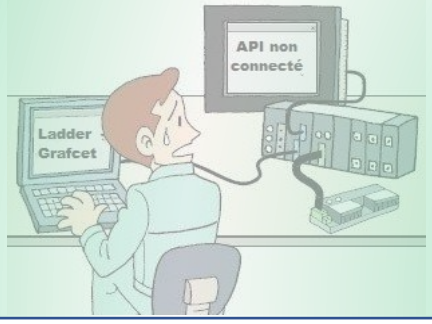
SEMAINE 4

LA PROGRAMMATION DES AUTOMATES

FICHE 12 : LES SYSTEMES DE NUMERATION ET DE CODAGE

Automation & Sense

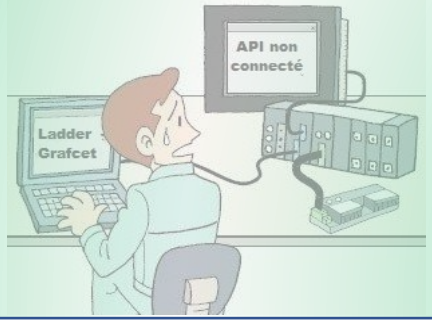
Novembre 2017 | www.automation-sense.com



Objectifs :

Après avoir consulté cette fiche, vous serez en mesure de:

- Définir c'est quoi un nombre décimal, binaire, octal et hexadécimal
- Faire la conversion vers les différents systèmes de numération
- Définir les termes bit, octet, mot, bit du poids le plus faible, bit du poids le plus fort etc…
- D'effectuer des calculs arithmétiques basiques de nombres binaires



SOMMAIRE

I) INTRODUCTION GENERALE

II) LE SYSTEME DE BASE 10 OU DECIMALE

III) LE SYSTEME DE BASE 2 OU BINAIRE

IV) LE SYSTEME DE BASE 8 OU OCTAL

V) LE SYSTEME DE BASE 16 OU HEXADECIMAL

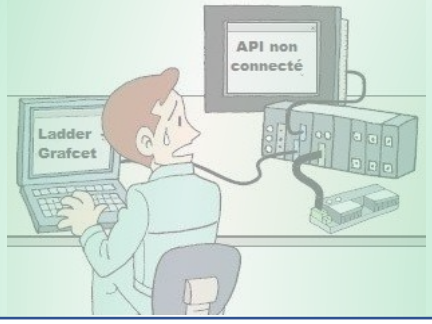
VI) LE SYSTEME DECIMAL CODE BINAIRE OU BCD

VII) LE CODE BINAIRE REFLECHI OU CODE GRAY

VIII) LE CODE ASCII

IX) LE BIT DE PARITE

X) CONVERSION ENTRE LES DIFFERENTS SYSTEMES



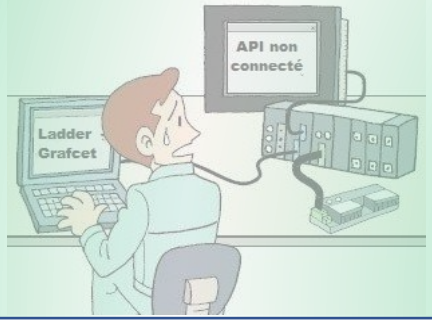
I) Introduction générale

Comme a pu le voir dans les précédentes fiches, du point de vue architecture matérielle, un automate programmable est composé quasiment de la même manière qu'un ordinateur traditionnel. Au niveau même d'un microprocesseur d'un ordinateur ou d'un automate, l'information est codé en binaire après compilation. C'est la raison pour laquelle nous allons voir dans cette fiche le binaire et les autres systèmes utilisés pour coder l'information au niveau des systèmes programmables.

En effet, outre le système décimal avec lequel on n'est tous habitué, il existe d'autres systèmes de numération comme le binaire, l'octal, l'hexadécimal, le BCD, le Gray, l'ASCII etc…

La connaissance des différents systèmes de numération et de codage est très utile lorsque vous travaillez avec les automates programmables. L'hexadécimal reste le système que vous rencontrerez le plus souvent surtout au niveau des logiciels de programmation d'automates.

.



II) Le système de base 10 ou décimale

Un système de numération est une manière de représenter la notion de quantité. En effet, pour représenter la quantité huit par exemple, il est fait usage du symbole « 8 ».Nous connaissons de la sorte dix symboles (0,1,2,3,4,5,6,7,8,9), d'où le nom de système de numération décimal ou système de base dix.

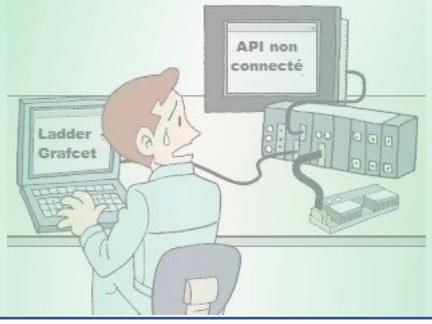
Le système décimal ou système de base 10 est le système que l'on connait le plus. Dans le système décimal, il existe 10 chiffres uniques qui permettent de représenter l'information. Ces chiffres vont de 0 à 9.

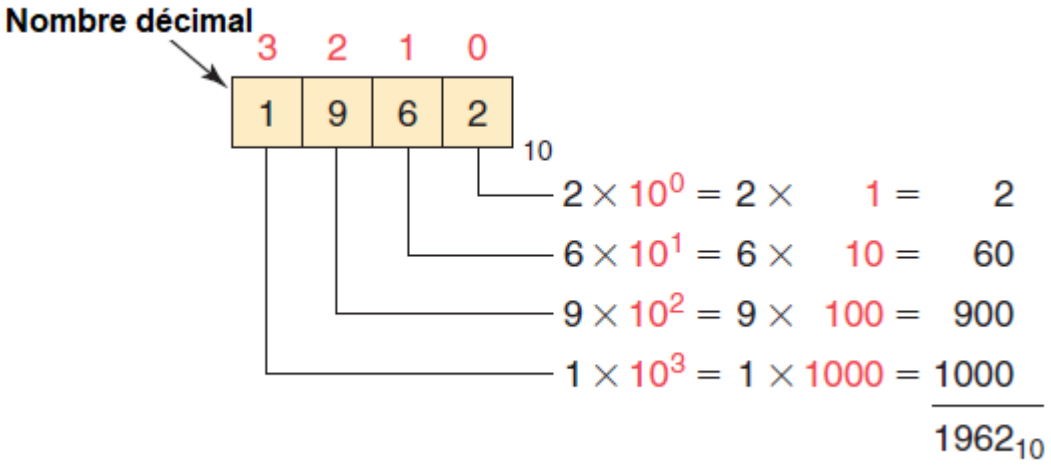
La valeur d'un nombre décimal dépend des chiffres qui le composent et du rang de chaque chiffre. En effet, un rang est attribué à chaque chiffre composant un nombre décimal. Ce rang s'établit de droite à gauche avec le rang 0 qui est attribué au chiffre le plus à droite, le rang 1 qui est attribué au deuxième, le rang 2 au troisième et ainsi de suite jusqu'au dernier chiffre.

Ainsi, lorsque nous voulons exprimer une quantité et écrivons 1962, cela représente : deux unités, six dizaines, 9 centaines et 1 millier ou :

$$1962 = 1000 + 900 + 60 + 2$$
$$= (1 \times 1000) + (9 \times 100) + (6 \times 10) + (2 \times 1)$$

Chaque chiffre correspond à une puissance de 10 lié à son rang. Par exemple, sur le schéma ci-dessous, on peut voir comment sont attribués les rangs au niveau du système décimal.





III) Le système de base 2 ou binaire

a) Généralités

L'exposé ci-dessous peut être généralisé pour toutes les bases. Un nombre quelconque N peut être mis sous la forme :

$N = (abc)n = a.n^2 + b.n^1 + c.n^0$ à la base n

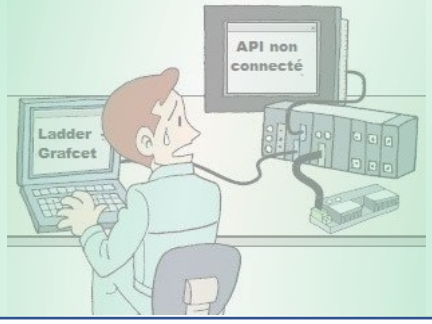
Le système binaire utilise le symbole 2 comme base. Les seuls symboles autorisés sont 0 et 1. Comme on pourra le voir un plutard, au niveau des circuits numériques les informations binaires sont codées avec les symboles 0 et 1.Le 0 symbolise l'absence de tension et le 1 la présence de tension.

Pour exprimer une quantité en système binaire, on peut écrire :

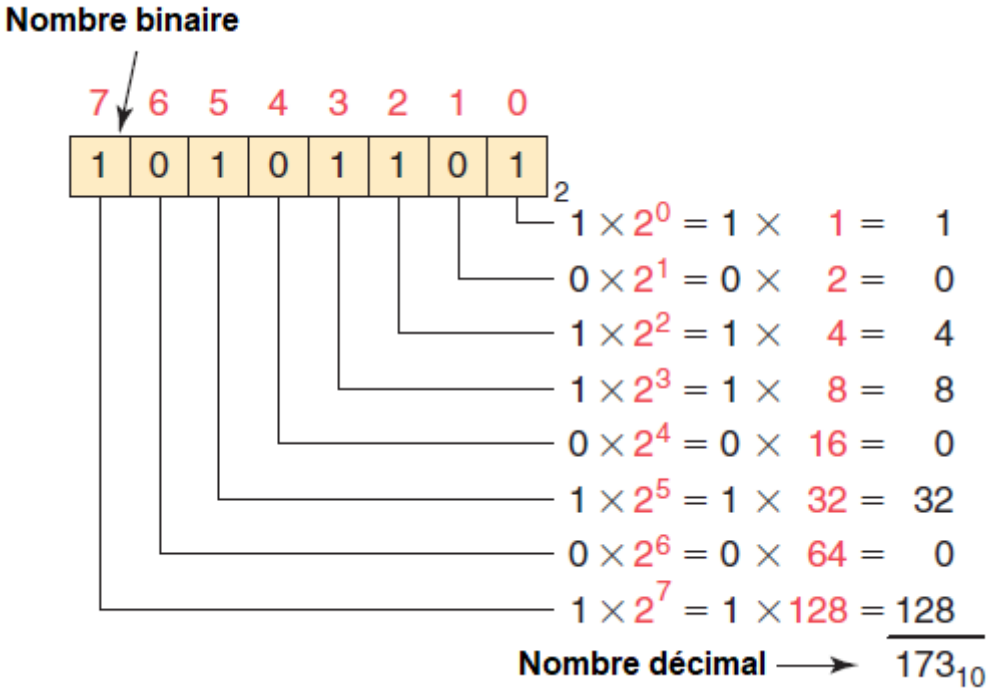
$(1001)_2 = 1.2^3 + 0.2^2 + 0.2^1 + 1.2^0$

$= (1 \times 8) + (0 \times 4) + (0 \times 2) + (1 \times 1) = (9)_{10}$

Cette facilité de représentation a pour inconvénient l'accroissement du nombre de chiffres nécessaires à la représentation du nombre.

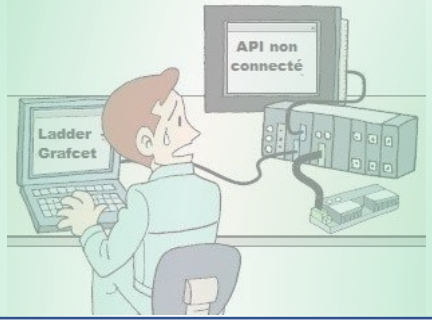


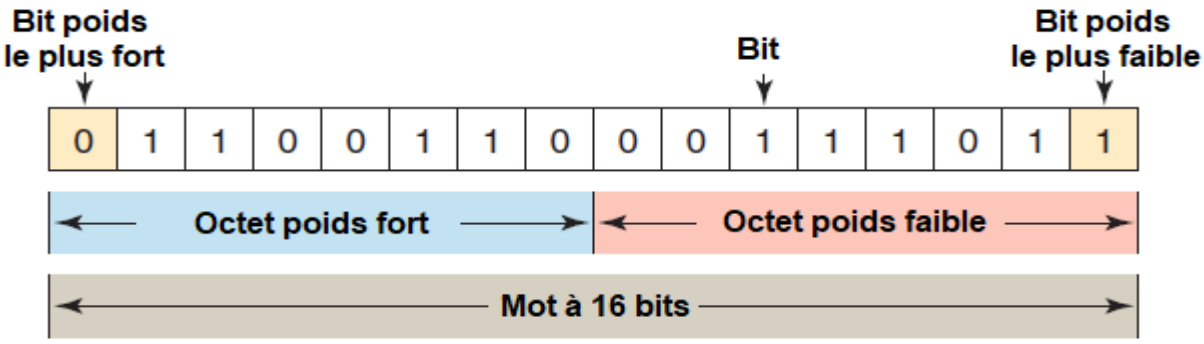
Par exemple comme on peut le voir sur l'image ci-dessous, $(10101101)_2$ correspond à $(173)_{10}$.



Au niveau du système binaire, les nombres peuvent être regroupés en groupes de bits. Un bit ou digit correspond au symbole 0 ou 1.En fonction des automates, un mot ou Word en anglais peut correspondre à un groupe de 16 bits ou un groupe de 32 bits. Un nombre binaire peut aussi être regroupé en groupe de 8 bits ou octet ou encore byte en anglais.

Ainsi un groupe de 8 bits est un octet et un groupe supérieur à 2 octets est un mot. Sur l'image ci-dessous, on peut voir un mot composé de 16 bits réparti en 2 octets.





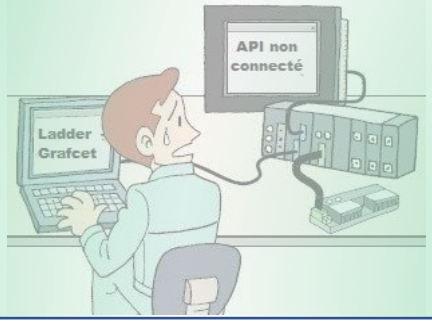
Le bit du poids le plus faible ou Least Significant Bit ou LSB en anglais est le chiffre qui a le rang le plus faible alors que le bit du poids le plus fort ou Most Significant Bit, ou MSB en anglais est le chiffre qui a le rang le plus élevé.

La mémoire d'un automate est organisée en utilisant des octets, des mots simples (WORD) ou des mots doubles (DWORD). La taille de la mémoire de l'automate programmable détermine la quantité d'information que l'utilisateur peut stocker dans celle-ci. Si la taille de la mémoire est de 1 Ko, on pourra y stocker 1024 mots ou 16 384 (1024 x 16) bits de données en utilisant des mots de 16 bits, ou 32 768 (1024 x 32) bits en utilisant des mots de 32 bits.

Même si le système binaire ne comporte que deux symboles, il peut être utilisé pour représenter n'importe quel nombre décimal. Aussi, comme énoncé plus haut, tous les automates travaillent avec le système binaire vu que les microprocesseurs qui les composent sont des composants électroniques qui comprennent uniquement le binaire. Si on consulte donc la mémoire d'un automate on pourra y voir une succession de 0 et de 1.

b) Codage des nombres binaires négatifs

Comme les nombres décimaux, les nombres binaires sont aussi signés. Par contre au niveau des automates, on ne peut pas utiliser les symboles (+) et (-) pour représenter le signe d'un nombre.



Une autre méthode est utilisée. Cela consiste à utiliser un digit supplémentaire qui sera le bit de signe. Ce digit est placé avant le bit du poids le plus fort du nombre. Un 0 indique que le nombre est positif alors qu'un 1 indique que le nombre est négatif. Par exemple 0111 correspond à +7 alors que 1111 correspond à -7.

Il existe une autre méthode pour exprimer un nombre négatif avec le système binaire. Cela consiste à utiliser le complément à DEUX du nombre. En binaire, pour déterminer le complément à DEUX d'un nombre, il suffit d'ajouter 1 à son complément à UN. On obtient le complément à UN d'un nombre en déterminant le complément à UN de chaque chiffre binaire qui le compose. Par exemple :

C_1 (1011) = 0100 (on inverse les chiffres qui composent le nombre un à un)

Ainsi, pour déterminer le complément à DEUX d'un nombre, on appliquera la formule suivante :

C_2 (N) = C_1 (N) + 1

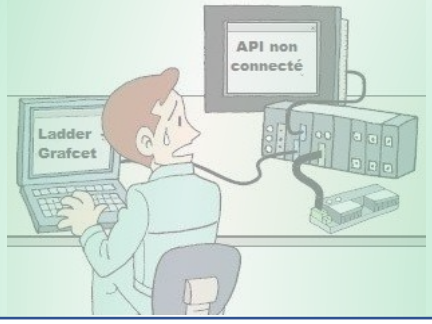
Exemple : C_2 (1011) = C_1(1011) + 1

= 0100 + 1 = 0101

L'utilisation du complément à DEUX facilite la tâche à l'automate et lui permet d'effectuer des opérations mathématiques comme on pourrait le faire avec le système décimal. L'automate sait qu'un nombre stocké dans sa mémoire est négatif si son bit du poids le plus fort est 1.A chaque fois que l'automate reçoit un nombre négatif, il le stocke en tant que complément à DEUX.

c) Les opérations arithmétiques entre nombres binaires

Comme pour les nombres décimaux, on peut additionner, soustraire, multiplier et diviser des nombres binaires. L'addition de deux nombres binaires suit le même principe que l'addition de deux nombres décimaux comme vous pouvez le voir sur l'image ci-dessous.



$$\begin{array}{rrrr} 0 & 1 & 0 & 1 \\ +0 & +0 & +1 & +1 \\ \hline 0 & 1 & 1 & 0 \end{array}$$ On retient 1

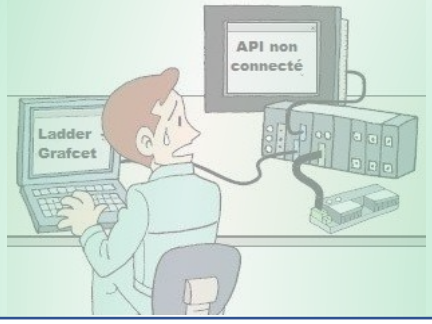
Addition	Soustraction
0 + 0 = 0	0 – 0 = 0
0 + 1 = 1	0 – 1 = 1 (avec retenue de 1)
1 + 0 = 1	1 – 0 = 1
1 + 1 = 10 (0 et un report de 1)	1 – 1 = 0

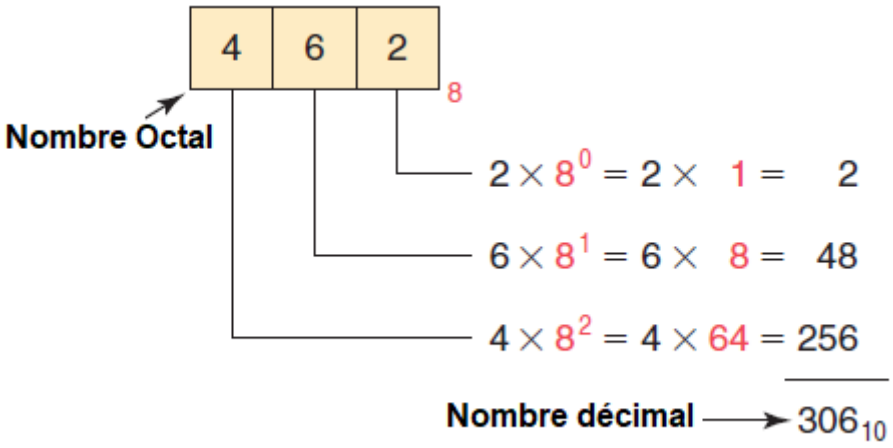
IV) Le système de base 8 ou octal

Exprimer un nombre dans le système binaire nécessite beaucoup plus de chiffres que dans le système décimal. Aussi un nombre binaire peut être très difficile à décortiquer et fastidieux à lire ou écrire. C'est la raison pour laquelle d'autres systèmes de numération connexes comme l'Octal sont utilisés.

Le système de numération octal ou système de base 8 est très pratique vu que 8 bits de données constituent un octet. Dans le système octal, les chiffres vont de 0 à 7; par conséquent, les chiffres 8 et 9 ne sont pas autorisés. Dans certains automates comme l'Allen Bradley SLC 5, le CPU utilise l'adressage en octal pour adresser ses E/S alors que pour l'automate SLC 500, le décimal est utilisé.

L'octal est un système de numération très simple : un nombre octal est plus facile à lire que son équivalent binaire. Comme dans tous les autres systèmes de numération, chaque chiffre d'un nombre octal a un poids dont la valeur dépend de son rang comme on peut le voir sur l'image ci-dessous.

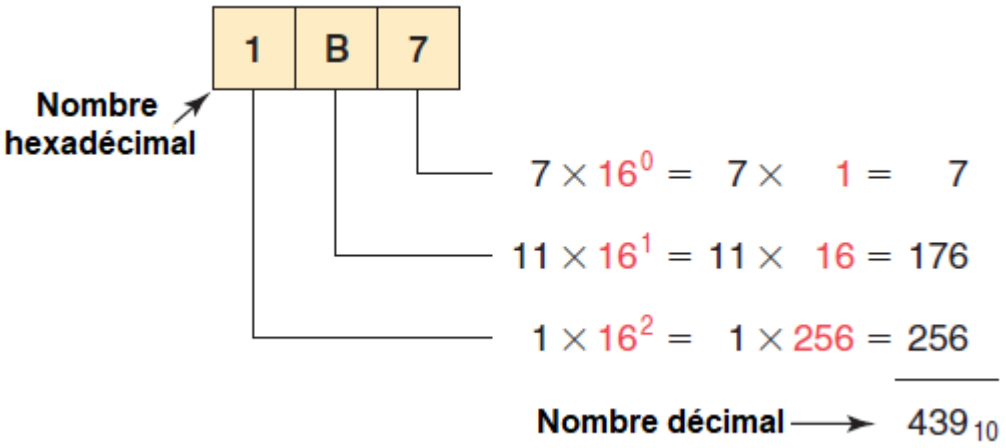


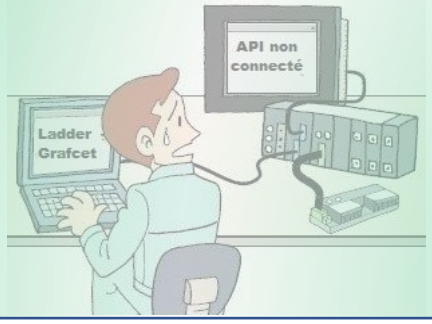


V) Le système de base 16 ou hexadécimale

Le système de numération hexadécimal (HEXA ou HEX) est très utilisé au niveau des contrôleurs et automates vu qu'un mot de données ou Word se compose souvent de 16 bits de données, ou deux octets de 8 bits.

Le système hexadécimal ou système de base 16 utilise les chiffres de 0 à 9 et les lettres de A à F sont utilisés pour représenter les nombres allant de 10 à 15. Le système de numération hexadécimale permet de représenter un grand nombre de bits dans un petit espace.





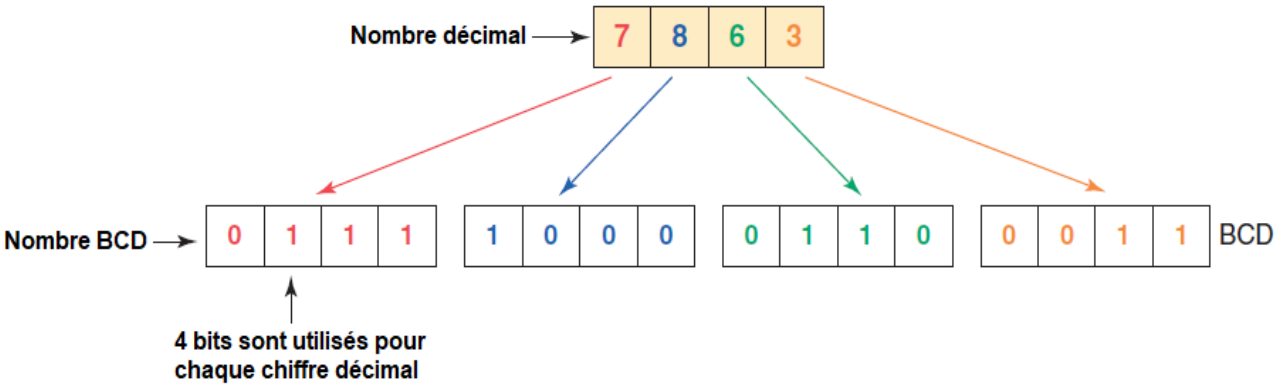
VI) Le système décimal codé binaire ou BCD (Binary Coded Decimal)

Le système BCD fournit un moyen facile permettant de manipuler des nombres binaires. Il est souvent utilisé au niveau des décodeurs des afficheurs 7 segments.

Le système BCD utilise 4 bits pour représenter chaque chiffre décimal. Les 4 bits utilisés sont les équivalents binaires des chiffres allant de 0 à 9.

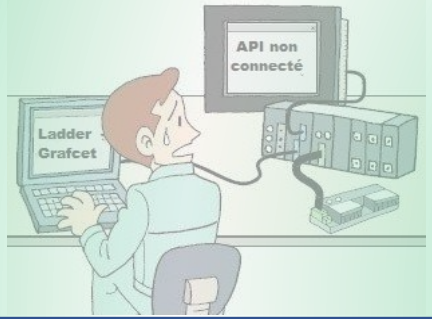
Dans le système BCD, le plus grand chiffre décimal qui peut être utilisé pour coder chacun des 4 bits composant le nombre est 9. La représentation BCD d'un nombre décimal est obtenue en remplaçant chaque chiffre décimal par son équivalent BCD. Pour distinguer le système de numération BCD d'un système binaire ou autre, une désignation BCD est accolée au nombre.

Sur l'image ci-dessous, on peut voir la représentation BCD du nombre 7863.



VII) Le code binaire réfléchi ou code Gray

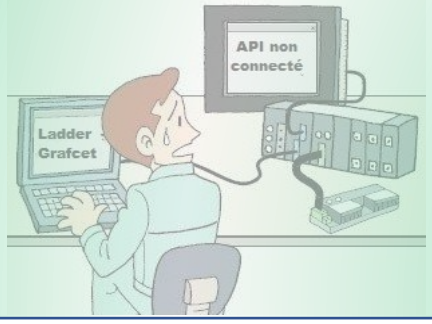
Le code Gray est un type spécial de code binaire qui n'utilise pas la pondération en fonction du rang. En d'autres termes, les chiffres qui composent un nombre n'ont pas de poids défini. Le code Gray est un code dans lequel la transition entre deux entiers consécutifs n'affecte qu'un seul bit à la fois.En d'autres termes, quand on passe d'un chiffre à un autre, un seul bit change. Le code Gray est idéal pour les codeurs. Par exemple, les codeurs absolus sont des capteurs de position qui utilisent le code Gray pour déterminer leur position angulaire.



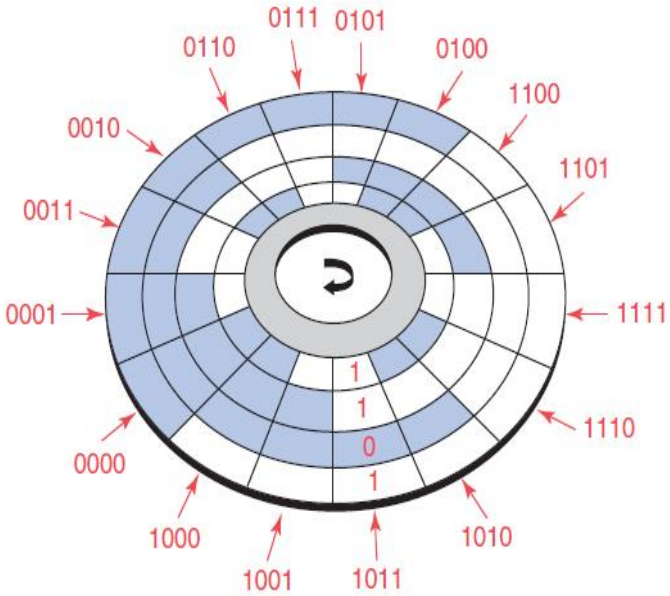
Code Gray	Code binaire
0000	0000
0001	0001
0011	0010
0010	0011
0110	0100
0111	0101
0101	0110
0100	0111
1100	1000
1101	1001
1111	1010
1110	1011
1010	1100
1011	1101
1001	1110
1000	1111

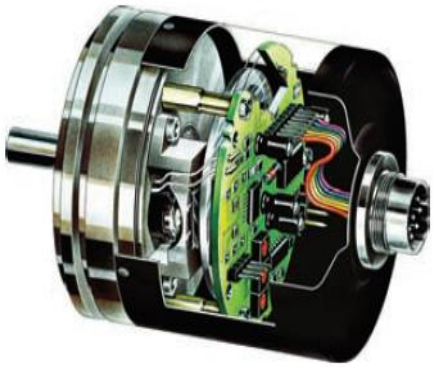
Contrairement au code Gray, en binaire, jusqu'à quatre chiffres peuvent changer à la fois. Par exemple, la transition du binaire 0111 vers 1000 (décimal 7 à 8) implique un changement dans les quatre chiffres. Ce genre de changement augmente la possibilité d'erreur dans certains circuits numériques. Pour cette raison, le code Gray est utilisé pour minimiser les erreurs.

Parce qu'un seul bit change à la fois, la vitesse de transition pour le code Gray est considérablement plus rapide que pour les codes tels que le BCD. Le code Gray est souvent utilisé avec les codeurs de position pour une précision du contrôle du mouvement des robots, des machines-outils et des servomécanismes. Il est également utilisé dans les tableaux de Karnaugh pour la simplification de fonction logique.



Sur l'image ci-dessous, on peut voir un encodeur optique utilisant un code Gray de 4 bits pour détecter les variations angulaires du disque.



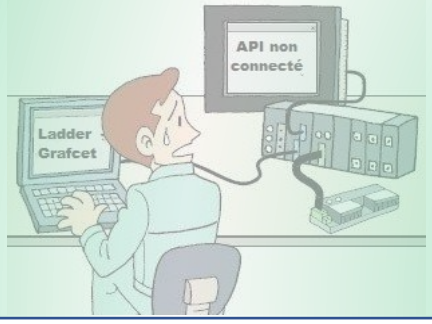


Dans cet exemple, le disque d'encodage est connecté à un arbre rotatif et émet un signal de code numérique de type Gray. Cela est utilisé pour déterminer la position de l'arbre.

VIII) Le code ASCII

Le code ASCII ou American Standard Code for Information Interchange est un code alphanumérique incluant des chiffres, des lettres et des caractères spéciaux.

Le code ASCII est composé de 10 chiffres, 26 lettres minuscules, 26 lettres majuscules et plus de 25 caractères spéciaux. On retrouve le code ASCII au niveau des ordinateurs dans lesquels les frappes sur le clavier sont converties en ASCII.



On verra plutard que l'on peut utiliser le codage ASCII pour coder l'information d'une trame de communication (Exemple : les trames du modbus ASCII).

IX) Le bit de parité

Certains protocoles de communication d'automates utilisent un chiffre binaire supplémentaire afin de vérifier l'exactitude de la transmission des données. Par exemple, lorsque les données sont transférées entre deux automates, un des bits transmis peut accidentellement changés de valeur. Cela peut arriver à cause d'un « bruit électrique » ou à cause de la défaillance d'une partie du réseau de transmission. Un bit de parité est alors utilisé pour détecter les erreurs qui peuvent se produire.

La parité est un système dans lequel chaque caractère transmis contient un bit supplémentaire. Ce bit est appelé bit de parité.

Il existe deux systèmes de parité : la parité impair (odd) et la parité pair (even).

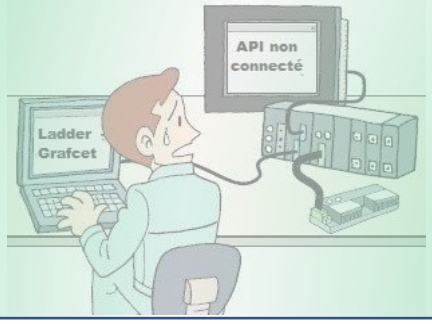
X) Conversion entre les différents systèmes

Pour faire la conversion entre les différents systèmes de numération, vous pouvez vous aider d'une calculatrice scientifique comme des méthodes qui sont montrées ci-dessous.

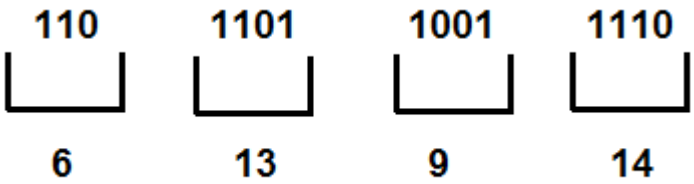
a) Conversion binaire/hexadécimal

La règle est :

- Partager le nombre en groupes de quatre chiffres en commençant par la droite
- Pour chaque groupe, écrire le nombre hexadécimal correspondant
- Remplacer chaque nombre supérieur à 9 par le symbole correspondant (A….F).



Exemple : $(110210110011110)_2$



$= (6D9E)_{16}$

L'astuce de conversion ici est la suivante :

Pour chaque groupe de bits, on dresse une matrice (8 4 2 1) comme ci-dessous :

8	4	2	1
1	1	1	0

Au niveau de la matrice, on additionne les chiffres pour lesquels on a un « 1 ».Par exemple, pour le groupe de bits 1110 on aura :

(8 + 4 + 2) = 14 *(en hexadécimal 14 correspond à E)*

On fait de même pour le groupe de bits (1001).

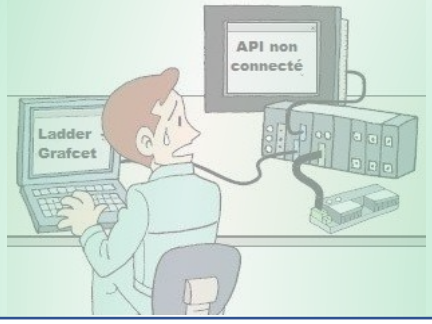
8	4	2	1
1	0	0	1

(8 + 1) = 9

Pour le groupe de bits (1101) on aura :

8	4	2	1
1	1	0	1

(8 + 4 + 1) = 13 *(en hexadécimal, 13 correspond à D)*



Pour le groupe de bits (0110) on aura :

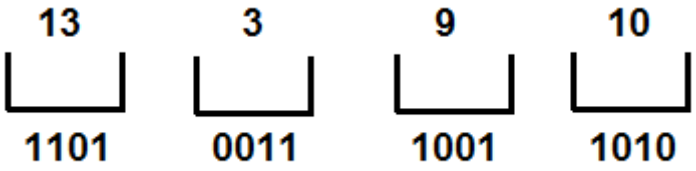
8	4	2	1
0	1	1	0

(4 + 2) = 6

b) Conversion hexadécimal/binaire

Nous procédons dans ce cas-ci à l'opération inverse. Le nombre $(D39A)_{16}$ devient

PS : Les symboles D et A correspondent respectivement à 13 et 10 en hexadécimal



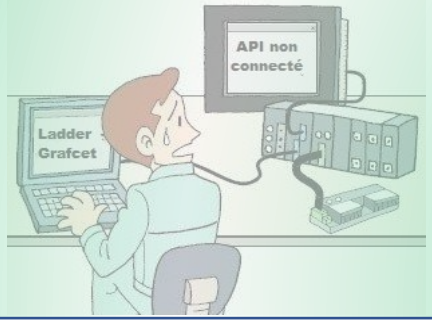
Pour effectuer la conversion, on dresse toujours la matrice

(8 4 2 1)

Pour traduire par exemple 10 en binaire. On se pose la question suivante : parmi les chiffres 8, 4 ,2 ,1 de la matrice, quels sont les chiffres à additionner pour obtenir 10 de la manière la plus rapide. On aura donc 8 et 2 (8 + 2 = 10).On affectera pour chaque chiffre choisi la valeur binaire « 1 », ce qui nous donnera au final (1010) en binaire.

8	4	2	1
1	0	1	0

On va appliquer la même méthode pour les valeurs 13,3 et 9.



c) Conversion décimal vers octal

Il s'agit de faire des divisions successives par 8 et de lire tous les restes dans le sens inverse. Soit à convertir 3817 en octal.

Opération	Quotient	Reste
3817/8	477	0,125
477/8	59	0,625
59/8	7	0,375
7/8	0	7

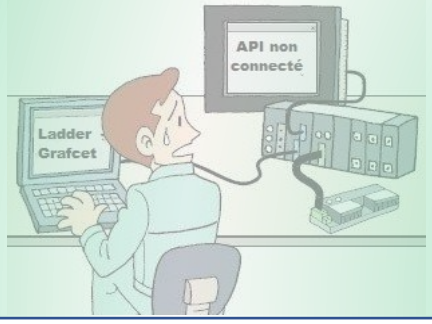
En lisant les restes à partir du bas, on aura les valeurs suivantes :

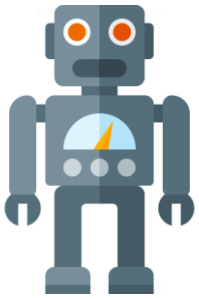
7 - 0,375 - 0,625 - 0,125.

Pour les valeurs décimales (avec virgule), on les multipliera par 8 .Ce qui va donner au final : $(7351)_8$.

7	7
0,375 x 8	3
0,625 x 8	5
0,125 x 8	1

Pour la conversion de décimal à binaire ou hexadécimal, on fera la même opération, mais au lieu de diviser par 8, on divisera par 2 en binaire et par 16 en hexadécimal.





Dans cette fiche vous avez pu découvrir les différents systèmes de numération et de codage qui peuvent être utilisés au niveau des automates programmables.

Vous pouvez tester vos connaissances en effectuant la fiche de test qui va suivre.