

3A - Microcontrôleur

TD1 - Codage et représentation des données.

Nicolas Delanoue

(1 séance de 2h40)

L'infographie est le domaine de l'informatique concernant la création et la manipulation des images numériques. Le processeur envoie les informations définissant l'image, de la mémoire vive (RAM) vers une carte graphique qui convertit les données pour les afficher sur l'écran.

On distingue 2 grandes catégories de codage d'images :

- Le codage vectoriel : l'image est codée par une chaîne de caractères qui décrivent les différentes primitives géométriques (segments, cercles, ...) présentes. Ce genre de fichier pour avoir l'extension svg, ps ...
- Le codage Bitmap ou matriciel : l'image est codée sous forme d'un tableau de valeurs qui décrivent les couleurs de chaque pixel (fichier pbm, bmp, jpeg ...).

Exercice 1 (Image vectorielle)

1. En ouvrant le fichier, nommé `exemple.svg`¹, avec un éditeur de texte, on voit apparaître le code suivant

```
<svg version="1.1"
  width="300" height="200"
  xmlns="http://www.w3.org/2000/svg">

  <rect width="100%" height="100%" fill="black" />
  <circle cx="150" cy="100" r="80" fill="gray" />
  <text x="150" y="125" font-size="60"
    text-anchor="middle" fill="white">SAGI</text>

</svg>
```

Quel serait le résultat si on décidait d'ouvrir ce fichier avec un navigateur web ? ou bien avec le logiciel Inkscape ?

2. Comment déplacer le cercle de 10 unités vers la droite ?
3. Pourquoi n'observe-t-on pas de "pixelisation" lors d'un zoom sur ce genre d'image ?

Exercice 2 (Une image matricielle)

Il existe plusieurs codages d'image matricielle. Pour les codages sans compression, le codage consiste à donner une liste de valeurs précisant la couleur pour de chaque pixel. Un des codages les plus simples est le format bmp (lire Bitmap pour « carte de bits »).

Dans cet exercice, nous allons nous focaliser sur le format pbm. Un fichier écrit avec ce codage respecte, dans cet ordre, le modèle suivant :

1. <https://perso-laris.univ-angers.fr/~delanoue/polytech/microcontroleur/exemple.svg>

- un chaîne de caractères dite « magique » précisant le format du reste du fichier : il indique le type de format (PBM, PGM, ou PPM) et la variante (binaire ou ASCII) ;
 - un caractère d'espace (espace, tabulation, nouvelle ligne) ;
 - la largeur de l'image (nombre de pixels, écrit explicitement sous forme d'un nombre en caractères ASCII) ;
 - un caractère d'espace ;
 - la hauteur de l'image (idem) ;
 - un caractère d'espace ;
 - les données de l'image : succession des valeurs associées à chaque pixel, l'image est codée ligne par ligne en partant du haut, chaque ligne est codée de gauche à droite. Pour un fichier pbm, un « 0 » correspond à un pixel blanc alors qu'une « 1 » code un pixel noir.
1. En ouvrant le fichier, nommé `exemple5.pbm`², avec un éditeur de texte, on voit apparaître le code suivant

```
P1
# Quelle image est codee par ce code pbm ?
24 7
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 1 1 1 1 0 0 0 0 1 0 0 0 1 1 1 1 0 0 1 0 0 0 0
0 1 0 0 0 0 0 0 1 0 1 0 0 1 0 0 0 0 0 1 0 0 0 0
0 1 1 1 1 0 0 1 0 0 0 1 0 1 0 1 1 0 0 1 0 0 0 0
0 0 0 0 1 0 0 1 1 1 1 1 0 1 0 0 1 0 0 1 0 0 0 0
0 1 1 1 1 0 0 1 0 0 0 1 0 1 1 1 1 0 0 1 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

2. Identifier dans ce fichier, à partir du modèle, les différentes parties du code (chaînes de caractères magique, ...).
3. Compléter la figure 1 afin d'afficher le résultat obtenu si l'on décidait d'ouvrir ce fichier avec un navigateur web ? ou bien avec le logiciel Gimp ?

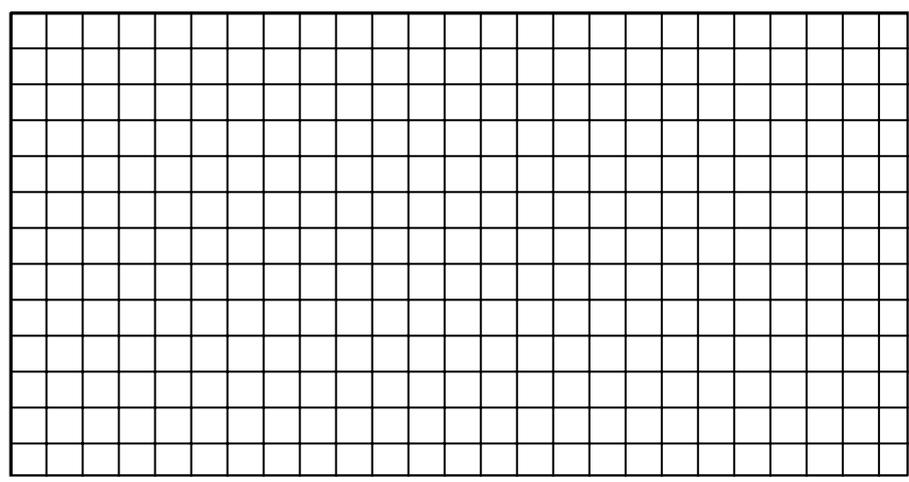


FIGURE 1 – Canevas afin de "dessiner" l'image décodé `exemple5.pbm`.

2. <https://perso-laris.univ-angers.fr/~delanoue/polytech/microcontroleur/exemple5.pbm>

Exercice 3 (Playstation ?)

Vous rencontrez un individu qui vous apprend sa manière de représenter les nombres entiers. Pour lui, les premiers nombres entiers s'écrivent comme illustré par la figure 2.

○
 △
 □
 ×
 △ ○
 △ △
 △ □
 △ ×
 □ ○
 □ △
 □ □
 □ ×
 × ○
 × △
 × □
 × ×

FIGURE 2 – Ses 15 premiers nombres entiers.

1. Quel est le nombre représenté par un carré ? par une croix ?
2. D'après vous, comment écrit-il le nombre que vous avez l'habitude d'écrire 16 ?
3. Quel est le nombre représenté par un triangle suivi d'un carré ? par un triangle suivi d'une croix ?
4. Compléter la figure 3 afin de réaliser l'addition :

$$\begin{array}{r}
 \triangle \square \\
 + \triangle \times \\
 \hline
 \dots
 \end{array}$$

FIGURE 3 – Une addition.

5. Est-ce cohérent ?
6. Finalement, dans quelle base représente-t-il les nombres ?
7. Est-ce que le nombre "triangle suivi d'une croix" est premier dans cette base ?
8. Est-ce que primalité d'un nombre dépend de sa représentation ?

Exercice 4 (Base 2)

1. Ecrire les 16 premiers nombres entiers en utilisant la base 2.
2. Compléter la table d'addition en base 2 suivante :

+	0	1
0	.	.
1	.	..

TABLE 1 – Table d'addition en base 2.

3. Décomposer cette table d'addition en deux tables, l'une pour le digit de poids faible, l'autre pour digit de poids fort.

+	0	1
0	.	.
1	.	.

TABLE 2 – Digit de poids faible (dans ce cas, le digit des unités)

+	0	1
0	.	.
1	.	.

TABLE 3 – Digit de poids fort (dans ce cas, on parle de retenue)

4. Compléter les deux tables vérités suivantes :

A	B	Digit des unités
0	0	.
0	1	.
1	0	.
1	1	.

TABLE 4 – Digit de poids faible (dans ce cas, le digit des unités)

A	B	La retenue
0	0	.
0	1	.
1	0	.
1	1	.

TABLE 5 – Digit de poids fort (dans ce cas, on parle de retenue)

5. Le câblage présenté sur la figure 4 réalise simultanément les deux tables de la question précédente. Tester en additionnant les deux nombres 1 et 1.

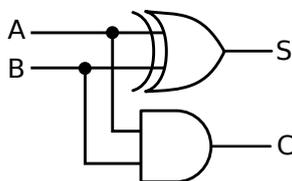


FIGURE 4 – Un demi-additionneur.

6. Afin de pouvoir mettre en série plusieurs additionneurs, on utilise classiquement un additionneur complet (full adder). Un exemple de câblage réalisant un additionneur complet est donné sur la figure 5. Tester en additionnant les deux nombres 1 et 0 avec une entrée C_{in} à 1.

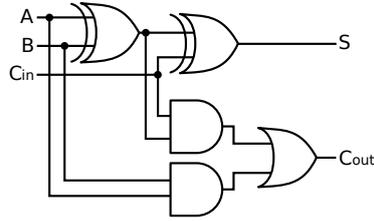


FIGURE 5 – Un additionneur complet.

7. Que represente l'entrée nommée C_{in} ?
8. Effectuer l'addition

$$\begin{array}{r}
 \\
 + \\
 \hline
 \\
 \\

 \end{array}$$

9. Le cablage représenté sur la figure 6 réalise-t-il matériellement l'addition de deux nombres entiers codés chacun sur 4 bits ?

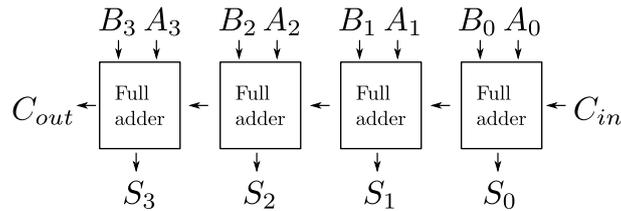


FIGURE 6 – Un additionneur pour deux nombres codés sur 4 bits.

10. Annoter la figure 6 en donnant la valeur de toutes les entrées et sorties lorsqu'on additionne les deux nombres de la question 8.
11. Créer un cablage capable d'additionner deux nombres codés en base 2 chacun sur un octet.

Exercice 5 (Premières représentations des nombres sur un ordinateur)

1. Compléter la représentation avec les indications suivantes :
 - (a) Le nombre 37 est stocké dans la case mémoire numéro 212 (on parle d'adresse 212) d'un ordinateur dans un octet en utilisant le codage « binaire non signé ».
 - (b) Le nombre -108 est stocké dans la case mémoire numéro 213 d'un ordinateur dans un octet en utilisant le codage « binaire signé complément à un ».
 - (c) Le nombre -108 est stocké dans la case mémoire numéro 214 d'un ordinateur dans un octet en utilisant le codage « binaire signé complément à deux ».

⋮	
212	→
213	→
214	→
⋮	

2. Lors de la lecture de la mémoire avec un éditeur hexadécimal, le résultat de la commande donne

⋮	
212	→ 0x25
213	→ 0x93
⋮	

Est-ce cohérent ?

3. Lors de la lecture de la mémoire avec un éditeur hexadécimal, on obtient le résultat suivant :

⋮	
107	→ 0x12
108	→ 0x34
⋮	

Quelle valeur est stockée dans la case 108 ? Est-elle positive ?

Exercice 6 (Accords de musique)

En musique, un accord peut être présenté en donnant les caractéristiques suivantes :

- code de la fondamentale (c'est à dire une des 12 notes),
- majeur ou mineur,
- accord 7 (septième de dominante) ou bien Δ (septième majeure).

Note pour les musiciens : ceci est une vue simplifiée pour les besoins de l'exercice.

1. Combien d'accords différents existe-il selon cette description ?
2. Est-il possible de coder un accord dans un octet ?
3. Proposer un codage binaire permettant de représenter tous ces accords.
4. De combien de bits votre codage nécessite-il ?