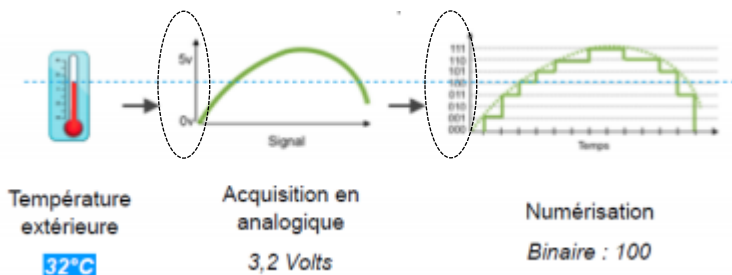
	Fiche de connaissance	Cycle 4
	LE LANGAGE INFORMATIQUE BINAIRE CONVERSIONS, CODAGE, CALCULS	Technologie
		IP

Le langage de l'ordinateur

Toutes les informations entrées dans un ordinateur, même les mots et les images, sont stockées et traitées sous forme de successions d'uniquement **deux valeurs numériques**, « 0 » ou « 1 » (d'où le mot « binaire »). Ces 2 valeurs sont appelées des « bits ».

La conversion des valeurs du signal dans notre système décimal en combinaisons de 0 et 1 (système binaire, appelée **codage binaire**, intervient lors de la numérisation (ou Conversion Analogique Numérique, CAN).



Le bit est la plus petite quantité d'information que peut traiter un ordinateur.

4 bits à la suite, par exemple 1001 forment un **nibble** ; 8 bits à la suite, par exemple 10011011 forment **1 octet**. L'octet est l'unité de base de stockage de l'information dans un ordinateur. On peut coder sur plusieurs octets (par exemple sur un CD, le son est codé sur 2 octets =16 bits).

Du système binaire au système décimal (nombres entiers)

Nos mathématiques classiques sont appelées « décimales » : tous les nombres peuvent être écrits avec 0,1,2,3,4,5,6,7,8,9 (soit 10 chiffres) et décomposés en puissances de 10.

En mathématiques binaires, tous les nombres peuvent être écrits avec 0 et 1 uniquement (2 chiffres) et décomposés en puissances de 2.

Pour convertir un nombre binaire en nombre décimal, il faut connaître les puissances de 2. Attention : $2^0=1$!

Exemples :

Que vaut en décimal le nibble **1101** ?

On utilise le tableau suivant :

Bit	1	1	0	1
Poids du bit	$2^3=8$	$2^2=4$	$2^1=2$	$2^0=1$
Valeur du bit x poids du bit	$1 \times 8=8$	$1 \times 4=4$	$0 \times 2=0$	$1 \times 1=1$
Décimal	$8+4+0+1=$ 13			

Que vaut en décimal l'octet **10101101** ?

On utilise le tableau suivant :

Bit	1	0	1	0	1	1	0	1
Poids du bit	$2^7=128$	$2^6=64$	$2^5=32$	$2^4=16$	$2^3=8$	$2^2=4$	$2^1=2$	$2^0=1$
Valeur du bit x poids du bit	$1 \times 128=128$	$0 \times 64=0$	$1 \times 32=32$	$0 \times 16=0$	$1 \times 8=8$	$1 \times 4=4$	$0 \times 2=0$	$1 \times 1=1$
Décimal	$128+0+32+0+8+4+0+1 =$ 173							

Si on rajoutait un bit à gauche de l'octet, son poids serait de 2^8 , etc.

Attention : le bit de plus faible poids est celui le plus à droite quand on écrit un nombre binaire.

Du système décimal (nombres entiers) au système binaire

Il faut décomposer le nombre décimal en puissances de 2.

Exemple : comment s'écrit en binaire *sur un octet* (8bits) 186 ?

$$186 = 0 \times 256 + 1 \times 128 + 1 \times 32 + 1 \times 16 + 1 \times 8 + 0 \times 4 + 1 \times 2 + 0 \times 1 = 1 \times 2^7 + 0 \times 2^6 + 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$$

→ l'octet résultant est : 01111010

Autre exemple : comment s'écrit en binaire *sur un octet* (8bits) ou sur un *nibble* (4bits) le nombre 13 ?

$$13 = 1 \times 8 + 1 \times 4 + 0 \times 2 + 1 \times 1$$

$$\text{ou encore } 13 = 0 \times 256 + 0 \times 128 + 0 \times 32 + 0 \times 16 + 1 \times 8 + 1 \times 4 + 0 \times 2 + 1 \times 1$$

→ Le nibble résultant est 1101

→ L'octet résultant est : 00001101

Le codage binaire

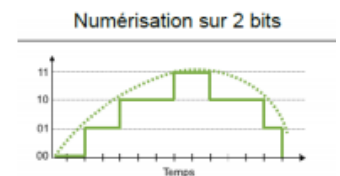
Pour être comprises et traitées par le microprocesseur de l'ordinateur, **toutes les informations doivent être converties ou codées en binaire**. En fonction du nombre de bits choisi pour le codage lors de la **numérisation**, c'est-à-dire de la **Conversion Analogique Numérique (CAN)**, le nombre de valeurs possibles en binaire est différent. Il correspond au nombre de combinaison de 0 et 1 différentes possibles.

Exemples de codage de nombres entiers :

Codage sur 1 bit : $2^1=2$ valeurs

0
1

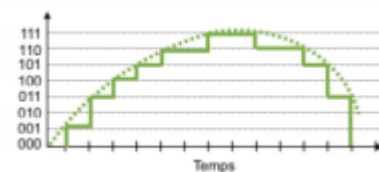
0	0	=0
0	1	=1
1	0	=2
1	1	=3



Codage sur 2 bits : $2^2=4$ valeurs possibles (de 0 à 3)

Codage sur 3 bits : $2^3=8$ valeurs possibles (de 0 à 7)

0	0	0	=0
0	0	1	=1
0	1	0	=2
0	1	1	=3
1	0	0	=4
1	0	1	=5
1	1	0	=6
1	1	1	=7



Codage sur 4 bits (1 nibble) : $2^4 = 16$ valeurs possibles (de 0 à 15)

Codage sur 8 bits (1 octet) : $2^8=256$ valeurs possibles (de 0 à 255)

Etc.

Conséquences :

- pour compter de 0 à 1, on n'a besoin de coder que sur 1 bit
- pour compter de 0 à 3, on n'a besoin de coder que sur 2 bits
- pour compter de 0 à 7, on n'a besoin de coder que sur 3 bits
- pour compter de 0 à 15, on n'a besoin de coder que sur 4 bits

Etc.