CODAGE DU TEXTE

1 GENERALITES

1.1 Objectifs recherchés :

De tout temps, les hommes ont cherché à transmettre l'information à distance. Avec l'apparition de technologies électriques puis électroniques/informatiques dans les pays occidentaux, l'alphabet de ces pays (romain) a été transformé en un code adapté à la technique utilisée.

Nous n'étudierons ici que les codages liés à la transmission de données par des réseaux.

1.2 Informations à coder :

- **Textes :** C'est le premier type de message que l'on a voulu transmettre, l'alphabet latin fut le premier utilisé. Le codage dépend de l'alphabet à transmettre et des performances du réseau.
- **Chiffres :** idem texte, on utilise habituellement les chiffres arabes et la base 10.
- **Sons :** transmis sans codage particulier dans les premiers réseaux.
- **Images :** le codage dépendra essentiellement de la qualité attendue (définition, couleur) et du débit possible.

1.3 Réseaux utilisés :

- TELEX : L'invention de l'électricité a permis la création du **Telex** afin de transmettre des messages**,** le débit étant très limité le codage se devait d'être compact.

- RADIO : Les premiers équipements radio ont permis de relier des mobiles (bateaux, avions), le débit initial étant faible on a utilisé un codage simple : le **Morse**.

- TELEPHONE : Les premiers réseaux téléphoniques ont permis de transmettre directement la parole puis des images (bélinographe). L'apparition du Modem vers 1960 permet la transmission de données à faible débit.

- Réseaux numériques : les débits initiaux étaient faibles et les besoins étaient essentiellement dans la transmission de textes. Avec l'explosion d'Internet, ces réseaux servent à transporter toutes sortes d'informations et on voit donc apparaître de multiples codages selon les applications. L'apparition de réseaux hauts débits permet d'envisager des applications de type « images temps réel ».

2 CODAGE DU TEXTE

2.1 Code MORSE :

Ce codage, utilisé surtout pour les transmissions radios, est particulièrement compact. Il n'utilise que 2 symboles (trait, point) et ne code que l'alphabet latin (majuscule) et les chiffres.

Ce codage est efficace car la taille du code n'est pas fixe et les lettres les plus courantes sont codées sur peu de symboles : E = 1 point… (comme un code de Huffman)

Lettre	Code	Lettre	Code	Lettre	Code	Chiffre	Code
A	• –	K	– • –	U	• • –	0	– – – – –
B	– • • •	L	• – • •	V	• • • –	1	• – – – –
C	– • – •	M	– –	W	• – –	2	• • – – –
D	– • •	N	– •	X	– • • –	3	• • • – –
E	•	O	– – –	Y	– • – –	4	• • • • –
F	• • – •	P	• – – •	Z	– – • •	5	• • • • •
G	– –•	Q	– – • –	Point	– • – • – •	6	– • • • •
H	• • • •	R	• – •	Erreur	• • • • • •	7	– – • • •
I	• •	S	• • •	Début	– • – • –	8	– – – • •
J	• – – –	T	–	Fin	• – • – •	9	– – – – •

2.2 Code BAUDOT :

Ce code binaire, aussi appelé AI n°2 (Alphabet International 2) ou ITA2 *(International Telegraph Alphabet)*, fut utilisé par le Télex. Ce code est sur 5 bits (32 combinaisons), il y a donc 2 jeux de codes : 1er jeu pour les lettres (majuscules) et 2ème jeu pour les chiffres et symboles.

Le Télex transmet à 50 Bauds, ce codage permet donc une transmission de 10 caractère/s compatible avec la vitesse de frappe des téléscripteurs électromécaniques.

Lettre	Chiffre	Code	Lettre	Chiffre	Code
A	-	**00011**	Q	1	**10111**
B	?	**11001**	R	4	**01010**
C	:	**01110**	S	'	**00101**
D	Qui ?	**01001**	T	5	**10000**
E	3	**00001**	U	7	**00111**
F	(1)	**01101**	V	=	**11110**
G	(1)	**11010**	W	2	**10011**
H	(1)	**10100**	X	/	**11101**
I	8	**00110**	Y	6	**10101**
J	Ring	**01011**	Z	+	**10001**
K	(	**01111**	CR (retour chariot)		**01000**
L	)	**10010**	LF (saut ligne)		**00010**
M	.	**11100**	**Lettre**		**11111**
N	,	**01100**	**Chiffre**		**11011**
O	9	**11000**	SP (espace)		**00100**
P	0	**10110**	(inutilisé)		**00000**

(1) Inutilisés en TELEX, on peut y trouver des codes nationaux (É, %, H en France).

2.3 Code ASCII :

ASCII = *American Standard Code for Information Interchange.* Ce code binaire, aussi appelé **AI n°5** (Alphabet International 5), fut créé pour la transmission de textes vers des terminaux ou des imprimantes. Ce code est sur 7 bits (128 combinaisons), il permet le codage des majuscules et minuscules. Les 32 premières combinaisons sont réservées au contrôles.

Ce code reste la base pour les codages de texte. On peut remarquer qu'entre les lettres majuscules et minuscules seul le bit 5 change !

Déci	Hexa	Car.	Déci	Hexa	Car.	Déci	Hexa	Car.	Déci	Hexa	Car.
0	00	**NUL**	32	20	**SP**	64	40	**@**	96	60	**`**
1	01	**SOH**	33	21	**!**	65	41	**A**	97	61	**a**
2	02	**STX**	34	22	**"**	66	42	**B**	98	62	**b**
3	03	**ETX**	35	23	**#**	67	43	**C**	99	63	**c**
4	04	**EOT**	36	24	**$**	68	44	**D**	100	64	**d**
5	05	**ENQ**	37	25	**%**	69	45	**E**	101	65	**e**
6	06	**ACK**	38	26	**&**	70	46	**F**	102	66	**f**
7	07	**BEL**	39	27	**'**	71	47	**G**	103	67	**g**
8	08	**BS**	40	28	**(**	72	48	**H**	104	68	**h**
9	09	**HT**	41	29	**)**	73	49	**I**	105	69	**i**
10	0A	**LF**	42	2A	*****	74	4A	**J**	106	6A	**j**
11	0B	**VT**	43	2B	**+**	75	4B	**K**	107	6B	**k**
12	0C	**FF**	44	2C	**,**	76	4C	**L**	108	6C	**l**
13	0D	**CR**	45	2D	**-**	77	4D	**M**	109	6D	**m**
14	0E	**SO**	46	2E	**.**	78	4E	**N**	110	6E	**n**
15	0F	**SI**	47	2F	**/**	79	4F	**O**	111	6F	**o**
16	10	**DLE**	48	30	**0**	80	50	**P**	112	70	**p**
17	11	**DC1**	49	31	**1**	81	51	**Q**	113	71	**q**
18	12	**DC2**	50	32	**2**	82	52	**R**	114	72	**r**
19	13	**DC3**	51	33	**3**	83	53	**S**	115	73	**s**
20	14	**DC4**	52	34	**4**	84	54	**T**	116	74	**t**
21	15	**NAK**	53	35	**5**	85	55	**U**	117	75	**u**
22	16	**SYN**	54	36	**6**	86	56	**V**	118	76	**v**
23	17	**ETB**	55	37	**7**	87	57	**W**	119	77	**w**
24	18	**CAN**	56	38	**8**	88	58	**X**	120	78	**x**
25	19	**EM**	57	39	**9**	89	59	**Y**	121	79	**y**
26	1A	**SUB**	58	3A	**:**	90	5A	**Z**	122	7A	**z**
27	1B	**ESC**	59	3B	**;**	91	5B	**[**	123	7B	**{**
28	1C	**FS**	60	3C	<	92	5C	\	124	7C	**|**
29	1D	**GS**	61	3D	=	93	5D	**]**	125	7D	**}**
30	1E	**RS**	62	3E	>	94	5E	^	126	7E	**~**
31	1F	**US**	63	3F	**?**	95	5F	**_**	127	7F	**DEL**

Caractères de contrôle			
Car.	**Sigle**	**Désignation**	**CTRL**
NUL	Null	Nul	**@**
SOH	Start Of Heading	Début d'en-tête	**A**
STX	Start Of Text	Début de texte	**B**
ETX	End Of Text	Fin de texte	**C**
EOT	End Of Transmission	Fin de transmission	**D**
ENQ	Enquiry	Interrogation	**E**
ACK	Acknowledge	Accusé de (bonne) réception	**F**
BEL	Bell	Sonnerie	**G**
BS	Back Space	Espace arrière	**H**
HT	Horizontal Tabulation	Tabulation horizontale	**I**
LF	Line Feed	Interligne (saut de ligne)	**J**
VT	Vertical Tabulation	Tabulation verticale	**K**
FF	Form Feed	Présentation de feuille (saut page)	**L**
CR	Carriage Return	Retour chariot	**M**
SO	Shift Out	Code spécial	**N**
SI	Shift In	Code normal	**O**
DLE	Data Link Escape	Échappement de la liaison	**P**
DC1	Device Control 1	Commande dispositif 1 (XON)	**Q**
DC2	Device Control 2	Commande dispositif 2	**R**
DC3	Device Control 3	Commande dispositif 3	**S**
DC4	Device Control 4	Commande dispositif 4 (XOFF)	**T**
NAK	Negative Acknowledge	Accusé de non réception	**U**
SYN	Synchronous idle	Synchronisation	**V**
ETB	End of Transmission Block	Fin de bloc de transmission	**W**
CAN	Cancel	Annulation	**X**
EM	End of Medium	Fin du support	**Y**
SUB	Substitute	Substitution	**Z**
ESC	Escape	Échappement	**[**
FS	File Separator	Séparateur de fichiers	****
GS	Group Separator	Séparateur de groupe	**]**
RS	Record Separator	Séparateur d'enregistrement	**^**
US	Unit Separator	Séparateur d'unité	**_**
SP	Space	Espace	
DEL	Delete	Effacement (d'un caractère)	

2.4 Code EBCDIC :

EBCDIC = *Extensed Binary Coded Decimal Interchange Code*
Ce code est une extension du code BCD, utilisé par IBM il est pratiquement abandonné.

	Poid Fort															
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**
0	NUL	DLE			SP	&							{	}	/	0
1	SOH	DC1					/		a	j			A	J		1
2	STX	DC2							b	k	s		B	K	S	2
3	ETX	DC3		SYN					c	l	t		C	L	T	3
4									d	m	u		D	M	U	4
5	HT	NL	LF						e	n	v		E	N	V	5
6		BS	ETB						f	o	w		F	O	W	6
7	DEL		ESC	EOT					g	p	x		G	P	X	7
8	EOM	CAM							h	q	y		H	Q	Y	8
9		EM							i	r	z		I	R	Z	9
A					[	]	|	:							
B	VT					$		#								
C	FF	FS		DC4	<	‘	%	"								
D	CR	GS	ENQ	NAC	(	)	-	'								
E	SO	RS	ACK		+		>									
F	SI	US	BEL	SUB	!	@	?	=								

2.5 ISO/IEC 8859-1 ou Latin-1 :

Ce codage sur 8 bits est une extension de l’ASCII, il permet l’affichage des caractères accentués de plusieurs langues et de symboles courants. Un codage multi-langue similaire appelé codage 850 existait sous MSDOS.

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

Les contrôles supplémentaires sont :

BPH *Break Permitted Here*, NBH *No Break Here*, NEL *Next Line*, SSA *Start of Selected Area*, ESA *End of Selected Area*, HTS *cHaracter Tabulation Set*, HTJ *cHaracter Tabulation with Justification*, VTS *Vertical Tabulation Set*, PLD *Partial Line Down*, PLU *Partial Line Up*, RI *Reverse Line feed*, SS2 *Single shift 2*, SS3 *Single shift 3*, DCS *Device Control String*, PU1 *Private Use 1*, PU2 *Private Use 2*, STS *Set Transmit State*, CCH *Cancel CHaracter*, MW *Message Waiting*, SPA *Start of Private Area*, EPA *End of Private Area*, SDS *Start of String*, SCI *Single Character Introducer*, CSI *Control Sequence Introducer*, ST *String Terminator*, OSC *Operating System Command*, PM *Privacy Message*, APC *Application Program Command*, NbSP *No Break Space*.

XML utilise les codes ISO 8859-1 (latin1) ou parfois UTF-8

2.6 VIDEOTEX :

Ce codage a été créé pour les services TELETEL et seuls les services « Minitel » en France l'ont véritablement utilisé. La transmission est de type 7 bits, parité paire et le codage est basé sur l'ASCII.

Les 32 premiers codes (00 à 1F hexadécimal) sont des caractères de contrôles et les 96 suivants (20 à 7F hexadécimal) appartiennent à 3 grilles : G0 pour les lettres (idem ASCII), G1 pour des caractères semi-graphiques et G2 pour les accents et spéciaux. Des codes de contrôle permettent de changer de grille.

	Jeu C		Jeu G0						Jeu G1				Jeu G2			
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**2**	**3**	**6**	**7**	**2**	**3**	**4**	**6**
0	NUL	DLE		**0**	**@**	**P**	–	**p**						°		
1	SOH	Con	**!**	**1**	**A**	**Q**	**a**	**q**						±	`	
2	STX	REP	**"**	**2**	**B**	**R**	**b**	**r**							′	
3	ETX	SEP	**#**	**3**	**C**	**S**	**c**	**s**					£		^	
4	EOT	Coff	**$**	**4**	**D**	**T**	**d**	**t**					$			
5	ENQ	NAK	**%**	**5**	**E**	**U**	**e**	**u**								
6		SYN	**&**	**6**	**F**	**V**	**f**	**v**					#			
7	BEL	$	**'**	**7**	**G**	**W**	**g**	**w**								
8	BS	CAN	**(**	**8**	**H**	**X**	**h**	**x**						÷	¨	
9	HT	**SS2**	**)**	**9**	**I**	**Y**	**i**	**y**								
A	LF	SUB	*****	**:**	**J**	**Z**	**j**	**z**								Œ
B	VT	ESC	**+**	**;**	**K**	**[**	**k**	**{**							ç	
C	FF		**,**	**<**	**L**	****	**l**	**|**					←	¼		
D	CR		**-**	**=**	**M**	**]**	**m**	**}**					↑	½		
E	**SO**	RS	**.**	**>**	**N**	**^**	**n**	**~**					→	¾		
F	**SI**	US	**/**	**?**	**O**	**_**	**o**						↓			

SO *(Shift Out)* permet de passer de G0 à G1 ; SI *(Shift In)* de G1 à G0 ; SS2 *(Single Shift 2)* permet un accès temporaire à G2.
Exemple : la suite 19_H *,* 41_H *,* 65_H *affiche le caractère « è »*

2.7 UNICODE :

La création d'un jeu de caractères universel nécessite un codage plus long, le chinois compte en effet 50000 signes, le coréen 12000… On doit aussi tenir compte du sens de lecture… Unicode différencie caractère et glyphe : un glyphe peut être composé de plusieurs caractères (accentués par exemple) et un même caractère peut posséder plusieurs glyphes (lettre arabe se dessinant différemment selon sa position dans le mot par exemple).

Le développement d'Unicode est synchronisé avec l'ISO 10646. L'incorporation d'Unicode dans les applications ou les sites WEB permet de satisfaire les demandes multi-langues.

Actuellement l'ISO 10646 propose 2 encodages :

- UCS2 (16 bits) *(Universal Character Set)*
- UCS4 (32 bits)

Dans le codage 16 bits, 63486 combinaisons sont utilisées pour représenter directement des caractères et 2048 combinaisons indiquent un codage sur 32 bits (UCS4).

Par exemple, les codes Latin-1 sont codés de 0000 à 00FF, le grec de 0370 à 03FF, l'arabe de 0600 à 06FF…

ASCII/8859-1 Text	
A	0100 0001
S	0101 0011
C	0100 0011
I	0100 1001
I	0100 1001
/	0010 1111
8	0011 1000
8	0011 1000
5	0011 0101
9	0011 1001
-	0010 1101
1	0011 0001
	0010 0000
t	0111 0100
e	0110 0101
x	0111 1000
t	0111 0100

Unicode Text	
A	0000 0000 0100 0001
S	0000 0000 0101 0011
C	0000 0000 0100 0011
I	0000 0000 0100 1001
I	0000 0000 0100 1001
	0000 0000 0010 0000
天	0101 1001 0010 1001
地	0101 0111 0011 0000
	0000 0000 0010 0000
س	0000 0110 0011 0011
ل	0000 0110 0100 0100
ط	0000 0110 0011 0111
ﻡ	0000 0110 0100 0101
	0000 0000 0010 0000
α	0000 0011 1011 0001
≤	0010 0010 0111 0000
γ	0000 0011 1011 0011

Par contre, si les codes sont transportés par octets, l'ordre n'est pas précisé par la norme (BE *Big Endian* comme dans les réseaux ou LE *Little Endian* comme dans les processeurs ix86) !

UTF-8 : *Unicode Transformation Format*, est une méthode permettant de convertir des caractères Unicode 16 bits en code 8 bits (ASCII par exemple) afin d'être compatible avec l'existant. Les 128 caractères ASCII sont sur 1 octet, 1920 codes de 2 octets sont utilisés pour le grec… et chinois/japonais sont sur 3 octets.

infos sur www.unicode.org

2.8 Base 64 :

Base 64 est en réalité un **transcodage** permettant de transporter des données binaires quelconques (image, programme … et ceci le plus souvent dans un courrier électronique : RFC2045) dans un document texte, type ASCII 7bits par exemple. Pour transporter du binaire on pourrait utiliser l'hexadécimal mais il faut alors 2 caractères « 0…F » pour chaque octet, le rendement est alors assez faible (50%).

Codage Base 64 : 3 octets binaires (24 bits) sont découpés en 4 blocs de 6 bits et chaque bloc de 6 bits est codé par un caractère. Il faut donc 4 caractères imprimables pour coder 3 octets binaires quelconques (rendement 75%).

Binaire	Lettre	Binaire	Lettre	Binaire	Lettre	Binaire	Lettre
000000	A	**010000**	Q	**100000**	g	**110000**	**w**
000001	B	**010001**	R	**100001**	h	**110001**	**x**
000010	C	**010010**	S	**100010**	i	**110010**	**y**
000011	D	**010011**	T	**100011**	j	**110011**	**z**
000100	E	**010100**	U	**100100**	k	**110100**	**0**
000101	F	**010101**	V	**100101**	l	**110101**	**1**
000110	G	**010110**	W	**100110**	m	**110110**	**2**
000111	H	**010111**	X	**100111**	n	**110111**	**3**
001000	I	**011000**	Y	**101000**	o	**111000**	**4**
001001	J	**011001**	Z	**101001**	p	**111001**	**5**
001010	K	**011010**	a	**101010**	q	**111010**	**6**
001011	L	**011011**	b	**101011**	r	**111011**	**7**
001100	M	**011100**	c	**101100**	s	**111100**	**8**
001101	N	**011101**	d	**101101**	t	**111101**	**9**
001110	O	**011110**	e	**101110**	u	**111110**	**+**
001111	P	**011111**	f	**101111**	v	**111111**	**/**