

LE BINAIRE ET LE CODAGE DES INFORMATIONS

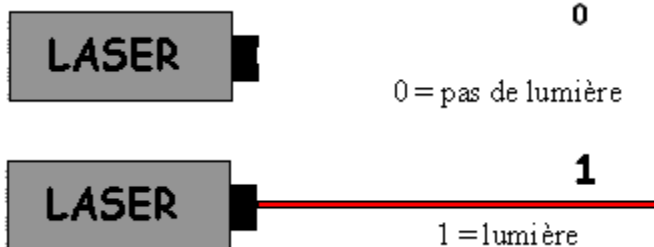
- Objectifs:**
- Connaître le système binaire, le bit et l'octet.
 - Comprendre le codage des informations en informatique

I LE SYSTEME BINAIRE

1) Le binaire

• L'informatique utilise des courants électriques, des aimantations, des rayons de lumière ... Chacun de ces phénomènes met en jeu **deux états possibles**:

- tension nulle ou tension non nulle (5V par ex),
- aimantation dans un sens ou dans l'autre sens,
- lumière ou pas de lumière.



• Il suffit donc de **deux chiffres** pour traduire ces deux états: c'est la **numération binaire** qui utilise les chiffres **0** et **1**. Un rayon de lumière peut parfaitement traduire ces deux valeurs:

- 0 = pas de lumière
- 1 = lumière.

- **Le système binaire** est un système de numération de position de **base deux**: les deux seuls chiffres qui le composent sont le "0" et le "1".
- **Le système binaire est le "langage" des ordinateurs**. Toutes les machines numériques utilisent le système binaire pour coder des informations (textes, sons, images, vidéos ...).

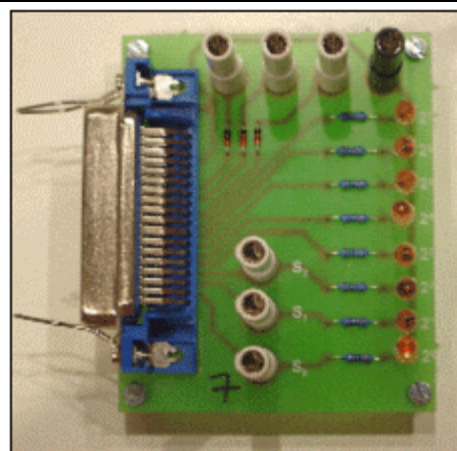
2) Découverte du langage binaire

• Le **module Entrée - Sortie 8 DEL**, ci-contre, va nous permettre de visualiser **en binaire** les nombres dans le système **décimal**.

A chaque **DEL** est associé un chiffre du système binaire:

- 0** si la DEL est **éteinte**
- 1** si la DEL est **allumée**

• La présence des **8 DEL** sur le module permet d'écrire des nombres binaires à 8 chiffres, de **0000 0000** à **1111 1111**.



• Relier le module au câble imprimante. Placer le module verticalement pour que les DELS soient situées en haut du module.


• Ouvrir le fichier Excel "**Module ES port imprimante.xls**" et cliquer sur l'onglet "Initiation au binaire": l'écran ci-contre apparaît alors.

• Pour visualiser en binaire un nombre décimal procéder de la façon suivante:


- entrer le nombre décimal puis ↵
- cliquer sur le bouton visualisation.

Initiation au langage binaire

| | | |
|-------------------------|-----------|---|
| Nombre décimal ≤ 255 | 56 | Visualisation en binaire du nombre décimal |
|-------------------------|-----------|---|



Module Entrée - Sortie à 8 DEL permettant de visualiser en binaire, le nombre décimal envoyée sur le port imprimante



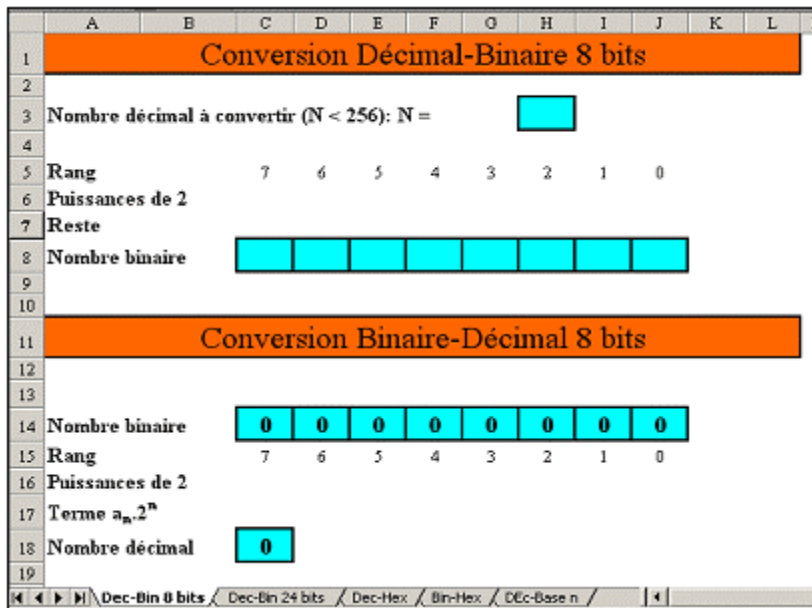
Visualisation en binaire d'un nombre décimal

- Les fonctions suivantes d'Excel permettent de réaliser ce type de décomposition:
 - **Quotient**(dividende; diviseur): donne **le quotient** de la division du **dividende** par le **diviseur**
 - **Mod**(dividende; diviseur): donne **le reste** de la division du **dividende** par le **diviseur**

Exemple: si dans une cellule on tape: = quotient(3240; 10) il s'affiche la valeur 324 car $3240 / 10 = 324$
 si dans une cellule on tape: = mod(3240; 10) il s'affiche la valeur 0 car le reste de la division $3240 / 10 = 324$ est égal à 0.

Remarque: pour que ces fonction soient actives dans Excel, il faut que les macros complémentaires:
 - Utilitaire d'analyse
 - Utilitaire d'analyse VBA
 soient sélectionnées.

- Charger le fichier "Conversion élève.xls". Cliquer sur l'onglet "Dec-Bin 8 bits". L'écran suivant apparaît:



- Dans la cellule **C6**, faire calculer la puissance: 2^7 en tapant = 2^{C5} (la cellule C5 a une valeur égale à 7). Tirer la cellule **C6** jusqu'à la cellule **J6**: cela copie et calcule toutes les autres puissances de 2.
- Le nombre décimal **N** à convertir est situé dans la cellule: **H3**.
 - aller en **C7**: faire calculer **le reste** de la division de **H3** par **C6**, en tapant = **mod(H3;C6)**
 - aller en **D7**: faire calculer **le reste** de la division de **C7** par **D6**. "Tirer" la cellule **D7** jusqu'à la cellule **J7**. Cela copie et calcule tous les autres restes.
- Aller en **C8**: faire calculer **le quotient** de la division de **H3** par **C6**, en tapant = **quotient(H3;C6)**
 Aller en **D8**: faire calculer **le quotient** de la division de **C7** par **D6**. "Tirer" la cellule **D8** jusqu'à la cellule **J8**. Cela copie et calcule tous les autres quotient.
- Vérifier la conversion avec le décimal **90** de l'exemple précédent. Vous venez de réaliser un **convertisseur décimal – binaire 8 bits**.

- Par quel chiffre se termine l'écriture binaire d'un nombre décimal pair ? Impair ?
- Si $N = 2^6$, quelle sera l'écriture décimale de N puis l'écriture binaire de ce nombre ?

4) Conversion Binaire - Décimal

Rappel: le nombre décimal N, en base binaire, s'écrit: $N = a_0.2^0 + a_1.2^1 + a_2.2^2 + \dots + a_n.2^n$.

N est donc la **somme de puissances de 2** affectées des termes a_n qui dans le système binaire valent **0** ou **1**.

- Remplir la ligne **16** du fichier Excel (Puissance de 2) en s'aidant de l'exemple précédent (ligne 6).
 - Aller en **C17**: faire calculer le terme $a_7.2^7$ en tapant **=C14*C16**. "Tirer" la cellule **C17** jusqu'à la cellule **J17**. Cela copie et calcule tous les autres termes $a_n.2^n$.

- Aller en **C18**: il faut ici faire la somme de tous les termes: $N = a_0.2^0 + a_1.2^1 + a_2.2^2 + a_3.2^3 + a_4.2^4 + a_5.2^5 + a_6.2^6 + a_7.2^7$. Taper alors: **=somme(C17 : J17)**.

- Vérifier la conversion binaire - décimal avec le nombre binaire **01011010** de l'exemple précédent.

- a) Quel est le nombre binaire le plus grand que l'on puisse écrire ? A quel décimal N_{max} correspond-il ?
- b) Écrire ce nombre sous la forme: $2^n - 1$ en déterminant la valeur de **n**.

5) Le bit d'information

- Le **bit** est l'abréviation de **binary digit**.
- Une **grandeur binaire** est codée sur **un bit**: elle ne peut prendre que deux états, "0" ou "1".
- Le **bit** est la **plus petite unité d'information** manipulable par un ordinateur. Plus le nombre de bits augmente plus le nombre d'états augmente.

a) Compléter le tableau suivant:

| Nombre de bits | Nombre d'états | États |
|----------------|----------------|-----------------|
| 1 | 2 | 0 1 |
| 2 | | 00 01 |
| 3 | | 000 001..... |
| 4 | | 0000 0001 |

- b) Quelle correspondance existe-t-il entre le **nombre d'états** et le **nombre de bits** ? Pour **n bits** quel est le **nombre d'états différents** ?
- c) Justifier les termes "**8 bits**" dans la conversion du paragraphe précédent.

6) L'octet et ses multiples

- En informatique, l'**unité d'information** composée de **8 bits** est appelée **octet**.

- a) Quel est le nombre d'états correspondant à un **octet** ? L'écrire sous la forme d'une **puissance de 2**.
- b) Un **kilo-octet** vaut **1 Ko = 2¹⁰ octets**. A quel décimal correspond **2¹⁰** ? A-t-on **1 Ko = 1000 octets** ?
- c) Un **méga-octet** vaut **1 Mo = 2²⁰ octets**. A quel décimal correspond **2²⁰** ? A-t-on **1 Mo = 1 000 000 octets** ?
- d) A combien d'octets correspond **un giga-octet (1 Go)** ?

- Pour évaluer les **capacités de stockage en informatique**, on utilise l'octet et ses multiples.
- e) Le premier ordinateur familial (ZX Sinclair) avait **1 Ko** de RAM ... Quelle est la RAM des ordinateurs aujourd'hui ?
- f) Quelle est l'ordre de grandeur de la capacité de stockage des disquettes 3 1/2, d'un CD-ROM, d'une clé USB, d'un disque dur ?

II LE CODAGE DES CARACTERES EN BINAIRE

• Le codage des caractères: notre alphabet contient **26** lettres de **a** à **z**. En différenciant les majuscules et minuscules, **52** caractères sont nécessaires, sans compter les lettres accentuées. Il faut, de plus, **10** chiffres de **0** à **9**. On doit compter les caractères des accents, des guillemets, de la ponctuation, les symboles mathématiques ... Mais un clavier, on l'a vu, comporte des commandes (flèches de curseur, tabulation, suppression, etc...). Il y a en tout plus de **100** éléments à coder.

1) Peut-on coder tous ces caractères sur **un seul octet** ? Pourquoi ?

• Un code a été créé, **le code ASCII** (American Standard Code for Informatic Information). A chaque valeur d'octet correspond un caractère ou une commande du clavier. En voici un extrait:

| | | | | | | | | | | | | | | | |
|----|----|----|---|----|---|----|---|-----|---|-----|---|-----|---|-----|---|
| 32 | | 48 | 0 | 64 | @ | 80 | P | 96 | ` | 112 | p | 128 | | 144 | |
| 33 | ! | 49 | 1 | 65 | A | 81 | Q | 97 | a | 113 | q | 129 | | 145 | |
| 34 | | 50 | 2 | 66 | B | 82 | R | 98 | b | 114 | r | 130 | é | 146 | |
| 35 | " | 51 | 3 | 67 | C | 83 | S | 99 | c | 115 | s | 131 | | 147 | |
| 36 | # | 52 | 4 | 68 | D | 84 | T | 100 | d | 116 | t | 132 | | 148 | |
| 37 | \$ | 53 | 5 | 69 | E | 85 | U | 101 | e | 117 | u | 133 | à | 149 | |
| 38 | % | 54 | 6 | 70 | F | 86 | V | 102 | f | 118 | v | 134 | | 150 | |
| 39 | & | 55 | 7 | 71 | G | 87 | W | 103 | g | 119 | w | 135 | ç | 151 | ù |
| 40 | ' | 56 | 8 | 72 | H | 88 | X | 104 | h | 120 | x | 136 | | 152 | |
| 41 | (| 57 | 9 | 73 | I | 89 | Y | 105 | i | 121 | y | 137 | | 153 | |
| 42 |) | 58 | : | 74 | J | 90 | Z | 106 | j | 122 | z | 138 | è | 154 | |
| 43 | * | 59 | ; | 75 | K | 91 | [| 107 | k | 123 | { | 139 | | 155 | |
| 44 | , | 60 | < | 76 | L | 92 | \ | 108 | l | 124 | | 140 | | 156 | £ |
| 45 | - | 61 | = | 77 | M | 93 |] | 109 | m | 125 | } | 141 | | 157 | |
| 46 | . | 62 | > | 78 | N | 94 | ^ | 110 | n | 126 | ~ | 142 | | | |
| 47 | / | 63 | ? | 79 | O | 95 | _ | 111 | o | 127 | Δ | 143 | | | |

Remarque: Le caractère 32 est "l'espace". Les autres cases vides correspondent à des caractères non représentés.

2) Traduire, en utilisant **le convertisseur binaire – décimal**, le monotone et mystérieux message suivant:

0100 0010 0101 0010 0100 0001 0101 0110 0100 1111 0010 0001.

Faire un tableau avec les nombres binaires, le nombre décimal correspondant et le code ASCII correspondant.

3) Ecrire votre **prénom** en binaire.

• Dans le Bloc-note de Windows (Programme → Accessoire → Bloc-notes) taper la lettre " a " et enregistrer le fichier sous le nom: " **1_caractère** "

Créer un fichier avec 10 caractères ASCII et enregistrer le fichier sous le nom: " **10_caractères**".

4) Quelle est la taille en octet de ces deux fichiers ? Sur combien d'octet(s) est codé un caractère ?

5) Combien de pages de **40** lignes, comportant chacune **80 caractères**, devrait-on pouvoir enregistrer sur une disquette de **1,44 Mo** ?

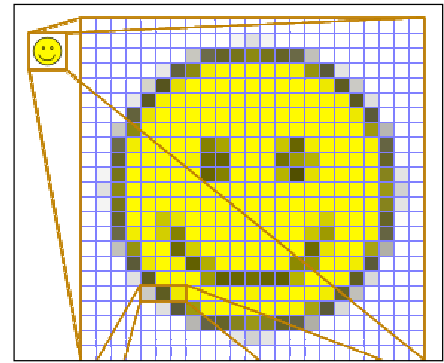
• Fermer le fichier Excel.

III. LE CODAGE DES IMAGES

1) Image bitmap

• Les **images bitmap** sont constituées d'un **ensemble de points** appelés **pixel**. Ce nom provient de la locution anglaise *picture element*, qui signifie "**élément d'image**" ou "**point élémentaire**".

• Chaque image est une succession de pixels (voir ci-contre). Les pixels sont codés sur un **certain nombre de bits**. Ces images ne sont **pas compressées**.



Pixels associés à l'image

• **Création d'une image bmp:**

Dans le bureau de Windows: Démarrer → Programmes → Accessoires → Paint

Dans le logiciel Paint: Image → Attribut

a) Relever la **largeur** et la **hauteur** du cadre image **en pixels** par défaut. (Vérifier que **pixel** et **couleur** sont choisis par défaut).

b) Quel est le nombre de pixels du cadre image ?

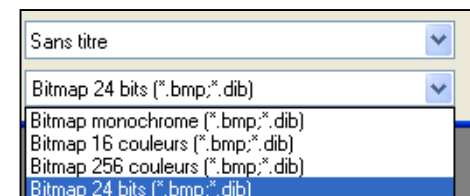
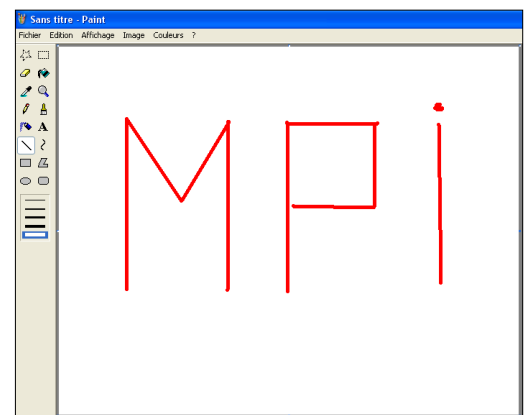
• Dans le cadre blanc vous pouvez dessiner avec les couleurs mises à disposition en bas de l'écran. Par exemple, écrire "MPI" avec la couleur de votre choix.

• Nous allons enregistrer cette image dans les différents formats donnés dans le cadre ci-contre.

Enregistrer une première fois l'image sous le nom: **img_24bits.bmp**.

Faire de même pour les trois autres formats d'image ci-contre en modifiant le nom selon le format.

c) Dans le poste de travail, retrouver les fichiers et déterminer leur taille en **Ko**. Puis compléter le tableau ci-dessous:



| Nom du fichier | Taille (en Ko) |
|--------------------|----------------|
| img_monochrome.bmp | |
| | |
| | |
| | |

2) Vérification de la taille des images par le calcul

• La **place mémoire** occupée par un **pixel** dépend du nombre de couleurs que l'on souhaite :

| Nombre de couleurs | Nombre de bits | Place mémoire en Ko |
|-----------------------------|----------------|---------------------|
| Noir et blanc: 2 couleurs | | |
| 16 couleurs | | |
| 256 couleurs | | |
| 63 536 couleurs | | |
| ≈ 16,7 millions de couleurs | | |

a) Sur combien **de bits** est codée une image où **chaque pixel** peut prendre:

2 couleurs 16 couleurs 256 couleurs 63 536 couleurs $\approx 16,7$ millions de couleur ?

Compléter alors le tableau précédent.

b) Rappeler le nombre de pixels des images enregistrées.

Sachant que: **1 octet = 8 bits** et que **1 Ko = 1024 octets**, calculer la place mémoire en **Ko** occupée par chaque image selon le nombre de couleurs utilisées et compléter le tableau précédent.

c) Comparer avec la **taille en Ko** de chaque image du 1). A-t-on concordance dans chaque cas ?

Remarque: la place mémoire réelle utilisée peut être en fait plus importante. Par exemple, en mode **16,7 millions de couleurs**, le pixel peut occuper **32 bits (4 octets)** au lieu de **24 bits (3 octets)**. L'octet supplémentaire est soit inutilisé soit utilisé pour une caractéristique supplémentaire de l'image: "transparence" ou "texture".

• Autres exemples suivant la définition de l'image:

| Définition de l'image en pixel | Noir et blanc (1 bit) | 256 couleurs (8 bits) | 65000 couleurs (16 bits) | True color (24 bits) |
|--------------------------------|-----------------------|-----------------------|--------------------------|----------------------|
| 320 × 200 | 7.8 Ko | 62.5 Ko | 125 Ko | 187.5 Ko |
| 640 × 480 | 37.5 Ko | 300 Ko | 600 Ko | 900 Ko |
| 800 × 600 | 58.6 Ko | 468.7 Ko | 937.5 Ko | 1.4 Mo |
| 1024 × 768 | 96 Ko | 768 Ko | 1.5 Mo | 2.3 Mo |

Cela met en évidence la quantité de mémoire vidéo que nécessite la carte graphique en fonction de la définition de l'écran (nombre de pixel affichés) et du nombre de couleurs. L'exemple montre ainsi qu'il faut une carte ayant au minimum 4 Mo de mémoire vidéo afin d'afficher une résolution de 1024 × 768 en true color...