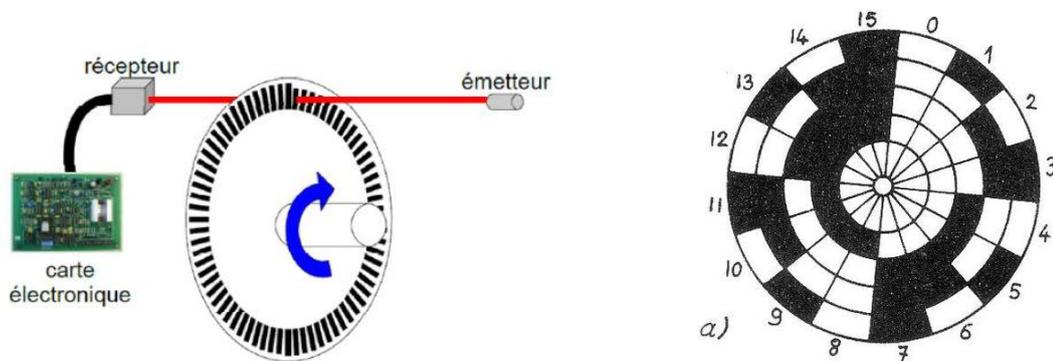


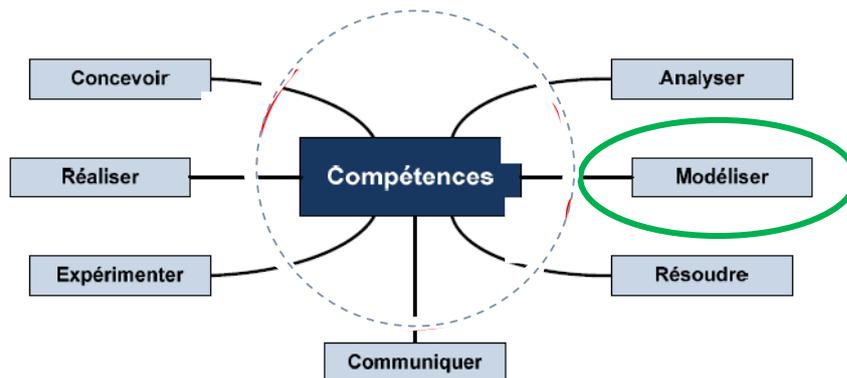
# Cycle 8: Modélisation des systèmes à évènements discrets

## Chapitre 1 : Numération et codage de l'information



### Compétences:

- Modéliser les systèmes à évènements discrets (tables vérité, fonctions logiques...)
- Traduire le comportement d'un système à évènements discrets



# Sommaire

1. <u>Introduction</u>	3
2. <u>Information analogique – numérique</u>	4
3. <u>Nécessité du codage de l'information</u>	5
4. <u>Tout est binaire</u>	6
5. <u>Mode de représentation : les bases</u>	7
5.1. Base 10 : décimal	7
5.2. Base 2 : binaire	7
5.3. Base 16 : hexadécimal	8
6. <u>Les codes</u>	9
6.1. Code binaire réfléchi : gray	9
6.2. Code BCD	10
6.3. Code n parmi p	10
6.4. Code barre « EAN »	12
6.5. Code barre 2D	13



## 1. Introduction

L'informatique est la science du traitement de l'information. Au sens étymologique, l'information est ce qui donne une forme à l'esprit.

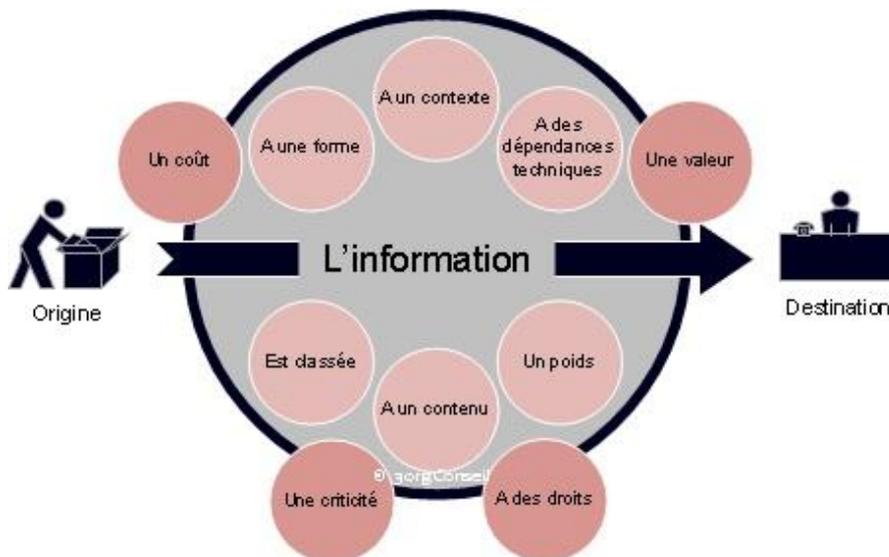
L'information est **par nature immatérielle**, et elle est **captée en permanence par nos sens**, comme:

- le bruit d'un animal dans un bois,
- le souvenir des vacances,
- la vibration d'un moteur,
- le goût d'un fruit,



L'information est étroitement liée aux notions de contrainte, communication, contrôle, donnée, formulaire, instruction, connaissance, signification, perception et représentation.

L'information désigne à la fois le message à communiquer et les symboles utilisés pour l'écrire.



Notre cerveau va traiter, mémoriser cette information puis essayer de la diffuser avec le moins de pertes possibles. Devant la quantité d'information à appréhender, nous avons conçus des **dispositifs techniques et des codes** destinés à **capter, mémoriser, transmettre et reproduire** cette information:

- *phonographe, platine disque, CD audio, MP3...*
- *télégraphe, téléphone filaire, mobile...*
- *boulier, machine à calculer de Pascal, calculateur, ordinateur...*

### Exemple de l'ordinateur:

- 1) **Echanger de l'information** (clavier, souris, bus, écran, imprimante, modem...).
- 2) **Mémoriser de l'information** (DD, RAM, ROM, DVD, CD, clé USB, ...).
- 3) **Calculer de l'information** (processeur).

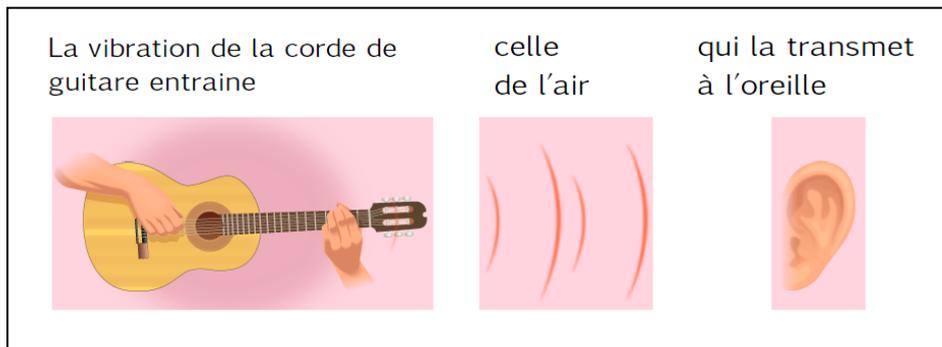


2. Information analogique-numérique

Information analogique:

Jusqu'à ces dernières décennies l'information et son support a été **analogique**, c'est à dire une **grandeur continue**. Les systèmes analogiques paraissent maintenant plus rudimentaires et manquent de précision (approximation, tolérance...) dès que le support qui sert à son écriture ou à sa transmission est de qualité insuffisante. A chaque manipulation des signaux analogiques **les erreurs d'imprécision viennent dégrader l'information**.

ex: chaîne de traitement analogique du son de la source à la restitution.



Si on souhaite enregistrer le son sur un disque vinyle, il est nécessaire de capter cette vibration à l'aide d'un micro, ce qui rajoute des parasites. De même, le passage du son dans les hauts parleurs à la lecture du disque rajoute aussi des parasites.



Information numérique:

Numériser l'information consiste à la quantifier, c'est à dire à **lui attribuer une valeur, prise dans un ensemble fini de valeurs**.

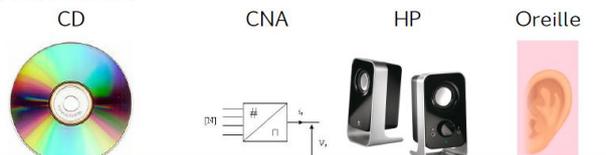
On note dans cette définition toute la difficulté et les limites de l'information numérique. Ainsi, pour notre exemple, il va falloir **définir le nombre de valeurs distinctes que peut prendre l'info** sonore et avec **quelle fréquence** afin que la restitution soit le plus proche possible de la réalité.

La chaîne de traitement de l'info sonore numérisée devient:

Pour l'enregistrement :

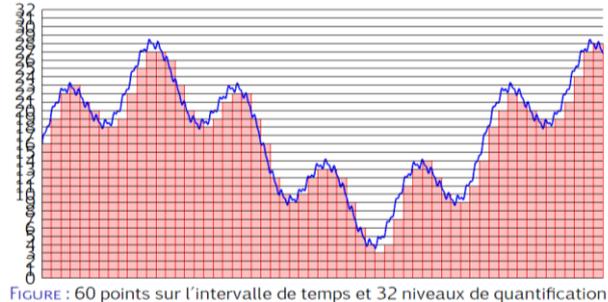
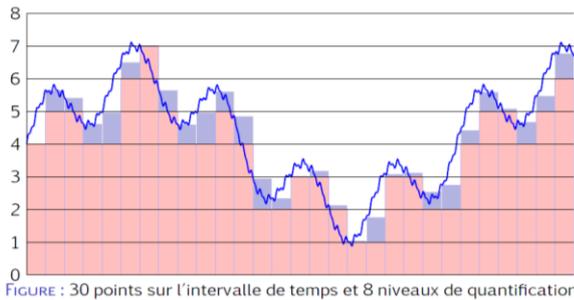


Pour l'écoute :



La numérisation du son comprend 2 étapes principales:

- **l'échantillonnage:** mesurer la valeur du son à des instants précis (nb échantillons / seconde = **fréquence d'échantillonnage en Hz**)
- **quantification:** l'intervalle maxi du signal (min-max) est divisé en N valeurs, le signal mesuré est arrondi à la valeur numérique la plus proche: c'est en général une puissance de 2 ( $2^8=256$ ,  $2^{16}=65536$ )

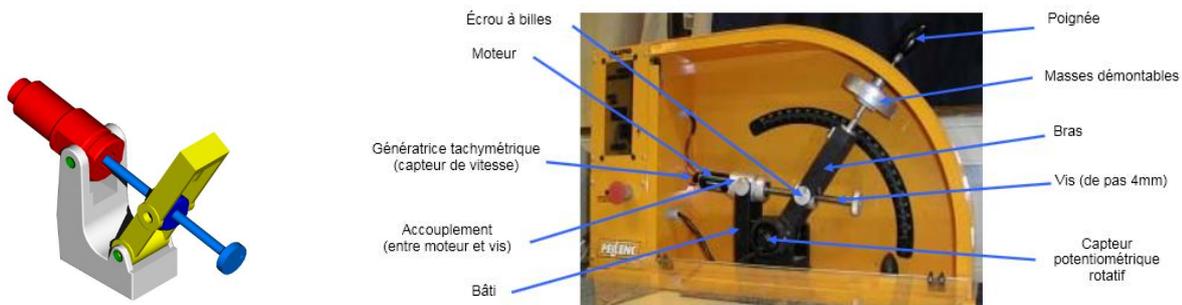


Attention: plus fréquence échantillonnage est petite, plus la représentation numérique sera précise **mais plus la quantité d'info à stocker sera grande.**

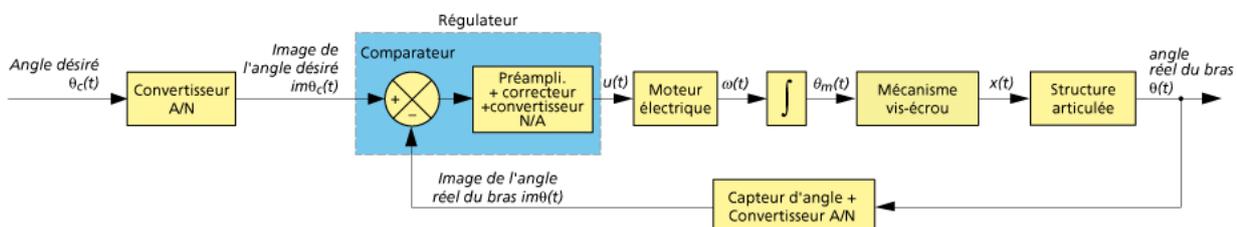
### 3. Nécessité du codage de l'information

La communication nécessite la **compréhension** entre les deux entités communicantes. L'émetteur envoie de l'information au récepteur qui doit savoir **l'interpréter** pour la comprendre. Ainsi, **le codage de l'information** est la première étape de toute communication.

Exemple du Maxpid dans le labo SII:



La chaîne fonctionnelle du Maxpid (vue en TP) est la suivante. L'information de "consigne de position" donnée par l'utilisateur (angle désiré  $\theta_c$ ) est numérisée, puis comparée à l'image numérisée de la position réelle du bras (via le capteur d'angle) puis transformée en tension pour commande moteur...etc.



L'information dans les systèmes actuels est présente sous diverses formes, et doit bien souvent **être codée pour être interprétée à différents niveaux et par différents composants.**



#### 4. Tout est binaire...

Depuis toujours, les **ordinateurs calculent en binaire**. Il s'agit de la **base 2** dans laquelle seul les **symboles 0 et 1**. Ils sont utilisés pour écrire les nombres, sans bien sûr que cela limite la taille des nombres représentables. Ceux ci vont être stockés dans des cases qui, elles, ont une taille fixée. L'ordinateur est donc **limité dans ses capacités à exprimer n'importe quel nombre**.

Chaque valeur "0 ou 1" est appelée « **bit** » (binary digit), mémorisée dans des cases de taille limitée. Il est possible de représenter physiquement cette information binaire par un signal électrique ou magnétique, qui, au-delà d'un certain seuil, correspond à la valeur 1.

Pour simplifier les grandeurs, on utilise des multiples de **l'octet** (ou Byte en anglais) = **ensemble de 8 bits**.

Un bit ne suffisant pas pour exprimer toutes les tailles de fichiers disponibles, des unités de mesures (comme le centimètre, le mètre et le kilomètre par exemple) ont été mises en place :

- *Le kilo-octet (Ko) : 1 Ko équivaut à 1000 octets (ou 1024 en puissance de 2)*
- *Le méga-octet (Mo) : 1 Mo = 1000 Ko.*
- *Le giga-octet (Go) : 1 Go = 1000 Mo.*
- *Le téra-octet (To) : 1 To = 1000 Go.*

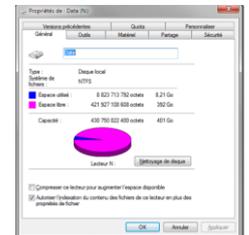
Attention : de nombreux logiciels (parfois même certains systèmes d'exploitation) utilisent toujours la notation antérieure à 1998 pour laquelle :

- *Un kibi-octet (kio) =  $2^{10}$  octets = 1024 octets*
- *Un Mébi-octet (Mio) =  $2^{20}$  octets = 1024 ko = 1 048 576 octets*
- *Un Gibi-octet (Gio) =  $2^{30}$  octets = 1024 Mo = 1 073 741 824 octets*
- *Un Tébi-octet (Tio) =  $2^{40}$  octets = 1024 Go = 1 099 511 627 776 octets*

#### Exercice:

Calculer le nombre de bits d'informations contenus dans un disque dur d'une capacité de 140 Giga Bytes (ancienne norme)

$$140 \text{ Giga Bytes} = 140 \times 2^{30} \times 8 = 1,2025 \times 10^{12} \text{ bits}$$



Les ordinateurs ont été conçus autour des propriétés de la **logique booléenne**. La logique booléenne ou algèbre de Boole est une logique qui ne traite que des variables ne pouvant prendre que **deux états** (vrai/faux, 0/1, haut/bas, appuyé/relâché, éclairé/éteint).



## 5. Mode de représentation des nombres: les bases

Le **numération** désigne le mode de représentation des nombres. Dans la vie de tous les jours, nous utilisons le **système décimal** c'est à dire un **codage avec 10 chiffres de 0 à 9**. Cependant il en existe d'autres :

- Le **système binaire** (algèbre de Boole, utilisé par les ordinateurs)
- Le **système hexadécimal** (utilisé en intermédiaire du système binaire)
- Le système sexagésimal (base 60, utilisé pour compter le temps et les angles)

### 5.1. La base 10: système décimal

C'est notre base naturelle de comptage, car nous avons 10 doigts aux mains.

On code avec **10 chiffres** (0 à 9), la place occupée par chaque chiffre dans un nombre est le **rang** (*on commence au rang 0*)

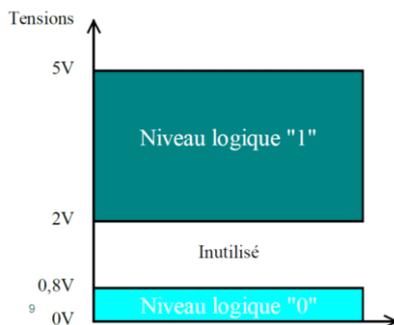
Le **poids** est la quantité représentée par une unité d'un rang donné : **Poids = Base<sup>Rang</sup>**.

En multipliant chaque chiffre par le poids de son rang, et en additionnant le tout, on obtient le **nombre**.

$$\text{ex: } 436_{(10)} = 4 \cdot 10^2 + 3 \cdot 10^1 + 6 \cdot 10^0$$

### 5.2. La base 2: système binaire

- Base : 2
- Chiffres (ou digits en informatique) : 0,1
- Poids : 1, 2, 4, 8, 16, 32, 64, 128, 256, ...



*Exemple : Dans les systèmes électroniques de commutation, les valeurs suivantes sont généralement admises (sauf spécifications contraires des constructeurs) pour les niveaux logiques 0 et 1.*

Un peu de vocabulaire :

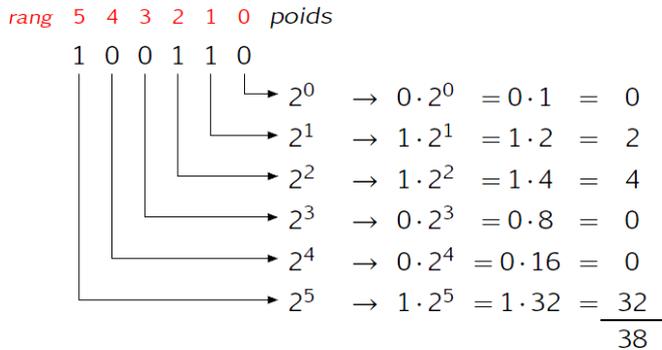
- **MSB** : Most significant Bit : **Bit de poids le plus fort**
- **LSB** : Low Significant Bit : **Bit de poids le plus faible**
- **Octet** (en anglais, byte) : nombre binaire composé de 8 bits
- Notation :  $(11101001)_2 = \%11101001$

Le tableau représentant les poids de chaque bit d'un octet :

Poids	$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$
	128	64	32	16	8	4	2	1



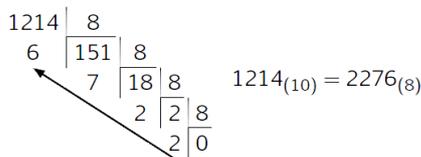
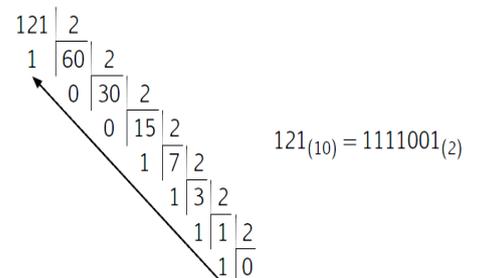
Transcodage de la base 2 vers base 10:



Transcodage de la base 10 vers base 2:

Pour passer de la base 10 vers la base 2, il suffit de réaliser des **divisions euclidiennes (entières) successives** jusqu'à ce que le quotient soit nul. le nombre binaire se lit du dernier reste vers le premier.

*Nota: on peut diviser le nombre décimal par 2, 4, 8, 16... (base 2, ...)  
pour avoir l'expression dans la base voulue.*



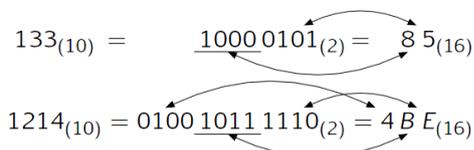
5.3. La base 16: système hexadécimal

- Base : 16
- Chiffres : 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F
- Poids : 1, 16, 256, 4096, ...
- Notation :  $(C48)_{16} = \$C48$

Le système **hexadécimal (la base 16)** est très souvent utilisé en informatique car il permet une représentation rapide et courte des nombres binaires. En effet, on peut en regroupant les bits du nombre en base 2 **par 4** en partant de la droite et en complétant par des 0 à gauche si nécessaire pour avoir un multiple de 4 bits, obtenir directement le code hexadécimal en remplaçant chaque groupe de 4 bits par le code hexadécimal correspondant

base															
10	0	1	2	3	4	5	6	7	8	9	10	11	12	13	
2	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	
16	0	1	2	3	4	5	6	7	8	9	A	B	C	D	

Transcodage de la base 2 vers base 16:

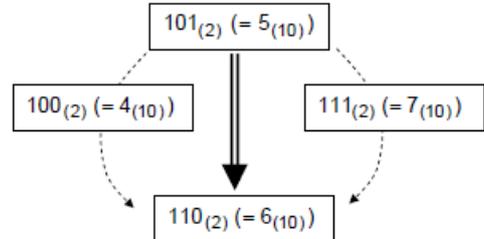


6. Les codes

6.1. Code binaire réfléchi ou « code gray »

Le code binaire naturel a pour inconvénient majeur de pouvoir introduire des erreurs entre 2 codes successifs car plusieurs bits peuvent changer d'état.

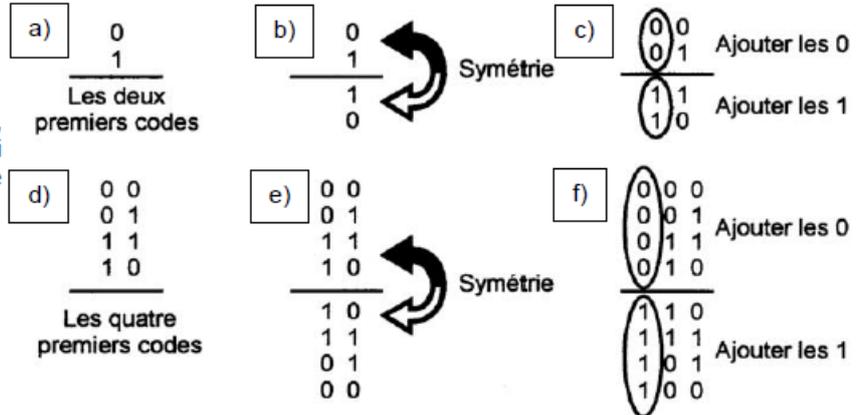
En effet entre le code  $101_{(2)} (= 5_{(10)})$  et le code  $110_{(2)} (= 6_{(10)})$ , deux bits changent d'état en même temps, ce qui est impossible physiquement : il existe deux transitions possibles,  $100_{(2)} (= 4_{(10)})$  ou  $111_{(2)} (= 7_{(10)})$ . Ainsi, pendant un court instant, un code parasite risque donc d'introduire une erreur, ce qui peut être très ennuyeux pour un système de codage de position par exemple.



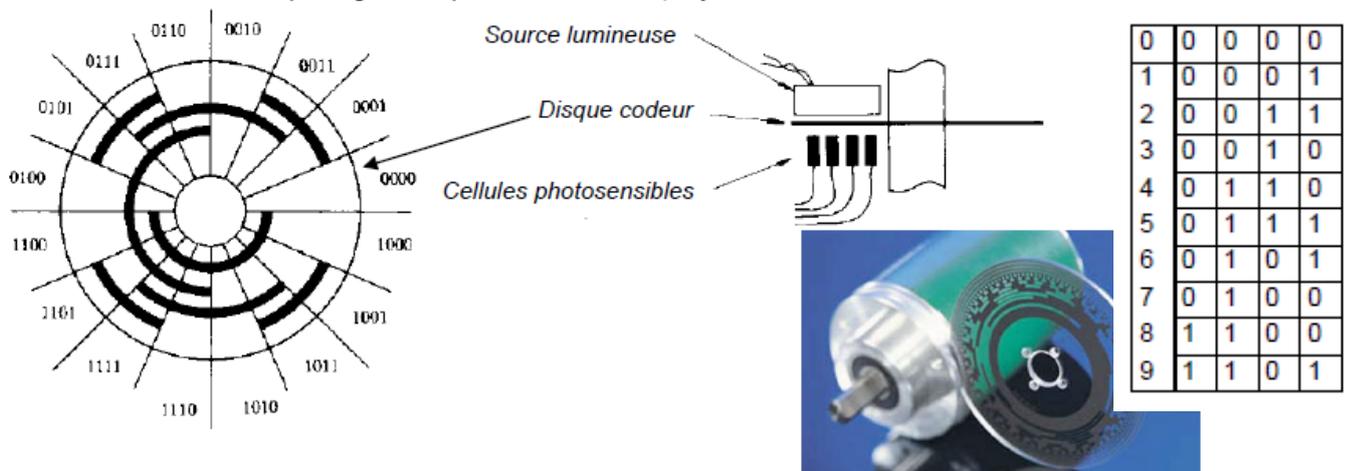
En revanche le code binaire réfléchi présente la particularité suivante :

*lorsque l'on passe d'une ligne à la suivante, seul un bit change d'état.*

Ce code, mis au point par Gray, prend le nom de binaire réfléchi car il existe des axes de symétrie dans la construction du code :



Ce code est utilisé pour la réalisation de capteurs numériques de position car il permet d'éviter toutes confusions de codes lors du passage d'une position à une autre, adjacente.





6.2. Code BCD « Binary Coded Decimal »

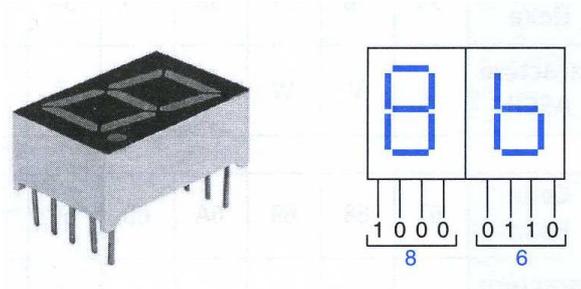
Le code B.C.D. est constitué de la manière suivante :

chaque chiffre du nombre décimal est codé en un nombre binaire pur de quatre bits.

Exemple :  $129_{(10)} = \underbrace{0001}_1 \underbrace{0010}_2 \underbrace{1001}_9$  (BCD)

Une application du codage BCD est l'**afficheur à 7 segments** : c'est un composant possédant quatre entrées logiques par chiffre, chaque chiffre étant codé en binaire. Lorsque l'on utilise N afficheurs, un nombre est représenté par la combinaison des N chiffres ce qui revient précisément à un codage BCD.

Dans le cas de deux afficheurs, les valeurs correspondantes peuvent être comprises entre 00 et 99 .



Codage BCD sur deux afficheurs à 7 segments.

Le code BCD présente deux inconvénients :

- il n'exploite pas toutes les possibilités du binaire, car il code seulement 10 chiffres sur 4 bits au lieu des 16 possibles (de 0000 à 1001 au lieu de 0000 à 1111)
- il faut le convertir pour effectuer des calculs arithmétiques, complexes sinon.

6.3. Code n parmi p

Le code p parmi n est un code à n bits dont p bits sont à 1 et (n-p) bits à 0.

Le nombre de

combinaisons répondant à cette définition est égale à  $(C_n^p !)$  arrangements.

Les particularités données par ce code sont les suivantes :

- code auto-correcteur : en effet, la lecture du code peut être associée à la vérification du nombre de 1 et de 0 dans l'information, ce qui permet un contrôle de l'information lue par la détection de code erronée.
- code personnel : en effet, il existe  $(C_n^p !)$  arrangements de la codification, ce qui permet de personnaliser son codage :

Le code 3 parmi 5 permet :

$$C_5^3 = \frac{5!}{2!3!} = 10 \text{ combinaisons}$$

Ce codage comporte trois « 1 » et deux « 0 », il existe également un code 2 parmi 5.

	b <sub>4</sub>	b <sub>3</sub>	b <sub>2</sub>	b <sub>1</sub>	b <sub>0</sub>
0	0	0	1	1	1
1	0	1	0	1	1
2	1	0	0	1	1
3	1	0	1	0	1
4	1	1	0	0	1
5	1	1	0	1	0
6	1	1	1	0	0
7	0	1	1	0	1
8	0	1	1	1	0
9	1	0	1	1	0

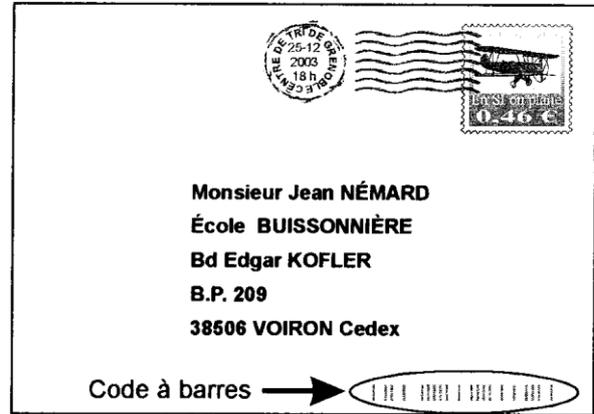
Table de vérité d'un code 3 parmi 5



**Application du code p parmi n : le Code POSTAL**

Dans les centres de tri de La Poste, les lettres de taille standard sont déposées en vrac dans une première trieuse qui les range toutes dans le même sens, adresse à l'endroit et vers l'avant.

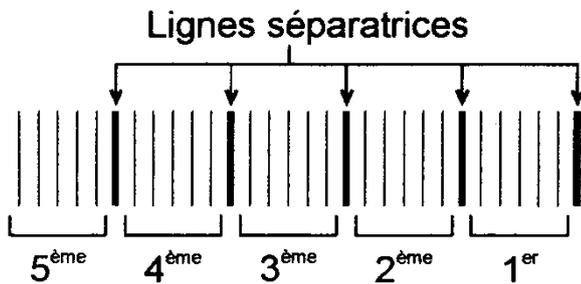
Les paquets de lettres sont alors déposés dans la deuxième trieuse qui a pour but d'imprimer un code à barres sur l'enveloppe. Un cliché est pris de chaque enveloppe, envoyé à un ordinateur qui doit déchiffrer le code postal inscrit sur l'enveloppe.



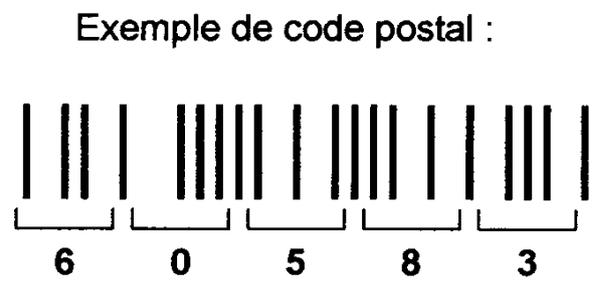
Si le code postal est reconnu et s'il y a compatibilité avec le nom de la ville, alors un code à barres correspondant au code postal est imprimé sur l'enveloppe.

Si le code postal n'est pas reconnu, un code à barres référence est imprimé sur l'enveloppe, le cliché est envoyé à une opératrice qui, sur sa console de vidéocodage, décide du code postal qui correspondra au code à barres référence.

Le dernier tri permet de déposer les lettres dans des casiers différents en fonction du code à barres lu.



Lecture des chiffres du Code Postal



Lire le code postal 38506

Le code postal utilise 3 barres parmi 5 pour coder un chiffre. Les barres sont « rosé fluo » et imprimées au bas des enveloppes. La lecture des codes à barres se fait de droite à gauche puisque les enveloppes se déplacent de gauche à droite dans les trieuses.

**Code Postal**

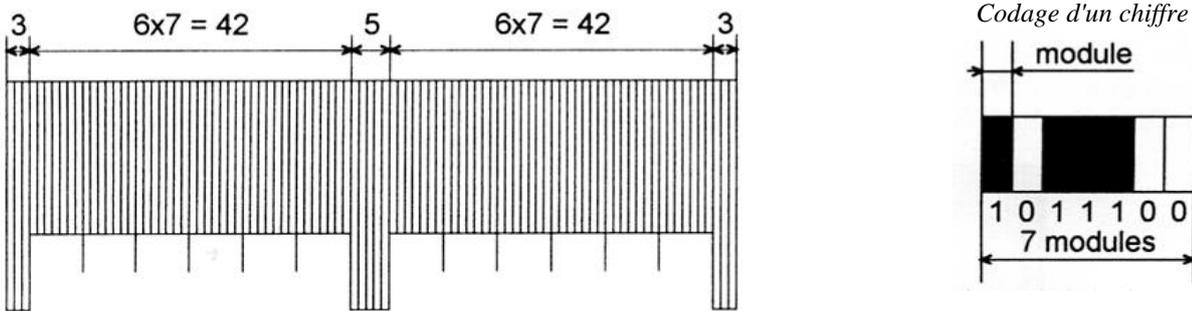




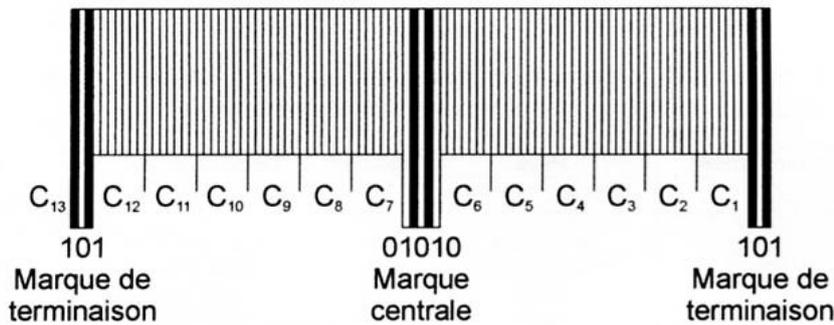
6.4. Code barre « EAN »

Le code EAN est le code à barres le plus utilisé dans le monde pour la distribution de produits d'alimentation: c'est un standard international. Il est composé de 8 ou 13 chiffres (code EAN 8 ou EAN 13).

Le code EAN 13 occupe 95 modules (3 + 6x7 + 5 + 6x7 + 3). Chaque module peut être blanc (0) ou noir (1). Un chiffre est codé par 7 modules :



Le code EAN 13 est délimité par deux marques de terminaison (101). Le code est structuré en deux zones de même taille séparées par une marque centrale (01010). Dans chaque zone sont représentés six chiffres (quatre pour le code EAN 8),  $C_1$  à  $C_6$  et  $C_7$  à  $C_{12}$  ( $C_1$  à  $C_4$  et  $C_5$  à  $C_8$  pour le code EAN 8) :



Aucune barre n'est attribuée au chiffre  $C_{13}$ , il est déterminé à partir des codages des chiffres  $C_7$  à  $C_{12}$  (voir plus loin).

Le chiffre  $C_1$  est une clé de contrôle calculée de la manière suivante :

$$C_1 = 10 - \left\{ \left[ 3 \times \sum_{i=1}^6 C_{2i} + \sum_{i=1}^6 C_{2i+1} \right] \text{MOD} 10 \right\}, \text{ où MOD } 10 \text{ signifie : « reste de la division entière par dix ».}$$

Chaque chiffre est codé sur 7 bits selon un des codes A, B ou C, suivant le tableau ci-dessous:

Chiffre	Code A	Code B	Code C
0	0001101	0100111	1110010
1	0011001	0110011	1100110
2	0010011	0011011	1101100
3	0111101	0100001	1000010
4	0100011	0011101	1011100
5	0110001	0111001	1001110
6	0101111	0000101	1010000
7	0111011	0010001	1000100
8	0110111	0001001	1001000
9	0001011	0010111	1110100

	$C_{13}$	$C_{12}$	$C_{11}$	$C_{10}$	$C_9$	$C_8$	$C_7$	$C_6$ à $C_1$
0	A	A	A	A	A	A	A	C
1	A	A	B	A	B	B	B	C
2	A	A	B	B	A	B	B	C
3	A	A	B	B	B	A	A	C
4	A	B	A	A	B	B	B	C
5	A	B	B	A	A	B	B	C
6	A	B	B	B	A	A	A	C
7	A	B	A	B	A	B	B	C
8	A	B	A	B	B	A	A	C
9	A	B	B	A	B	A	A	C



Modélisation des systèmes à évènements discrets: Numération - Codage

Il n'y a pas de confusion possible entre un code A, B ou C.

Ces codages permettent une lecture bidirectionnelle, puisqu'un code inversé n'est pas reconnu comme un code.

Les chiffres C<sub>1</sub> à C<sub>6</sub> sont codés en code C.

Les chiffres C<sub>7</sub> à C<sub>12</sub> sont codés en Code A ou B : la configuration de ces codes détermine le treizième chiffre, conformément au tableau droit ci-dessus.

Les chiffres C<sub>8</sub> à C<sub>13</sub> identifient l'entreprise et le pays.

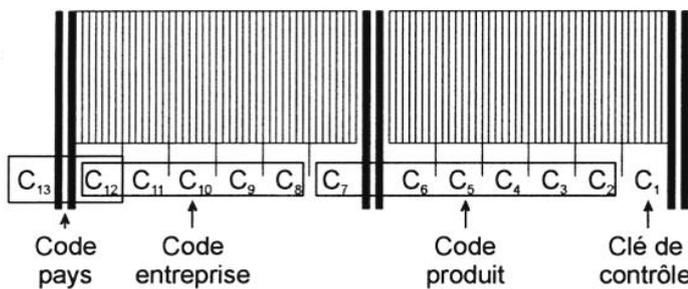
Les chiffres C<sub>2</sub> à C<sub>7</sub> identifient le produit dans l'entreprise.

En France, les chiffres C<sub>13</sub> et C<sub>12</sub> identifient le pays d'origine (ex: 0 pour la France) et les chiffres C<sub>12</sub> à C<sub>8</sub> identifient l'entreprise.

Le codage des entreprises est géré par Gencod - EAN France.

SIGNIFICATION CODE BARRE

Imprimerie IMAP



000 à 019							
030 à 039	Etats-Unis	520	Grèce	625	Jordanie	784	Paraguay
060 à 139							
300 à 379	France	528	Liban	626	Iran	786	Équateur
380	Bulgarie	529	Chypre	627	Koweït	789 et 790	Brésil
383	Slovénie	530	Albanie	628	Arabie Saoudite	800 à 839	Italie
385	Croatie	531	Macédoine	629	Émirats Arabes Unis	840 à 849	Espagne
387	Boznie-Herzégovine	535	Malte	640 à 649	Finlande	850	Cuba
400 à 440	Allemagne	539	Irlande	690 à 695	Chine	858	Slovaquie
450 à 459							
490 à 499	Japon	540 à 549	Belgique & Luxembourg	700 à 709	Norvège	859	République tchèque
460 à 469	Russie	560	Portugal	729	Israël	860	Serbie Monténégro
470	Kirgistan	569	Islande	730 à 739	Suède	865	Mongolie
471	Taïwan	570 à 579	Danemark	740	Guatemala	867	Corée du Nord
474	Estonie	590	Pologne	741	Salvador	869	Turquie
475	Lettonie	594	Roumanie	742	Honduras	870 à 879	Pays-Bas
476	Azerbaïdjan	599	Hongrie	744	Costa Rica	880	Corée du Sud
477	Lituanie	600 et 601	Afrique du Sud	745	Panama	884	Cambodge
478	Ouzbékistan	603	Ghana	746	République Dominicaine	885	Thaïlande
479	Sri Lanka	608	Bahrein	750	Mexique	888	Singapour
480	Philippines	609	Île Maurice	754 à 755	Canada	890	Inde
481	Biélorussie	611	Maroc	759	Venezuela	893	Vietnam
482	Ukraine	613	Algérie	760 à 769	Suisse	899	Indonésie
484	Moldavie	616	Kenya	770	Colombie	900 à 919	Autriche
485	Arménie	618	Côte d'Ivoire	773	Uruguay	930 à 939	Australie
486	Géorgie	619	Tunisie	775	Pérou	940 à 949	Nouvelle-Zélande
487	Kazakhstan	621	Syrie	777	Bolivie	950	Madagascar
489	Hong Kong	622	Égypte	779	Argentine	955	Malaisie
500 à 509	Royaume Uni	624	Libye	780	Chili	958	Macao



6.5. Code barre 2D « flash code »



Le code **DataMatrix** est un code bidimensionnel à haute densité, permettant de représenter une quantité importante d'informations sur une surface réduite.

Il se compose d'un motif extérieur permettant le repérage des lignes et des colonnes et d'une matrice de données située à l'intérieur.

Grâce à l'écriture redondante des informations, il offre un niveau de sécurité maximal : lecture possible d'un symbole partiellement effacé jusqu'à environ 20% de la surface totale, au contraire d'un code unidimensionnel qui n'offre aucune sécurité si le symbole est dégradé.

La taille du DataMatrix dépend du nombre de caractères numériques à coder. Pour un message de 8 chiffres, la matrice intérieure sera constituée d'un carré de 10 lignes et 10 colonnes. La surface totale du DataMatrix pour un message de 8 chiffres sera alors de  $10\text{mm}^2$ , alors que le même message codé avec le code barre linéaire 39 occuperai plus de  $30\text{mm}^2$ .

2.1	2.2	3.6	3.7	3.8	4.3	4.4	4.5		
2.3	2.4	2.5	5.1	5.2	4.6	4.7	4.8		
2.6	2.7	2.8	5.3	5.4	5.5	1.1	1.2		
1.5	6.1	6.2	5.6	5.7	5.8	1.3	1.4		
1.8	6.3	6.4	6.5	8.1	8.2	1.6	1.7		
7.2	6.6	6.7	6.8	8.3	8.4	8.5	7.1		
7.4	7.5	3.1	3.2	8.6	8.7	8.8	7.3		
7.7	7.8	3.3	3.4	3.5	4.1	4.2	7.6		