Spécialité NSI

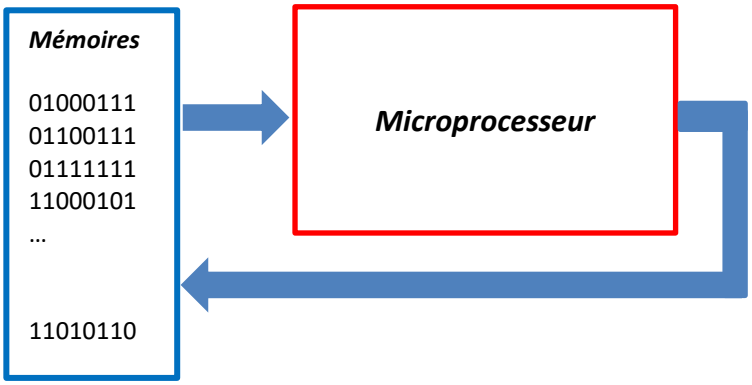
LYCÉE de l'ÉLORN
le choix de votre réussite

Fiche de cours
Représentation des données
Numération et codage

Documents disponibles sur sciencesdelingenieur.fr ▸ menu ressources pédagogiques ▸ Spécialité NSI

Pourquoi ce cours ?

Prenez conscience qu'un système numérique traite uniquement des données au format binaire. Les grandeurs numériques sont généralement manipulées par paquets de 8 bits (mots de 8, 16, 32, 64 ou 128 bits) suivant la puissance du microprocesseur.



1- La numération

1.1 Rappel sur le système décimal

La base 10 utilise 10 symboles différents : 0, 1, 2, 3, 4, 5, 6, 7, 8 et 9. Un nombre N (entier positif) exprimé dans ce système de numération est défini par le polynôme :

$$N = a_n \times 10^n + a_{n-1} \times 10^{n-1} + \dots + a_1 \times 10^1 + a_0 \times 10^0$$ (où a_n est un chiffre de rang n)

Exemple : $N = 2023$

$N = 2 \times 10^3 + 0 \times 10^2 + 2 \times 10^1 + 3 \times 10^0$

Les puissances de 10 sont appelées les **poids** ou les **valeurs de position**. Le poids est égal à la base élevée à la puissance de son **rang**.

	Unités	***Dizaines***	***Centaines***	***Milliers***	***10×Milliers***	***100×Milliers***
Chiffre	a_0	a_1	a_2	a_3	a_4	a_5
Rang	0	1	2	3	4	5
Poids	10^0	10^1	10^2	10^3	10^4	10^5

1.2 Système binaire (binaire pur)

Le système binaire est le système de base 2, il utilise deux symboles différents, le 0 et le 1. Chacun des éléments est appelé bit (contraction de binary digit). Nous préciserons la notation d'un nombre binaire avec le préfixe 0b ou en notant entre parenthèse avec en indice la valeur 2.

Dans ce système, le poids est une puissance de 2. Pour N = 0b110110 ou $(110110)_2$ le polynôme s'écrit :

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

$N = (54)_{10}$

Rappel des puissances de 2 :

n	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
2^n	1	2	4	8	16	32	64	128	256	512	1024	2048	4096	8192	16384	32768	65536

☞ Convertir en décimal les nombres :

N = 0b1011 = ( )$_{10}$
N = 0b00001011 = ( )$_{10}$
N = 0b11110000 = ( )$_{10}$
N = 0b11111110 = ( )$_{10}$
N = 0b11111111 = ( )$_{10}$
N = 0b1111111111 = ( )$_{10}$

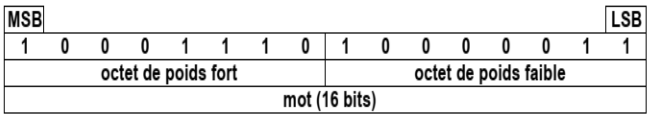
1.3 Format des nombres binaire, définitions :

Octet (byte en anglais) : Nombre binaire formé de 8 bits. *ex : 0b00000010, 0b10101111*

Mot (word) : En dehors de l'unité de transfert usuelle (octet), des regroupements plus importants sont couramment utilisés : le mot de 16 bits = 2 octets (**word)**, le mot de 32 bits = 4 octets (**double word**), et le mot de 64 bits = 8 octets (**quad word**).

MSB, LSB : Dans un mot binaire, le bit situé le plus à gauche est le bit le plus significatif. Il sera désigné comme étant le MSB (Most Significant Bit).Celui situé le plus à droite est le bit le moins significatif, LSB (Least Significant Bit).

Remarque : Pour faciliter les manipulations des mots ont peut le diviser en plusieurs octets. Le mot ci-dessous est constitué de 2 octets, celui situé à gauche sera désigné comme étant l'octet de poids fort, et celui situé à droite, l'octet de poids faible.



Quartet : nombre binaire formé de 4 éléments binaires. *Ex : 0001, 1001 ,1111*

Capacités : La capacité en octets des différents constituants tels que les circuits mémoires ou les disques durs est souvent importante. Il devient indispensable d'utiliser **des unités multiples de l'octet.** Historiquement les préfixes « kilo », « méga », « giga » ... représentaient cette capacité naturellement exprimée par une puissance de 2 mais de nouveaux noms ont été créés pour noter ces valeurs. On parle désormais de «kibi», « mébi », « gibi » …

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 kibi-octet = 1 048 576 octets
Gio (GiB) = Gibi-octet (GibiByte) = 2^{30} octets = 1024 Mébi-octet = 1 073 741 824 octets
Tio (TiB) = Tébi-octet (TebiByte) = 2^{40} octets = 1024 Gibi-octet

k, M, G, T, … = multiple du système international
b=bit, B=Byte, bi=binary

1.4 Etendue des valeurs en binaire

En utilisant **n** bits, on peut former $\mathbf{2^n}$ nombres différents et le plus grand d'entre eux est égal à $\mathbf{(2^n-1)}$.

Par exemple si n = 8, $N_{max} = (2^8-1) = 255$
On peut former 256 nombres évoluant de 0 0b00000000 à 255 0b11111111.

☞ Quelle est l'étendue d'un nombre définie sur 11 bits ?

☞ On souhaite dénombrer 2 millions de valeurs, quelle doit être la grandeur du mot binaire utilisé ?

1.5 Système hexadécimal

Le système hexadécimal est de **base 16** et utilise donc 16 symboles différents. On y trouvera les dix premiers chiffres décimaux : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 et les 6 premières lettres de l'alphabet : A, B, C, D, E, F qui correspondent respectivement aux nombres décimaux de 10 à 15.

Base 10	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Base 16	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**	**10**	**11**	**12**	**13**
Base 2	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111				

Nous préciserons la notation d'un nombre hexadécimal avec le préfixe 0x ou en notant entre parenthèse avec en indice la valeur 16.

Exemple : $N = 0xAC53 = (AC53)_{16}$
$N = A \times 16^3 + C \times 16^2 + 5 \times 16^1 + 3 \times 16^0$ *Le poids est une puissance de 16*
$N = 10 \times 16^3 + 12 \times 16^2 + 5 \times 16^1 + 3 \times 16^0$
$N = 44115$

Etendue des valeurs en hexadécimale : En utilisant **n** symboles on peut former $\mathbf{16^n}$ nombres différents et le plus grand d'entre eux est égal à $\mathbf{(16^n-1)}$.

☞ *Donnez la valeur en décimale des nombres suivants :*

N = 0x1234 = N = 0xABCD = N = 0xFFFF =

2 Changement de base

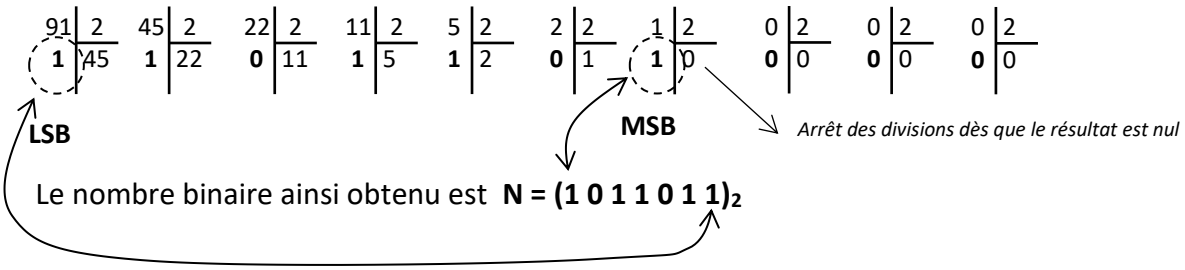
2.1 Conversion d'un nombre décimal en un nombre d'un système d'une autre base :

Un nombre **N** étant donné en base 10, cherchons à l'écrire dans un système de base **b**.

1ère méthode : la division successive

Nous divisons le nombre décimal à convertir par la base b et nous conservons le reste (division entière). Le quotient obtenu est ainsi successivement divisé tant qu'il n'est pas nul. Les restes successifs sont écrits, en commençant par le dernier, de la gauche vers la droite pour former l'expression de N dans le système de base b.

Exemple : Conversion de N = 91 en un nombre du système binaire (b=2).



☞ Avec cette méthode, convertissez en binaire puis en hexadécimale 29, 210 et 2023.

2nd méthode : Approximation successive *(méthode du « compte est bon »)*

On reconstitue le mot binaire en utilisant les poids rencontrés dans la base souhaitée.

☞ Avec cette méthode, convertissez en binaire 42 et 203

Poids	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
	128	**64**	**32**	**16**	**8**	**4**	**2**	**1**
binaire								
binaire								

2.2 Passerelle entre la notation binaire et hexadécimale

Conversion d'un nombre hexadécimal en un nombre binaire :
Chaque symbole du nombre hexadécimal est remplacé par son équivalent écrit sur 4 bits dans le système binaire.

Exemple : $N = (B\ F\ 8)_{16}$
$N = (1011\ \ 1111\ \ 1000)_2$
B F 8

Conversion d'un nombre binaire en un nombre hexadécimal :
C'est l'inverse de la précédente. Il faut donc regrouper les bits du nombre par quartets en commençant par la droite, puis chaque groupe est remplacé par le symbole hexadécimal correspondant à sa valeur.

Exemple : $N = (100001101111)_2$
$N = (1000\ \ 0110\ \ 1111)_2 = (86F)_{16}$

2.3 Fonction de conversion en Python

Exemple sur la console :

Base10 -> base 2	bin(55) donne 0b110111
Base10 -> base 16	hex(55) donne 0x37
Base2 -> base 10	int('110111',2) ou 0b110111 donne 55
Base16 -> base 10	int('37',16) ou 0x37 donne 55

3- Le codage

3.1 Le code ASCII

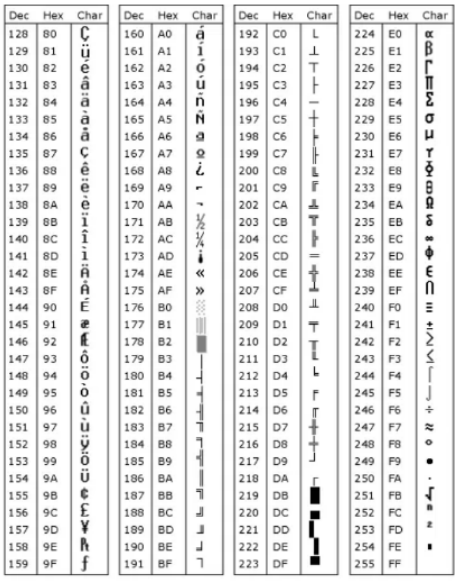
Le jeu de caractères codés ASCII (American Standard Code for Information Interchange) est la norme de codage de caractères en informatique la plus connue, la plus ancienne et la plus compatible.

Le code ASCII initial est codé sur 7 bits (valeurs 0 à 127), il permet de définir :

- des caractères imprimables universels : lettres minuscules et majuscules, chiffres, symboles,…
- des codes de contrôle non imprimables : indicateur de saut de ligne, de fin de texte, codes de contrôle de périphériques, …

Decimal	Hexadecimal	Binary	Octal	Char
0	0	0	0	*[NULL]*
1	1	1	1	*[START OF HEADING]*
2	2	10	2	*[START OF TEXT]*
3	3	11	3	*[END OF TEXT]*
4	4	100	4	*[END OF TRANSMISSION]*
5	5	101	5	*[ENQUIRY]*
6	6	110	6	*[ACKNOWLEDGE]*
7	7	111	7	*[BELL]*
8	8	1000	10	*[BACKSPACE]*
9	9	1001	11	*[HORIZONTAL TAB]*
10	A	1010	12	*[LINE FEED]*
11	B	1011	13	*[VERTICAL TAB]*
12	C	1100	14	*[FORM FEED]*
13	D	1101	15	*[CARRIAGE RETURN]*
14	E	1110	16	*[SHIFT OUT]*
15	F	1111	17	*[SHIFT IN]*
16	10	10000	20	*[DATA LINK ESCAPE]*
17	11	10001	21	*[DEVICE CONTROL 1]*
18	12	10010	22	*[DEVICE CONTROL 2]*
19	13	10011	23	*[DEVICE CONTROL 3]*
20	14	10100	24	*[DEVICE CONTROL 4]*
21	15	10101	25	*[NEGATIVE ACKNOWLEDGE]*
22	16	10110	26	*[SYNCHRONOUS IDLE]*
23	17	10111	27	*[ENG OF TRANS. BLOCK]*
24	18	11000	30	*[CANCEL]*
25	19	11001	31	*[END OF MEDIUM]*
26	1A	11010	32	*[SUBSTITUTE]*
27	1B	11011	33	*[ESCAPE]*
28	1C	11100	34	*[FILE SEPARATOR]*
29	1D	11101	35	*[GROUP SEPARATOR]*
30	1E	11110	36	*[RECORD SEPARATOR]*
31	1F	11111	37	*[UNIT SEPARATOR]*
32	20	100000	40	*[SPACE]*
33	21	100001	41	!
34	22	100010	42	"
35	23	100011	43	#
36	24	100100	44	$
37	25	100101	45	%
38	26	100110	46	&
39	27	100111	47	'
40	28	101000	50	(
41	29	101001	51	)
42	2A	101010	52	*
43	2B	101011	53	+
44	2C	101100	54	,
45	2D	101101	55	-
46	2E	101110	56	.
47	2F	101111	57	/
48	30	110000	60	0
49	31	110001	61	1
50	32	110010	62	2
51	33	110011	63	3
52	34	110100	64	4
53	35	110101	65	5
54	36	110110	66	6
55	37	110111	67	7
56	38	111000	70	8
57	39	111001	71	9
58	3A	111010	72	:
59	3B	111011	73	;
60	3C	111100	74	<
61	3D	111101	75	=
62	3E	111110	76	>
63	3F	111111	77	?
64	40	1000000	100	@
65	41	1000001	101	A
66	42	1000010	102	B
67	43	1000011	103	C
68	44	1000100	104	D
69	45	1000101	105	E
70	46	1000110	106	F
71	47	1000111	107	G
72	48	1001000	110	H
73	49	1001001	111	I
74	4A	1001010	112	J
75	4B	1001011	113	K
76	4C	1001100	114	L
77	4D	1001101	115	M
78	4E	1001110	116	N
79	4F	1001111	117	O
80	50	1010000	120	P
81	51	1010001	121	Q
82	52	1010010	122	R
83	53	1010011	123	S
84	54	1010100	124	T
85	55	1010101	125	U
86	56	1010110	126	V
87	57	1010111	127	W
88	58	1011000	130	X
89	59	1011001	131	Y
90	5A	1011010	132	Z
91	5B	1011011	133	[
92	5C	1011100	134	\
93	5D	1011101	135	]
94	5E	1011110	136	^
95	5F	1011111	137	_
96	60	1100000	140	`
97	61	1100001	141	a
98	62	1100010	142	b
99	63	1100011	143	c
100	64	1100100	144	d
101	65	1100101	145	e
102	66	1100110	146	f
103	67	1100111	147	g
104	68	1101000	150	h
105	69	1101001	151	i
106	6A	1101010	152	j
107	6B	1101011	153	k
108	6C	1101100	154	l
109	6D	1101101	155	m
110	6E	1101110	156	n
111	6F	1101111	157	o
112	70	1110000	160	p
113	71	1110001	161	q
114	72	1110010	162	r
115	73	1110011	163	s
116	74	1110100	164	t
117	75	1110101	165	u
118	76	1110110	166	v
119	77	1110111	167	w
120	78	1111000	170	x
121	79	1111001	171	y
122	7A	1111010	172	z
123	7B	1111011	173	{
124	7C	1111100	174	|
125	7D	1111101	175	}
126	7E	1111110	176	~
127	7F	1111111	177	*[DEL]*

Un code étendu existe pour les principaux caractères latins.



☞ À l'aide de la table ASCII ci-dessus, coder la chaîne de caractère « *Landerneau 2020* » en décimale puis en hexadécimale.

3.2 UNICODE et son encodage

Pour coder de manière universelle l'ensemble des symboles utilisés quelle que soit la langue (anglais, français, grec, chinois…) il est possible d'attribuer un identifiant numérique à tout caractère ou symbole (émoticônes). Ce type de codage est proposé par la norme **Unicode** dont la plage définie permet d'attribuer jusqu'à 1114112 caractères

Chaque symbole d'écriture est représenté par un nom et une valeur hexadécimale de 4 à 6 caractères préfixée par « **U+** ».

A	**« lettre majuscule latine A »**	**U+0041**
é	**« lettre minuscule latine e accent aigu »**	**U+00E9**
€	**« symbole euro »**	**U+20AC**
	**« Emoji rolling on the floor laughing»**	**U+1F923**

Encodage

Pour stocker sur un support informatique un texte constitué de caractères Unicode il faut choisir un procédé transformant chaque définition Unicode en une suite d'octets. C'est ce que l'on appelle le **processus d'encodage.** Actuellement le système d'encodage le plus couramment utilisé est l'**UTF-8** (**U**nicode **T**ransformation **F**ormat)

Définition du nombre d'octets utilisés en UTF-8

1 octet codant 1 à 7 bits → **0**xxxxxxx
2 octets codant 8 à 11 bits → **110**xxxxx **10**xxxxxx
3 octets codant 12 à 16 bits → **1110**xxxx **10**xxxxxx **10**xxxxxx
4 octets codant 17 à 21 bits → **11110**xxx **10**xxxxxx **10**xxxxxx **10**xxxxxx

Exemple d'encodage en UTF-8

Type	Caractère	Point de code (hexadécimal)	Valeur scalaire		Codage UTF-8	
			décimal	binaire	binaire	hexadécimal
Contrôles	[NUL]	U+0000	0	*0000000*	*0*0000000	00
	[US]	U+001F	31	*0*011111	*0*0011111	1F
Texte	[SP]	U+0020	32	*0*100000	*0*0100000	20
	A	U+0041	65	1000001	*0*1000001	41
	~	U+007E	126	1111110	*0*1111110	7E
Contrôles	[DEL]	U+007F	127	1111111	*0*1111111	7F
	[PAD]	U+0080	128	*00*010 000000	**11***000*010 **10**000000	C2 80
	[APC]	U+009F	159	*00*010 011111	**11***000*010 **10**011111	C2 9F
Texte	[NBSP]	U+00A0	160	*00*010 100000	**11***000*010 **10**100000	C2 A0
	é	U+00E9	233	*00*011 101001	**11***000*011 **10**101001	C3 A9
	[illegible]	U+07FF	2047	11111 111111	**11**011111 **10**111111	DF BF
	[illegible]	U+0800	2048	*0000* 100000 000000	**1110***0000* **10**100000 **10**000000	E0 A0 80
	€	U+20AC	8 364	*0*010 000010 101100	**1110**0010 **10**000010 **10**101100	E2 82 AC
	[illegible]	U+D7FF	55 295	1101 011111 111111	**1110**1101 **10**011111 **10**111111	ED 9F BF

Ex : Pour le symbole € codé en unicode U+20AC = $(8364)_{10}$ ou $(10\ 0000\ 1010\ 1100)_2$ soit 14 bits
3 octets sont utilisés : 11100010 **10**000010 **10**101100 soit E2 82 AC

☞ Donnez le code UTF-8 de l'émoticône « rolling on the floor laughing» U+1F923

https://www.compart.com/fr/unicode/U+1F923

Remarque utile :

Pour insérer un caractère ASCII sur PC , maintenez la touche ALT enfoncée tout en tapant le code du caractère.
Pour insérer un caractère Unicode, appuyez sur ALT et sur X, tapez le code du caractère.