Numérisation de l'information

Table des matières

La numérisation est la conversion des informations d'un support (texte, image, audio, vidéo) ou d'un signal électrique en données numériques que des dispositifs informatiques ou d'électronique numérique pourront traiter. Les données numériques se définissent comme une suite de caractères et de nombres qui représentent des informations. On utilise parfois le terme franglais digitalisation (digit signifiant chiffre en anglais).

1. Transmission des informations

On appelle information un ensemble de connaissances qui peuvent être codés de plusieurs façons Le transfert d'une information nécessite une chaîne de transmission. Elle comporte un encodeur, qui code l'information, qui la transmet à un émetteur (qui éventuellement la crypte, la compresse,la module,...) qui la transmet à un récepteur (qui éventuellement la décrypte, la décompresse, la démodule,...) qui la décode et la restitue.

Exemple : Chaîne de transmission en téléphonie

La transmission peut être soit analogique (signal continu) ou numérique (signal discret).

La façon de transmettre l'information a évolué au niveau du milieu de transmission et de la nature des signaux.

- Si on utilise l'atmosphère comme milieu, les signaux peuvent être : des sons, des ultra-sons, des ondes électromagnétiques,...
- Si on utilise la fibre optique, les signaux sont alors des ondes électromagnétiques.

Exemple : Lorsqu'un usager téléphone, une ligne le relie à son correspondant.

Elle assure le transport de la voix, dans les deux sens, jusqu'à ce que la communication soit terminée. Cette liaison provisoire est créée par la compagnie de téléphone, grâce à des opérations de commutation effectuées dans les centraux téléphoniques.

Dans l'histoire du téléphone, la commutation a d'abord été réalisée de façon manuelle, puis électromécanique, puis enfin informatique.

Entre l'invention du téléphone par A.G.Bell (en 1877) et les années soixante, la voix fut transmise de manière analogique, sous forme d'un signal électrique se propageant sur des fils de cuivre. Puis les compagnies de téléphone commencèrent à utiliser la transmission numérique entre les centraux.

La transformation du signal, analogique vers numérique et inversement, est assurée par des "codecs" (Codeur/DECodeur).

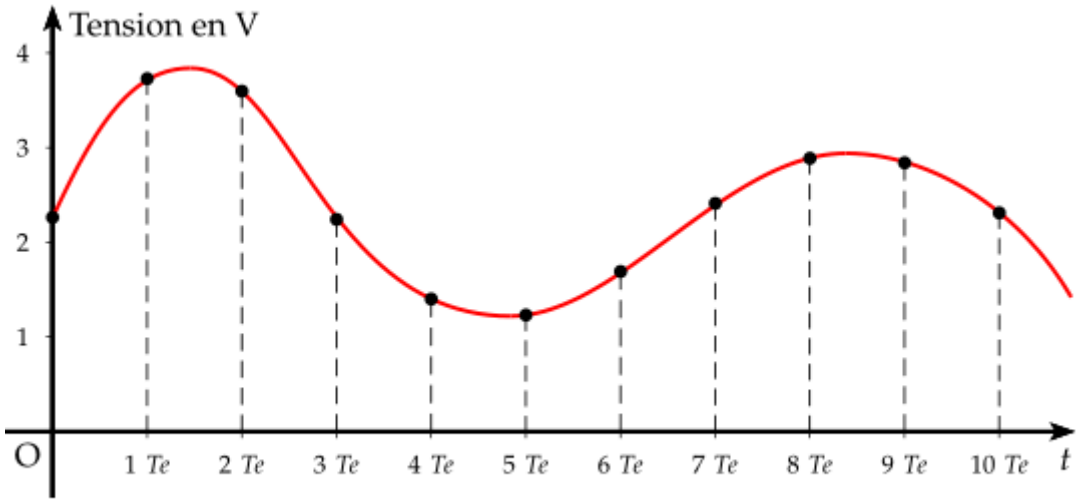
2. La numérisation

Un signal est la représentation physique d'une information qui est transportée avec ou sans transformation, de la source jusqu'au destinataire. Il en existe deux catégories :

- les signaux analogiques, qui varient de façon continue dans le temps (intensité sonore, intensité lumineuse, pression, tension), c'est-a-dire qu'ils peuvent prendre une infinité de valeurs différentes.
- les signaux numériques qui transportent une information sous la forme de nombres.

Le signal analogique à convertir est une tension électrique variable issue d'un capteur (microphone par exemple) ou d'un circuit électrique.

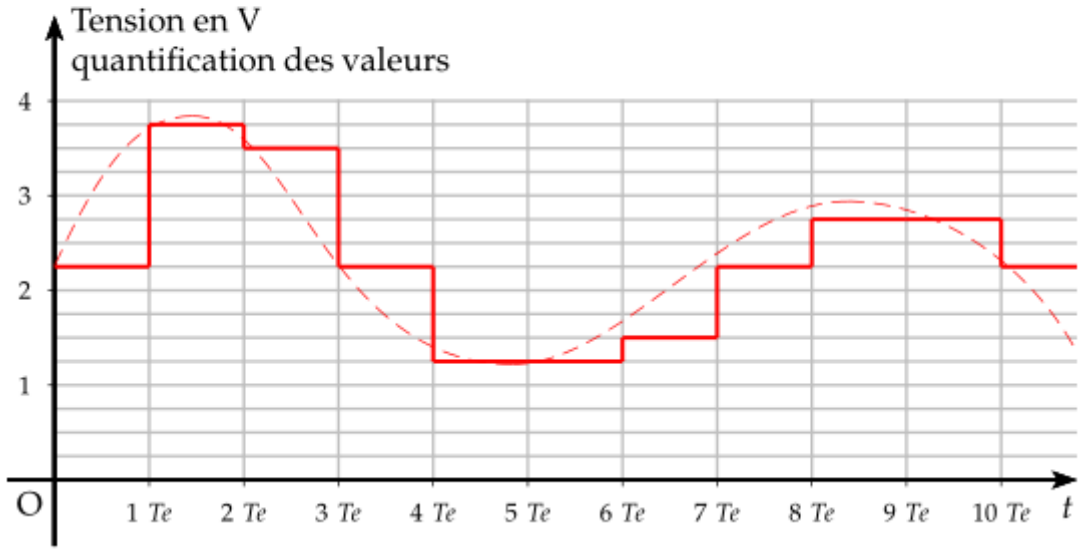
On obtient alors la courbe suivante représentant le signal analogique :



signal analogique

Numériser un signal analogique consiste à transformer les grandeurs continues dans le temps en des grandeurs discontinues qui varient par palier en prenant des valeurs à intervalle de temps régulier : période d'échantillonnage Te.

La numérisation est faites à l'aide d'un convertisseur analogique-numérique (CAN).



signal numérique : résolution de 0,25 V

Remarque : La numérisation est d'autant meilleure que le signal numérique se rapproche du signal analogique initial.

La numérisation d'un signal nécessite trois étapes :

1. L'échantillonnage

2. La quantification
3. Le codage

On appelle BIT (BInary digiT, C. Shannon 1938) le plus petit élément d'information stockable par un ordinateur. Le bit est la particule élémentaire d'information. Un bit ne peut prendre que deux valeurs (0 ou 1) correspondant à deux états possibles d'un élément de circuit électrique (tension présente ou nulle aux bornes d'un dipôle). L'opération qui consiste à transformer (ou coder) une information en une suite de bits est appelée NUMÉRISATION.

- **La numération décimale** utilise 10 symboles ou CHIFFRES : 0, 1,2, 3,4, 5,6, 7, 8 et 9.

  Exemple : 459 = 400 + 50 + 9 = $4x10^2 + 5x10^1 + 9x10°$
- **La numération binaire** utilise 2 symboles ou CHIFFRES : 0 et 1.

  Exemple: $(101)_2 = 1x2^2 + 0x2^1 + 1x2^0 = (5)_{10}$

Les chiffres d'un nombre représentent la décomposition du nombre selon les **puissances croissantes** de la **base** de **numération** considérée. Exemple $(357)_{10}$, signifie que le nombre 357 est exprimé en base 10.

Le nombre binaire 101 sera notée $(101)_2$ ce qui signifie 101 en base binaire (ou base 2). Ce nombre binaire vaut 5 en base décimal : $(101)_2 = (5)_{10}$

Un octet (byte en anglais) est constitué de **8 bits**.

Exemple : valeur d'un octet : 1101 0110

2.1. Le poids d'un bit

Dans un nombre binaire, la valeur d'un bit, appelée **poids**, dépend de la **position du bit** dans le nombre binaire. A la manière des dizaines, des centaines et des milliers pour un nombre décimal, le poids d'un bit croît d'une puissance de deux en allant de la droite vers la gauche comme le montre le tableau suivant :

Nombre binaire	1	1	1	1	1	1	1	1	1
Poids	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
nombre décimal correspondant	256	128	64	32	16	8	4	2	1

Exemple: $(1101\ 0110)_2 = 2^7 + 2^6 + 2^4 + 2^2 + 2^1 = (214)_{10}$

2.2. Conversion binaire/décimale

Si l'on numérote les bits de 0 à 7, le bit numéro 0 est le bit de poids faible, et le bit numéro 7 est le bit de poids le plus fort. Si l'on considère un octet comme un nombre écrit en base 2, sa valeur numérique en base 10 vaut :

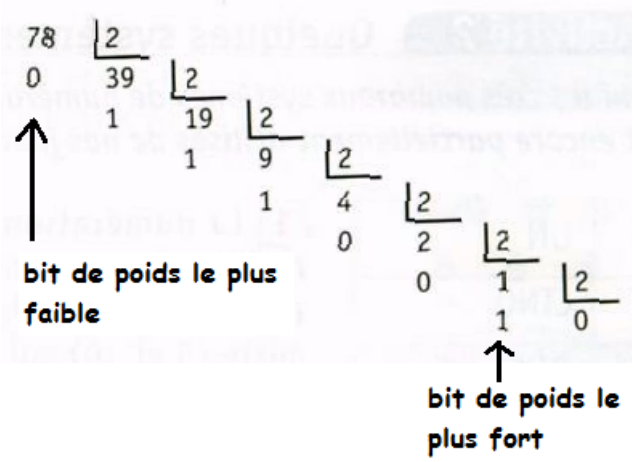
$$\sum_{n=0}^{n=7} b_n.2^n$$ avec b_n valeur du nième bit.

On peut également utiliser le tableau précédent pour effectuer la conversion binaire décimale.

2.3. Conversion décimale/binaire

Pour la **conversion décimale → binaire,** on procède par division successives par 2 : les restes des divisions sont les chiffres binaires de la conversion. Le bit « de poids faible » (le plus à droite) est le premier reste obtenu.

Exemple : 78 (décimal) = 0100 1110 (binaire). Cette conversion est illustrée ci-dessous.



2.4. L'échantillonnage

On appelle période d'échantillonnage Te (en s), le temps entre deux mesures successives.

La fréquence d'échantillonnage fe, correspond au nombre de mesures effectuées par seconde. On a :

$$f_e = \frac{1}{T_e}$$

Remarque : Le choix de la fréquence d'échantillonnage est crucial afin de reproduire fidèlement le signal étudié. En effet si le signal analogique varie trop vite par rapport à la fréquence d'échantillonnage, la numérisation donnera un rendu incorrect.

Théorème de Shannon :

Pour un signal périodique (comme un son) la fréquence échantillonnage fe doit être au moins le double de la fréquence maximale f max du signal : fe > f max

Exemple : Les fichiers audio sont couramment échantillonnés à 44,1 kHz, car cela permet de restituer des sons dont la fréquence peut aller jusqu'à 22,05 kHz, c'est-à-dire un peu au-delà de la fréquence maximale audible par l'Homme (20 kHz).

2.5. La quantification

Un signal numérique ne peut prendre que certaines valeurs : c'est la quantification. Elle s'exprime en bits.

Cette quantification est assurée par un convertisseur (CAN). Chaque valeur est arrondie à la valeur permise la plus proche par défaut.

On appelle alors résolution, ou pas, l'écart (constant) entre deux valeurs permises successives.

Remarque : Un bit (de l'anglais binary digit) est un chiffre binaire (0 ou 1). C'est la plus petite unité de numérisation.

On définit alors un multiple du bit : l'octet. Un octet est un ensemble de 8 bits.

On peut donc quantifier $2^8 = 256$ valeurs avec un octet. Par exemple 0100 1001.

Plus la quantification est grande, plus l'amplitude du signal numérique sera proche de celle du signal analogique.

Exemple : Quantification sur différents support de sons

Type de support	**Quantification choisie**	**nombre de valeurs**
CD audio	16 bits	65 536
DVD	24 bits	16 777 216
Téléphonie	8 bits	256
Radio numérique	8 bits	256

Le nombre d'octets qui vont être nécessaires pour numériser le signal sur un support de stockage (disque dur, clé USB, DVD,...) n'est pas illimités, ce qui explique les quantifications choisies. De plus, en ce qui concerne la radio numérique, il faut du temps pour écrire toutes ces données. Le "flux" n'est pas aussi illimitée.

On appelle calibre l'intervalle des valeurs mesurables des tensions analogiques à numériser (par exemple ± 5 V).

On appelle plage d'un convertisseur, la largeur de l'intervalle entre la plus petite et la plus grande valeur du calibre. (pour un calibre de ± 5 V, la plage est alors de 10 V).

Le pas p d'un convertisseur de n bits et de plage donnée, est alors défini par :

$$p = \frac{plage}{2^n}$$

Exemple : Le convertisseur (CAN) d'une carte d'acquisition possède les caractéristiques suivantes : calibre ± 4,5 V sur 12 bits. Déterminer le pas du convertisseur.

La plage est donc de 9 V. Le pas est alors de : $p = \frac{9}{2^{12}} = 2{,}2.10^{-3} V$

2.6. Le codage

On appelle codage la transformation des différentes valeurs quantifiées en langage binaire.

Remarque : On définit les multiples (SI) et binaires de l'octet suivants :

Préfixes SI

Nom	Symbole	Valeur
kilooctet	ko	10^3
mégaoctet	Mo	10^6
gigaoctet	Go	10^9
téraoctet	To	10^{12}
pétaoctet	Po	10^{15}

Préfixes binaires

Nom	Symbole	Valeur
kibioctet	kio	2^{10}
mébioctet	Mio	2^{20}
gibioctet	Gio	2^{30}
tébioctet	Tio	2^{40}
pébioctet	Pio	2^{50}

Cette distinction n'est malheureusement pas appliqué par le grand public ou les fabricants : on parle ainsi kilooctet à la place de kibioctet. Cela crée des confusions : un disque de 100 gigaoctets à la même capacité qu'un disque de 93,13 gibioctets.

Exemple : Le nombre N d'octets nécessaires pour "décrire" numériquement une minute de son est :

$$N = f \times \frac{q}{8} \times 60 \times n$$

- f : fréquence échantillonnage en Hz
- q : quantification en bits
- n : nombre de voies (si le son est stéréo, n = 2; en mono : n = 1)

Déterminer le nombre d'octet nécessaires pour une minute d'en CD audio (44,1 kHz et 16 bits, stéréo).

$$N = 44\,100 \times \frac{16}{8} \times 60 \times 2 = 10\,584\,000\,octets$$

$$N = \frac{10\,584\,000}{1024} = 10\,335\,kio = \frac{10\,335}{1024} = 10{,}9\,Mio$$

Soit 10,9 Mo pour le grand public!

3. Caractéristiques d'une image numérique

3.1 Pixellisation

Une image numérique est un ensemble discret de points appelés Pixels (contraction de PICTure ELements). Elle a pour vocation d'être affichée sur un écran.

Chaque pixel possède une couleur.

Pour fabriquer une image numérique (à partir d'un appareil photo, scanner, caméra numérique), il faut des capteurs qui sont de petites cellules photoélectriques placées en quadrillage.

L'appareil découpe l'image en un quadrillage ou trame. Chaque case est un pixel.

Le pixel est une portion de surface élémentaire permettant d'échantillonner spatialement une image. A chaque pixel est affecté un nombre binaire correspondant à la couleur de la case.

La définition de l'image est le nombre de pixels qu'elle contient. C'est le nombre de pixels contenus dans la dalle de capteurs d'un appareil numérique.

La résolution de l'image est le nombre de pixels par unité de longueur. Elle s'exprime en ppp (pixel par pouce) ou dpi (dot per inch). Le pouce (inch en anglais) vaut 2,54 cm.

Exemple : une feuille A4 (21 x 29,7) numérisée en 300 ppp correspond à une trame de

$\frac{300}{2,54} \times 21,0 = 2480\ pixels$ sur $\frac{300}{2,54} \times 29,7 = 3508\ pixels$

Le fichier est composé de 2480 × 3508 = 8699840 pixels soit à peu près 8,7 Mpixels.

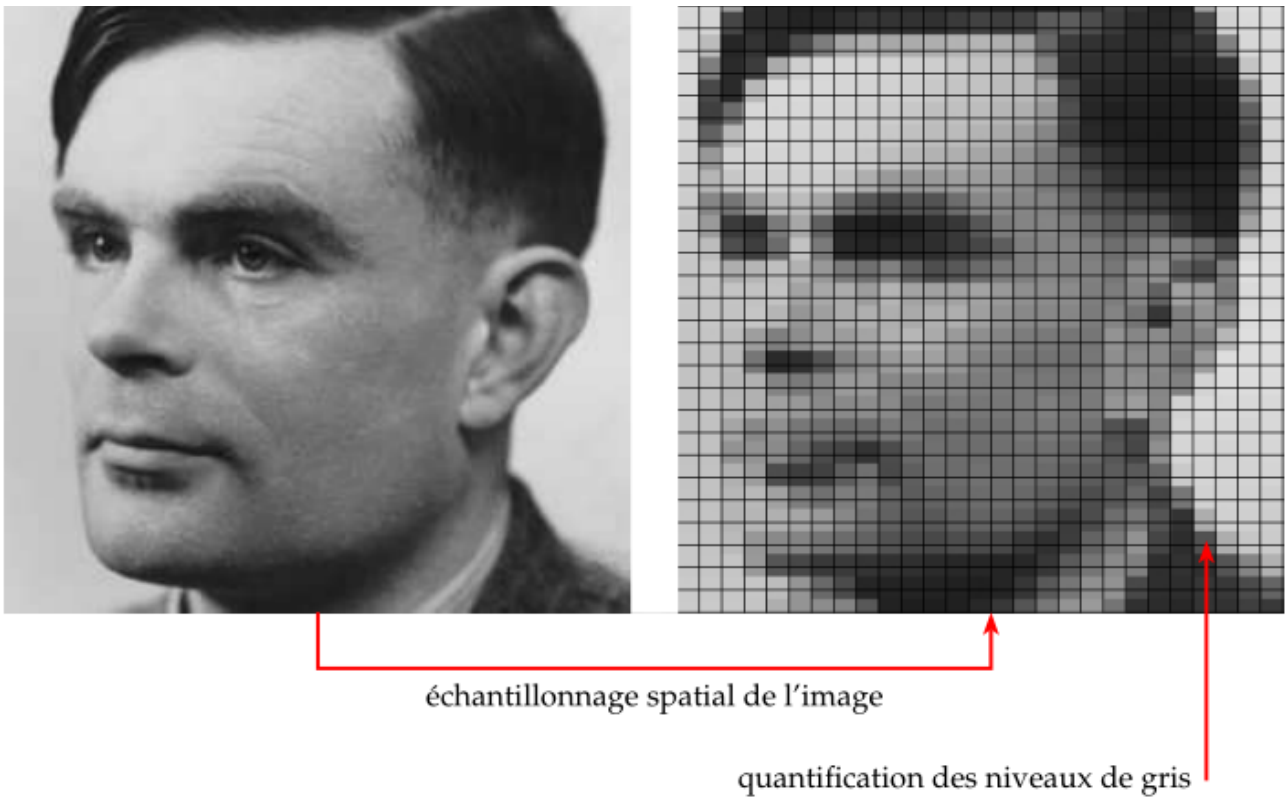
3.2. Codage en niveaux de gris

Chaque cellule du capteur mesure l'intensité lumineuse moyenne correspondant au pixel.

L'intensité lumineuse, grandeur analogique, est convertie par la cellule en un signal analogique sous forme de tension électrique.

Elle est ensuite quantifiée, puis numérisée. A chaque valeur d'intensité lumineuse correspond un niveau de gris codé numériquement.

A LAN T URING mathématicien, fondateur de la sciences informatique

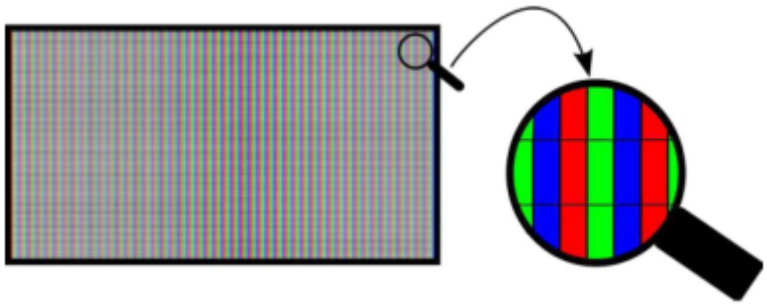


Un système à 4 bits permet de coder $2^4 = 16$ niveaux de gris.

Un octet (8 bits) permet, lui, de coder pour chaque pixel $2^8 = 256$ niveaux de gris. La valeur numérique codant l'intensité lumineuse et la position du pixel sont stockées dans la mémoire. L'image est reconstruite par l'ordinateur à partir des données collectées et numérisées.

3.3. Le codage RVB

Il existe plusieurs système de codage des couleurs dont le plus utilisé est le système RGB (rouge,vert,bleu). Ce système utilise les trois couleurs primaires. La superposition de ces trois couleurs permet de recréer toutes les autres couleurs.



Pour coder les couleurs d'un pixel dans le système RVB, le fichier image associe à chaque pixel 3 octets correspondant aux trois couleurs primaires. Il y a donc 256 valeurs pour chaque couleur, soit en tout : $256^3 = 16\ 581\ 375$ couleurs possibles.

Remarque : d'autre système sont possible comme le cmjn (cmyk en anglais) : cyan, magenta, jaune et noir (quadrichromie), le système TSL (HSL) teinte, saturation, luminosité qui est basé sur le cercle chromatique.

On appelle définition d'une image, le nombre de pixels qui la compose. Par exemple pour une image de 640 colonnes sur 240 lignes, l'image est composée de : $640 \times 240 = 153\ 600$ pixels

On appelle taille d'une image, le produit de sa définition par le nombre d'octet par pixel. Par exemple une image RGB de 640 colonnes sur 24

4. Codage du texte

4.1. Les différents types de code

Le texte est constitué de caractères (lettre, chiffre signe de ponctuation). Chaque caractère est représenté par un entier. Il existe de nombreux codages des caractères ; les principaux codages pour les occidentaux sont :

1. Le code ASCII (ISO 646)
2. Les codes ISO 8859-1 / ISO 8859-15 (symbole e)
3. Le code Unicode
4. Les codes UTF-8 / UTF-16 / UTF-32

4.2. Le code ASCII

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040		Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	DEL

L'ASCII (American standard code for information interchange) a été créé au début des années 60. Son principe consiste à associer à chaque lettre, chiffre ou caractère d'un clavier d'ordinateur un entier compris entre 0 et 127, donc représentable sur 7 bits. Avec ce code on peut représenter les chiffres, les lettres latines, et les principaux symboles de ponctuation. Exemple : lorsqu'on tape dans un texte sur la barre d'espace (space), l'ordinateur enregistre dans sa mémoire vive (RAM) le code hexadécimal $(20)_{hex}$. Le codage binaire sur 7 bits correspondant est : $(010\ 0000)_2 = (32)_{10}$

4.3. Le codage ISO 8859-1

Le codage ASCII suffit pour coder un texte anglais mais ne suffit pas pour les autres langues. Par exemple, les lettres accentuées ne figurent pas dans le code ASCII. Le code **ISO 8859-1** a été créé dans les années 80, il est appelé également **'latin-1'**. Il permet de représenter les caractères accentués. A chaque caractère alphanumérique (lettre, chiffre, ponctuation etc..) est associé un nombre compris entre 0 et 255. On a donc besoin de 8 bits pour représenter l'ensemble des caractères.

Exemple : le codage du point d'interrogation est $(3F)_{hex} = (0011\ 1111)_2$

ISO-8859-1																
	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1x	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2x	SP	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3x	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4x	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5x	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6x	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7x	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DEL
8x	PAD	HOP	BPH	NBH	IND	NEL	SSA	ESA	HTS	HTJ	VTS	PLD	PLU	RI	SS2	SS3
9x	DCS	PU1	PU2	STS	CCH	MW	SPA	EPA	SOS	SGCI	SCI	CSI	ST	OSC	PM	APC
Ax	NBSP	¡	¢	£	¤	¥	¦	§	¨	©	ª	«	¬		®	¯
Bx	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
Cx	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
Dx	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
Ex	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
Fx	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

4.4. L'Unicode

D'autres codages ont été définis pour les autres langues : le chinois, l'arabe etc.. Ces codes sont compatibles avec l'ASCII mais ne le sont pas entre eux. Comment écrire un texte multilingue ? On a créé un **unique codage universel l'Unicode.** Les valeurs entières associées étaient initialement comprises entre 0 et 65235 donc codées sur 16 bits. Cependant ce codage soufre de nombreux défauts (car mis en place par l'homme). Maintenant il est codé sur 32 bits. Toutes les langues connues (sur Terre) sont représentées dans Unicode.

Problème ; le é peut se représenter soit par le caractère 'é' du latin-1, soit par la suite de caractère « 'e » .

Pour pallier les problèmes de l' UNICODE on a construits des codages tels que l'UTF-8.