

1 Rappels de numération

1.1 Décimal (Base 10)

Rang	n-1	...	5	4	3	2	1	0
Poids	10^{n-1}	...	10^5	10^4	10^3	10^2	10^1	10^0
Ex						2	5	6

Ex :

Le nombre **256** en Décimal:

$$256 = (2 \times 10^2) + (5 \times 10^1) + (6 \times 10^0)$$

$$256 = (2 \times 100) + (5 \times 10) + (6 \times 1)$$

$$256 = 200 + 50 + 6$$

$$256 = 256_{(10)}$$

1.2 Binaire (Base 2)

Rang	n-1	...	5	4	3	2	1	0
Poids	2^{n-1}	...	2^5	2^4	2^3	2^2	2^1	2^0
Ex			1	1	0	0	1	0

Ex :

Le nombre **110010** en Binaire:

$$110010 = (1 \times 2^5) + (1 \times 2^4) + (1 \times 2^1)$$

$$110010 = 32 + 16 + 2$$

$$110010_{(2)} = 50_{(10)}$$

(2) et (10) sont ici utilisés pour préciser la base

1.3 Hexadécimal (Base 16)

Rang	n-1	...	5	4	3	2	1	0
Poids	16^{n-1}	...	16^5	16^4	16^3	16^2	16^1	16^0
Ex						3	D	F

Ex :

Le nombre **3DF** en Hexadécimal:

$$3DF = (3 \times 16^2) + (D \times 16^1) + (F \times 16^0)$$

$$3DF = (3 \times 256) + (13 \times 16) + (15 \times 1)$$

$$3DF = 768 + 208 + 15$$

$$3DF_{(16)} = 991_{(10)}$$

Pour plus de détails voir ([code hexadécimal](#))

1.4 Binaire ↔ Hexadécimal

On passe du binaire à l'hexadécimal en convertissant chaque QUARTET binaire en un chiffre hexadécimal en commençant par la droite.

Ex :

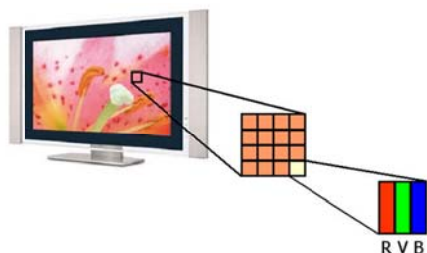
D	3	
┌───┴───┐		
11010011 ₍₂₎	=	D3 ₍₁₆₎ (0011 = 3 et 1101 = D)
101011 ₍₂₎	=	2B ₍₁₆₎ (1011 = B et 10 = 2)

On passe de l'hexadécimal au binaire en convertissant chaque chiffre hexadécimal en un QUARTET binaire.

Ex :

B8 ₍₁₆₎	=	101110000 ₍₂₎ (B = 1011 et 8 = 1000)
4F2C ₍₁₆₎	=	100111100101100 ₍₂₎

2 Le pixel



La dalle d'un écran ou d'un moniteur vidéo est constituée de pixels. Chaque pixel est constitué de 3 sous-pixels Rouge, Vert et Bleu.

La couleur d'un pixel est le résultat de la synthèse additive des trois couleurs RVB de base réalisée par notre cerveau.



En informatique, la technique consiste à envoyer à la carte graphique la valeur numérique correspondant au niveau d'intensité de chaque composante RVB de chaque pixel constituant l'image à afficher. Cette valeur numérique est ensuite convertie en une grandeur électrique analogique afin d'exciter chaque sous pixel de façon plus ou moins intense.

3 Numérisation d'une image

Dans une image numérique, chaque sous pixel RVB d'un pixel est codé en utilisant une valeur numérique binaire proportionnelle à son intensité.

Les intensités RVB sont généralement exprimées par 3 octets (3 x 8bits). On peut donc définir pour chaque niveau RVB, 256 valeurs comprises entre 0 et 255 en décimal ou 00 et FF en hexadécimal ou 00000000 et 11111111 en binaire.

Chaque pixel est donc représenté par un nombre de 24 bits. Exemple :

R	V	B
10001110	00001110	11110010
00001110	1110010	

On représente généralement cette valeur en hexadécimal :

8E1EF2

Un pixel ainsi codée sur 24 bits (3 octets) permet d'obtenir 2^{24} (ou 16^6) soit **16777216** couleurs différentes.

4 Pondération des chiffres

Un OCTET (8 bits) est composé de deux QUARTETS (4 bits). En hexadécimal, le chiffre de droite le moins significatif a un poids égal à 16^0 . Celui de gauche le plus significatif a un poids égal à 16^1 . Le quartet de gauche est donc 16 fois plus significatif que celui de droite.

Binaire	Hexadécimal	Décimal
0000 0000	00	0
0000 0001	01	1
.....
0000 1111	0F	15

La variation de 1 à F (0001 à 1111) du quartet de poids faible fait varier la valeur finale de 1 à 15.

0000 0000	00	0
0001 0000	10	16
.....
1111 0000	F0	240

La variation de 1 à F (0001 à 1111) du quartet de poids fort fait varier la valeur finale de 16 à 240.

Le quartet de poids fort à 16 fois plus de poids que celui de poids faible.

5 Convention de notation binaire & hexadécimale

	Notation préfixée	Notation suffixée
Hexadécimal	0x123C, &h123C, \$123C 0h123C, 16#123C	123Ch, 123C ⁽¹⁶⁾
Binaire	0b110100 2#110100	110100b, 110100 ⁽²⁾

Afin de différencier les nombres binaires, décimaux et hexadécimaux, plusieurs conventions sont utilisées en informatique et en arithmétiques. Certaines utilisent un préfixe d'autres un suffixe.