

1 Nombres binaires :

1.1 Systèmes de numération pondérés :

Pour compter des objets et les représenter par des nombres, on utilise le plus souvent des systèmes de numération pondérés.

La définition d'un système pondéré repose sur les trois notions de **base**, de **digit** et de **poids**.

- La base B est le nombre de valeurs possibles pour un DIGIT. En base 10 B=10
- Le Digit D est la valeur du chiffre qui occupe une position. D prend les valeurs 0 à B-1.
- Le rang R : C'est la position du digit noté de 0 à n de droite à gauche.
- Le poids P : C'est la valeur représentée par un digit de valeur 1 : $P = B^R$.

En posant ces définitions, la valeur du nombre en système décimal s'écrit :

$$(N)_{10} = \sum D_i \times P_i \text{ pour } i=0 \text{ à } n \text{ si il y a } n \text{ rang.}$$

exemple 1 : nombre **décimal** (base 10) 2023.

Rangs (R)	Rang 3	Rang 2	Rang 1	Rang 0
Poids (P) = B^R				
Valeur des DIGITS (D)				
Valeur en décimal : $(N)_{10} =$				=

exemple 2 : nombre **binnaire** (base 2) 1001.

Rangs (R)	Rang 3	Rang 2	Rang 1	Rang 0
Poids (P) = B^R				
Valeur des DIGITS (D)				
Valeur en décimal : $(N)_{10} =$				=

MSB

(most significant bit ou bit de poids fort)

LSB

(least significant bit ou bit de poids faible)

exemple 3 : nombre **octal** (base 8) 1047.

Rangs (R)	Rang 3	Rang 2	Rang 1	Rang 0
Poids (P) = B^R				
Valeur des DIGITS (D)				
Valeur en décimal : $(N)_{10} =$				=

exemple 4 : nombre **hexadécimal** (base 16) 134A.

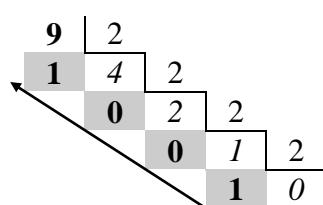
Rangs (R)	Rang 3	Rang 2	Rang 1	Rang 0
Poids (P) = B^R				
Valeur des DIGITS (D)				
Valeur en décimal : $(N)_{10} =$				=

1.2 Méthodes de conversion inverse : expression d'un décimal $(N)_{10}$ dans une autre base.

a Méthode des divisions successives :

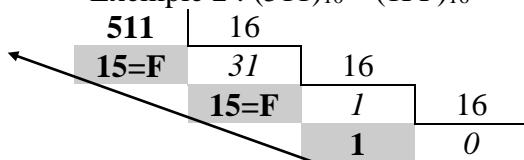
Il suffit de poser la division par B (ici 2) et continuer jusqu'à un résultat nul. La suite des restes donne le nombre binaire.

Exemple 1 :



On lit les restes dans le sens de la flèche et on obtient le nombre binaire : $(9)_{10} = (1001)_2$

Exemple 2 : $(511)_{10} = (1FF)_{16}$



b Méthode du bit de poids fort :

On cherche le poids le plus fort inférieur au nombre décimal que l'on veut convertir, puis après l'avoir soustrait on continue avec le reste.

Exemple 1 :

$$(9)_{10} = \begin{array}{cccc} 8 & 4 & 2 & 1 \\ (1 & 0 & 0 & 1)_2 \\ \text{reste} & 1 & 1 & 0 \end{array}$$

Exemple 2 :

$$(141)_{10} = \begin{array}{ccc} 64 & 8 & 1 \\ (2 & 1 & 5)_8 \\ \text{reste} & 13 & 5 & 0 \end{array}$$

1.3 Tableau de conversion directe entre bases sur 4 bits :

Le tableau ci-contre, donne à partir des règles énoncées précédemment, la correspondance entre les différents systèmes pondérés pour les 16 chiffres hexadécimaux.

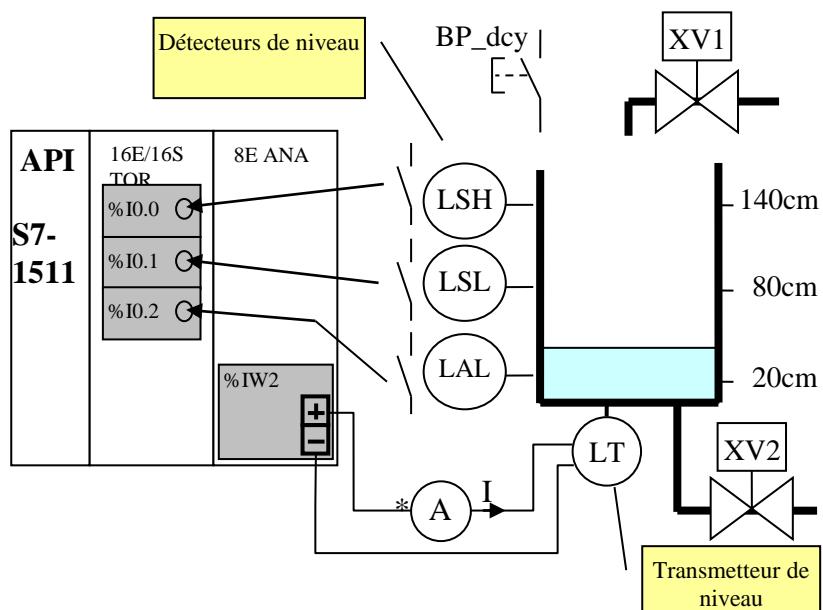
Décimal	Binaire	Octal	Hexadécimal
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F
16	10000	20	10
.....

Exemple d'application : stockage des valeurs issues d'un transmetteur de niveau

On utilise des détecteurs de niveau LAL, LSL et LSH pour contrôler le niveau dans une cuve.

Le transmetteur de niveau LT transmet l'information de niveau à l'automate en 4-20mA. LT étant connecté à la voie 0 du module analogique, l'information de niveau est récupérée dans le mot %IW2 suivant l'échelle suivante :

I (mA)	%IW2
4	0
20	27648



- Proposer un programme permettant de disposer de l'information de niveau en cm dans le mot **Niv_cm** d'adresse %MW10.
- Cette solution est-elle réaliste ?
- Proposer un programme permettant de disposer de l'information de niveau dans le mot **Niv_mm** d'adresse %MW12.

Niv : RobIA1	R120 : STRUCTURE MATERIELLE DES API Numération et codage - Rappel	Cours n°3 Page 3 sur 6
--------------	--	---------------------------

2 Nombres binaires signés:

2.1 Signe d'un nombre binaire

Lorsqu'on désire travailler sur des nombres binaires *signés* (c'est-à-dire affectés d'un signe), se pose le problème de l'expression de ce signe à l'aide des seuls symboles disponibles en binaire, le 0 et le 1.

→ Par convention, le signe d'un nombre binaire est représenté par son bit de poids fort (MSB), 0 si le nombre est positif, 1 s'il est négatif.

Exemples : sur 4 bits, le nombre 0101 est positif, le nombre 1000 est négatif.

2.2 Complément à 2

Il y a plusieurs conventions possibles pour représenter un nombre binaire négatif. La plus utilisée est celle du *complément à 2*. *Le complément à 2 d'un nombre binaire, c'est le nombre qu'il faut lui ajouter pour obtenir la puissance de 2 immédiatement supérieure.*

Exemple : chercher le complément à 2 de 1001.

La puissance de 2 immédiatement supérieure est 1 0000 (soit, en décimal, 2^4). Pour trouver le nombre binaire cherché, il faut effectuer la soustraction $1\ 0000 - 1001$. Celle-ci est plus facile si l'on remarque que $1\ 0000 = 1111 + 1$; en effet, la soustraction se fait alors sans retenue. On posera alors $1111 - 1001$ (a) puis on ajoutera 1 (b) :

$$(a) 1111 - 1001 = 0110$$

Le résultat intermédiaire (0110) est appelé *complément à 1*. Il est très facile à obtenir : il suffit de complémer chaque bit du nombre initial. Attention : si les nombres sont codés sur 4 bits, il faut tenir compte des 0 non significatifs.

$$(b) 110 + 1 = 111.$$

Le complément à 2 de 1001 est 111. On peut le vérifier en faisant la somme du nombre et de son complément à 2 : $1001 + 111 = 1\ 0000$

Exemple d'application : dérive du transmetteur de niveau

On suppose que le détecteur de niveau **LSL** commute de l'état 0 à l'état 1 à exactement $h = 80\text{cm}$. On veut utiliser cette propriété pour mesurer la dérive du transmetteur niveau dans une cuve.

- Proposer un programme permettant de disposer de l'information de l'erreur de niveau dans le mot **Ecart_mm** d'adresse %MW14.
- A quel écart de niveau correspond la valeur de %MW14. ci-contre :

Name	Address	Display format	Monitor value
"Ecart_mm"	%MW14	Hex	16#FFFD

3 Nombres à virgule flottante (réels)

3.1 Norme IEEE 754

La plupart des calculs scientifiques ne peuvent se faire efficacement en utilisant des entiers. Les mots doubles ne peuvent coder des grandeurs $> 2^{32}$. Il faut utiliser pour cela de nombres à virgule flottante. La norme IEEE 754 (reprise par la norme internationale CEI 60559) spécifie deux formats de nombres en virgule flottante (et deux formats étendus optionnels) et les opérations associées.

La répartition des bits est la suivante, où $1 \leq M < 2$:

Encodage	Signe	Exposant	Mantisse	Valeur d'un nombre
----------	-------	----------	----------	--------------------

Niv : RobIA1	R120 : STRUCTURE MATERIELLE DES API			Cours n°3
Rép : R120	Numération et codage - Rappel			Page 4 sur 6

Simple précision	32 bits	1 bit	8 bits	23 bits	$(-1)^S \times M \times 2^{(E-127)}$
Double précision	64 bits	1 bit	11 bits	52 bits	$(-1)^S \times M \times 2^{(E-1023)}$

Remarque 1 : Les nombres flottants ne sont pas tout à fait des nombres réels : la mantisse ne peut prendre qu'un nombre fini de valeurs, alors que les réels peuvent prendre une infinité de valeurs.

3.2 Format des nombres binaires dans les API =

a **taille de nombres réels:** les nombres à virgule flottante (float), sont dans les automates des mots de 32 bits. Ils sont donc stockés dans des mots doubles. exemple : %MD100.

b **valeur immédiate de nombres réels:**

Type	Taille	Exemple de valeur	valeur mini	valeur maxi
Real -	32bits	1.32E12	-3.402824E+38	3.402824E+38

Exemple d'application : valeur du niveau en m

On veut pouvoir afficher dans une supervision la valeur du niveau en mètres (m). Il faut pour cela convertir la mesure de niveau en , c'est-à-dire en nombre à virgule flottante IEC754, 32 bits.

- Proposer un programme permettant de disposer de l'information de niveau en mètre dans le mot double **Niv_m** d'adresse %MD100.
- A quelle valeur de niveau correspond la valeur de %MD10 ci-contre :

Name	Address	Display format	Monitor value
"Niv_m"	%MD100	Hex	16#3F19_999A

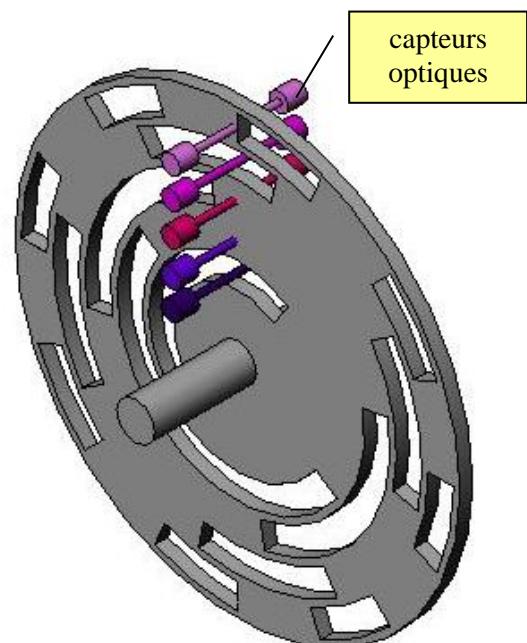
4 Codes binaires

On appelle "codes" les systèmes de numération non pondérés, c'est-à-dire les systèmes ne pouvant pas s'écrire sous la forme : $(N)_{10} = \sum D_i \times P_i$. Ces systèmes ne sont pas adaptés à l'exécution d'opérations mathématiques.

4.1 Code binaire réfléchi ou code GRAY

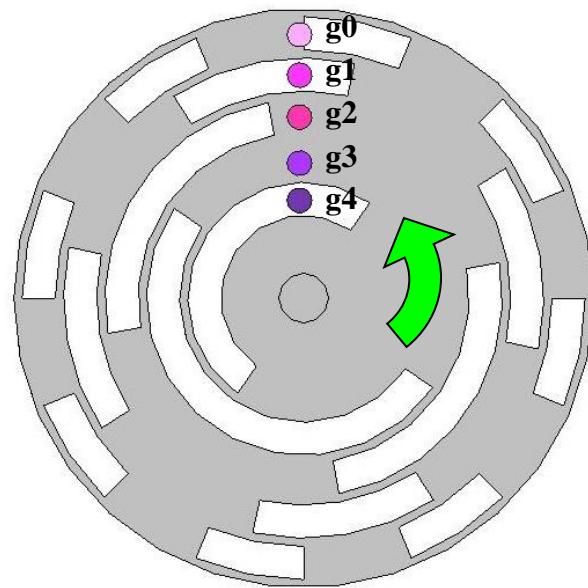
Dans certains cas, pour les codes étudiés précédemment, le passage d'un nombre à l'autre entraîne le changement de plusieurs bits. Le changement de tous les bits ne pouvant se faire simultanément, des codes erronés peuvent apparaître pendant la transition. Le code GRAY se caractérise par le fait que lorsque l'on passe d'un nombre au suivant, il n'y a qu'un bit qui change à la fois, évitant ainsi les aléas de transition. Le code GRAY est principalement utilisé dans les codeurs absolus.

Un codeur rotatif absolu, à la différence d'un codeur rotatif incrémental, délivre un code numérique binaire. Si un codeur possède N pistes, associés à N capteurs optiques, alors il délivrera à l'API un mot de N bits. La position et le sens sont connus dès la mise sous tension. Il n'y a pas nécessité d'opérer une prise d'origine comme c'est le cas lorsque l'on utilise un codeur incrémental.



Exemple de codeur absolu 5 bits

position	BINAIRE PUR	GRAY g ₄ g ₃ g ₂ g ₁ g ₀	CODE BCD
0	00000	00000	0000
1	00001	00001	0001
2	00010	00011	0010
3	00011	00010	0011
4	00100	00110	0100
5	00101	00111	0101
6	00110	00101	0110
7	00111	00100	0111
8	01000	01100	1000
9	01001	01101	1001
10	01010	01111	0001 0000
11	01011	01110	0001 0001
12	01100	01010	0001 0010
13	01101	01011	0001 0011
14	01110	01001	0001 0100
15	01111	01000	0001 0101
...	
...			
...			

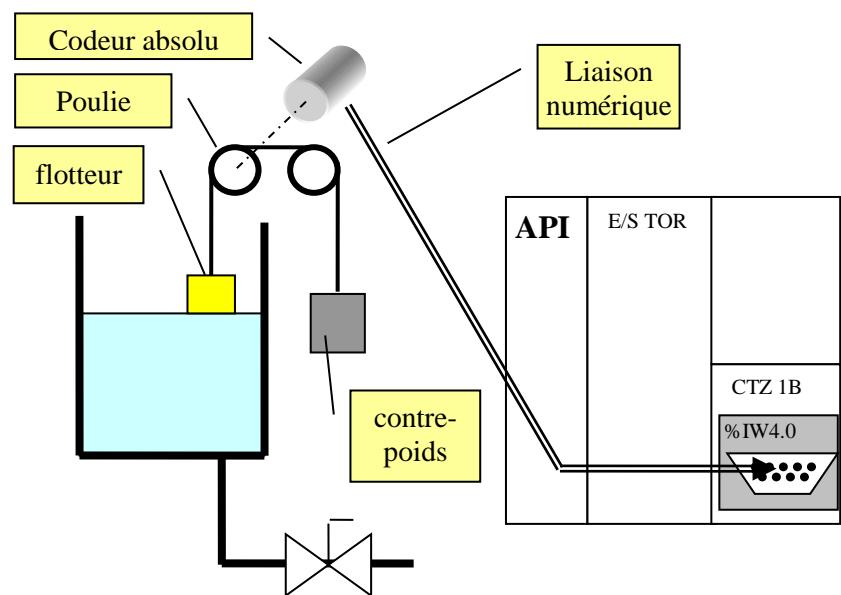


Une carte de conversion spéciale est souvent nécessaire pour réceptionner le mot binaire en provenance du codeur. Les bits g₄ à g₀ forment un mot utilisé pour donner la position sur l'axe, après conversion par l'automate en entier binaire.

Exemple d'application : mesure du niveau dans la cuve

On utilise un codeur absolu 12 bits afin de déterminer la hauteur dans la cuve. Un module permet de récupérer l'information de niveau dans un mot %IW40 écrit en code GRAY.

Quel est l'intérêt d'utiliser ce type de transmission plutôt que celle étudiée précédemment? Quel doit être le diamètre de la poulie pour que le codeur fasse un tour entre les niveaux 20 et 140 cm ? On récupère le mot %IW40 = 100001010101 dans l'API. Quel est le niveau de liquide dans la cuve ?

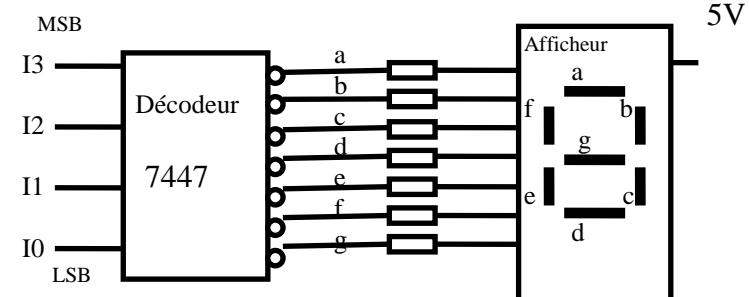


4.2 Code B.C.D. (Binary Coded Decimal)

Chaque chiffre d'un nombre décimal est codé en binaire pur. Comme un chiffre décimal nécessite 4 bits, on aura autant de groupes de 4 bits qu'il y a de chiffres dans le nombre. Ce type de code est utilisé pour afficher des valeurs sur des afficheurs de type 7 segments.

Exemple : 1264 codé en B.C.D. donne (0001 0010 0110 0100)_{BCD}

1	2	6	4
0001	0010	0110	0100



Exemple d'application :
affichage du niveau en m

Exemple de raccordement d'un afficheur 7 segments.
I3 à I0 codent le chiffre qui doit être affiché en code BCD.

On veut pouvoir afficher sur 4 afficheurs 7 segments présents sur le tableau électrique la valeur du niveau en mètres (m). L'information est envoyée aux afficheurs par le mot de sortie %QW0

- A quelle valeur de niveau correspond la valeur de %QW10 ci-dessous :

Name	Address	Display format	Monitor value
"Affichage_Niv_m"	%QW0	Bin	2#0001_0001_1001_0101

4.3 Code A.S.C.I.I. (American Standard Code for Information Interchange)

C'est le code le plus couramment utilisé pour décrire les principaux caractères utilisés dans les ordinateurs, systèmes programmables... Le tableau 2 ci-dessous donne le code ASCII pour des mots numériques de sept bits (128 combinaisons).

ASCII	b ₇	b ₆	b ₅	000	001	010	011	100	101	110	111
b ₄ b ₃ b ₂ b ₁	Hexa.	0	1	2	3	4	5	6	7		
0000	0	(NUL)	(DLE)	espace	0	@	P	`	p		
0001	1	☺ (SOH)	(DC1)	!	1	A	Q	a	q		
0010	2	(STX)	↑ (DC2)	"	2	B	R	b	r		
0011	3	♥ (ETX)	!! (DC3)	#	3	C	S	c	s		
0100	4	♦ (EOT)	¶ (DC4)	¤	4	D	T	d	t		
0101	5	♣ (ENQ)	_ (NAK)	%	5	E	U	e	u		
0110	6	♠ (ACK)	(SYN)	&	6	F	V	f	v		
0111	7	• (BEL)	↑ (ETB)	'	7	G	W	g	w		
1000	8	(BS)	↑ (CAN)	(8	H	X	h	x		
1001	9	○ (HT)	↓ (EM))	9	I	Y	i	y		
1010	A	(LF)	→ (SUB)	*	:	J	Z	j	z		
1011	B	(VT)	← (ESC)	+	;	K	[k	{		
1100	C	(FF)	(FS)	,	<	L	\	l			
1101	D	(CR)	↔ (GS)	-	=	M]	m	}		
1110	E	(SO)	(RS)	.	>	N	^	n	-		
1111	F	⊗ (SI)	(US)	/	?	O	—	o	DEL		

exemple d'application : Message reçu du capteur numérique de la cuve. Le capteur envoie une suite de caractères. Ces caractères sont stockés à partir de l'octet %MB30 dans l'API

%MB	30	31	32	33	34	35
Valeur hexadécimale	44	65	66	61	75	74

Quel est le message reçu ?