

But du TP :- Comprendre le principe de codage des informations .
- Savoir passer d'un système de numérotation à un autre.
- Construire un tableau de conversion avec Excel.

Lire le cours suivant puis compléter le.

I°) Le codage des informations :

1°) Qu'est-ce qu'une information :

Un Microprocesseur manipule des codes binaires 0 ou 1 appelés bits (binary digit).

Chaque bit prend donc la valeur 0 ou 1. En électronique les 0 et les 1 correspondent à des niveaux électriques :

- La valeur 0 quand le courant ne passe pas (niveau bas).
- La valeur 1 quand le courant passe (niveau haut).

Chaque information (lettres majuscules ou minuscules, chiffres, caractères de ponctuation ...) est un ensemble de 0 ou 1.

Avec 1 fil on peut coder deux caractères (0 ;1)

Avec 2 fils on peut coder 4 caractères (00 ;01 ;10 ; 11) soit $2^2=4$ caractères

Avec 3 fils on peut coder 8 caractères (000,001, 010, 100, 011, 101,110,111) soit $2^3=8$ caractères

Le nombre de caractères codés est donc une puissance de 2

Les micro-ordinateurs utilisent 8 fils. Quel est le nombre de caractères que l'on peut coder ? $2^8=256$ caractères

On numérote ces caractères de 0 à 255 (0 inclus)

Chaque caractère sera constitué de 8 bits : c'est à dire d'un **octet**.

Dans ce cas un mot ou un nombre par exemple sera constitué de plusieurs **octets** qui transiteront d'un circuit intégré à un autre par une nappe de fils électriques appelés bus. Plus l'horloge de l'ordinateur à une fréquence élevée, plus le microprocesseur sera rapide dans le traitement des informations. Actuellement 64 bits (soient $64/8=8$ caractères) sont transmis simultanément dans le bus à chaque top de l'horloge.

De cette façon, il est possible de connaître le débit maximal du bus (ou taux de transfert maximal), c'est-à-dire la quantité de données qu'il peut transporter par unité de temps, en multipliant sa largeur par sa fréquence. Un bus d'une largeur de 16 bits, cadencé à une fréquence de 133 MHz possède donc un débit égal à : $16 * 133.10^6 = 2128.10^6 \text{ bit/s} = 266.10^6 \text{ o/s} = 266 \text{ Mo/s}$

Remarque : l'octet est l'unité de base pour calculer le contenu des supports informatiques (disquette, disque dur, Cédérom...)

1Ko (kilooctet) = 2^{10} octets soit 1024 octets arrondi à 1000.

1Mo (mégaoctets) = 2^{20} soit 1048576 arrondi à 1000 000.

1 Go (gigaoctets) = 2^{30} octets soit 1 073 741 824 octets arrondi à 1 000 000 000.

2°) Le code A.S.C.I.I (Américan standard code international interchange) :

Ce code est celui qui est utilisé pour transformer les signaux électriques reçus du clavier en caractères affichables à l'écran ou à l'imprimante. Cette transformation dépend de la police utilisé.

Pour un octet, il existe 256 possibilités de remplissage à l'aide de 0 ou de 1 allant de 0 0 0 0 0 0 0 0 à 1 1 1 1 1 1 1 1

exemple : Lorsque vous tapez un **a** sur votre clavier, ce dernier renvoie à l'ordinateur le mot suivant 0 1 1 0 0 0 0 1 .

Pour arriver à ce résultat le caractère **a** est reconnu dans le code ASCII par le chiffre 97 et ce dernier par le code 0 1 1 0 0 0 0 1 dans le monde binaire.

De même, pour imprimer la lettre **a** (dans la police ARIAL, Times New Roman... mais pas Symbole) le microprocesseur va placer le code ASCII 97 (avec la police correspondante) sur le port de l'imprimante.

3°) Comment transite l'information entre l'ordinateur et le monde extérieur ?

Il existe de nombreuses possibilités pour qu'une information émise par le microprocesseur parvienne à un périphérique.

Citons deux types de liaisons :

- une liaison série, l'information transite par un seul fil (plus la masse). Les bits sont alors envoyés un par un.
- La liaison parallèle, l'information transite sur 8 fils (plus la masse). Les 8 bits de l'octet sont envoyés en même temps.

4°) Les systèmes de numérotation utilisés en informatique :

1) Les bases utilisées en informatique :

Il existe différentes manières de compter les objets :

binaire : 0 1

décimal : 0 1 2 3 4 5 6 7 8 9

hexadécimal : 0 1 2 3 4 5 6 7 8 9 A B C D E F

A) Base décimale (base 10).

Exemple : 138. Chaque chiffre n'a pas le même **poids suivant sa position**. 8 représente les unités, 3 les dizaines et 1 les centaines.

Unité : $1 = 10^0$ dizaine : $10 = 10^1$ centaine : $100 = 10^2$ etc..

$138 = 1 \times 10^2 + 3 \times 10^1 + 8 \times 10^0 = 1 \times 100 + 3 \times 10 + 8 \times 1$.

B) Binaire (base 2).

Exemple : 10001010. Comme en base 10, le poids d'un chiffre dépend de sa position.

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	0	0	0	1	0	1	0

En décimal, 10001010 s'écrit : $1 \times 2^7 + 0 \times 2^6 + 0 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = 128 + 8 + 2 = 138$

C) Base hexadécimale :

Binaire : 0 1

Décimal : 0 1 2 3 4 5 6 7 8 9

Hexadécimal : 0 1 2 3 4 5 6 7 8 9 A B C D E F

Pour indiquer que le nombre est en hexadécimale, il est précédé de \$.

Exemple : $\$89 = 8 \times 16^1 + 9 \times 16^0 = 8 \times 16 + 9 \times 1 = 137$

2) Avantages et inconvénients des différentes bases (changements de base) :

Exemple : Traduction d'un nombre exprimé dans une base quelconque en base 10.

Se souvenir du rôle des chiffres dans le nombre, ainsi 1987 en base 10 représente :

$7 \times 10^0 + 8 \times 10^1 + 9 \times 10^2 + 1 \times 10^3$

3C1F en hexadécimal vaut $15 \times 16^0 + 1 \times 16^1 + 12 \times 16^2 + 3 \times 16^3$ soit $15 + 16 + 3072 + 12288 = 15391$

1011001 vaut $1 \times 2^0 + 0 \times 2^1 + 0 \times 2^2 + 1 \times 2^3 + 1 \times 2^4 + 0 \times 2^5 + 1 \times 2^6$ soit $1 + 8 + 16 + 64 = 89$

La base 10 est utilisée habituellement.

La base 2 est proche du langage machine mais il faut écrire beaucoup de chiffres.

La base 16 (hexadécimale) permet de coder un octet sur deux symboles.

A) Conversion d'un nombre décimal en binaire :

Exemple : 115

$$115 = 64 + 32 + 16 + 2 + 1$$

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

$$115 = 01110011$$

On divise par la base et on réécrit les restes dans l'ordre inverse ainsi le codage de 138 en base 2

donne

138 | 2

0 | 69 | 2

1 | 34 | 2

0 | 17 | 2

1 | 8 | 2

0 | 4 | 2

0 | 2 | 2

0 | 1

Soit 10001010

Autre méthode possible

On soustrait au nombre de départ les puissances de 2 possibles de la plus grande à la plus petite

Ainsi $138 = 128 + 8 + 2$

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	0	0	0	1	0	1	0

B) Conversion décimale-hexadécimale :

Exemples : 115 : $115 = 7 \times 16 + 3 = \73
 : 251 : $251 = 15 \times 16 + 11 = \FB

C) Conversion binaire-hexadécimale.

Exemple : 11001110 en binaire = ? en hexadécimale.

2^3	2^2	2^1	2^0	2^3	2^2	2^1	2^0
8	4	2	1	8	4	2	1
1	1	0	0	1	1	1	0

Il faut séparer les 8 bits en deux paquets de 4.

Paquet de droite : $2 + 4 + 8 = 14 = E$

Paquet de gauche : $4 + 8 = 12 = C$

11001110 en binaire = CE en hexadécimal

Exercices :

- Les premiers ordinateurs codaient leur information sur 6 bits, combien de signes différents pouvaient-ils identifier ? $2^6=64$ signes différents (0, 1, 2, ..., 9, A, B ... a, b ... soit les 10 chiffres + 27*2 lettres en majuscule+ minuscule)
- Ecrire les nombres suivants dans le système demandé :
 10011001 et 00110110 (donné en binaire) dans la base 10
 $10011001 = 1 \times 2^0 + 0 \times 2^1 + 0 \times 2^2 + 1 \times 2^3 + 1 \times 2^4 + 0 \times 2^5 + 0 \times 2^6 + 1 \times 2^7 = 153$
 $00110110 = 0 \times 2^0 + 1 \times 2^1 + 1 \times 2^2 + 0 \times 2^3 + 1 \times 2^4 + 1 \times 2^5 + 0 \times 2^6 + 0 \times 2^7 + 0 \times 2^8 = 54$
 129, 258 (donné en base 10) dans la base 2 ou binaire.
 $129 = 128 + 1 = 0 \times 2^0 + 0 \times 2^1 + 0 \times 2^2 + 0 \times 2^3 + 0 \times 2^4 + 0 \times 2^5 + 0 \times 2^6 + 1 \times 2^7$ soit 10000001 en binaire
 $258 = 256 + 2 = 0 \times 2^0 + 1 \times 2^1 + 0 \times 2^2 + 0 \times 2^3 + 0 \times 2^4 + 0 \times 2^5 + 0 \times 2^6 + 0 \times 2^7 + 1 \times 2^8$ soit 100000010 en binaire
 15F et 3B6 (donné en hexadécimal) dans la base 10.
 $15F = 15 \times 16^0 + 5 \times 16^1 + 1 \times 16^2 = 351$
 $3B6 = 6 \times 16^0 + 11 \times 16^1 + 3 \times 16^2 = 950$
- Transformer les quantités suivantes dans l'unité indiquée.
 5Mbits en Ko $(5 \times 10^6 / (8 \times 10^3)) \approx 625$ Ko en fait 610 Ko (1Ko=1024 bits)
 500 Ko = $500 \times 10^3 \times 8 \approx 4.10^3$ Kbits en fait 4096 Kbits (1Ko=1024 bits)
- Déterminer la quantité d'information contenue dans une page de texte de 80 par 66 caractères et ce en Ko. Définir le nombre de page que l'on peut écrire en stockant nos informations sur une disquette de 1,44 Mo puis sur une clef USB de 1Go.

80*66=5280 caractères ce qui correspond à 5280 octets soit environ 5,15 Ko pour une page.
On peut donc écrire $1048576 * 1.44 / (5,15.10^3)$ environ 293 pages sur une disquette de 1,44 Mo.
Sur une clef de 1Go on peut écrire $10^9 / (5,15.10^3)$ environ $1,9.10^5$ pages

5°) Un Minitel reçoit des informations par la ligne téléphonique qui se comporte comme une liaison série débitant à 110 bauds (110 bits/s). L'écran comporte 25 lignes de 40 colonnes. (Le codage de l'information se fait sur 8 bits comme pour les ordinateurs). Indiquer le temps qu'il faudrait pour charger une page remplie d'informations. Remarque.

$25 * 40 = 10^3$ caractères possibles soit 10^3 octet (soit 8 kbits)

Pour charger une page d'information (avec l'écran entièrement rempli de caractères) il fallait $8.10^3 / 110$ soit environ 72 s $\approx$ 1 min 12 s. Heureusement que le minitel n'est plus utilisé !!!! vive l'ADSL 2+ et bientôt la fibre optique à 100 Mbps.

6°) Un fournisseur d'accès propose l'ADSL2+ avec un débit de 15 Mbps (megabits par seconde). 1 Ko (kilooctet) représente 2^{10} octets. Combien de kilooctets sont transférés en 1 s ?

Une page de texte contient 6400 caractères. Chaque caractère est codé sur un octet. Calculer la durée nécessaire pour transférer cette page.

1Ko=1024 octet et un octet correspond à 8 bits donc 1 Ko vaut 8192 bits, ce qui fait une vitesse de transfert de $15.10^6 / 8192 \approx 1,8.10^3$ Ko soit 1,8 Mo par seconde (un peu plus d'une disquette transférée à la seconde).

$6400 * 8 = 5,12.10^4$ bits pour une page de texte soit un temps de transfert de $5,12.10^4 / (15.10^6) \approx 0.0034$ s (à peine 3/1000 de seconde)

Partie pratique :

1. Travail sur ordinateur.

En utilisant la fonction CAR() d'EXCEL, faire un tableau qui représente l'ensemble des 256 caractères de la police ARIAL.

1^{ère} colonne : nombres de 0 à 31. 2^{ème} colonne : symboles correspondant (avec la fonction CAR().

3^{ème} colonne : nombres de 32 à 63. 4^{ème} colonne : symboles correspondant (avec la fonction CAR().

etc... jusqu'à 255 (16 colonnes en tout).

Le travail doit tenir dans une page.

Titre : code ASCII, titre secondaire : Police ARIAL.

Même travail pour la police SYMBOL.

2. Changement de bases :

Ecrire 10011001 et 00110110 (donnés en binaire) en base 10.

Ecrire 129, 243, 12, 154 (donnés en décimal) en binaire (base 2).

3. Travail sur ordinateur :

Utiliser les fonctions DECIBIN () d' EXCEL pour construire un tableau de conversion des 256 premiers nombres décimaux en binaire.

1^{ère} colonne : nombres de 0 à 31 (en décimal). 2^{ème} colonne : nombre correspondant en binaire avec la fonction DECIBIN().

3^{ème} colonne : : nombres de 32 à 63 (en décimal). 4^{ème} colonne : nombre correspondant avec la fonction DECIBIN().

etc... jusqu'à 255 (16 colonnes en tout).

Titre : Conversion décimal-binaire (mise en page en format paysage pour faire une seule page.

4. Changement de bases :

Ecrire 129, 243, 12, 154 (donnés en décimal) en hexadécimal (base 16).

Ecrire \$FF, \$CD, \$88, \$7A (donnés en hexadécimal) en décimal.

Ecrire 01101110, 11111101, 10101010 (donnés en binaire) en hexadécimal.

5. Travail sur ordinateur :

Utiliser la fonction DECHEX () d'EXCEL pour construire un tableau de conversion des 256 premiers nombres décimaux en hexadécimal. Titre : Conversion décimal-hexadécimal.

6. Majuscule-minuscule :

Police ARIAL.

1^{ère} colonne : nombres décimaux correspondant aux 26 lettres majuscules.

2^{ième} colonne : nombres binaires correspondant aux 26 lettres majuscules (utiliser la fonction DECBIN).

3^{ième} colonne : lettres majuscules ; (utiliser la fonction CAR).

4^{ième} colonne : nombres décimaux correspondant aux 26 lettres minuscules.

5^{ième} colonne : nombres binaires correspondant aux 26 lettres minuscules (utiliser la fonction DECBIN).

6^{ième} colonne : lettres minuscules ; (utiliser la fonction CAR).