

# C PARTY : Un exposé applicatif du langage C pour les microcontrôleurs

## Sommaire

---

### Chapitre 1 : PRESENTATION GENERALE

- 1 C pourquoi ? La genèse de C PARTY
- 2 C pour qui ?
- 3 C facile ?
- 4 C pratique ?
- 5 C long ?

### Chapitre 2 : NUMERATION ET CONVERSIONS

1. Décimal, Binaire, bit, hexa, BCD, octets, etc. C par ici...
2. La logique combinatoire : ET, OU, NAND, NOR, EXOR
3. Décalage à droite, décalage à gauche, masquages
4. Les nombres signés et non signés

### Chapitre 3 : LE LANGAGE DE PROGRAMMATION C

1. Structure d'un programme
2. Les variables
3. Les constantes
4. Les tableaux
5. Les opérateurs arithmétiques, relationnels, logiques, d'affectations, conditionnels
6. Les tests de sélections → If et Switch
7. Les boucles → for(), while(), do... while()
8. Préprocesseur: opérateurs et commandes associés
9. Les fonctions
10. Mixer du C et de l'assembleur.

### Chapitre 4 : RESSOURCES MICROCONTROLEURS 8 bits

1. Caractéristiques générales des microcontrôleurs
2. Alimentation
3. Mémoires RAM, EEPROM, FLASH EEPROM
4. Horloges
5. Ports d'entrée/sorties
6. Timers 8 bits et 16 bits / Capture-Compare-PWM (CCP)
7. Watchdog
8. Reset
9. Les interruptions
10. Convertisseur Analogique-Numérique (ADC) 8bits, 10 bits et 12 bits
11. Comparateurs Analogiques
12. Le port série (USART)
13. Le bus I2C
14. Le mode dodo (Sleep mode)

## Chapitre 5 : APPLICATIONS

### 1) LEDES

- Une led qui clignote
- Une led qui clignote à une fréquence précise (timer + interruption)
- Deux leds qui clignent à des vitesses différentes (timer+ interruption)
- Une led + un bouton poussoir → Variation de luminosité (PWM)
- 8 leds + un bouton poussoir → variations lumineuses, effet scanner, etc.
- Une led bicolore
- Un stroboscope à leds
- Un vu-mètre à leds (voltmètre/ampèremètre)
- Matrice de leds ou comment optimiser les ports du  $\mu\text{C}$
- Un dé électronique
- Mise en œuvre d'une led de forte puissance

### 2) Afficheur 7 segments

- Compteur/Décompteur 1 digit
- Compteur/Décompteur 2 digits (technique du multiplexage)
- Compteur/Décompteur 4 digits (technique du multiplexage)
- Voltmètre/Ampèremètre 4 digits (multiplexage et ADC)
- Horloge temps réel → DS1337+2 boutons+4 digits (I2C, interruptions, multiplexage)

### 3) Afficheur LCD alphanumériques

- Câblage, initialisation, mise en œuvre
- Affichage de caractères spéciaux (utilisation CGRAM)
- Voltmètre/Ampèremètre
- Fréquence-mètre, Période-mètre
- Capacimètre, inductancemètre, milliOhmmètre
- Mesure de distance avec des ultra-sons
- Mesure de distance avec des leds infrarouges
- Une calculatrice

### 4) Buzzer et haut-parleurs

- Interfaçages
- Générer des fréquences
- Un mini piano
- Une sirène Américaine

### 5) Port série (USART)

- RS232 Câblages
- RS485 Câblages
- Piloter son  $\mu\text{C}$  par PC simplement.

### 6) Bootloader

- Bootloader pour PIC18F8722

## Chapitre 1 :PRESENTATION GENERALE

---

### 1 C pourquoi ? La genèse de C PARTY

Si vous tapez sous Google les mots clés suivants : C+microcontrôleur vous obtenez, à la date où sont écrits ces lignes, plus de 954.000 résultats, donc tout porterait à croire qu'apprendre le C orienté microcontrôleurs est à la portée de tous et que tout cela est d'une banalité totale.

Il n'en est rien.

Il suffit pour s'en convaincre de piocher au hasard quelques résultats pour en tirer quelques constats et émettre quelques critiques :

- Cours très complets sur le C mais aussi très théoriques, très généralement illustrés d'exemple orientés vers la console PC.
- Cours survolant (trop) les bases fondamentales de numération, les vocabulaires ou les méthodes de raisonnements, d'organisation, de choix.
- Applications vers les microcontrôleurs n'expliquant que rarement les méthodes de réflexions, les calculs ou encore les mécanismes d'utilisation ou de configurations des ressources matériels.
- Trop souvent des codes extrêmement mal commentés ou, nous verrons que ce point est très important en programmation.
- Trop souvent des codes issus de compilateurs non C ANSI et donc non portable d'un  $\mu$ C à l'autre, alors que c'est un des arguments fort de ce langage à condition de faire ce qu'il faut.
- Des documents issus de l'enseignement, biens faits mais qui répondent à des exigences de programmes scolaires avec des choix trop souvent orientés commercialement (marque de compilateur, marque de microcontrôleurs, etc) et ne répondant pas toujours à mon sens à une nécessaire universalité ou diversité...
- Ou encore ne traitant que d'aspects trop limitatifs pour avoir une vision globale.
- Je passe sur les sites ou blogs de bricoleurs du dimanche qui disséminent un tas d'énormités, d'inexactitudes qui étrangement reçoivent plus de crédits et pénètrent plus durablement, plus insidieusement les esprits.

Aujourd'hui la très grande majorité des sites dédiés aux microcontrôleurs est essentiellement axée sur les PIC, laissant croire aux novices qu'il n'existerait que cette marque, que cette solution.

Pire encore, que cette référence est incontournable, la meilleure preuve étant sa prolifération.

C'est évidemment faux.

Juste que commercialement ils ont mieux fait leur boulot sans doute et aussi que Microchip est arrivé au bon moment et avec de bons arguments, à l'époque tout du moins...

Comme dans ce domaine il n'est nulle éternité, ni supériorité indéniable et incontournable, il est donc souhaitable de ne pas rester prisonnier et d'avoir les idées aussi larges que possible en ce domaine.

Les novices sont souvent très effrayés par l'assembleur, même si de très bon cours existent et notamment celui de Bigonoff qui fait référence en la matière sur le net.

L'assembleur possède de nombreux avantages et aussi quelques restrictions notamment d'être lié à une marque de microcontrôleur voir à une famille dans une même marque et intransportable en terme de code.

Egalement il ne permet pas une maintenance aisée dans des applications lourdes.

L'époque a changé, autrefois les émulateurs coûtaient très chers et l'assembleur était souvent le moyen bon marché de faire du développement dans les petites structures, les microcontrôleurs avaient des ressources très limités ce qui imposait naturellement l'emploi de l'assembleur.

Et puis il y avait surtout quelque chose d'assez nouveau dans le milieu de l'électronique qui historiquement était depuis toujours dominé par la compétence analogique : l'émergence de la vague numérique avec l'apparition de gens à compétences informatique dont l'assembleur est la langue maternelle.

Bref, aujourd'hui pour des raisons d'efficacités et de rendement, par le fait que les microcontrôleurs d'aujourd'hui possèdent d'énormes ressources mémoires, l'assembleur perd du terrain au profit des langages dit « évolués » tels que C, Pascal, Basic.

Nous verrons dans cet exposé que la meilleure efficacité se trouve souvent dans l'usage conjoint de l'assembleur et du C, le C étant le langage le plus proche de l'assembleur, langue native du microcontrôleur.

Le C comme tous langages évolués masquent en partie l'intimité du matériel ce qui la plupart du temps est plutôt un avantage quand on doit écrire une application complexe mais qui peut nécessiter l'usage de l'assembleur pour par exemple optimiser l'usage de certaines ressources ou écrire des *drivers* performants et rapides.

Par contre quand il s'agit de faire des calculs complexes le C s'avère bien plus efficient, vous l'aurez compris il n'est pas question d'abandonner l'assembleur mais d'exploiter à bon escient les avantages respectifs de l'un et de l'autre.

## 2 C pour qui ?

- C'est bien sur pour tous et plus largement pour tous ceux qui se promettent depuis des années de s'y mettre, tout en remettant aux calendes grecques la date effective...
- L'étudiant de terminale y trouvera un (très constructif) complément aux cours qu'il reçoit.
- L'électronicien à la retraite mais passionné par son métier et qui n'a jamais eu l'occasion de se mettre à la page de cette technologie.
- L'électronicien qui cherche à ajouter une nouvelle corde à son arc, ou plutôt à son CV.
- Le non électronicien qui a pour loisirs l'électronique.
- Le scientifique qui n'a pas une formation spécifique à l'électronique mais qui a des besoins qui nécessitent dans son corps de métier des compétences en microcontrôleurs, par exemple dans le cadre de matériels pour compléter ou réaliser une expérience, un outil spécifique.
- L'informaticien qui viendra chercher ici davantage d'informations pratiques sur le coté matériels des microcontrôleurs.
- Plus globalement tous les gens curieux d'apprendre quelque chose d'actuel et de particulièrement généralisé dans les technologies modernes des équipements qui nous entourent.
- Pourquoi pas aussi des enseignants cherchant des exemples d'applicatifs pour leurs cours, cet exposé étant libre de droit et aucunement restrictif en terme d'usage.
- Ceux qui pensent que l'on ne peut pas compter jusqu'à 10 avec deux doigts seulement (lol).
- Enfin ceux qui connaissent un autre langage comme basic et qui ne réalisent pas encore qu'ils vont se faire contaminer grave, qu'ils n'en ressortiront pas indemnes.
- Les adeptes de l'assembleur... non je déconne, ceux-là sont quasiment en religion avec l'assembleur donc peu de chance de les voir débouler ici.
  
- Et j'ai le droit de rêver un peu ... si cela pouvait convaincre le monde de l'enseignement de bien vouloir bouter énergiquement hors de nos écoles tous ces pseudos logiciels comme flowcode par exemple, dits intelligents, qui n'enseignent rien, n'amènent à rien, ne servent à rien sauf à engourdir des esprits qui ne demandent qu'à progresser et comprendre le sens des choses.

Pour conclure sur le sujet de l'intelligence :

**«L'intelligence c'est ce qui se passe quand rien n'empêche l'intelligence de fonctionner »** Louis Pauwels.

Et je partage à 100% cette idée d'où les fondements de cet exposé.

### 3 **C facile ?**

Question légitime que les plus anxieux se posent déjà j'en suis sûr.

Ben « oui et non » comme disent les Normands.

Oui si on a un minimum de bon sens et de logique.

Non car seul le temps apporte l'expérience qui servira de véritable ciment à ces nouvelles connaissances pas forcément naturelles au premier abord.

Mais rassurez-vous, le principe que j'adopte est le même que j'emploie lors des nombreuses formations que j'ai été amené à conduire dans le cadre de mes activités professionnelles depuis 25 ans. J'essaye de me mettre toujours à la place d'un débutant absolu, j'estime qu'il n'y a jamais de mauvaises questions quand on apprend, bien souvent la « mauvaise » question vient d'un manque de clarté de la part de celui qui enseigne ou forme.

« Ce que l'on conçoit bien s'énonce clairement » telle est ma devise.

Bon, même si ce n'est pas toujours évident d'être en forme ou inspiré chaque jour...

Je dirais que pour pouvoir suivre cet exposé il faut les dispositions suivantes :

- L'envie
- La curiosité
- La pugnacité

Accessoirement si vous avez un niveau de 3<sup>ème</sup> vous êtes aptes à tout comprendre.

En tout cas si vous savez compter jusqu'à 10 vous êtes bien équipé pour suivre jusqu'au bout.

La différence avec un exposé devant de vrais gens est que d'ici je n'ai aucun recul ni aucun retour en temps réel de la manière dont les gens perçoivent l'exposé.

C'est pour cela que le document sera en permanence remis en forme, enrichi et complété.

La version est toujours visible en haut de chaque page, donc la mise à jour consistera à tout simplement écraser une version antérieure.

N'imprimez pas !

Vous allez gâcher du papier inutilement, en tout cas dans un premier temps, les versions vont évoluer très rapidement donc attendez au moins le début des applications pour imprimer ce qui précède.

### 4 **C pratique ?**

Ca c'est un des objectifs de ce projet.

J'ai prévu de faire des fiches très complètes et individuelles sur chaque sujet.

Par exemple sur les décalages et le masquage en un coup d'œil vous savez à quoi ça correspond, j'ai favorisé le côté visuel, manière bande dessinée, même un non francophone pourra comprendre donc à priori même un rappeur...

L'intérêt également est que le plan de l'exposé n'est pas restrictif, ce qui nous permettra selon l'évolution des choses à enrichir progressivement les exemples d'applications.

Celui qui voudra réaliser un exemple, aura un schéma complet, des explications complètes, un code 100% fonctionnel, il sera à même de pouvoir le modifier.

Mieux qu'un livre, là il suffira d'imprimer uniquement la section concernée.

Même pas peur d'Hadopi, vous pouvez télécharger autant que vous voulez, vous ne devrez rien à personne.

### 5 **C long ?**

Ben ça ne dépendra que de vous, du temps que vous y consacrerez.

Quelqu'un de normalement constitué devrait en moins de un mois avoir acquis les principes de bases lui permettant de reconnaître tous les éléments de base du langage, être capable de faire clignoter une led voir plusieurs.

Je suis persuadé que la grande majorité sera bien au-delà dans un mois, en tout cas que le virus du C vous aura contaminé au point que vous ne voudrez plus autre chose pour programmer vos précieux microcontrôleurs.

Il n'y aura plus que votre imagination pour limiter vos projets les plus fous, bientôt vous parlerez couramment le C à votre  $\mu$ C adoré.

Pour atteindre notre objectif et arriver dans le vif du sujet équipé des outils indispensables nous allons passer par une petite phase de révision des principes élémentaires de la numération afin de bien comprendre comment et de quoi est composé un nombre, comment le manipuler afin de le traduire dans le langage natif de la machine et réciproquement pour exploiter ses résultats.

Chaque partie sera illustrée d'exemples commentés et conclue par un résumé des notions les plus indispensables pour aborder le chapitre suivant.

## Chapitre 2 : NUMERATION ET CONVERSIONS

Depuis que l'homme possède deux mains, et donc 10 doigts à moins accident, il compte en base 10 que l'on appelle communément le système décimal.

Le système décimal permet de prendre en compte 10 valeurs distinctes : 0,1,2,3,4,5,6,7,8,9.

Depuis que l'ordinateur existe celui-ci ne comprend que deux états possibles : VRAI ou FAUX ou encore OUI ou NON, deux états possibles d'où l'appellation binaire relative à la base 2.

Il est donc naturel et suffisant pour lui de manipuler deux valeurs numériques distinctes: 0 et 1

En base 10 on appelle chaque élément des chiffres.

En base 2 on appelle chaque élément des chiffres binaires qui en anglais se dit : *binary digit*.

Ce terme contracté est plus couramment nommé : bit

Pour que l'homme puisse dialoguer avec sa machine il va donc être nécessaire d'effectuer des conversions numériques pour passer de la base 10 à la base 2 et vice-versa.

Voyons simplement comment un nombre décimal se compose en faisant un petit rappel mathématique (quelques minutes de transpiration indispensables)

Par exemple prenons au hasard le nombre **1024 en base 10**.

On peut exprimer littéralement que 1024 c'est **une fois 1 millier plus 0 centaine plus deux dizaines plus quatre unités**.

Ce qui mathématiquement se traduit par la forme canonique :  $(1 \times 1000) + (0 \times 100) + (2 \times 10) + (4 \times 1)$

On sait par ailleurs que  $1000 = 10 \times 10 \times 10$  ce qui peut s'écrire plus aisément :  $10^3$  (3 représente le nombre de fois que l'on multiplie 10 par lui-même)

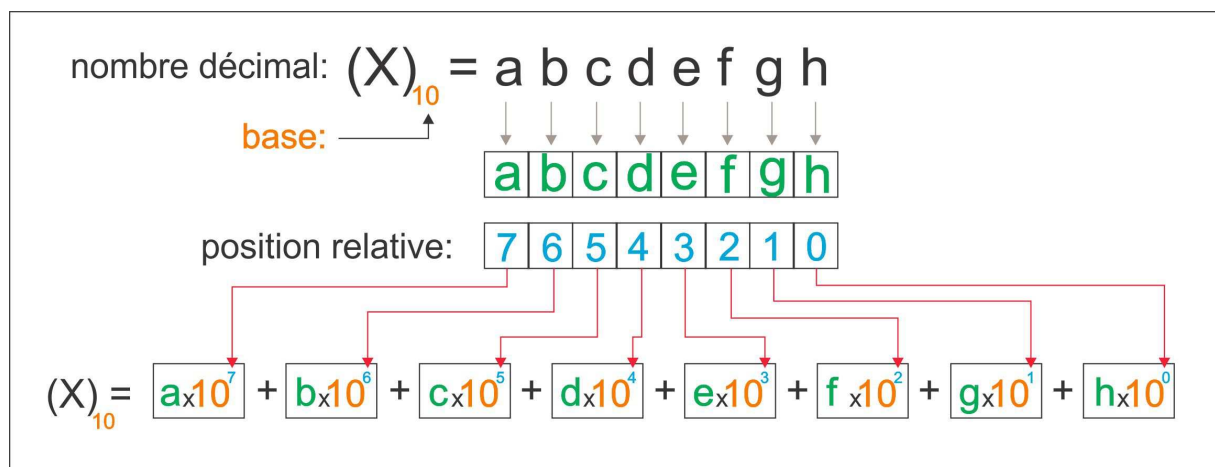
De même  $100 = 10 \times 10 = 10^2$  puis  $10 = 10^1$  et enfin  $1 = 10^0$

Nous pouvons donc écrire que 1024 en base décimale se traduit mathématiquement par :

$$1024 = 1 \times 10^3 + 0 \times 10^2 + 2 \times 10^1 + 4 \times 10^0 \quad 1,0,2,4 \text{ étant les quantités de chaque rang.}$$

Cette écriture purement mathématique n'a d'intérêt que de faire apparaître clairement le format de la base ainsi que la quantité propre à chaque rang.

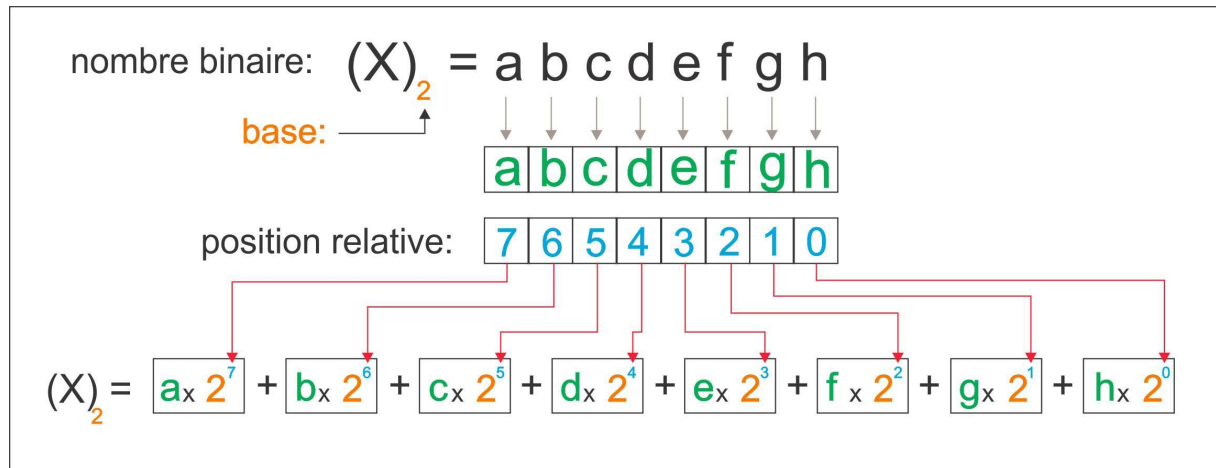
Le tableau suivant rassemble ce que nous venons de voir pour un nombre décimal composé de 8 chiffres quelconques :



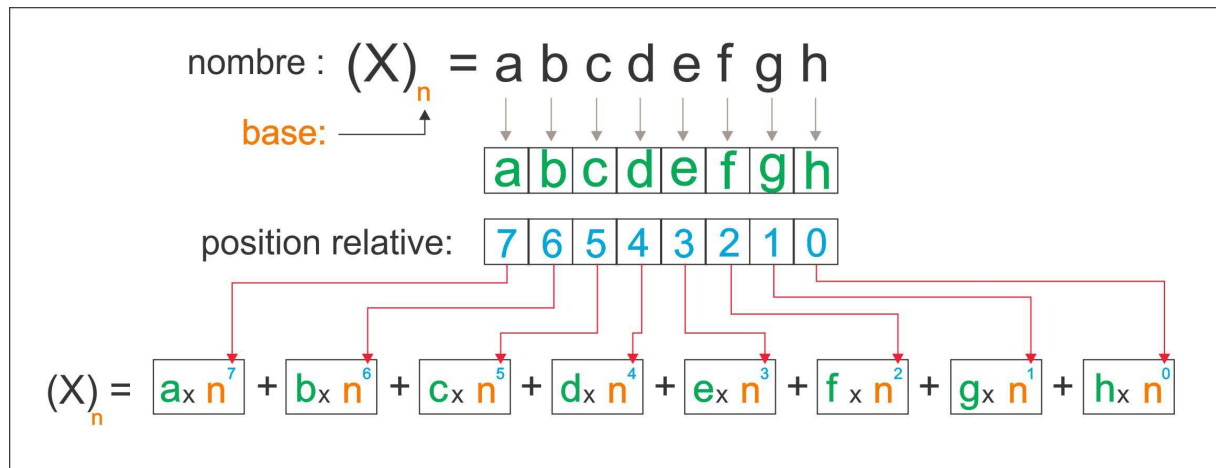
De ce qui précède nous pouvons déjà déduire un certain nombre de choses à retenir :

1. Une base B est composée de N chiffres distincts (base 10 → 10 chiffres, base 2 → 2 chiffres).
2. Le chiffre le plus grand dans cette base B vaut N-1 (base 10 → 9 base 2 → 1)
3. Un nombre X peut se décomposer par une somme de produits, chaque produit étant composé de la quantité du rang multiplié par la base B à la puissance correspondant à ce rang (la puissance exprime le nombre de fois que l'on multiplie la base B par elle même).

Voici le même type de représentation pour un nombre X en base binaire :



Et enfin pour un nombre X dans une base quelconque :



Riche de ces enseignements nous pouvons traduire un nombre binaire en son équivalent décimal.

