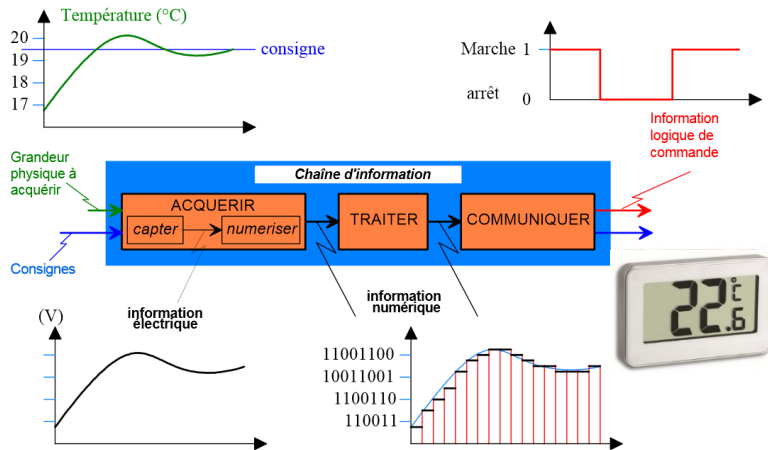


CODAGE DE L'INFORMATION

1. NATURE DE L'INFORMATION



La grandeur physique est la température dans la maison.

Un capteur traduit cette grandeur physique en un signal électrique proportionnel (information analogique).

Un convertisseur traduit l'information analogique en une information numérique (nombre binaire codé sur plusieurs fils). Cette information est traitée pour le calcul et l'affichage de la température.

L'utilisateur règle la température voulue. Cette consigne est comparée à la température mesurée pour produire un signal logique qui active ou stoppe le chauffage

2. GENERALITES

2.1. POIDS FORT ET POIDS FAIBLE

Dans un nombre, le poids faible correspond au chiffre ayant l'influence la plus petite (c'est le chiffre le plus à droite). Dans un nombre, le poids fort correspond au chiffre ayant l'influence la plus grande (c'est le chiffre le plus à gauche).

Ex: dans le nombre décimal 4 589 237 : le chiffre de poids fort est : ____, le chiffre de poids faible est : ____

2.2. BASE D'UN NOMBRE

Dans l'étude des systèmes automatisés il est souvent nécessaire d'utiliser plusieurs bases de numérotation pour un même nombre. Les nombres sont convertis le plus souvent en trois bases.

2.3. BASE 10 : DECIMALE

C'est la manière de numérotter la plus courante, mais pas la plus pratique à utiliser en informatique.

Un nombre en base 10 est représenté sans préfixe. Ex 2356.

Il se décompose comme suit : $2 \times \text{Base}^{\text{Rang}} + 3 \times \text{Base}^{\text{Rang}} + 5 \times \text{Base}^{\text{Rang}} + 6 \times \text{Base}^{\text{Rang}} = 2 \times 10^3 + 3 \times 10^2 + 5 \times 10^1 + 6 \times 10^0$

2.4. BASE 2 : BINAIRE

Dans la base 2 tous les nombres sont uniquement formés par deux chiffres 0 et 1.

Les minuscules interrupteurs électroniques des microprocesseurs ne connaissent que deux états, ouvert ou fermé (0 ou 1). Le langage interne de tout système automatisé est donc basé sur le système binaire. Il existe plusieurs variétés de codes binaires, les plus utilisés seront étudiés dans cette leçon. Un nombre écrit en binaire naturel est représenté avec un préfixe 0b ou 0B. Ex 0b110011

Un nombre écrit en binaire BCD est représenté avec un préfixe (BCD). Ex (BCD)110011

2.5. BASE 16 : HEXADECIMALE

Dans la base 16 il faut 16 signes différents pour représenter un chiffre 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F. les lettres A à F ont été utilisées pour compléter ce code.

Ce code est très pratique à utiliser en informatique

Un nombre écrit en hexadécimal est représenté avec un préfixe 0x ou 0X

Ex 0x1AC2BF

REMARQUE :

Il est très important de connaître les différents préfixes car les nombres 10110, 0b10110, 0x10110 et (BCD)10110 sont tous différents.

3. LE CODAGE DES NOMBRES NATURELS

3.1. VOCABULAIRE

Une variable binaire est appelé **BIT**. Un bit peut prendre la valeur 0 ou 1

Un **QUARTET** est un nombre formé de 4 bit. (Il peut prendre 2^4 combinaisons soit 16 valeurs)

Un **OCTET** (BYTE) est un nombre formé de 8 bit. (Il peut prendre _____ valeurs)

Un **MOT** (WORD) est un nombre formé de 16 bit. (Il peut prendre _____ valeurs)

Un **MOT LONG** (DWORD) est un nombre formé de 32 bit. (Il peut prendre _____ valeurs)

Longtemps l'informatique s'est singularisée par l'utilisation de différentes valeurs pour les unités du système international. Ainsi beaucoup d'informaticiens ont appris que 1 kilooctet valait 1024 octets. Or, depuis décembre 1998, l'organisme international IEC a statué sur la question. Voici donc les unités standardisées :

- Un kilooctet (ko ou kB) = 1000 octets
- Un Mégaoctet (Mo ou MB) = 1000 ko = 1 000 000 octets
- Un Gigaoctet (Go ou GB) = 1000 Mo = 1 000 000 000 octets
- Un Téraoctet (To) = 1000 Go = 1 000 000 000 000 octets

Attention ! De nombreux logiciels (parfois même certains systèmes d'exploitation) utilisent toujours la notation antérieure à 1998 pour laquelle :

- Un kilooctet (ko) = 2^{10} octets = 1024 octets
- Un Mégaoctet (Mo) = 2^{20} octets = 1024 ko = 1 048 576 octets ...

L'IEC a également défini le kilo binaire (kibi), le méga binaire (Mébi), le giga binaire (Gibi), le téra binaire (Tebi).

Voici leurs définitions :

- Un kibioctet (kio ou kiB) vaut 2^{10} = 1024 octets
- Un Mébioctet (Mio ou MiB) vaut 2^{20} = 1 048 576 octets
- Un Gibioctet (Gio ou GiB) vaut 2^{30} = 1 073 741 824 octets
- Un Tébioctet (Tio ou TiB) vaut 2^{40} = 1 099 511 627 776 octets

3.2. LE CODE BINAIRE NATUREL (BASE 2)

C'est le code le plus simple. Il est utilisé pour le calcul par les microprocesseurs

3.2.1. PASSAGE DU CODE BINAIRE NATUREL AU CODE DECIMAL

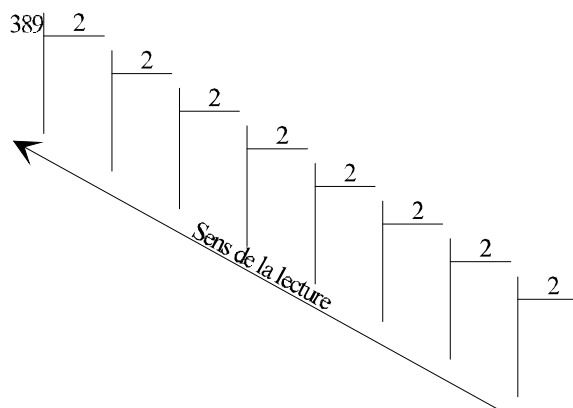
Il faut utiliser les puissances de deux.

poids	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
	256	128	64	32	16	8	4	2	1
0b	1	1	0	0	0	0	1	0	1

$$0b110000101 = 1 \times 256 + 1 \times 128 + 0 \times 64 + 0 \times 32 + 0 \times 16 + 0 \times 8 + 1 \times 4 + 0 \times 2 + 1 \times 1 = \text{-----}$$

3.2.2. PASSAGE DU CODE DECIMAL AU CODE BINAIRE NATUREL

a) Le transcodage se réalise par des divisions par 2 successives



$$389 = 0b \text{-----}$$

b) Transcodage par addition : exemple $67 = 64 + 3 = 64 + 2 + 1$

poids	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
	256	128	64	32	16	8	4	2	1
0b									

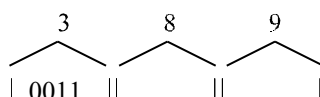
CODE BINAIRE NAT.				Code déc.
d	c	b	a	
0	0	0	0	
0	0	0	1	
0	0	1	0	
0	0	1	1	
0	1	0	0	
0	1	0	1	
0	1	1	0	
0	1	1	1	
1	0	0	0	
1	0	0	1	
1	0	1	0	
1	0	1	1	
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	

3.3. LE CODE BINAIRE CODE DECIMAL (BCD)

C'est le code utilisé dans l'électronique digitale. Il est constitué par une juxtaposition du code binaire limité à 9 (0b1001)

3.4. PASSAGE DU CODE DECIMAL AU CODE BCD

Chaque chiffre décimal est codé individuellement en son équivalent en binaire naturel sur 4 bits



$$389 = (\text{BCD}) \text{-----}$$



CODE BCD				Code déc.
d	c	b	a	
0	0	0	0	
0	0	0	1	
0	0	1	0	
0	0	1	1	
0	1	0	0	
0	1	0	1	
0	1	1	0	
0	1	1	1	
1	0	0	0	
1	0	0	1	

3.5. PASSAGE DU CODE BCD AU CODE DECIMAL

La méthode de transcodage est la même que précédemment, mais dans le sens contraire. On groupe le code BCD par quartets (à partir de la droite) et on converti chaque quartet en un chiffre décimal.

(BCD)1 000 0110 1001 = _____

(BCD)11 1001 0110 = _____



CODE BINAIRE NAT.				Code Hex.
d	c	b	a	
0	0	0	0	
0	0	0	1	
0	0	1	0	
0	0	1	1	
0	1	0	0	
0	1	0	1	
0	1	1	0	
0	1	1	1	
1	0	0	0	
1	0	0	1	
1	0	1	0	
1	0	1	1	
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	

3.6. LE CODE HEXADECIMAL (BASE 16)

Le code binaire est si éloigné de la pensée humaine que les informaticiens ont dû imaginer un moyen plus simple pour représenter des nombres binaires. En effet dans les microprocesseurs toutes les informations sont codées en binaire. Les langages qui permettent de programmer ces microprocesseurs sont appelés. Il est impensable de programmer en binaire naturel. Le code hexadécimal est un moyen de compacter le code en binaire naturel afin que l'écriture des programmes soit plus lisible.

3.6.1. PASSAGE DE L'HEXADECIMAL AU CODE BINAIRE

Chaque chiffre décimal est codé individuellement en son équivalent en binaire naturel sur 4 bits

0x 4 C 8 F

0x4C8F = 0b _____

RGB Hex Triplet Color Chart			
E-mail-ware...What a concept!			
FFFFFF	FFCCFF	FF99FF	FF66FF
FFFFCC	FFCCCC	FF99CC	FF66CC
FFFF99	FFCC99	FF9999	FF6699
EEEEEE	FFFF66	FFCC66	FF9966
			FF6666

3.6.2. PASSAGE DU CODE BINAIRE NATUREL A L'HEXADECIMAL

La méthode de transcodage est la même que précédemment mais dans le sens contraire. On groupe le code binaire naturel par quartets (à partir de la droite) et on converti chaque quartet en un chiffre hexadécimal.

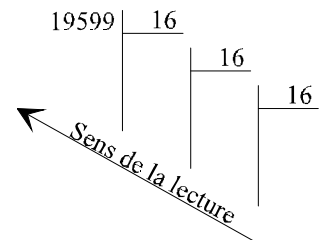
0b1100 0110 1100 = 0x _____

0b11 1001 1110 = 0x _____

3.6.3. PASSAGE DU CODE DECIMAL AU CODE HEXADECIMAL

Le transcodage se réalise par des divisions successives par 16

19599 = 0x _____



3.6.4. PASSAGE DU CODE HEXADECIMAL AU CODE DECIMAL

Il faut utiliser les puissances de 16

0x4C8F = ____ x 16³ + ____ x 16² + ____ x 16¹ + ____ x 16⁰

0x4C8F = _____

4. LE CODAGE DES CARACTERES

En informatique il est nécessaire d'utiliser des caractères (pour un texte). Il est donc impératif d'utiliser un code convertissant les caractères en binaire naturel (ou hexadécimal). Le code utilisé est le code ASCII (American Standard Code for Information Interchange). Avant d'être utilisé en informatique ce code était utilisé pour la communication de messages à distance (TELEX) C'est pour cela qu'il comporte des caractères non imprimables (de 0x0 à 0x1F). Ces codes non imprimables sont des protocoles de mise en page.

Ex: 0x7 correspond à un "bip" sonore (<BEL>)

0xD correspond au retour chariot

0xA correspond au passage à la ligne suivante

Un code ASCII est codé sur 8 bits

Ex: (ASCII) 0x35 = _____

(ASCII) 0x52 = _____

(ASCII) 35 = _____



Hexa-decimal	0	1	2	3	4	5	6	7
0	<NUL>			0	@	P	~	p
1		<DC1>	!	1	A	Q	a	q
2		<DC2>	"	2	B	R	b	r
3		<DC3>	#	3	C	S	c	s
4		<DC4>	\$	4	D	T	d	t
5			%	5	E	U	e	u
6			&	6	F	V	f	v
7	<BEL>			7	G	W	w	
8	<BS>	<CAN>	(	8	H	X	h	x
9	<HT>		)	9	I	Y	i	y
A	<LF>		*	:	J	Z	j	z
B	<VT>	<ESC>	+	;	K	[	k	{
C	<FF>		,	<	L	\	l	
D	<CR>		-	=	M	]	m	}
E	<SO>		.	>	N	^	n	~
F	<SI>		/	?	O	_	o	

5. CODAGE DES NOMBRES NEGATIFS

5.1. LA CONVENTION DU SIGNE

Un nombre ayant son bit de poids fort à 0 est dit positif.

Un nombre ayant son bit de poids fort à 1 est dit négatif.

Un octet signé ne contient que _____ bits pour coder la valeur du nombre.

Un octet non signé peut prendre _____ combinaisons.

Un octet signé peut prendre _____ combinaisons positives et prendre _____ combinaisons négatives.

5.2. LE CODE COMPLEMENT A UN

Dans cette méthode, tous les entiers positifs sont représentés en format binaire correct. Par exemple, "+3" est, comme d'habitude, représenté par 00000011. La représentation de son complément "-3" s'obtient en inversant chaque bit de la représentation initiale. Chaque 0 est transformé en 1, et chaque 1 en 0. Dans notre exemple la représentation en complément à un de "-3" sera donc : 11111100.

Autre exemple :

+2	s'écrit	00000010
-2	s'écrit	11111101

Remarquons que, dans cette représentation, les nombres positifs commencent sur la gauche, Par un « 0 », et les nombres négatifs par un « 1 ».

Exercices : Trouver le complément à un sur un octet de 55 : _____
Trouver le complément à un sur un octet de 118 : _____

Réalisons le test consistant à ajouter "-4" à "+6"

-4	s'écrit	11111011
+6	s'écrit	00000110
<hr/>		
La somme est	(1)	00000001

où (1) représente le report.

Le résultat correct devrait être 2, soit 00000010. Autrement dit cela ne marche pas. Cette représentation permet de représenter des nombres positifs et négatifs. Cependant, leur addition ne fournit pas toujours un résultat correct. Nous sommes donc contraints de faire appel à une autre représentation, déduite du complément à un : le complément à deux.

5.3. LE CODE COMPLEMENT A DEUX

Dans la représentation en complément à deux, de la même manière qu'en complément à un, les nombres positifs se représentent en binaire signé habituel. La différence réside dans la représentation des nombres négatifs. **La représentation en complément à deux d'un nombre négatif s'obtient en calculant d'abord le complément à un, puis en ajoutant 1.** Exemple : +3 se représente en binaire signé par 00000011. Son complément à un s'écrit 11111100. Le complément à deux s'obtient en ajoutant 1. Soit donc 11111101.

Essayons une **addition** : 3 + 5

	(3)	00000011
+	(5)	00000101
=	(8)	00001000

Le résultat est correct.

Essayons une **soustraction** : 3 - 5

	(3)	00000011
+	(-5)	11111011
=	(-2)	11111110

Le résultat est correct.

Identifions le résultat, en calculant le complément à deux.

Le complément à un de 11111110 est 00000001 on ajoute 1, le complément à deux est donc 00000010, soit +2. Notre résultat précédent, "11111110", représente -2. Le résultat est correct. Ainsi, les résultats obtenus lors d'une addition et d'une soustraction sont exacts (en ignorant le report). La méthode est bonne.