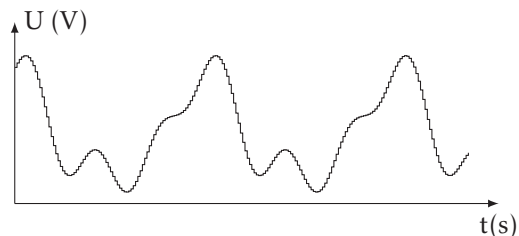
Je vous laisse vérifier que le codage est bien :

```
111111111101100001000100011001100011001000110111100110101010110111111111111000
110001101010110010100110010010010001001101010111110111111110100101000110101
```

Finalement, le signal pourrait ressembler à :



puis à



### Qu'en est-il aujourd'hui ?

Les ordinateurs actuels codent les informations sur 64 voire 128 bits, c'est-à-dire une suite de 64 ou 128 zéros ou uns par échantillon et à des fréquences avoisinant le MHz.

Cela laisse imaginer le nombre d'informations à stocker !

## V Transmission

Transmettre des informations signifie les faire transiter à travers un "tuyau". Aujourd'hui, les câbles (notamment les fibres optiques) permettent de faire passer plusieurs signaux simultanément.

### Définition

On appelle **débit binaire**, le nombre de bits transmis par unité de temps. Il s'exprime en  $bits.s^{-1}$

