UM2-L2 EEA : HLEE407 – Programmation des microcontrôleurs

1 – L'origine des chiffres

1. Notions de mathématiques binaires :

On peut définir de nombreux éléments de la vie quotidienne qui nous entourent par deux états:

« Ça marche ou ça ne marche pas »

Il est facile de dire que la lumière est allumée ou éteinte, que le verre est plein ou vide, qu'il fait jour ou qu'il fait nuit. On raisonne avec deux états opposés sans se préoccuper de la variation pour passer d'un état à un autre.

On résume :

Lumière : **Allumée** ou **Éteinte**

On peut affecter une valeur numérique par exemple 1 pour allumée et 0 pour éteinte :

1 **0**

Il sera alors facile de noter l'état de la lumière par 0 ou 1; ceci correspond réellement au passage du courant. Avec cette définition on a besoin que de deux éléments pour noter les états : d'où la notion de **« binaire » (0 et 1)**. Le chiffre binaire ou unité s'appelle le « BIT » la contraction de Binary Digit d'après Shannon en 1938.

2. Histoire de la notation décimale :

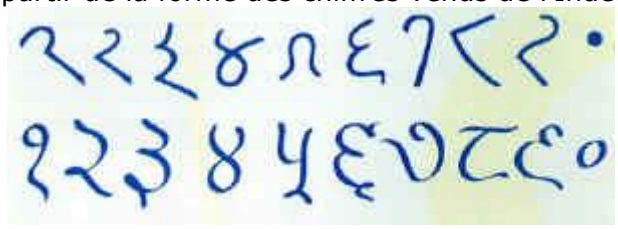
Mais au fait pourquoi appelle-t-on également cela mathématique ou électronique digitale ? ... on compte avec ses doigts, et on a dix doigts, d'où le système décimal qui comporte 10 chiffres ou unités:

0, 1, 2, 3, 4, 5, 6, 7, 8, 9

Ces chiffres nous ont été importés de l'Inde par les Arabes en l'an 1000

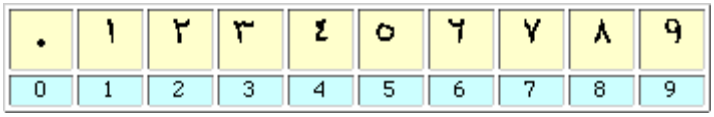
*Le mot "chiffre" vient d'ailleurs de "sifr" qui signifie "**vide**" en arabe, terme utilisé au départ pour désigner le zéro.*
Le système décimal a été transmis à l'Occident par l'intermédiaire de l'Andalousie musulmane.
Le graphisme des chiffres que nous connaissons n'est pas celui qu'ils avaient à l'origine.
Leur forme s'est progressivement modifiée, épurée.
Aujourd'hui dans les pays arabes d'Orient, les chiffres (étiquettes dans les magasins) sont incompréhensibles de nous, car ils restent graphiquement proches de leurs ascendants indiens.

A partir de la forme des chiffres venus de l'Inde...



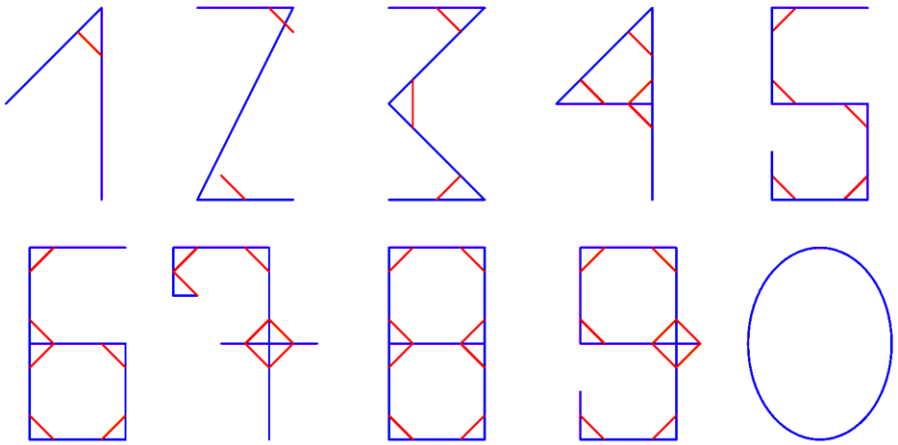
...Voici celle des chiffres "arabes" s'est européanisée vers le XV e siècle.

Les chiffres arabes actuels :



1 – L'origine des chiffres

Une des représentations possibles des symboles actuels pourrait bien venir des arabes qui comptaient les angles comme sur cette figure :



On attribut l'apparition du Zéro aux arabes également « Sifre » le cercle (qui n'a pas d'angle). En Chine, Confucius a eu également l'idée de marquer la position des chiffres par un symbole « 0 ».

3. Position des chiffres dans le nombre :

La valeur des chiffres en fonction de leur place dans le nombre correspond à la base puissance du rang.
On appelle ce rang le POIDS des chiffres, à droite le moins significatif aura le poids ou le rang « 0 »
Ici la base est 10 (décimale), chaque poids du chiffre correspond à une puissance de sa base :

MSB Most signifiant bit = le bit de poids le plus fort (à gauche) son poids est 7 pour un octet
LSB Least signifiant bit = le bit de poids le plus faible (à droite) son poids est 0

Exemple de décomposition sur une base décimale d'un nombre à 4 chiffres (du rang 3 au rang 0)

1 2 3 4 = 4 x 10^0 = 4
+ 3 x 10^1 = 30
+ 2 x 10^2 = 200
+ 1 x 10^3 = 1000
= 1234

Exemple de décomposition en utilisant une base binaire (du bit 3 au bit 0)

1 1 0 1 = 1 x 2^0 = 1
+ 0 x 2^1 = 0
+ 1 x 2^2 = 4
+ 1 x 2^3 = 8
= 13

Exemple de décomposition en utilisant une base 16 (hexadécimale) en 1 octet

$23 = 3 x 16^0 = 3
+ 2 x 16^1 = 32
= 35

1 – Les systèmes numériques

4. Notions de mathématiques binaires :

On peut définir de nombreux éléments de la vie quotidienne qui nous entourent par deux états:

« Ça marche ou ça ne marche pas »

Il est facile de dire que la lumière est allumée ou éteinte, que le verre est plein ou vide, qu'il fait jour ou qu'il fait nuit. On raisonne avec deux états opposés sans se préoccuper de la variation pour passer d'un état à un autre.

On résume:

Lumière : **Allumée** ou **Eteinte**

On peut affecter une valeur numérique par exemple 1 pour allumée et 0 pour éteinte

Lumière: **1** **0**

Il sera alors facile de noter l'état de la lumière par 0 ou 1; ceci correspond réellement au passage du courant. Avec cette définition on a besoin que de deux éléments pour noter les états: d'où la notion de **« binaire » (0 et 1)**. Le chiffre binaire ou unité s'appelle le « BIT » la contraction de Binary Digit.

Mais au fait pourquoi appelle-t-on également cela mathématique ou électronique digitale ? ... on compte avec ses doigts, et on a dix doigts, d'où le système décimal qui comporte 10 chiffres ou unités:

0, 1, 2, 3, 4, 5, 6, 7, 8, 9

En utilisant la précédente définition des deux états d'une ampoule électrique, on a besoin de seulement 2 valeurs 0 et 1. On peut regrouper ces unités (ou bits) en nombre (ou mot) comme on le ferait en base décimale. Ces nombres s'appellent:

Quartets	(Half) codé sur 4 bits
Octets	(Byte) codé sur 8 bits
Mots	(Word) codé sur 16 bits
Longs mots	(Long word) codé sur 32 bits

Ces nombres (ou mots informatiques) répondent aux mêmes règles mathématiques que les nombres décimaux. (Addition, soustraction, multiplication, division).

Pour des facilités d'utilisation on coupe les octets, en deux quartets, les mots en quatre quartets, les longs mots en huit quartets.

Le système de chiffre arabe que nous utilisons ne possédant que 10 signes (de 0 à 9) on utilisera les signes de l'alphabet pour compléter une description des mots en 4 bits. Effectivement, si l'on décrit l'information avec 4 bits en puissance 2 (binaire) on obtient 16 combinaisons possibles. (2^4 = 16).

Le nouveau système pour décrire toutes les possibilités de 4 bits s'écrit donc avec les 16 valeurs du code hexadécimal (base 16).

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

1 – Les systèmes numériques

5. Les notations normalisées :

- pour les nombres décimaux exprimés en base 10, on écrira le nombre tel quel exemple : **125**
- pour les nombres binaires exprimés en base 2, on écrira les bits précédés du signe « % »
  ou suivi de la lettre B exemple **%1110 0011** ou **1110 0111B**
- pour les nombres octaux exprimés en base 8, on écrira les chiffres précédés du signe « @ »
  ou suivi de la lettre O exemple **@1275** ou **1275O**
- pour les nombres hexadécimaux exprimés en base 16, on écrira les nombres précédés du signe « $ » ou précédé de « 0x » ou suivi de « H » exemple **$12AF** ou **0x12AF** ou **12AFH**

6. Table de conversion binaire / hexadécimal / décimal /octal :

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

7. Table de conversion Hexa --> décimal et inverse :

Quatrième chiffre		**Troisième chiffre**		**Deuxième chiffre**		**Premier chiffre**	
Hexa	Décimal	Hexa	Décimal	Hexa	Décimal	Hexa	Décimal
0	0	0	0	0	0	0	0
1	4.096	1	256	1	16	1	1
2	8.192	2	512	2	32	2	2
3	12.288	3	768	3	48	3	3
4	16.384	4	1.024	4	64	4	4
5	20.480	5	1.280	5	80	5	5
6	24.576	6	1.536	6	96	6	6
7	28.672	7	1.792	7	112	7	7
8	32.768	8	2.048	8	128	8	8
9	36.864	9	2.304	9	144	9	9
A	40.960	A	2.560	A	160	A	10
B	45.056	B	2.816	B	176	B	11
C	49.152	C	3.072	C	192	C	12
D	53.248	D	3.328	D	208	D	13
E	57.344	E	3.584	E	224	E	14
F	61.440	F	3.840	F	240	F	15

1 – Les systèmes numériques

8. Le code ASCII :

Pour échanger des informations deux systèmes doivent avoir en commun un codage identique des informations. Le code ASCII (American Standard Inter-exchange Intercommunication) permet de normaliser des liaisons. Une partie de ce code comporte les caractères alphanumériques et les signes de ponctuations, une deuxième partie désigne des caractères de contrôle de l'échange et la mise en page. Il est à noter que ce code ne comporte que 7 bits; le code ANSI en comporte 8 donc 128 caractères de plus.

LSB/MSB	0	1	2	3	4	5	6	7
0	NUL	DLE	SP	0	@	P	`	p
1	SOH	DC1	!	1	A	Q	a	q
2	STX	DC2	«	2	B	R	b	r
3	ETX	DC3	#	3	C	S	c	s
4	EOT	DC4	$	4	D	T	d	t
5	ENQ	NAK	%	5	E	U	e	u
6	ACK	SYN	&	6	F	V	f	v
7	BEL	ETB	‘	7	G	W	g	w
8	BS	CAN	(	8	H	X	h	x
9	HT	EM	)	9	I	Y	i	y
A	LF	SUB	.	:	J	Z	j	z
B	VT	ESC	+	;	K	[	k	{
C	FF	FS	,	<	L	\	l	|
D	CR	GS	-	=	M	]	m	}
E	SO	RS	.	>	N	^	n	~
F	SI	US	/	?	O	_	o	DEL

Il est à noter quelques caractères de contrôle important à connaître :

- NUL Caractère qui doit être ignoré
- STX Début de transmission
- ETX Fin de transmission
- EOT Fin de texte
- ENQ Demande de transmission
- ACK Acquittement (transmission correcte)
- NAK Erreur de transmission
- BEL Indicateur sonore (beep)
- BS Retour arrière (curseur à gauche)
- HT Tabulation horizontal (curseur à droite)
- LF Ligne suivante (curseur vers le bas)
- VT Tabulation vertical (curseur vers le haut)
- FF Page suivante
- CR Retour chariot (touche entrée) (curseur en début de ligne)
- ESC Echappement
- DEL Effacement

1 – Les systèmes numériques

9. Le système BCD :

Avec un système binaire si on veut représenter des nombres en valeur décimale on utilise 4 bits pour exprimer les chiffres de 0 à 9 ; on appelle ce système Décimal Codé en Binaire.

10. Les poids des mots :

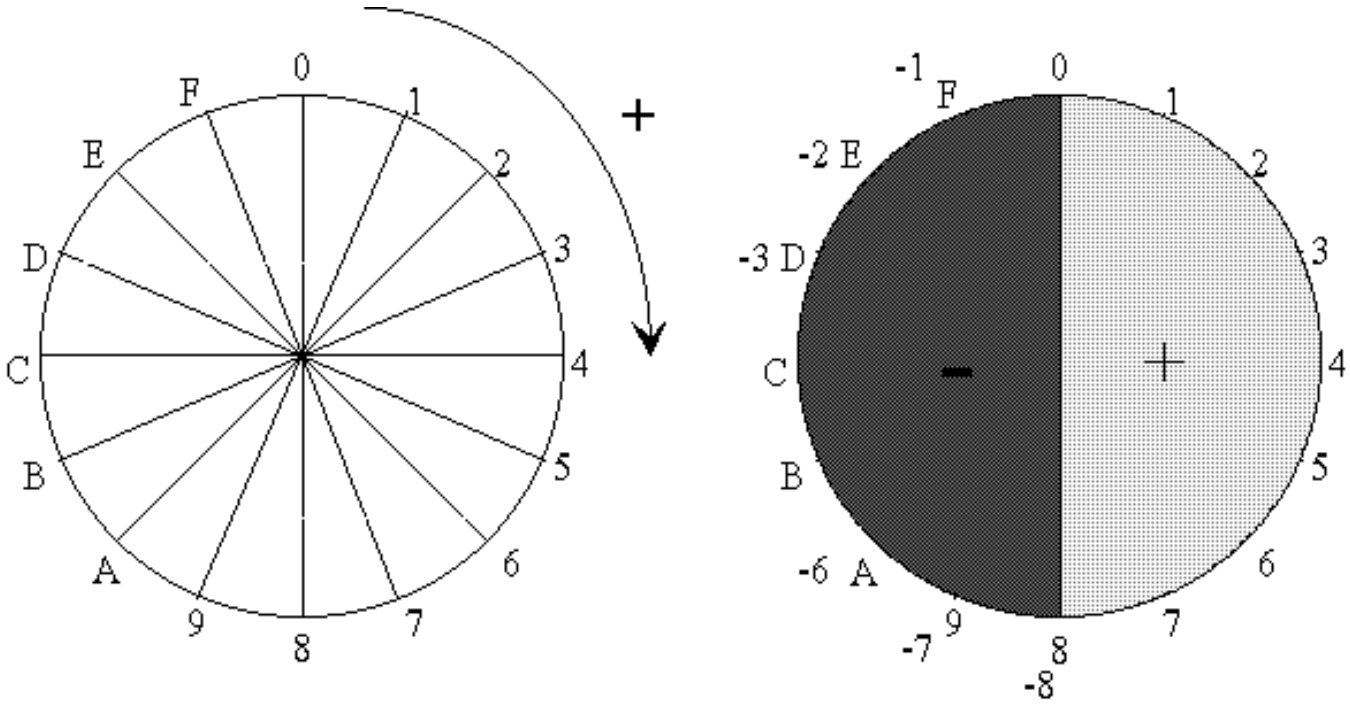
La position d'un bit dans un mot est appelée « **poids** » ; c'est à dire la grandeur qu'il entraine lors de son changement de 0 à 1 ou de 1 à 0

MSB Most signifiant bit = le bit de poids le plus fort (à gauche) son poids est de 2^7 (=128) pour un octet
LSB Least signifiant bit = le bit de poids le plus faible (à droite) son poids est 0 (moins significatif)
On a l'habitude de numéroter les bits en partant de la droite par le bit 0 - le LSB - le bit de poids 2^0 (=1).

11. Les valeurs signées :

Si on utilise 8 bits pour coder des informations il est possible de définir une liste de 256 états de 0 à 255
Si notre choix est de n'utiliser que des entiers non signés la numération d'événement se fera de 0 à 255.
Par contre si toujours avec les 8 bits on désire effectuer des calculs avec des entiers signés, on utilisera alors que 7 bits pour définir la valeur et 1 bit (le MSB) pour définir le signe.

12. Représentation schématique d'un système à 16 valeurs (4 bits) :



N.B.: les opérations (addition et soustractions) s'effectuent de la même façon pour des valeurs signées ou non.

13. Complémentation et négation :

La complémentation à 1 ou l'inverse est l'opération d'inversion de tous les bits à « 1 » par des bits à « 0 » et vice versa.
La négation est la valeur négative correspondant au nombre telle que ces deux valeurs additionnées donnent le résultat 0. C'est également le complément à 2.
On réalise une négation (ou complément à 2) en additionnant 1 au complément à 1.
Exemple : %0000 0010 = 2 ➔ -2 = %1111 1101 +1 = %1111 1110 (=$FE)
Vérification : %0000 0010 + %1111 1110 = %(1) 0000 0000

1 – Les systèmes numériques

11- Rappel des fonctions logiques simples

La fonction ET (AND)

La fonction OU inclusif (OR)

La fonction OU exclusif (XOR)

La fonction Pas (NOT)

La fonction Non ET (NAND)

La fonction Non Ou (NOR)

La fonction Egalité (XNOR)

1 – Les systèmes numériques

Mots Clé :

Chiffre, Nombre, Base, Décimal, Binaire, Bit, MSB, LSB, Hexadécimal, Mot, Octet, Byte, Cardinal, Ordinal
Complément, Inverse, Négation, Complément à 2, ko, Mo, Go
Table de vérité, ET, OU, XOU, NOT. Code ASCII, Parité

A Savoir :

Hexadécimal → (**0x**) ou (**$**) ; binaire → (**0b**) ou (**%**) ; Décimale → (aucune notation particulière)
Poids d'un chiffre, poids d'un bit
1ko= 1024 octets 1Mo= 1024ko 1Go= 1024Mo
Puissance de 16 : 16^0=1 16^1=16 16^2=256 16^3=4096
Addition hexadécimale, Soustraction hexadécimale, opération logique (bit à bit)
Représentation de la valeur négative en complément à 2.
Attention en 8 bits max = 255 (0xFF)
Codes ascii : *« CR » =0x0D « LF » =0x0A « SPC »=0x20 « A »=0x41 « 0 »=0x30*
Calcul de parité paire ou de parité impaire

En Plus :

Les nombres négatifs par la représentation d'une valeur signée : ***-1 → $FF***
$FF+1 =0 *=> mais dépassement de la capacité mémoire. Visible sur bit de retenu (carry)*