

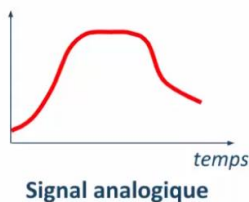
Langage Binaire

Nature du signal électrique: analogique ou numérique

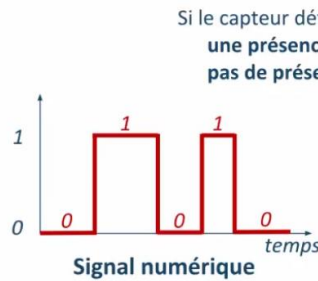
Le capteur détecte une grandeur physique, la mesure et la transforme en un **signal** qui est soit **analogique**, soit **numérique**.



Un **signal analogique** varie de manière **continue** dans le **temps**.



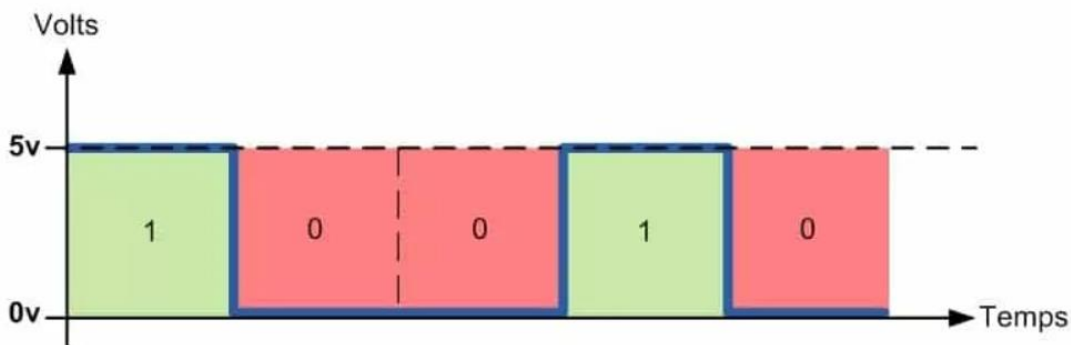
Un **signal numérique** n'a que **deux valeurs possibles** 0 ou 1.



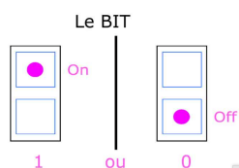
Le signal numérique

- valeur **0** = 0 volt sur le fil électrique
- valeur **1** = 5 volts sur le fil électrique

Ce qui nous donne :



les Bits



- Le bit (Binary Digit) est l'unité élémentaire utilisée en informatique.
- C'est un interrupteur qui contient l'information la plus simple : elle ne contient que deux valeurs : vrai ou faux, traduit par 1 ou 0.

Les octets

1 OCTET = 8 BITS



- Une succession de 8 bits correspond à un octet : c'est l'unité de base de l'information qu'un processeur d'ordinateur traite en même temps.
exemple : 11001100 ou 10111001.
- La valeur d'un octet peut représenter alors une lettre ou un chiffre

Unité de taille de l'information : en octet

L'octet est l'unité d'information que l'on retrouve par exemple pour donner la taille d'un fichier

1 Kilo-octet	1 Ko	= 1000 octets	1000 octets	Quelques lignes de textes
1 Méga-octet	1Mo	= 1000 Ko	1000.000 octets	Une petite photo
1 Giga-octet	1 Go	= 1000 Mo	1.000.000.000 octets	Plusieurs films videos
1 Téra-octet	1 To	= 1000 Go	1.000.000.000.000 octets	Une grosse bibliothèque de videos

Les combinaisons de bits dans un octet.

00000000 représente l'octet 0

00000001 représente l'octet 1

00000010 représente l'octet 2

00000011 représente l'octet 3

00000100 représente l'octet 4 ... on va ainsi jusqu'à

11111111 représente l'octet 255

avec 8 bits, on peut combiner les « 0 » et les « 1 » de 256 façons différentes,

Les octets représentent des lettres (et symboles)

Au début de l'informatique, on a attribué à chaque lettre minuscule, chaque lettre majuscule, chaque chiffre, chaque caractère spécial (/ * - + , ; ...) une combinaison particulière.

On utilisait alors une table « ASCII » qui faisait la correspondance entre une combinaison dans un octet et un caractère.

Par exemple la lettre « A » correspond à l'octet « 01000001 », ce qui correspond à l'octet 65.

Mais 256 combinaisons ne suffisaient pas à coder tous les lettres et caractères utilisés dans le monde : caractères chinois, indiens, russes, accents nationaux, ...

On a créé une norme Unicode pour pouvant contenir plus d'un million de caractères : la norme UTF8 est codée sur 4 octets.

Par exemple le caractère « € » ([euro](#)) est le 8365^e caractère du répertoire Unicode.

La principale caractéristique d'UTF-8 est qu'elle est rétro-compatible : tout caractère ASCII se code en UTF-8 sous forme d'un unique octet, identique au code ASCII. Par exemple « A » se code avec l'octet « 65 » en Ascii et en UTF-8..

Les octets peuvent représenter des nombres

- ❑ Le système binaire est un système en base 2, c'est-à-dire qu'il ne contient que les chiffres 0 ou 1:
- ❑ Nous calculons avec un système en base 10, c'est-à-dire qui contient les chiffres 0,1,2, ..., 7,8,9. (Probablement parce que nous avons 10 doigts).
- ❑ En base 10, le nombre **4237** correspond à $4 \times 1000 + 2 \times 100 + 3 \times 10 + 7 \times 1$

les milliers
les centaines
les dizaines
les unités
- ❑ En système en base 2, c'est le même principe mais les dizaines sont remplacés par 2, les centaines par $2 \times 2 = 4$, les milliers par $2 \times 2 \times 2 = 8$, etc

- ❑ Par exemple le nombre **11011** en base 2 correspond en base 10 à 27

$$1 \times 16 + 1 \times 8 + 0 \times 4 + 1 \times 2 + 1 \times 1 = 27$$

$2 \times 2 \times 2 \times 2$
 $2 \times 2 \times 2$
 2×2
2
les unités

Le système hexadécimal

- L'hexadécimal est un système d'écriture de nombres en base 16, avec les symboles 0,1,2,3,4,5,6,7,8,9,a,b,c,d,e,f.
- Celles-ci sont divisées en groupes de quatre bits et converties en nombres hexadécimaux (voir table ci-contre)
- Le système hexadécimal est utilisé pour faciliter la lisibilité de grands nombres comme les longues chaînes de bits.
- Ce système a été adopté massivement en informatique car il permet d'écrire 1 octet (8 bits) avec 2 caractères hexadécimaux
- Exemple : l'octet 01101100 s'écrit 6A
en effet 0110 pour 6 et 1100 pour A
- On retrouve cette notation dans le code qui identifie les cartes réseaux dans les ordinateurs ou les box.

Exemple : **01:00:5E:6H:FF:02**

Nombre décimal	Nombre binaire de 4 bits	Nombre hexadécimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

Utilisation des octets pour une photo noir et blanc à l'écran :

- Un écran est constitué de petits points (Pixel) que l'on peut éclairer plus ou moins fortement.
- Ces pixels sont alignés sur des lignes et dans des colonnes
- Un écran (télé, ordinateur, téléphone) contient environs un million de pixels (par exemple : 1000 lignes fois 800 colonnes).
- Dans une image en noir et blanc, on va associer la puissance de la lumière de chaque point de l'écran (pixel) avec la valeur d'un octet.
- Ainsi, Si la puissance de la lumière est faible, la valeur de l'octet sera proche de 0, et à l'inverse, si elle la lumière est forte, blanche, son code sera proche de 255 (voir tableau ci-contre)
00000000 = 0 représente un pixel noir
11111111 = 255 représente un pixel blanc
- Pour un écran en couleur, on procède de la même façon mais un pixel noir et blanc est remplacé par 3 pixels, l'un d'une couleur bleu, l'autre d'une couleur rouge, et le dernier d'une couleur verte (qui remplace le jaune difficile à obtenir)
- Il suffit d'éclairer plus ou moins les trois pixels de couleur différente pour obtenir des couleurs intermédiaires
- Pour obtenir un film, il suffit d'afficher des séries d'images (en général, 20 images / secondes)

0	3	6	9	12	15	18	21
24	27	30	33	36	39	42	45
48	51	54	57	60	63	66	69
72	75	78	81	84	87	90	93
96	99	102	105	108	111	114	117
120	123	126	129	132	135	138	141
144	147	150	153	156	159	162	165
168	171	174	177	180	183	186	189
192	195	198	201	204	207	210	213
216	219	222	225	228	231	234	237
240	243	246	249	252	255		128

Les fichiers

- Un fichier est un ensemble d'octets que peut traiter un ordinateur, une tablette, un téléphone.
- Il y a plusieurs types de fichier : les fichiers peuvent contenir du texte, une photo, un programme, une vidéo, un son, ...
- On vient de voir que selon le type d'information contenu dans un octet (image, chiffre, caractère), le processeur va travailler différemment.
- Pour savoir quel type de fichier est traité, il suffit de regarder après le nom du fichier, les lettres (en général 3) qui suivent le point :

Exemple : « monlivre.doc » « maphoto.jpg » « mavideo.mp4 »

- On associera donc un type de fichier à une, ou plusieurs applications, qui permettent de traiter convenablement ce fichier.

Exemple :
« monlivre.doc » → traité par l'application Word
« maphoto.jpg » → traité par une application de traitement d'image
« mavideo.mp4 » → traité par une application de traitement du son
« monprogramme.exe » → correspond à un programme exécutable
« monpowerpoint.ppt » → traité par l'application Power-point
« monexcel.xls » → traité par l'application E