 Lycée Victor Hugo	LM	**Numération et changement de base**	SI
	Cours		**1h**

Dans la vie de tous les jours, nous sommes entourés de **valeurs analogiques**. L'homme a vite choisi d'utiliser **des grandeurs numériques**, faciles à traiter et à transmettre. Selon l'usage que l'on souhaite en faire, un codage trouve une application plus appropriée par rapport à tel autre : par exemple, nous avons une base de 10 chiffres pour compter (0.. 9), une base 24 ou 60 pour nous repérer dans le temps (heures, minutes), 12 pour les mois.

I) Les systèmes de numération

Définition : un système de numération est un ensemble de règles permettant de représenter les nombres. Dans les systèmes numériques, les bases les plus utilisées sont la base 2, la base 10 et la base 16.

1) Le système de numération décimale (base 10)

Le système de numération **décimale** utilise les dix chiffres **0**, **1**, **2**, **3**, **4**, **5**, **6**, **7**, **8**, **9**.
On a l'habitude de compter l'argent, les quantités, les masses, les distances, etc., dans la base 10.
On pourrait décomposer par exemple $1342_{(10)} = 1000 + 300 + 40 + 2 = 1.10^3 + 3.10^2 + 4.10^1 + 2.10^0$
On peut généraliser pour toutes les bases $abcd_{(n)} = a.n^3 + b.n^2 + c.n^1 + d.n^0$

2) Le système de numération binaire (base 2)

Le système de numération **binaire** utilise exclusivement les deux chiffres **0** et **1**. (appelé **bit**)
On peut décomposer par exemple $1\,0110_{(2)} = 1.2^4 + 0.2^3 + 1.2^2 + 1.2^1 + 0.2^0$

3) Le système de numération hexadécimale (base 16)

Le système de numération **hexadécimale** utilise les chiffres de 0 à 9 suivi par les 6 lettres de A à F :

0, **1**, **2**, **3**, **4**, **5**, **6**, **7**, **8**, **9**, **A**, **B**, **C**, **D**, **E**, **F**.

$A2D_{(16)}$ (autres notations : **0x**A2D ou A2D**$**)

II) Correspondance entre les bases usuelles

Décimal	Binaire	Hexadécimal
0	0000 0000	0
1	0000 0001	1
2	0000 0010	2
3	0000 0011	3
4	0000 0100	4
5	0000 0101	5
6	0000 0110	6
7	0000 0111	7
8	0000 1000	8
9	0000 1001	9
10	0000 1010	A
11	0000 1011	B
12	0000 1100	C
13	0000 1101	D
14	0000 1110	E
15	0000 1111	F
16	0001 0000	10

III Les changements de base

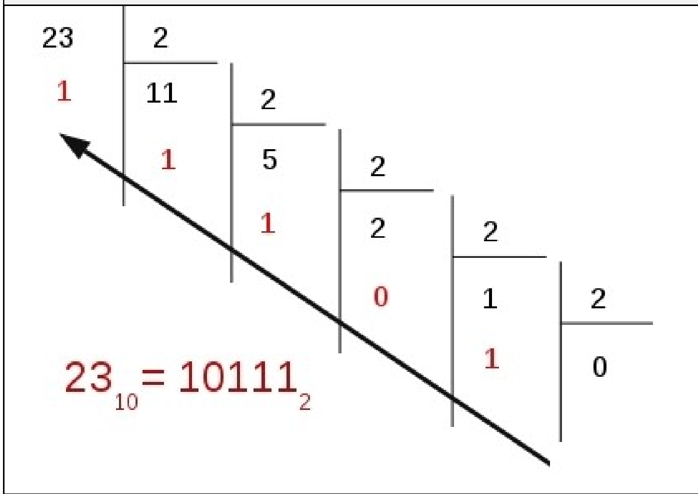
De nombreuses calculatrices proposent le changement de base (celle Windows en mode programmeur).

1) **De la base 2 à la base 10**

   Il suffit d'appliquer la décomposition de base

   $1\ 0110_{(2)} = 1.2^4 + 0.2^3 + 1.2^2 + 1.2^1 + 0.2^0 = 16 + 0 + 4 + 2 + 0 = 22_{(10)}$

2) **De la base 10 à la base 2**

<table>
<tr><th>Par divisions successives</th><th>Par soustractions successives</th></tr>
<tr><td>


</td><td>

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

120 – 64 = 56
56 – 32 = 24
24 – 16 = 8
8 – 8 = 0

$120_{(10)} = 0111\ 1000_{(2)}$
</td></tr>
</table>

3) **De la base 2 à la base 16**

La technique consiste à former des groupes de 4 bits et de convertir directement à l'aide du tableau de correspondance.

$101011_{(2)} = 0010\ \ 1011 = 2B_{(16)}$

Décimal	Binaire	Hexa
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

4) **De la base 16 à la base 2**

La technique consiste à remplacer les caractères en hexadécimal par leur équivalent binaire.

$EDF_{(16)} = 1110\ 1101\ 1111_{(2)}$

5) **De la base 16 à la base 10**

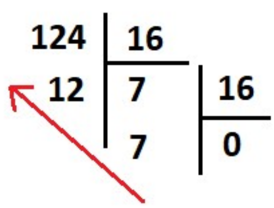
On applique la décomposition de base

$F8A_{(16)} = 15.16^2 + 8.16^1 + 10.16^0 = 3840 + 128 + 10 = 2978_{(10)}$

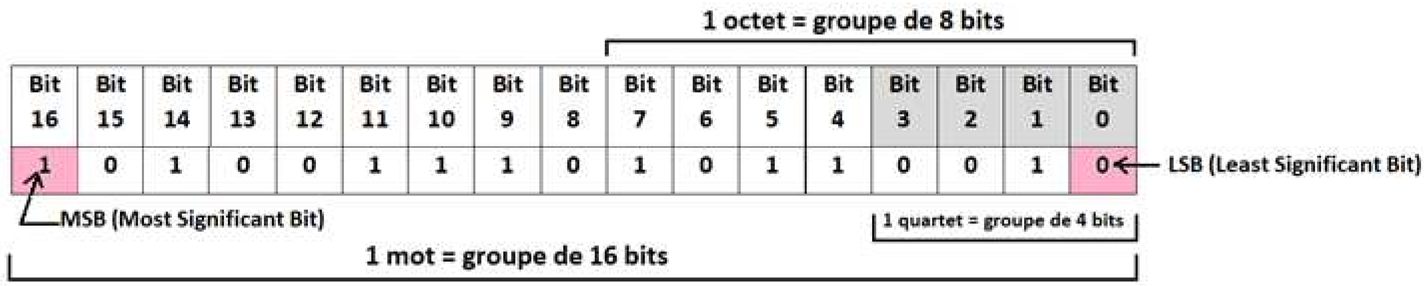
6) **De la base 10 à la base 16**

On applique la méthode de la division successive :

$124_{(10)} = 7C_{(16)}$



IV les termes utiles en binaire



- **Le bit** est la plus petite information manipulable par un microprocesseur (Contraction de Binary dig**it)**.
- **Le quartet** est un groupement de 4 bits
- **L'octet** (en anglais *byte*) est un groupement de 8 bits.
- **Le MSB** est le premier bit en partant de la gauche, il est le bit de poids le plus fort.
- **Le LSB** est le premier bit en partant de la droite, c'est le bit de poids le plus faible.

Codage sur	Nom français	Nom anglais	Intervalle de valeurs	Nombre de valeurs possibles
1 bit	**Bit**	Bit	0 à 1	$2\ valeurs\ (2^1)$
4 bits	Quartet	Quartet	0 à 15	$16\ valeurs\ (2^4)$
8 bits	Octet	**Byte**	0 à 255	$256\ valeurs\ (2^8)$
16 bits	Mot	Word	0 à 65 535	$65\ 536\ valeurs\ (2^{16})$
32 bits	Double mot	Double Word	0 à 4 294 967 295	$4\ 294\ 967\ 296\ valeurs\ (2^{32})$
n bits	-	-	0 à 2^n-1	$2^n\ valeurs$

Les préfixes

Décimaux

Nom	Symbole	Valeur
kilooctet	ko	10^3
mégaoctet	Mo	10^6
gigaoctet	Go	10^9
téraoctet	To	10^{12}
pétaoctet	Po	10^{15}
exaoctet	Eo	10^{18}
zettaoctet	Zo	10^{21}
yottaoctet	Yo	10^{24}

Les préfixes décimaux sont des puissances de 10.

Les préfixes binaires sont des puissances de 2.

Il est important de bien différencier les 2.

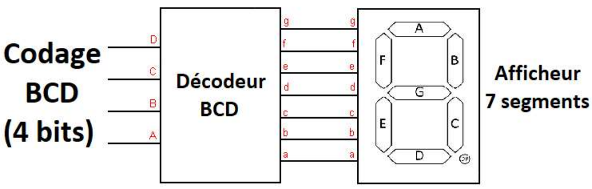
Binaires

Nom	Symbole	Valeur
kibioctet	Kio	2^{10}
mébioctet	Mio	2^{20}
gibioctet	Gio	2^{30}
tébioctet	Tio	2^{40}
pébioctet	Pio	2^{50}
exbioctet	Eio	2^{60}
zébioctet	Zio	2^{70}
yobioctet	Yio	2^{80}

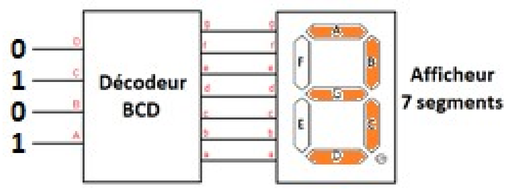
V) Le codage numérique

1) Le code BCD (Binary Coded Decimal)

Ce codage est utilisé principalement pour les **afficheurs 7 segments**.
Un nombre que l'on veut afficher (397 par exemple) est décomposé en chiffres (3, 9 et 7).
Chaque chiffre est ensuite codé sur 4 bits (un quartet).
Chaque quartet est exploité par un décodeur BCD.



Principe du codage BCD			
Nombre	397		
Décomposition en chiffres	3	9	7
Codage sur 4 bits	0101	1001	0111



2 Le code ASCII

Le codage ASCII (American Standard Code for Information Interchange) ou code américain normalisé pour l'échange d'informations, inventé par Bob BERNER en 1961, est la norme de codage de caractères en informatique la plus connue, la plus ancienne et la plus largement compatible.
Le code ASCII est un code sur 7 bits (valeurs 0 à 127), il permet de définir :
- des caractères imprimables universels : lettres minuscules et majuscules, chiffres, symboles,…
- des codes de contrôle non imprimables : indicateur de saut de ligne, de fin de texte, codes de contrôle, …

LSB	MSB	0 000	1 001	2 010	3 011	4 100	5 101	6 110	7 111
0	0000	NUL	DLE	SP	0	@	P	`	p
1	0001	SOH	DC1	!	1	A	Q	a	q
2	0010	STX	DC2	"	2	B	R	b	r
3	0011	ETX	DC3	#	3	C	S	c	s
4	0100	EOT	DC4	$	4	D	T	d	t
5	0101	ENQ	NAK	%	5	E	U	e	u
6	0110	ACK	SYN	&	6	F	V	f	v
7	0111	BEL	ETB	'	7	G	W	g	w
8	1000	BS	CAN	(	8	H	X	h	x
9	1001	HT	EM	)	9	I	Y	i	y
A	1010	LF	SUB	*	:	J	Z	j	z
B	1011	VT	ESC	+	;	K	[	k	}
C	1100	FF	FS	,	<	L	\	l	|
D	1101	CR	GS	-	=	M	]	m	{
E	1110	SO	RS	.	>	N	^	n	~
F	1111	SI	US	/	?	O	_	o	DEL

Par exemple si l'on veut écrire un A,
Dans le tableau on relève
MSB = 100 et FSB = 0001
Ce qui donne $100\ 0001_{(2)} = 65_{(10)}$
Ouvrez le bloc note et en maintenant la touche ALT enfoncée, tapez 65 et vous allez voir apparaître A.

3) Le code Gray

C'est un codage qui a la particularité de ne changer qu'un bit lors d'un passage d'une valeur à une autre.
il y a un risque de passage transitoire lorsque plusieurs bits changent en même temps, ce que le code Gray évite. Ce qui permet de fiabiliser le système.
Ce code est surtout utilisé pour des capteurs de positions absolue (codeur angulaire pour antenne radar).

Décimal	Code binaire	Code Gray
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000