

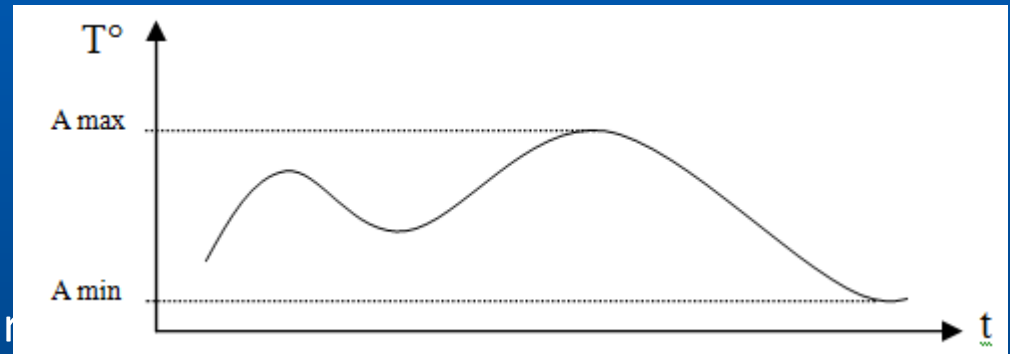
Numération

Informatique et Science du Numérique



La courbe ci-contre représente le signal délivré par un capteur de température.

Ce signal est analogique (il peut prendre n'importe quelle valeur). Pour être traité par l'ordinateur il doit être converti en un mot binaire.



00000000 correspond à 0 en décimal

00000001 correspond à 1 en décimal

00000010 correspond à 2 en décimal

...

11111111 correspond à 255 en décimal

Numération

Informatique et Science du Numérique



Base 2 ou binaire

Un mot est la plus petite unité manipulée par un microprocesseur. Sa taille s'exprime en bits (binary digit*) ou en octets (1 octet est constitué de 8 bits).

Les ordinateurs utilisent des données de 8, 16, 32 ou 64 bits :
Ex : 01101111 11001100 10101111 11110111 (32 bits ou 4 octets).

* Anglicisme Informatique : Chiffre, symbole graphique.

Numération

Informatique et Science du Numérique



A chaque bit du mot en base 2 correspond une puissance de 2.

	Poids							
	forts ← MSD							→ faibles LSD
Puissances	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Valeurs en décimal du poids	128	64	32	16	8	4	2	1
exemple	0	1	1	0	1	1	1	0
Valeur en décimal correspondante	$64+32+8+4+2=110$							

Le digit de poids le plus faible est appelé LSD (Least Significant Digit).

Exercice : Convertir le mot binaire %11001001 en décimal.

Numération

Informatique et Science du Numérique



Comment convertir 211 en binaire ?

Méthode par soustraction

On soustrait successivement la plus grande puissance de 2 et inférieure au nombre décimal à convertir.

$211 - 128 = 83$, puis $83 - 64 = 19$, puis $19 - 16 = 3$, puis $3 - 2 = 1$, reste 1

211 est donc composé de $128+64+16+2+1 = \%11010011$

Exercice : Convertir le nombre décimal 195 en binaire.

Numération

Informatique et Science du Numérique



Méthode par division

On divise par 2 autant de fois que nécessaire pour obtenir un quotient nul et l'on écrit les restes dans l'ordre inverse où ils ont été obtenus.

$$\begin{array}{r} 211 \ 2 \\ \underline{1} \ 105 \ 2 \\ \quad \underline{1} \ 52 \ 2 \\ \quad \quad \underline{0} \ 26 \ 2 \\ \quad \quad \quad \underline{0} \ 13 \ 2 \\ \quad \quad \quad \quad \underline{1} \ 6 \ 2 \\ \quad \quad \quad \quad \quad \underline{0} \ 3 \ 2 \\ \quad \quad \quad \quad \quad \quad \underline{1} \ 1 \end{array}$$

Exercice : Convertir le nombre décimal 103 en binaire.

Numération

Informatique et Science du Numérique



Base 16 ou Hexadécimal

Le codage en Hexadécimal (système de numération à base 16) permet de faciliter la représentation d'un mot binaire.

Ainsi l'octet %11010011 s'écrit D3 en hexadécimal

Ou encore : \$D3 ou 0xD3 ou D3h ...

A chaque quartet du mot binaire correspond un symbole en hexadécimal

Exercice : Convertir %01101011 en hexadécimale.

Binaire %	Hexadécimal \$ ou Ox	Décimal
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010	A	10
1011	B	11
1100	C	12
1101	D	13
1110	E	14
1111	F	15

Numération

Informatique et Science du Numérique



Conversion hexa → décimal :

à chaque digit du mot en hexa correspond une puissance de 16

	Poids				
	forts ←				→ faibles
Puissances	16^4	16^3	16^2	16^1	16^0
Valeurs en décimal du poids	65536	4096	256	16	1
Digits	0	0	0	6	B
Valeur en décimal	$6*16+ 11*1= 107$				

Exercice : Conv

Numération

Informatique et Science du Numérique



Base 8 ou octal

Le système octal possède l'avantage de ne pas requérir de symbole supplémentaire pour ses chiffres.

A partir de la numération binaire, on groupe les chiffres consécutifs en triplets.

Ainsi, la représentation binaire du nombre décimal 74 est 1001010 , que l'on groupe en $(00)1\ 001\ 010$ s'écrit 112 en octal.

Ou encore : 0112

Exercice : Convertir $0x67$ en octal.

Numération

Informatique et Science du Numérique



Code BCD (binaire codé décimal)

Il permet d'écrire rapidement un mot binaire en décimal.
A chaque chiffre du nombre décimal on lui fait correspondre le quartet binaire.

Exemple : le décimal 211 s'écrit

- 0010 0001 0001 en BCD
- 11010011 en binaire

Exercice : Ecrire 254 en BCD puis en binaire puis en HEXA

Décimal	Binaire
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

Numération

Informatique et Science du Numérique



Le jeu de caractères codés ASCII [aski:] (American Standard Code for Information Interchange, « Code américain normalisé pour l'échange d'information ») permet de représenter sur **7 bits** un caractère :

G est codé \$47
ce qui donne
% 100 0111

Table des 128 caractères ASCII

PDF : fr en v · d · m	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
000	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
001	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
002	SP	!	"	#	\$	%	&	'	()	*	+	,	-	.	/
003	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
004	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
005	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
006	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
007	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL

Exercice : donner le code ASCII de @

Numération

Informatique et Science du Numérique



Avec le développement des protocoles de communication basés sur 8 bits et les logiciels écrits en langues occidentales, il est apparu plus simple d'avoir à coder les caractères sur un seul octet.

On définit ainsi une table de jeu de caractères étendus en fonction de chaque pays.

ISO-8859-1																
	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
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	ı	ç	£	¤	¥	¦	§	¨	©	*	«	¬	¦	@	—
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	ø	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

Jeu de caractères étendus pour la France

Numération

Informatique et Science du Numérique



UTF-8 (Universal character set Transformation Format 8 bits) est un codage de caractères informatique conçu pour coder l'ensemble des caractères internationaux en restant compatible avec la norme ASCII.

Type	Caractère	Point de code (hexadécimal)	Valeur scalaire		Codage UTF-8	
			décimal	binaire	binaire	hexadécimal
Contrôles	[NUL]	U+0000	0	0000000	00000000	00
	[US]	U+001F	31	0011111	00011111	1F
Texte	[SP]	U+0020	32	0100000	00100000	20
	A	U+0041	65	1000001	01000001	41
	~	U+007E	126	1111110	01111110	7E
Contrôles	[DEL]	U+007F	127	1111111	01111111	7F
	[PAD]	U+0080	128	00010 000000	11000010 10000000	C2 80
	[APC]	U+009F	159	00010 011111	11000010 10011111	C2 9F
Texte	[NBSP]	U+00A0	160	00010 100000	11000010 10100000	C2 A0
	é	U+00E9	233	00011 101001	11000011 10101001	C3 A9
	☹	U+07FF	2047	11111 111111	11011111 10111111	DF BF
	☹	U+0800	2048	0000 100000 000000	11100000 10100000 10000000	E0 A0 80
	€	U+20AC	8 364	0010 000010 101100	11100010 10000010 10101100	E2 82 AC
	☹	U+D7FF	55 295	1101 011111 111111	11101101 10011111 10111111	ED 9F BF

Numération

Informatique et Science du Numérique



Nombres signés/non signés

1 octet : **0** 0 0 0 0 0 0 0

Bit de poids fort = bit de signe

- 0 : nombre +
- 1 : nombre -

$$\mathbf{0} 0 0 0 0 0 0 0 = 0$$

$$\mathbf{0} 0 0 0 0 0 0 1 = 1$$

...

$$\mathbf{0} 1 1 1 1 1 1 1 = 127$$

$$\mathbf{1} 0 0 0 0 0 0 0 = -128$$

$$\mathbf{1} 0 0 0 0 0 0 1 = -127 = -128 + 1$$

...

$$\mathbf{1} 1 1 1 1 1 1 1 = -1 = -128 + 127$$

Numération

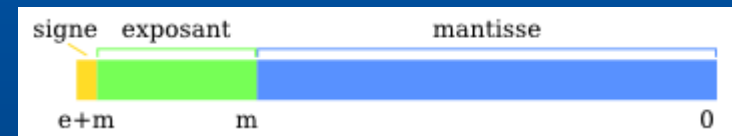
Informatique et Science du Numérique



Nombres flottants

$$(-1)^S \times M \times 2^{(E-d)}$$

- S : bit de signe (0 | 1)
- M : **mantisse** (23 | 52 bits)*
- E : **exposant** (8 | 11 bits)**
- d : décalage = $2^E - 1$ (127 | 2043)



En faisant varier l'exposant, on fait « flotter » la virgule

Ex : 00111110 10110000 00000000 00000000

$$(-1)^0 \times (1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-3}) \times 2^{(125-127)} = 1,375 \times 2^{-2} = 0,34375$$

* simple ou double précision

** dénormalisé si tout à 0 ou 1