Table des matières

Représentation d'un entier positif

1 Le système décimal :

Dame nature nous ayant affublés de deux mains comportant chacune cinq doigts, nous avons fini par nous mettre à compter en **base 10** comme le disent les mathématiciens : c'est le système décimal. Concrètement, cela signifie que l'on se sert de 10 symboles pour écrire des nombres : ce sont les 10 chiffres que l'on a appris dès notre plus jeune âge : 0 1 2 3 4 5 6 7 8 9.

▷ *Alors que l'on sait bien faire la différence entre lettre et mot, il y a souvent une confusion entre chiffre et nombre. Il ne faut pas la faire ! Les chiffres permettent d'écrire un nombre, au même titre que les lettres permettent d'écrire un mot.*

Pour écrire un nombre plus grand que 9, **on attribue aux chiffres un poids plus ou moins grand suivant la position occupée par le chiffre à l'intérieur du nombre** en question :

Position :	**Milliers**	**Centaines**	**Dizaines**	**Unités**	**Valeur :**
Poids :	10^3	10^2	10^1	10^0	
soit :	1000	100	10	1	
Exemple 1		1	2	5	0x1000 + 1x100 + 2x10 + 5x1=125
Exemple 2	1	2	5	0	1x1000 + 2x100 + 5x10 + 0x1=1250
Exemple 3	0	0	7	2	
Exemple 4		9	9	9	

→ Compléter les exemples 3 et 4.

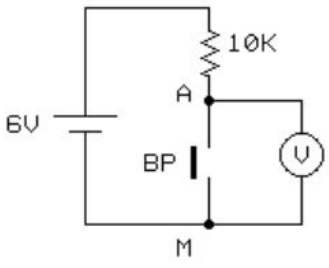
2 Le système binaire

2.1 Fonctionnement binaire

Un système binaire ne peut exister que dans l'une des deux configurations qui lui sont possibles :

- Vraie / Fausse pour une proposition
- Allumée / Eteinte pour une lampe
- Bloqué / Passant pour un transistor de commutation
- … etc

Le montage ci-dessous est constitué d'une alimentation continue réglable de 0 à 15V, d'une résistance de valeur 10kΩ, et d'un bouton poussoir. Un multimètre est utilisé en mode voltmètre pour mesurer la tension entre les points A et M :
Schéma théorique :



Le fonctionnement d'un bouton poussoir est **binaire**. Cela signifie qu'il n'y a que deux états pour lui :

- soit il est dans l'état « appuyé » (et il assure le contact électrique entre ses deux bornes)
- soit il est dans l'état « relâché » (et ce contact est rompu)

Voici ce que donne la mesure de la tension U_{AM} présente entre les deux bornes du bouton poussoir dans différents cas :

	Alimentation réglée à 6,0 V	
BP	**U_{AM}(V)**	**Niveau logique en A**
Appuyé	**0,0**	**« bas » ou « 0 »**
Relâché	**6,0**	**« haut » ou « 1 »**

	Alimentation réglée à 12,0 V	
BP	**U_{AM}(V)**	**Niveau logique en A**
Appuyé	**0,0**	**« bas » ou « 0 »**
Relâché	**12,0**	**« haut » ou « 1 »**

En électronique numérique, les circuits utilisés n'acceptent que deux niveaux de tension : ils travaillent donc en binaire :

- le niveau haut lorsque la tension est voisine de la tension d'alimentation : il est généralement représenté par le chiffre 1
- le niveau bas lorsque la tension est voisine de 0 volt : ce niveau est généralement représenté par le chiffre 0

2.2 Codage en binaire :

Les circuits logiques, (avec leurs deux niveaux : haut et bas) ne comptent que sur deux doigts, donc en **base deux** ("en binaire"). Dans cette façon de compter, on ne dispose que de deux symboles (deux chiffres) pour écrire un nombre : **les chiffres 0 et 1**.

Voici un nombre écrit en binaire : 1 1 0 1 0 0 1 0

Pour en comprendre la signification, il faut utiliser la même technique de pondération des chiffres en fonction de la position vue avec le système décimal, cela donne :

Position :	**b7**	**b6**	**b5**	**b4**	**b3**	**b2**	**b1**	**b0**	**Valeur en décimal :**
Poids :	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	
soit :	128	64	32	16	8	4	2	1	
Exemple 1	0	1	1	1	1	1	0	1	0x128+1x64+1x32+1x16+1x8+1x4+0x2+1x1=125
Exemple 2	1	1	1	1	1	0	1	0	
Exemple 3	1	1	0	1	0	0	1	0	
Exemple 4									
Exemple 5									
Exemple 6									170
Exemple 7									85

b0 à b7 sont appelés des *bits* (on dit « un » bit). le symbole du bit est la lettre « b »

Bit : Binary digit = Chiffre binaire

L'association de 8 éléments binaires (comme ici b0 à b7) constitue **un octet**

b0 est appelé bit de poids faible (LSB) alors que b7 est le bit de poids fort (MSB).

LSB : Least Significant Bit = bit de poids (le plus) faible
MSB : Most Significant Bit = bit de poids (le plus) fort

→ Déterminer la valeur en décimal de l'octet représenté dans les exemples 2 et 3

→ Représenter dans l'exemple 4 l'octet ayant la valeur la plus petite et donner cette valeur en décimal.

→ Représenter dans l'exemple 5 l'octet ayant la valeur la plus grande et donner cette valeur en décimal.

→ Combien de valeurs différentes peut-on donc représenter avec un octet ?

→ Faire l'exemple 6, puis l'exemple 7.

 BYTE = terme anglais qui sert à définir l'unité de base de stockage d'une donnée informatique. Les premiers microprocesseurs fonctionnaient avec 4 bits, le Byte valait donc 4 bits.
Assez rapidement une « normalisation » a eu lieu avec l'arrivée de microprocesseurs 8 bits.
Depuis le Byte représente donc 8 bits… et donc 1 Byte = 1 octet.
Le symbole du Byte est la lettre « B »
L'octet est devenu l'unité de base de stockage des données informatiques.

2.3 Convention d'écriture ?

On aura compris au travers de ces différents exemples que des activités liées au numérique peuvent nécessiter parfois de travailler en binaire (ou même en hexadécimal qui est en fait une forme d'écriture plus compacte, mais pas au programme de TS) plutôt que dans notre système décimal usuel. Cela pourrait être source de confusion : par exemple, que signifie l'écriture du nombre « 1000 » ?

- en décimal : mille
- en binaire l'écriture 1000 a comme équivalent décimal le nombre 8
- en hexadécimal, l'écriture 1000 a comme équivalent décimal le nombre 4096

Pour éviter les risques de confusions, on précisera la base dans laquelle le nombre est écrit. Malheureusement il n'y a pas de convention unique… la précision de la base se faisant par un préfixe ou par un suffixe, et pas toujours le même !!!

Quelques exemples… on a choisi des exemples que l'on peut rencontrer assez souvent :

- dans un texte on peut rencontrer un suffixe numérique en indice (car c'est facile à faire avec un traitement de texte)
- les calculettes Casio utilisent une lettre en préfixe
- en programmation, rien d'unifié non plus, cela dépend du langage.

	Traitement de texte :	**Casio :**	**Python, C :**	**HTML (codage en hexadécimal des couleurs) :**
binaire	1000_2	**b**1000	**0b**1000	
décimal	1000_{10}	**d**1000	1000	
hexadécimal	1000_{16}	**h**1000	**0x**1000	#001000

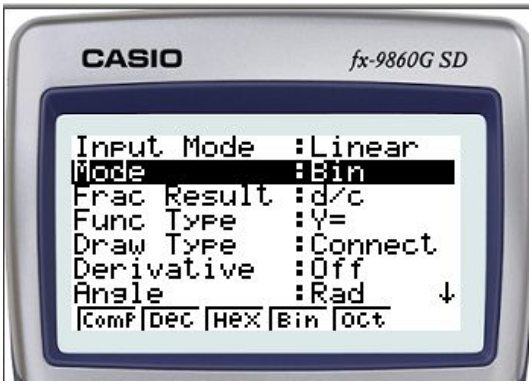
2.4 Outils de conversion :

2.4.1. Avec une calculette graphique :

Il est possible de faire une conversion d'une base dans une autre ou même des calculs dans une autre base que la base 10 avec les calculettes graphiques. Les fonctions de conversion peuvent être

installées de base dans la calculette ou devront y être implantées (par programmation ou en cherchant sur le net)

Pour la calculette Casio :

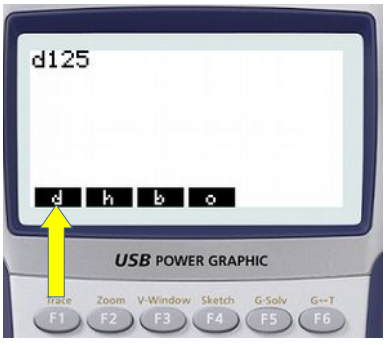


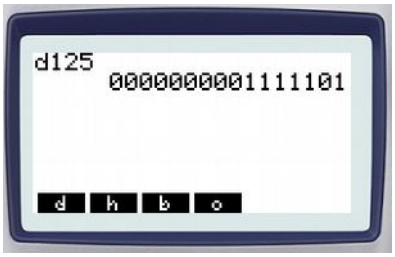
- aller dans le menu de réglage (« Set up ») et se déplacer sur la ligne « Mode ».

- Choisir alors le système dans lequel on veut obtenir le résultat des opérations à venir. Par exemple on a choisi ici le mode binaire en appuyant sur la touche F4 puis on valide avec la touche EXE.

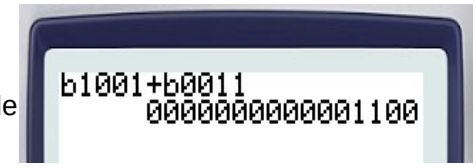
Appuyer sur la touche F1 (menu « d~o » pour dérouler le menu contenant les 4 options : d pour décimal, h pour hexadécimal, b pour binaire et enfin o pour octal) puis sélectionner le mode dans lequel on va entrer le nombre:

Par exemple pour convertir le nombre 125_{10} en binaire : appuyer sur F1 pour sélectionner une entrée en décimal puis entrer au clavier le nombre 125 et appuyer sur la touche EXE pour avoir la conversion :





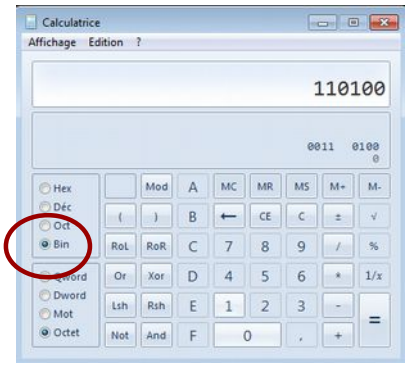
On peut également faire des opérations. Par exemple pour faire $1001_2 + 0011_2$, entrer ces deux nombres binaires précédés de la lettre b (touche F3) sans oublier le symbole de l'opération à réaliser entre les deux, puis exécuter le calcul (touche EXE) :

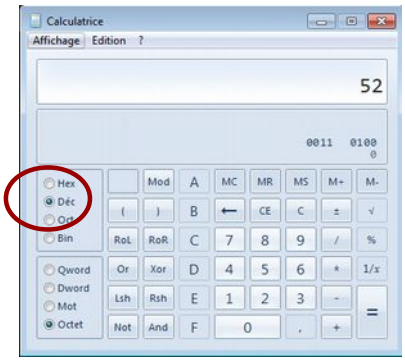


2.4.2. Avec une calculette système :

Les systèmes d'exploitation fournissent une calculatrice disposant de plusieurs niveaux d'utilisation tels que : basique, scientifique, financier et programmeur. C'est dans ce dernier mode que l'on pourra faire des conversions entre les différentes bases (2, 10 et 16, mais aussi 8 pour le système « octal »).

Ci-dessous un exemple avec la calculette Windows pour convertir le nombre binaire 00110100 :





On a ici commencé par mettre la calculette en mode Binaire (« Bin »). Remarquer qu'alors seuls les chiffres 0 et 1 sont disponibles sur le clavier.

Après avoir entré le nombre bianire à convertir, il suffit de sélectionner soit l'option « Dec » soit l'option « Hex » pour avoir la conversion en décimal ou en hexadécimal. Remarquer également les chiffres disponibles sur le clavier dans ces autres systèmes : 0 à 9 en décimal et 0 à F en hexadécimal.

On pourra s'entraîner à manipuler cette calculatrice en réutilisant les différents exemples proposés

3 Entiers de grande taille

Avec un octet (8 bits) on peut représenter des entiers positifs allant de __ à ____ soit _____ valeurs possibles. Ce nombre de valeurs possibles peut se calculer par l'opération :

$$\mathbf{Nbre\_de\_valeurs = 2^{nbre\_de\_bits}}$$

Exemple en 8 bits : Nbre_de_valeurs = 2^8 = 256

Pour représenter des nombres plus grands, il suffit d'utiliser davantage de bits. Comme on va le voir dans l'exemple ci-dessous, il sera plus simple d'avoir des « mots » dont la largeur est multiple de 8. Par exemple 16, ou 24, ou 32, ou 64 bits.

3.1 Cas d'un entier sur 16 bits

Prenons le cas d'un entier positif codé sur 16 bits. Compléter le tableau suivant, et donner la valeur en décimal puis en hexadécimal du mot de 16 bits donné en ligne 4 :

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
								2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
								128	64	32	16	8	4	2	1
1	**1**	**0**	**0**	**1**	**1**	**0**	**0**	**0**	**1**	**1**	**0**	**1**	**1**	**0**	**1**
= **en décimal**															

3.2 Généralisation :

Compléter le tableau suivant :

mot	Nbre d'octets	Nbre de valeurs	$(\text{Valeur mini})_{10}$	$(\text{Valeur maxi})_{10}$
8 bits	1	$2^8 = 256$	0	$2^8 - 1 = 255$
16 bits			0	
24 bits			0	
32 bits			0	
64 bits			0	