Systèmes de numération (Comment on compte)

I) Le système décimal

C'est celui qu'on utilise tous les jours. Un système est basé sur des chiffres (10 chiffres ==> base 10).

1) la base

Les chiffres 0,1,2,3,4,5,6,7,8,9
Il y a un ordre 0 est plus petit que 1, qui est plus petit que 2 et….
On passe d'un chiffre à son successeur en ajoutant 1:
0+1==>1 1+1==> 2 2+1==>3

2) Les nombres.

Que faire avec 9+1 lorsque l'on ne possède que les 10 chiffres 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

$$\begin{array}{rr} & 9 \\ + & 1 \\ \hline 1 & 0 \end{array}$$

On prend le deuxième chiffre De la base cela fait 1 dizaine.

On prend le premier chiffre de la base Et cela fait 0 unités

II) Le système binaire (base 2)

1) la base

Elle est composée de deux chiffres 0 et 1
Il y a un ordre 0 est plus petit que 1
On passe d'un chiffre au suivant en ajoutant 1: 0 + 1 ==> 1

2) Les nombres.

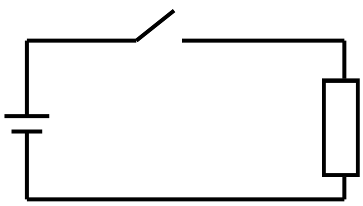
Que faire avec 1+1 quand l'on ne possède que les deux chiffres 0, 1,

$$\begin{array}{rr} & 1 \\ + & 1 \\ \hline 1 & 0 \end{array}$$

On prend le deuxième chiffre de la base cela fait 1 deuzaine.

On prend le premier chiffre de la base Et cela fait 0 unités

3) Pourquoi utilise-t-on le système binaire?



Les phénomènes électriques fonctionnent souvent en tout ou rien. Par exemple 0 le courant ne passe pas et 1 si le courant passe. Cela a imposé l'utilisation du système binaire.

III) Le système Hexadécimal

1) Nécessité d'une autre base

Maintenant pour différentier un nombre binaire (en base 2) d'un nombre décimal, je ferai précéder le nombre binaire du symbole % .
Voici le nombre binaire: %1110000011101 et le même en décimal: 7197
%1110000011101 est difficile à retenir (imaginer que ce soit le code de votre carte bleue, et 7197 n'est pas compris par un microcontrôleur.
L'idéal serait de pouvoir passer de l'un à l'autre rapidement, Mais malheureusement ce n'est pas le cas.
Par exemple voyons comment convertir %1110000011101 en décimal:
Pour cela réalisons le tableau ci-dessous. Il contient autant de colonnes que le nombre binaire comporte de chiffres. On place un 1 dans la colonne la plus à droite. Pour les autres colonnes on multiplie par 2 le nombre de la colonne de gauche.

4096	2048	1024	512	256	128	64	32	16	8	4	2	1

Plaçons notre nombre binaire sur la deuxième rangée:

4096	2048	1024	512	256	128	64	32	16	8	4	2	1
1	1	1	0	0	0	0	0	1	1	1	0	1

Dans la troisième rangée, inscrivons les résultats des multiplications des deux rangée du haut.

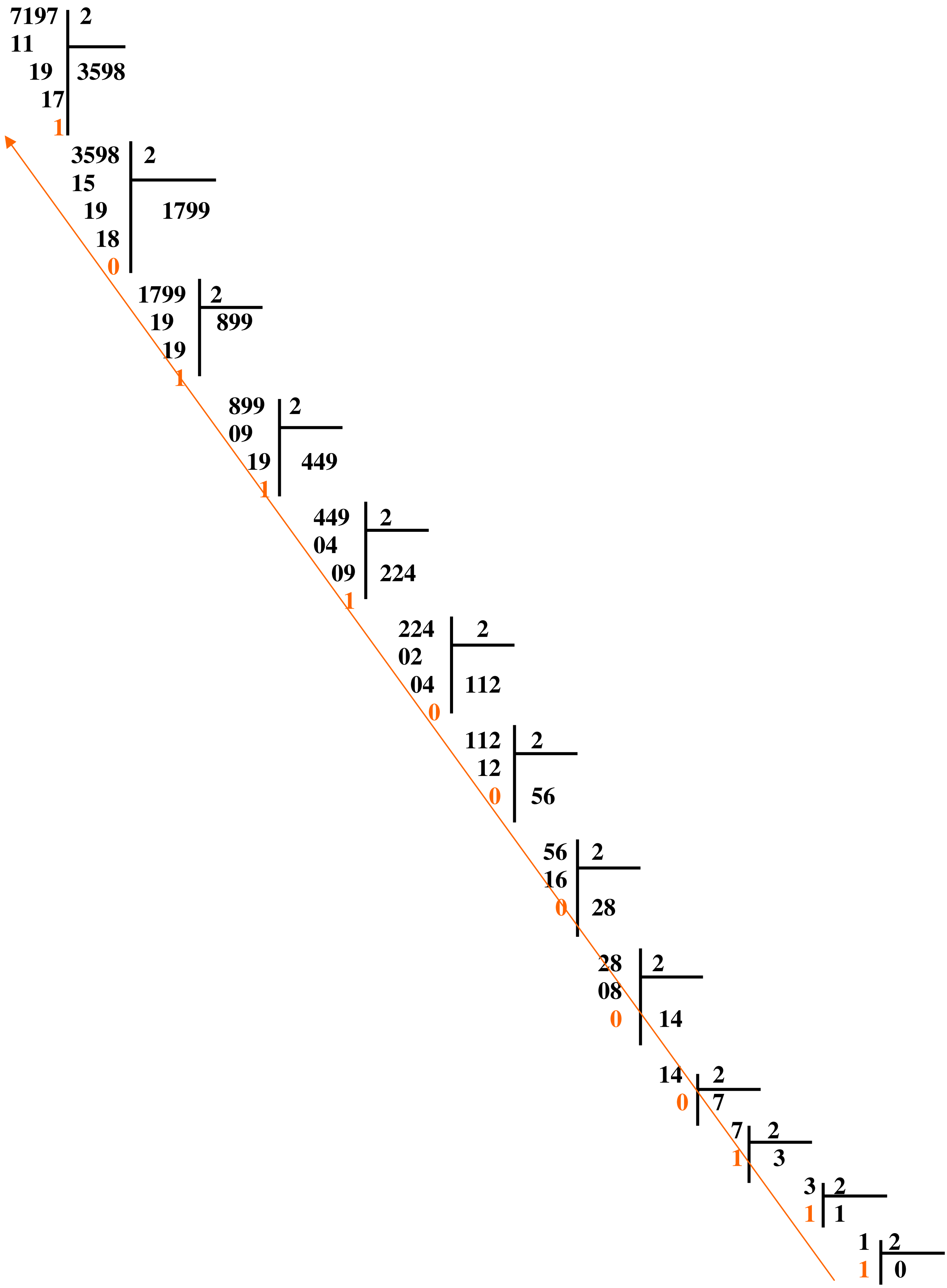
4096	2048	1024	512	256	128	64	32	16	8	4	2	1
1	1	1	0	0	0	0	0	1	1	1	0	1
4096	2048	1024	0	0	0	0	0	16	8	4	0	1

Il suffit ensuite d'ajouter les nombres de la troisième rangée.
4096 + 2048 + 1024 + 16 + 8 + 4 +1 = 7197
Bon on voit que cela n'est pas très pratique.
Pour les forts en math il existe la formule (en utilisant la notation de Einstein): $a_\mu 2^\mu$

La conversion inverse pour obtenir le nombre binaire à partir du décimal est aussi assez longue

<u>Voyons comment on peut convertir 7197 en binaire:</u>

Il faut faire une succession de divisions par 2



En lisant les restent des divisions dans le sens de la flèche, on retrouve bien %1110000011101

Il est plus facile de faire ces conversions en utilisant par exemple la calculatrice de windows en mode scientifique ou en mode programmateur. Pour cela choisir la base de départ, écrire le nombre dans cette base et enfin cliquer sur la base de destination.

Heureusement, il existe la base 16 (hexadécimal), Dans cette base, les nombres sont faciles à retenir et la conversion en binaire et très facile si on retiens par cœur la conversion des 16 chiffres hexadécimaux.

2) La base hexadécimale

Elle est formée de 16 chiffres: 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F
Les chiffres sont ordonnés: 9 est plus grand que 8 , A est plus grand que 9, C est plus grand que B
On passe d'un chiffre à son successeur en ajoutant 1:
8+1 ==> 9 9+1==>A A+1==>B B+A==>C C+1==>D D+1==>E E+1==>F
Il est utile de connaitre par cœur le tableau suivant de correspondance entre les chiffres (les deux premières colonnes):

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

J'ai mis les nombres binaires sous la forme de 4 chiffres avec des zéros inutiles, car c'est surtout cette configuration que l'on rencontrera.

Remarque:
Les anglo-saxons utilisent le mot Bit pour désigner un chiffre binaire et Byte un chiffre hexadécimal.

3) Les nombre hexadécimaux

Je ferai précéder les nombres hexadécimaux (en base 16) du symbole $ (Symbole de dollar), bien que la plupart des revues utilisent le préfixe 0x. Exemple :
$124 est le nombre 124 en base 16. (ou encore 0x124).
292 est le chiffre décimal 292
%000100100100 est un nombre en binaire. Il représentent tous le même nombre 292.

4) Intérêt des nombres hexadécimaux

Le nombre binaire %1110000011101   est $1C1D en hexadécimal et 7197 en base 10.
La conversion entre nombre hexadécimal et décimal reste toujours aussi difficile à faire.
Mais la conversion binaire <==> hexadécimal est rapide
Exemple n°1: convertir $1C1D en binaire.

1	**C**	**1**	**D**	**<— Le nombre hexa**
0001	**1100**	**0001**	**1101**	**<— Le nombre binaire**

Donc $1C1D = %1110000011101

Exemple n°2:   convertir %110110110
On sépare en série de 4 bits à partir de la droite: 1 1011 0110
On complète à 4 bits à gauche avec des 0:   0001 1011 0110
On remplace par les équivalents sut par cœur dans le tableau: 1B6
C'est notre nombre hexadécimal: $1B6
Connaitre ce nombre en base 10 n'est pas très utile, l'important étant de pouvoir convertir et de retenir. Un autre interet des nombres hexadécimaux, c'est qu'ils sont petits que les décimaux:
%110110110 = $1B6 = 438
Les conversion entre décimal et hexadécimal se font comme entre le binaire et le décimal, mais il faut multiplier ou diviser par 16 au lieu de 2.
Exemple convertir $1B6 en décimal:

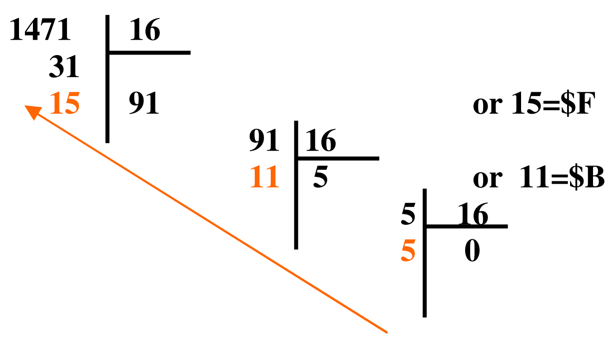
4096	**256**	**16**	**1**
	1	**B**	**6**
	256	**16*11**	**6**

<== On multiplie par 16

<== Le chiffre $B correspond à 11

Le résultat est: 256 + 176 + 6 = 438

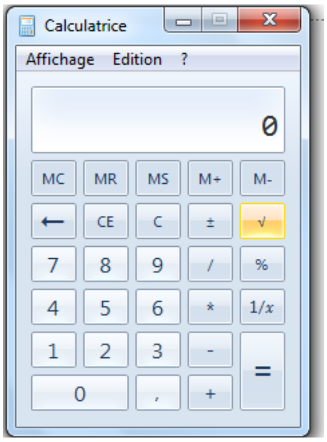
Avec les microcontrôleurs, nous n'auront pas besoin d'avoir de tableaux plus grand.
Pour convertir 1471 en hexadécimal, il faut faire les divisions:



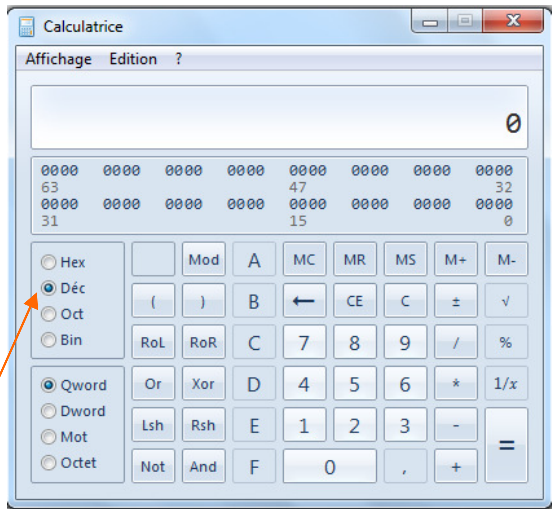
Le résultat se lit suivant la flèche et en tenant compte que les nombres décimaux 10 = $A, 11=$B, 12=$C, 13=$D, 14=$E, 15=$F . Ce qui donne $5BF =1471
On peut voir que la conversion en binaire est imédiate: %0001 0100 0111 0001
Nous allons voir maintenant comment faire toutes ces conversion avec la calculatrice de windows.

5) Utilisation de la calculatrice de windows pour faire les conversions.

La calculatrice se trouve dans: menu démarrer/tous les programmes/accessoires/calculatrice. Si vous avez une version de windows antérieure à windows 7 mettez la en mode scientifique. Sous windows 7 il faut la mettre en mode programmateur.

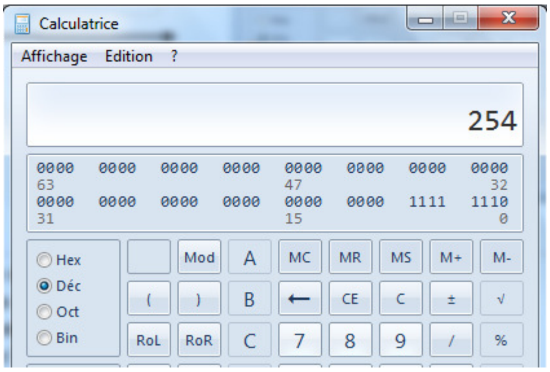


Cliquez sur affichage et choisir le mode programmateur ou scientifique si version de windows antérieu-re à 7

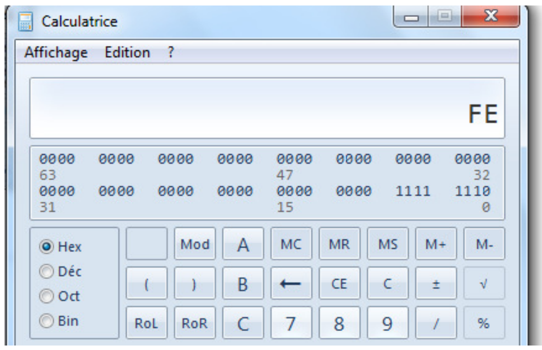


Pour convertir à partir d'un nombre décimal (base 10)
Vérifiez que le mode décimal est bien coché (Déc)

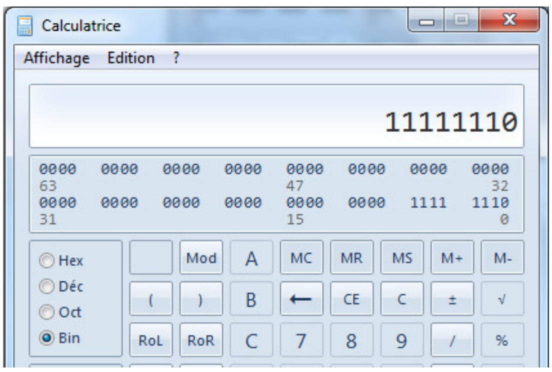
Pour avoir le nombre binaire, cochez Bin



Tapez le nombre décimal, ici 254.



Cochez Hex pour avoir le nombre hexa-décimal (ici $FE)



Cochez Bin pour avoir le nombre en binaire
Ici %11111110

Procedez de façon similaire pour faire les autres types de conversion. CC'est tout de même bien plus agréable qu'à la main.

IV) Les calculs

1) adition

Elles se font comme en base 10,

Exemple: 156 + 137

```
    1                 1 1              1   1 1
  1 5 6               $ 9 C          % 1 0 0 1 1 1 0 0
+ 1 3 7             + $ 8 9          % 1 0 0 0 1 0 0 1
 ______              _____         +  ____________

  2 9 3             $  1 2 5         % 1 0 0 1 0 0 1 0 1
```

2) La soustraction

La soustraction n'existe pas en tant que telle, on ajoute le complément à 2

3) Le complément à 1

Il est très utilisé cela reviens à remplacer tout les 1 par des 0 et les 0 par des 1.

Exemple:

% 111000 ==> %_000111 celui de % 10101010 ==> % 01010101

4) Le complément à 2

On prend le complément et on ajoute 1

Exemple: % 10110011 ==> % 01001100 +1 ==> % 01001101

Application à la soustraction: 73—69 (% 01001001 - % 01000101)

Le complément à 1 de % 01000101 ==> % 10111010

Le complément à 2 est ==> % 10111010 + 1 = % 10111011

On ajoute le complément à 2 :

```
     1 1 1 1   1 1
% 0 1 0 0 1 0 0 1
% 1 0 1 1 1 0 1 1
+ _______________
% 0 0 0 0 0 1 0 0
```

(Ne pas tenir compte de la retenue débordant le plus grand nombre)

Or % 100 <==> 4

5) le ET (AND)

E1	E2	S
0	0	0
0	1	0
1	0	0
1	1	1

La sortie S est à 1 si l 'entrée E1 et l'entrée E2 sont à 1.

C'est très utilisé pour éliminer des chiffres indésirables.

Exemple: soit % 1 0 1 0 1 0 1 0

Pour masquer les quatre derniers bits on fait un et:

% 1010 1010 AND 1111 000 ==> 1010 000

Pour s'en souvenir, multiplier les bits