



# TRAITEMENT NUMÉRIQUE

Aujourd'hui nos ordinateurs, téléphones et autres appareils savent manipuler aussi bien des nombres et du texte que des images, de la vidéo ou de la musique... Mais comment représenter, au sein d'un système numérique, cette diversité des objets du monde réel ou virtuel ?

**Quelles sont les techniques utilisées pour représenter numériquement les grandeurs qui nous entourent ?**

## 1. UNITÉ DE CODAGE

Les composants constituant un système informatique réagissent, de manière interne, à des signaux « ..... ». On représente les deux états stables ainsi définis par les symboles « 0 » et « 1 » ou encore par « L » (Low) et « H » (High).

Le système de numération adaptée à la représentation de tels signaux est la base 2, on parle alors de codage binaire. L'unité de codage de l'information est un élément ne pouvant prendre que les valeurs 0 ou 1 ; le bit (contraction de Binary Digit).

## 2. UNITÉ DE TRANSFERT

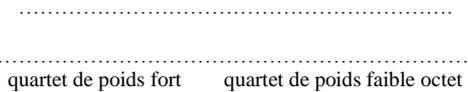
Pour les échanges de données, les informations élémentaires (bits) sont manipulées par groupes qui forment ainsi des mots binaires. La taille de ces mots est le plus souvent un multiple de .....

L'unité de transfert utilisée pour les échanges de données est le mot de 8 bits appelé octet. Exemples : (2 octets)

Remarque :

Un octet est un byte (chiffre binaire) particulier contenant 8 bits.

Pour faciliter les manipulations, un octet peut être divisé en deux mots de 4 bits que l'on appelle des quartets : celui situé à gauche est le quartet de poids fort, MSQ (Most Significant Quartet), et celui situé à droite, le quartet de poids faible, LSQ (Less Significant Quartet).



## 3. MOTS BINAIRES

Dans un mot binaire, le bit situé le plus à gauche est le bit le plus significatif, MSB (Most Significant Bit), celui situé le plus à droite est le bit le moins significatif, LSB (Less Significant Bit).

Exemple :



Dans les notations de quantité binaires « kilo », « méga », ... sont utilisés pour exprimer des multiples en puissances de 2, mais cet usage est contraire aux normes SI (Système International).

- ko (kB) = kilo-octet (kiloByte) = 10<sup>3</sup> octets = 1000 octets
- Mo (MB) = Méga-octet (MegaByte) = 10<sup>6</sup> octets = 1000 ko
- Go (GB) = Giga-octet (GigaByte) = 10<sup>9</sup> octets = 1000 Mo
- To (TB) = Téra-octet (TeraByte) = 10<sup>12</sup> octets = 1000 Go
- kio (kiB) = kibi-octet (kibiByte) = 2<sup>10</sup> octets = 1024 octets
- Mio (MiB) = Mébi-octet (MebiByte) = 2<sup>20</sup> octets = 1024 ko
- Gio (GiB) = Gibi-octet (GibiByte) = 2<sup>30</sup> octets = 1024 Mo
- Tio (TiB) = Tébi-octet (TebiByte) = 2<sup>40</sup> octets = 1024 Go



## 4. EXERCICES D'APPLICATIONS

### EXERCICE N°1

La fiche technique d'un disque dur indique une capacité de 320 GB.  
Question : Exprimer cette capacité en Mio.



### EXERCICE N°2

Votre FAI vous annonce un débit descendant de 8 192 kibits/s.  
Vous faites une mesure de débit réel et vous trouvez une moyenne de 3 280 kibits/s.



Question :  
Quelle sera le temps théorique minimal de téléchargement d'une application de taille égale à 25 Mo ?

## 5. Les systèmes de numération

Les bases de numération utilisées sont :

5.1 Le décimal (Base 10) : { .....

Le système décimal est le système universellement utilisé. C'est la base de référence, ce qui signifie qu'un nombre est de manière implicite décimal dès lors qu'il est écrit sans précision de sa base.

5.2 Le binaire (Base 2) : { .....

C'est la base de numération couramment utilisé en électronique. C'est un système à base 2 qui est donc composé des caractères 0 et 1. Chacun de ces chiffres est appelé 'Bit', contraction des mots Binary Unit ou Binary Digit.

5.3 L'hexadécimal (Base 16). Cette base utilise des nombres et des lettres : { .....

Ce système à base 16 est le plus utilisé en électronique numérique car il permet une manipulation de quartets en représentation compacte. Ce qui, dans les systèmes actuels à grande capacité mémoire par exemple, est un avantage non négligeable. La base 16 est une forme contractée de la base 2.

Lorsqu'on écrit un nombre, il faut spécifier dans quelle base il est écrit, sinon il y a des risques de confusion. Par convention, on indique la base en indice :

✚ 3A9(16) = ..... base 16 : hexadécimal  
✚ 238(10) = ..... base 10 : décimal  
✚ 0100 1101(2) = ..... base 2 : binaire

## 6. Les conversions

### 6.1 Le codage

C'est le passage de la base 10 vers une autre base b.

Pour convertir un nombre décimal N en un nombre en base b il faut diviser le nombre décimal par la base b et répéter cette opération jusqu'à ce que le quotient soit nul. Les restes successifs sont écrits, en commençant par le dernier, de la gauche vers la droite pour former l'expression de N dans les systèmes de base b.



### 6.1.2 Le codage en base de 2

Exemple : Conversion de  $N = (236)_{10}$  en un nombre binaire ( $b = 2$ )

La suite des divisions successives conduit au résultat suivant :

$$N = 236 = (11101100)_2$$



### 6.1.3 Le codage en base de 16

Exemple : Conversion de  $N = (3786)_{10}$  en un nombre hexadécimal ( $b = 16$ )

La suite des divisions successives conduit au résultat suivant :

$$N = 3786 = (ECA)_{16}$$



## 6.2 Le décodage.

C'est le passage d'une base quelconque à la base 10.

Pour convertir un nombre d'une base  $b$  en son équivalent décimal, il faut multiplier chaque chiffre du mot à convertir par le poids qui lui est affecté (ainsi qu'il a été montré dans la présentation des différentes bases).

### 6.2.1 Le décodage d'un nombre binaire.

Exemple : Conversion de  $N = (10110101)_2$  en un nombre décimal

### 6.2.2 Le décodage d'un nombre hexadécimal.

Exemple : Conversion de  $N = (2AF)_{16}$  en un nombre décimal.

## 6.3 Le transcodage.

Le transcodage est le passage d'une base quelconque (autre que la base 10) à une autre base (également différente de la base 10). Il est essentiellement utilisé pour passer du système binaire au système hexadécimal et inversement.



### 6.3.1 Le transcodage base 16 vers base 2.

Chaque symbole du nombre hexadécimal est remplacé par son équivalent écrit dans le système binaire.

### 6.3.2 Le transcodage base 2 vers base 16.

Il faut regrouper les bits du nombre binaire par quatre (en commençant par la droite), puis chaque groupe est remplacé par le symbole hexadécimal correspondant.

## 7. Les Codes

### 7.1 Binaire pur

Toute donnée entrant, par exemple, dans un ordinateur ou un automate est transmis en code binaire pur. Le code binaire pur utilise la numération binaire. La construction se fait par copie du code binaire pur en partant du poids faible (LSB) en rajoutant les bits de poids supérieur pour atteindre le bit de poids fort.

### 7.2 Binaire réfléchi ou code gray

Dans ce codage, un seul bit change d'état entre deux valeurs voisines. Il ne peut pas exister de résultat temporaire aberrant entre deux valeurs voisines. Du fait de cette propriété, le code Gray (ou code réfléchi) est très utilisé, notamment dans les codeurs absolus de position.

	Binaire pur	Binaire réfléchi
0	0000	
1	0001	
2	0010	
3	0011	
4	0100	
5	0101	
6	0110	
7	0111	

### 7.3 Code BCD (Binary Code Décimal) – Décimal codé en binaire

Ce code conserve les avantages du système binaire naturel et du système décimal. Chaque chiffre du code décimal est représenté par un quartet binaire, mais on compte en base 10, ce qui veut dire que la valeur la plus élevée dans un quartet est  $9(10) = 1001(2)$ .

Le chiffre 857 sera donc représenté par :



## 7.4 Code ASCII



Le code ASCII (American Standard Code for Information Interchange) est principalement utilisé pour la transmission en série des caractères alphanumériques (Chiffres, lettres, caractères spéciaux) dans les systèmes informatiques.

Table ASCII sur 7 bits :

Binaire				b6	0	0	0	0	1	1	1	1	
				b5	0	0	1	1	0	0	1	1	
Hexadécimal				b4	0	1	0	1	0	1	0	1	
					0	1	2	3	4	5	6	7	
b3	b2	b1	b0	Décimal	0	16	32	48	64	80	96	112	
0	0	0	0	0	+0	NUL (DEL)	TC7	SP	0	@	P	·	p
0	0	0	1	1	+1	TC1 (SOH)	DC1	!	1	A	Q	a	q
0	0	1	0	2	+2	TC2 (STX)	DC2	"	2	B	R	b	r
0	0	1	1	3	+3	TC3 (ETX)	DC3	#	3	C	S	c	s
0	1	0	0	4	+4	TC4 (EOT)	DC4	\$	4	D	T	d	t
0	1	0	1	5	+5	TC5 (ENO)	TC8 (NAK)	%	5	E	U	e	u
0	1	1	0	6	+6	TC6 (ACK)	TC9 (SYN)	&	6	F	V	f	v
0	1	1	1	7	+7	BEL (ETB)	TC10	'	7	G	W	g	w
1	0	0	0	8	+8	FE0 (BS)	CAN	(	8	H	X	h	x
1	0	0	1	9	+9	FE1 (HT)	EM	)	9	I	Y	i	y
1	0	1	0	A	+10	FE2 (LF)	SUB	*	:	J	Z	j	z
1	0	1	1	B	+11	FE3 (VT)	ESC	+	;	K	[	k	é
1	1	0	0	C	+12	FE4 (FF)	IS4 (FS)	,	<	L	\	l	ù
1	1	0	1	D	+13	FE5 (CR)	IS3 (GS)	-	=	M	]	m	è
1	1	1	0	E	+14	SO	IS2 (RS)	.	>	N	^	n	-
1	1	1	1	F	+15	SI	IS1 (US)	/	?	O	_	o	DEL

### 7.4.1 EXERCICES D'APPLICATIONS

#### EXERCICE N°1

Question :

Décrypter la chaîne ASCII ci-dessous représentée sous la forme d'une suite d'octets :

0011 0001 0101 0011 0100 1001

#### EXERCICE N°2

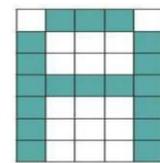
Question :

Retrouver la chaîne ASCII sous la forme décimale du texte page suivante.

J'aime la SI.



## 8. Les images.



Lettre pixelisée

Après le texte, l'image est le support le plus utilisé pour communiquer.

Dans le domaine du numérique, les images sont constituées d'une ..... (Largeur x Hauteur) de points élémentaires que l'on nomme généralement ..... (abréviation de Picture Element).

Chaque pixel a une couleur codée sur un nombre plus ou moins grand de bits.

Le périphérique de sortie (écran, imprimante, ...) se doit de restituer les pixels de manière ordonnée en fonction de leur position respective (x, y) et de leur couleur.

### Exemple :

Le caractère ASCII \$41 peut être simplement représenté sur une matrice 5 x 7 en « allumant » les pixels adéquats. La « couleur » peut-être ici matérialisée par un unique bit (pixel allumé ou éteint).

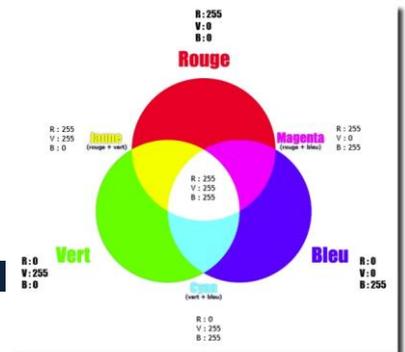
Le codage d'une image peut donc dans un premier temps se résumer en la succession des codages des pixels suivant un ordre bien défini (balayage lignes-colonnes en partant du coin supérieur gauche).

## 9. DE LA COULEUR

Le mode de représentation RVB (Rouge, Vert et Bleu, ou en anglais RGB) correspond à celui fourni par la plupart des caméras couleur, il est naturellement utilisé pour la reproduction de couleurs sur écran (base noire). C'est le mode de composition des couleurs basé sur le principe des couleurs additives : le rouge, le vert et le bleu sont les trois primaires utilisés dans la constitution de couleurs à partir de sources lumineuses.

### Exemple : (codage RVB)

Pixel blanc ® codage RVB = .....



## EXERCICES D'APPLICATIONS

### EXERCICE N°1

#### Question 1 :

Quelle est la taille (en octets) d'une image non compressée, de définition 640 x 480 et de profondeur 24 bits (RVB) ?

L'image est incorporée à un document destiné à être distribué sous forme de photocopies noir et blanc.

#### Question 2 :

Quelle économie de taille réalisez-vous en convertissant l'image en 256 niveaux de gris ?

### EXERCICE N°2

Votre ordinateur affiche sans problème des images de définition 1024 x 768 et de profondeur 32 bits (RVBA).

#### Question :

Que pouvez-vous en déduire quant à la taille mémoire de votre carte vidéo ?

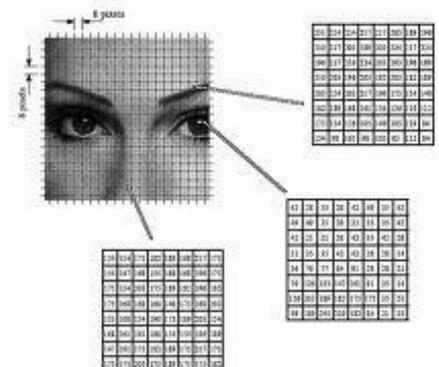


FIGURE 27-6  
JPEG image division. JPEG transform compression starts by breaking the image into 8x8 groups, each containing 64 pixels. Three of these 8x8 groups are retained in this figure, showing the values of the individual pixels, a single-byte value between 0 and 255.



La numération



# TRAITEMENT NUMÉRIQUE

Aujourd'hui nos ordinateurs, téléphones et autres appareils savent manipuler aussi bien des nombres et du texte que des images, de la vidéo ou de la musique... Mais comment représenter, au sein d'un système numérique, cette diversité des objets du monde réel ou virtuel ?

**Quelles sont les techniques utilisées pour représenter numériquement les grandeurs qui nous entourent ?**

## 10. UNITÉ DE CODAGE

Les composants constituant un système informatique réagissent, de manière interne, à des signaux « tout ou rien ». On représente les deux états stables ainsi définis par les symboles « 0 » et « 1 » ou encore par « L » (Low) et « H » (High).

Le système de numération adaptée à la représentation de tels signaux est la base 2, on parle alors de codage binaire. L'unité de codage de l'information est un élément ne pouvant prendre que les valeurs 0 ou 1 ; le bit (contraction de Binary Digit).

## 11. UNITÉ DE TRANSFERT

Pour les échanges de données, les informations élémentaires (bits) sont manipulées par groupes qui forment ainsi des mots binaires. La taille de ces mots est le plus souvent un multiple de  $8 = 2^3$ .

L'unité de transfert utilisée pour les échanges de données est le mot de 8 bits appelé octet.

Exemples : (2 octets)  
1111 0011  
1010 1111

Remarque :

Un octet est un byte (chiffre binaire) particulier contenant 8 bits.

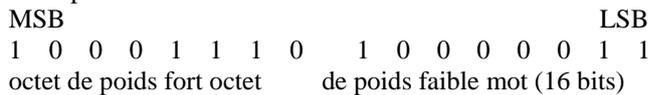
Pour faciliter les manipulations, un octet peut être divisé en deux mots de 4 bits que l'on appelle des quartets : celui situé à gauche est le quartet de poids fort, MSQ (Most Significant Quartet), et celui situé à droite, le quartet de poids faible, LSQ (Less Significant Quartet).



## 12. MOTS BINAIRES

Dans un mot binaire, le bit situé le plus à gauche est le bit le plus significatif, MSB (Most Significant Bit), celui situé le plus à droite est le bit le moins significatif, LSB (Less Significant Bit).

Exemple :



Dans les notations de quantité binaires « kilo », « méga », ... sont utilisés pour exprimer des multiples en puissances de 2, mais cet usage est contraire aux normes SI (Système International).

- ko (kB) = kilo-octet (kiloByte) =  $10^3$  octets = 1000 octets
- Mo (MB) = Méga-octet (MegaByte) =  $10^6$  octets = 1000 ko
- Go (GB) = Giga-octet (GigaByte) =  $10^9$  octets = 1000 Mo
- To (TB) = Téra-octet (TeraByte) =  $10^{12}$  octets = 1000 Go
- kio (kiB) = kibi-octet (kibiByte) =  $2^{10}$  octets = 1024 octets
- Mio (MiB) = Mébi-octet (MebiByte) =  $2^{20}$  octets = 1024 kio
- Gio (GiB) = Gibi-octet (GibiByte) =  $2^{30}$  octets = 1024 Mio
- Tio (TiB) = Tébi-octet (TebiByte) =  $2^{40}$  octets = 1024 Gio



## 13. EXERCICES D'APPLICATIONS

### EXERCICE N°1

La fiche technique d'un disque dur indique une capacité de 320 GB.  
Question : Exprimer cette capacité en Mio.



$$320 \text{ GB} = 320 \cdot 10^9 \text{ octets} = 320 \cdot 10^9 / 2^{20} = 305175,78125 \text{ Mio}$$

### EXERCICE N°2

Votre FAI vous annonce un débit descendant de 8 192 kibits/s.  
Vous faites une mesure de débit réel et vous trouvez une moyenne de 3 280 kibits/s.



Question :  
Quelle sera le temps théorique minimal de téléchargement d'une application de taille égale à 25 Mo ?

$$3\,280 \text{ kibits/s} = 3\,280 \cdot 2^{10} \text{ bits/s} = 3\,280 \cdot 2^{10} / 8 \text{ octets/s} = 419\,840 \text{ octets/s}$$
$$25 \text{ Mo} = 25 \cdot 10^6 \text{ octets}; \quad 25 \cdot 10^6 / 419\,840 = 59.5 \text{ s}$$

## 14. Les systèmes de numération

Les bases de numération utilisées sont :

5.1 Le décimal (Base 10) : {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

Le système décimal est le système universellement utilisé. C'est la base de référence, ce qui signifie qu'un nombre est de manière implicite décimal dès lors qu'il est écrit sans précision de sa base.

14.2 Le binaire (Base 2) : {0, 1}

C'est la base de numération couramment utilisée en électronique. C'est un système à base 2 qui est donc composé des caractères 0 et 1. Chacun de ces chiffres est appelé 'Bit', contraction des mots Binary Unit ou Binary Digit.

5.3 L'hexadécimal (Base 16). Cette base utilise des nombres et des lettres : {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F}

Ce système à base 16 est le plus utilisé en électronique numérique car il permet une manipulation de quartets en représentation compacte. Ce qui, dans les systèmes actuels à grande capacité mémoire par exemple, est un avantage non négligeable. La base 16 est une forme contractée de la base 2.

Lorsqu'on écrit un nombre, il faut spécifier dans quelle base il est écrit, sinon il y a des risques de confusion. Par convention, on indique la base en indice :

$$\begin{aligned} \updownarrow & 3A9(16) = (3A9)_{16} \quad \text{base 16 : hexadécimal} \\ \updownarrow & 238(10) = (238)_{10} \quad \text{base 10 : décimal} \\ \updownarrow & 0100\,1101(2) = (0100\,1101)_2 \quad \text{base 2 : binaire} \end{aligned}$$

## 15. Les conversions

### 6.1 Le codage

C'est le passage de la base 10 vers une autre base b.

Pour convertir un nombre décimal N en un nombre en base b il faut diviser le nombre décimal par la base b et répéter cette opération jusqu'à ce que le quotient soit nul. Les restes successifs sont écrits, en commençant par le dernier, de la gauche vers la droite pour former l'expression de N dans les systèmes de base b.

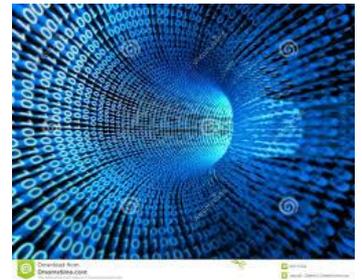
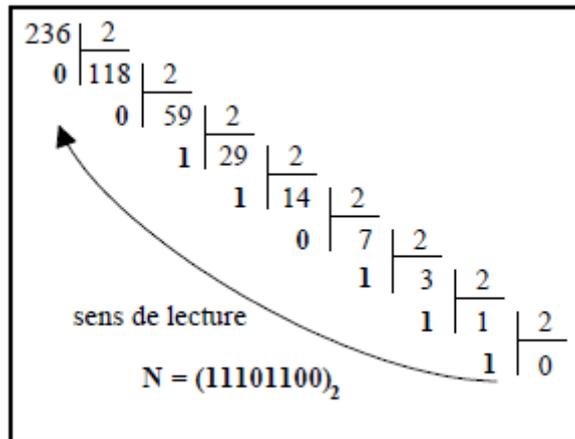


### 6.1.2 Le codage en base de 2

Exemple : Conversion de  $N = (236)_{10}$  en un nombre binaire ( $b = 2$ )

La suite des divisions successives conduit au résultat suivant :

$$N = 236 = (11101100)_2$$

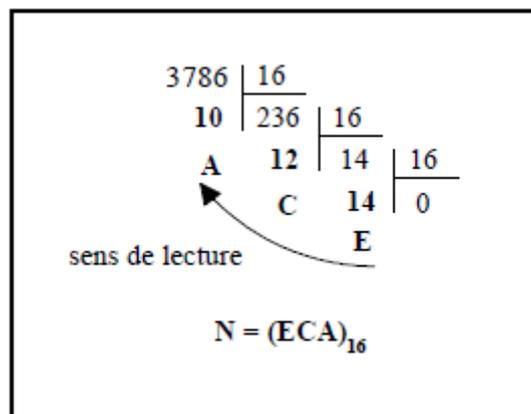


### 6.1.3 Le codage en base de 16

Exemple : Conversion de  $N = (3786)_{10}$  en un nombre hexadécimal ( $b = 16$ )

La suite des divisions successives conduit au résultat suivant :

$$N = 3786 = (ECA)_{16}$$



## 15.2 Le décodage.

C'est le passage d'une base quelconque à la base 10.

Pour convertir un nombre d'une base  $b$  en son équivalent décimal, il faut multiplier chaque chiffre du mot à convertir par le poids qui lui est affecté (ainsi qu'il a été montré dans la présentation des différentes bases).

### 6.2.1 Le décodage d'un nombre binaire.

Exemple : Conversion de  $N = (10110101)_2$  en un nombre décimal

$$N = (1 \ 0 \ 1 \ 1 \ 0 \ 1 \ 0 \ 1)_2$$

$$N = 1 \times 2^7 + 0 \times 2^6 + 1 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$$

$$N = 128 + 0 + 32 + 16 + 0 + 4 + 0 + 1 = (181)_{10}$$

### 6.2.2 Le décodage d'un nombre hexadécimal.

Exemple : Conversion de  $N = (2AF)_{16}$  en un nombre décimal.

$$N = (2 \ A \ F)_{16} = 2 \times 16^2 + 10 \times 16 + 15 = (687)_{10}$$



### 15.3 Le transcodage.

Le transcodage est le passage d'une base quelconque (autre que la base 10) à une autre base (également différente de la base 10). Il est essentiellement utilisé pour passer du système binaire au système hexadécimal et inversement.

#### 15.3.1 Le transcodage base 16 vers base 2.

Chaque symbole du nombre hexadécimal est remplacé par son équivalent écrit dans le système binaire.

$$\text{Exemple : } N = (6AE3)_{16} = (\underbrace{0110}_6 \underbrace{1010}_A \underbrace{1110}_E \underbrace{0011}_3)_2 = (110101011100011)_2$$

#### 15.3.2 Le transcodage base 2 vers base 16.

Il faut regrouper les bits du nombre binaire par quatre (en commençant par la droite), puis chaque groupe est remplacé par le symbole hexadécimal correspondant.

$$\text{Exemple : } N = (10001101111)_2 = (\underbrace{1000}_4 \underbrace{0110}_6 \underbrace{1111}_F)_2 = (46F)_{16}$$

## 16. Les Codes

### 7.1 Binaire pur

Toute donnée entrant, par exemple, dans un ordinateur ou un automate est transmis en code binaire pur. Le code binaire pur utilise la numération binaire. La construction se fait par copie du code binaire pur en partant du poids faible (LSB) en rajoutant les bits de poids supérieur pour atteindre le bit de poids fort.

### 16.2 Binaire réfléchi ou code gray

Dans ce codage, un seul bit change d'état entre deux valeurs voisines. Il ne peut pas exister de résultat temporaire aberrant entre deux valeurs voisines. Du fait de cette propriété, le code Gray (ou code réfléchi) est très utilisé, notamment dans les codeurs absolus de position.

	Binaire pur	Binaire réfléchi
0	0000	
1	0001	
2	0010	
3	0011	
4	0100	
5	0101	
6	0110	
7	0111	

### 7.3 Code BCD (Binary Code Décimal) – Décimal codé en binaire

Ce code conserve les avantages du système binaire naturel et du système décimal. Chaque chiffre du code décimal est représenté par un quartet binaire, mais on compte en base 10, ce qui veut dire que la valeur la plus élevée dans un quartet est  $9(10) = 1001(2)$ .

Le chiffre 857 sera donc représenté par :



## 7.4 Code ASCII



Le code ASCII (American Standard Code for Information Interchange) est principalement utilisé pour la transmission en série des caractères alphanumériques (Chiffres, lettres, caractères spéciaux) dans les systèmes informatiques.

Table ASCII sur 7 bits :

Binaire				b6	0	0	0	0	1	1	1	1	
				b5	0	0	1	1	0	0	1	1	
Hexadécimal				b4	0	1	0	1	0	1	0	1	
					0	1	2	3	4	5	6	7	
b3	b2	b1	b0	Décimal	0	16	32	48	64	80	96	112	
0	0	0	0	0	+0	NUL	TC7 (DEL)	SP	0	@	P	·	p
0	0	0	1	1	+1	TC1 (SOH)	DC1	!	1	A	Q	a	q
0	0	1	0	2	+2	TC2 (STX)	DC2	"	2	B	R	b	r
0	0	1	1	3	+3	TC3 (ETX)	DC3	#	3	C	S	c	s
0	1	0	0	4	+4	TC4 (EOT)	DC4	\$	4	D	T	d	t
0	1	0	1	5	+5	TC5 (ENO)	TC8 (NAK)	%	5	E	U	e	u
0	1	1	0	6	+6	TC6 (ACK)	TC9 (SYN)	&	6	F	V	f	v
0	1	1	1	7	+7	BEL	TC10 (ETB)	'	7	G	W	g	w
1	0	0	0	8	+8	FE0 (BS)	CAN	(	8	H	X	h	x
1	0	0	1	9	+9	FE1 (HT)	EM	)	9	I	Y	i	y
1	0	1	0	A	+10	FE2 (LF)	SUB	*	:	J	Z	j	z
1	0	1	1	B	+11	FE3 (VT)	ESC	+	;	K	[	k	é
1	1	0	0	C	+12	FE4 (FF)	IS4 (FS)	,	<	L	\	l	ù
1	1	0	1	D	+13	FE5 (CR)	IS3 (GS)	-	=	M	]	m	è
1	1	1	0	E	+14	SO	IS2 (RS)	.	>	N	^	n	-
1	1	1	1	F	+15	SI	IS1 (US)	/	?	O	_	o	DEL

### 7.4.1 EXERCICES D'APPLICATIONS

#### EXERCICE N°1

Question :

Décrypter la chaîne ASCII ci-dessous représentée sous la forme d'une suite d'octets :

0011 0001 0101 0011 0100 1001

**ISI**

#### EXERCICE N°2

Question :

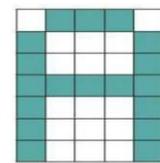
Retrouver la chaîne ASCII sous la forme décimale du texte page suivante.

J'aime la SI.

**74 39 97 105 109 101 32 108 97 32 83 73 46**



## 17. Les images.



Lettre pixelisée

Après le texte, l'image est le support le plus utilisé pour communiquer.

Dans le domaine du numérique, les images sont constituées d'une **matrice L x H** (Largeur x Hauteur) de points élémentaires que l'on nomme généralement **des pixels** (abréviation de Picture Element).

Chaque pixel a une couleur codée sur un nombre plus ou moins grand de bits.

Le périphérique de sortie (écran, imprimante, ...) se doit de restituer les pixels de manière ordonnée en fonction de leur position respective (x, y) et de leur couleur.

### Exemple :

Le caractère ASCII \$41 peut être simplement représenté sur une matrice 5 x 7 en « allumant » les pixels adéquats. La « couleur » peut-être ici matérialisée par un unique bit (pixel allumé ou éteint).

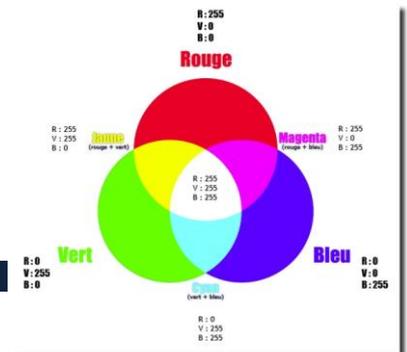
Le codage d'une image peut donc dans un premier temps se résumer en la succession des codages des pixels suivant un ordre bien défini (balayage lignes-colonnes en partant du coin supérieur gauche).

## 18. DE LA COULEUR

Le mode de représentation RVB (Rouge, Vert et Bleu, ou en anglais RGB) correspond à celui fourni par la plupart des caméras couleur, il est naturellement utilisé pour la reproduction de couleurs sur écran (base noire). C'est le mode de composition des couleurs basé sur le principe des couleurs additives : le rouge, le vert et le bleu sont les trois primaires utilisés dans la constitution de couleurs à partir de sources lumineuses.

### Exemple : (codage RVB)

Pixel blanc @ codage RVB = \$FFFFFF



## EXERCICES D'APPLICATIONS

### EXERCICE N°1

#### Question 1 :

Quelle est la taille (en octets) d'une image non compressée, de définition 640 x 480 et de profondeur 24 bits (RVB) ?

**Taille de l'image = 640 x 480 x 3 octets = 921 600 octets**

L'image est incorporée à un document destiné à être distribué sous forme de photocopies noir et blanc.

#### Question 2 :

Quelle économie de taille réalisez-vous en convertissant l'image en 256 niveaux de gris ?

**Taille de l'image en niveaux de gris = 1 octet par pixel soit 640 x 480 x 1 = 307 200 octets**

**Économie = 921 600 - 307 200 = 614 400 octets soit 66,6 %**

### EXERCICE N°2

Votre ordinateur affiche sans problème des images de définition 1024 x 768 et de profondeur 32 bits (RVBA).

#### Question :

Que pouvez-vous en déduire quant à la taille mémoire de votre carte vidéo ?

**La taille mémoire vidéo doit être supérieur à 1024\*768=786432 pixels**

**32/8=4 octets 786432\*4 = 3145728 octets = 25165824 bits.**

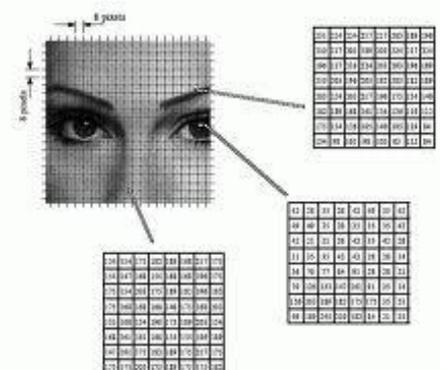


FIGURE 21-6 JPEG image division. JPEG transform compression starts by breaking the image into 8x8 groups, each containing 64 pixels. Three of these 8x8 groups are enlarged in this figure, showing the values of the individual pixels, a single-byte value between 0 and 255.

