

CHAPITRE II : LE MICROCONTROLEUR PIC 16F876A

I- Introduction

Comme nous l'avant mentionner dans le premier chapitre notre carte de commande est basé sur le microcontrôleur 16F876. Etant donné l'utilité de ce composant nous avons jugés nécessaire de consacré un chapitre qui décrit ce type de circuit intégré.

La connaissance des différentes caractéristiques de ce composant va nous aider à mieux concevoir notre montage.

L'avènement des microcontrôleurs, qui associent au microprocesseur de base un programme intégré au circuit, ainsi que des périphériques et de la mémoire, a permis de faire évoluer les montages vers encore plus de simplicité et de rapidité, les périphériques étant intégrés au circuit. Les microcontrôleurs sont aujourd'hui implantés dans la plupart des réalisations grand public ou professionnelles, ils gèrent au plus juste et au plus vite les applications. Il existe aujourd'hui de nombreuses familles de microcontrôleurs.

La société américaine Microchip Technologie a mis au point dans les années quatre-vingt-dix un microcontrôleurs CMOS (Complementary Metal-Oxide Semi-conducteur), le PIC. Ce composant, très utilisé à l'heure actuelle, est un compromis entre simplicité d'emploi, rapidité et prix de revient.

Les PIC font partie de la famille des microcontrôleurs, ils possèdent notamment un jeu d'instructions réduit, ce qui caractérise la famille des circuits RISC (Reduced Instruction Set Computer). Ils sont appréciés pour leur rapidité d'exécution et leur simplicité de mise en œuvre

II- Les microcontrôleurs :

Un microcontrôleur se présente comme étant une unité de traitement de l'information de type microprocesseur contenant tous les composants d'un système informatique [7], à savoir microprocesseur, des mémoires et des périphériques (ports, timers, convertisseurs...). Chaque fabricant a sa ou ses familles de microcontrôleur. Une famille se caractérise par un noyau commun (le microprocesseur, le jeu d'instruction...). Ainsi les fabricants peuvent présenter un grand nombre de pins qui s'adaptent plus au moins à certaines tâches. Mais un programmeur connaissant une famille n'a pas besoin d'apprendre à utiliser chaque membre, il lui faut connaître juste ces différences par rapport au père de la famille. Ces différences sont souvent, la taille des mémoires, la présence ou l'absence des périphériques et leurs nombres (voir annexe).

L'utilisation des microcontrôleurs pour les circuits programmables à plusieurs points forts et bien réels. Il suffit pour s'en persuader, d'examiner la spectaculaire évolution de l'offre des fabricants de circuits intégrés en ce domaine depuis quelques années.

Nous allons voir que le nombre d'entre eux découle du simple sens.

- Tout d'abord, un microcontrôleur intègre dans un seul et même boîtier ce qui, avant nécessitait une dizaine d'éléments séparés. Il résulte donc une diminution évidente de l'encombrement de matériel et de circuit imprimé. [8]
- Cette intégration a aussi comme conséquence immédiate de simplifier le tracé du circuit imprimé puisqu'il n'est plus nécessaire de véhiculer des bus d'adresses et de donnée d'un composant à un autre.
- L'augmentation de la fiabilité du système puisque, le nombre des composants diminuant, le nombre des connexions composants/supports ou composants/circuits imprimer diminue.
- Le microcontrôleur contribue à réduire les coûts à plusieurs niveaux : [8]
 - Moins cher que les autres composants qu'il remplace.
 - Diminuer les coûts de main d'œuvre.
- Réalisation des applications non réalisables avec d'autres composants.

III- Caractéristiques générales du PIC 16F876A

Le PIC 16F876 fait partie des sous-familles des PICS 16F87X. Leurs caractéristiques techniques sont les suivantes :

III-1- Mémoire [9]

Les mémoires du PIC 16F876 sont divisées en trois parties :

- **Mémoire Programmable**

Une mémoire contenant le programme à exécuter par le microcontrôleur. Cette mémoire a la particularité de sauvegarder en permanence les informations qu'elle contient même en absence de tension. La mémoire programme de notre PIC 16F876 est de type flash qu'on peut effacer électriquement, alors on peut la reprogrammer plusieurs fois. La capacité de cette mémoire est de 8 KO.

- **EEPROM**

La mémoire EEPROM (Electrical Erasable Programmable Read Only Memory) est constituée de 256 octets, qu'on peut lire et écrire depuis notre programme. Ces octets sont conservés après une coupure de courant et sont très utiles pour conserver des paramètres semi-permanents. Leur utilisation implique une procédure spéciale, car ce n'est pas de la RAM, mais bien une ROM de type spéciale. Il est donc plus rapide de la lire que d'y écrire.

- **RAM**

Une mémoire vive également appelée RAM (Random Access Memory). Cette mémoire permet de sauvegarder temporairement des informations. Il est à noter que le contenu d'une RAM n'est sauvegardé que pendant la phase d'alimentation du circuit. La mémoire RAM disponible du PIC 16F876 est de 368 octets. Elle est organisée en 4 banques, dans chacune des banques nous allons trouver des cases mémoires spéciales appelés **registres spéciaux** et des cases mémoires libres.

III-2- Registre :[7]

Il existe plusieurs types de registres. Parmi les registres les plus utilisés on peut citer :

- Registre OPTION.
- Registre STATUS.
- Registre INTCON.

IV- Caractéristiques principales du PIC 16F876A :[8],[7]

Les caractéristiques principales du PIC16F876 sont les suivantes :

- Fréquence de fonctionnement élevée : DC-20 MHz.
- Période d'exécution d'une instruction : DC-200 ns.
- De plus de mémoire de données RAM – 368 x 8 Octets (répartie sur 4 banques).
- De plus de mémoire de données FLASH – 8k mots de 14Bits.
- De plus de mémoire de données EEPROM – 256 x 8 Octets.
- De plus de ports d'entrée/sortie (22 E/S) : PORTA (6 lignes), PORTB (8 lignes), PORTC (8 lignes).
- Chaque sortie peut fournir un courant maximum de 25mA.
- Trois temporisateurs (Timers) :

Timer 0 (8 bits)

Il peut être incrémenté par des impulsions extérieures via la broche (TOCKI/ RA4) ou par l'horloge interne ($F_{osc}/4$).

Timer1 (16 bits)

Il peut être incrémenté soit par l'horloge interne, par des impulsions sur la broche T1CKI/RC0 ou par un oscillateur (RC ou quartz) connecté sur les broches TOSO/RCO et T1OSI/RC1.

Timer 2 (8 bits)

Il est incrémenté par l'horloge interne, celle qui peut être pré-divisé.

- De nouvelles fonctionnalités (port série MSSP ou USART).
- Module convertisseur analogique/numérique à plusieurs canaux (AN0 jusqu'à AN4) d'une résolution de 10 bits.

Deux modules (CCP1, CCP2) : Capteur / Comparateur / PWM.

- 1- Capteur de 16-bit, résolution maximale : 12.5 ns.
- 2- Comparateur de 16-bit, résolution maximale : 200 ns.
- 3- PWM (Modulation de largeur d'impulsions), résolution maximale : 10 bit.
 - Utilisent une architecture RISC (Reduce Instruction Set Computer → Processeur à jeu d'instructions réduit). Haut performance du CPU (RISC CPU).
 - Option de sélection de l'Oscillateur plus facile (Oscillateur programmable).
 - Faible consommation, mémoire FLASH/EEPROM haute technologie CMOS.
2mA à 5V, pour : $F_{quartz} = 4\text{Mhz}$. 20 μA à 3V, pour : $F_{quartz} = 32\text{Khz}$.

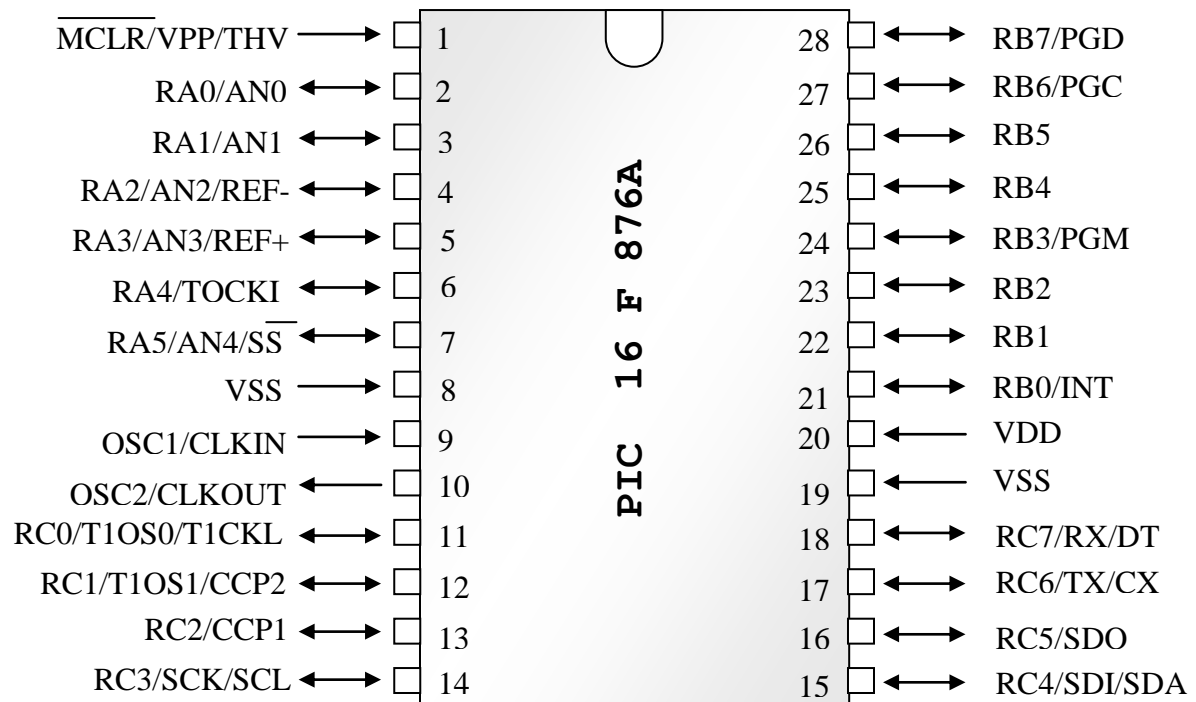
VI-1- Description des broches :[9]

Figure Ch.2-1 Branchage de PIC 16F876.

PORTA (6 lignes)

Le port A peut être configuré comme une Entrée/Sortie numérique bidirectionnelle ou bien :

- Comme des canaux d'entrées analogiques (AN0 à AN4).
- Comme entrées des tensions de références (RA2/AN2/ V_{REF-} , RA3/AN3/ V_{REF+}).
- Comme un sélecteur de port série synchrone (\overline{SS}).

PORTB (8 lignes)

Le port B peut être configuré comme une Entrée/Sortie numérique bidirectionnelle ou bien :

- Comme une broche d'alimentation de programmeur (PGM).
- Comme une broche d'horloge de programmation (PGC).
- Comme une broche des données de programmeur (PGD).

Les lignes PGM, PGC, PGD permettent de programmer le PIC avec basse tension (5V).

PORTC (8 lignes)

Le port C peut être configuré comme une Entrée/Sortie numérique bidirectionnelle ou bien :

- Comme sortie d'oscillateur, entrée d'horloge de TIMER1 (RC0/T10S0/T1CKI).
- Comme entrée oscillateur de TIMER1 ou entrée de capture 2 / sortie de comparateur 2 / sortie de PWM2 (RC1/T10SI/CCP2).
- Comme entrée/sortie de l'horloge de synchronisation série de module MSSP (SDA/SDO).
- Comme entrées/sortie de module de transmission série des données (RC6/TX/CK), (RC7/RX/DT).

VI-2- Interruptions :[7],[9]

Le PIC 16F876 dispose de plusieurs sources d'interruptions.

- Une interruption externe, action sur la broche INT/RB0.
- Débordement du TIMER0.
- Changement d'état logique sur une des broches du PORTB (RB4 à RB7).
- Une interruption d'un des périphériques (PEIE).
 - Fin de conversion analogique numérique (ADIE).
 - Réception d'une information sur la liaison série (RCIE).
 - Fin de programmation d'une case mémoire de l'EPROM.
 - Changement d'état sur le PORTD (PSPIE).
 - Fin d'émission d'une information sur la liaison série (TXIE).
 - Interruption SPI ou I2C du module MSSP (SSPIE).
 - Interruption du registre de capture et/ou de comparaison 1 (CCPI1E).
 - Interruption du registre de capture et/ou de comparaison 2 (CCPI2E).
 - Débordement du TIMER1 (TMR1E).
 - Débordement du TIMER2 (TMR2E).
 - Collision de BUS (BCLIE).

VI-3- Organisation de la mémoire :[9]

Le plan mémoire est linéaire, les adresses vont de 0000h à 1FFFh (8k mots de 14 bits), par page de 2K mots. On peut remarquer que le vecteur de reset est figé en 0000h.

Les PICs n'ont qu'un seul vecteur d'interruption en 0004h. Lors d'une interruption, le sous programme associé devra déterminer quel périphérique a demandé une interruption.

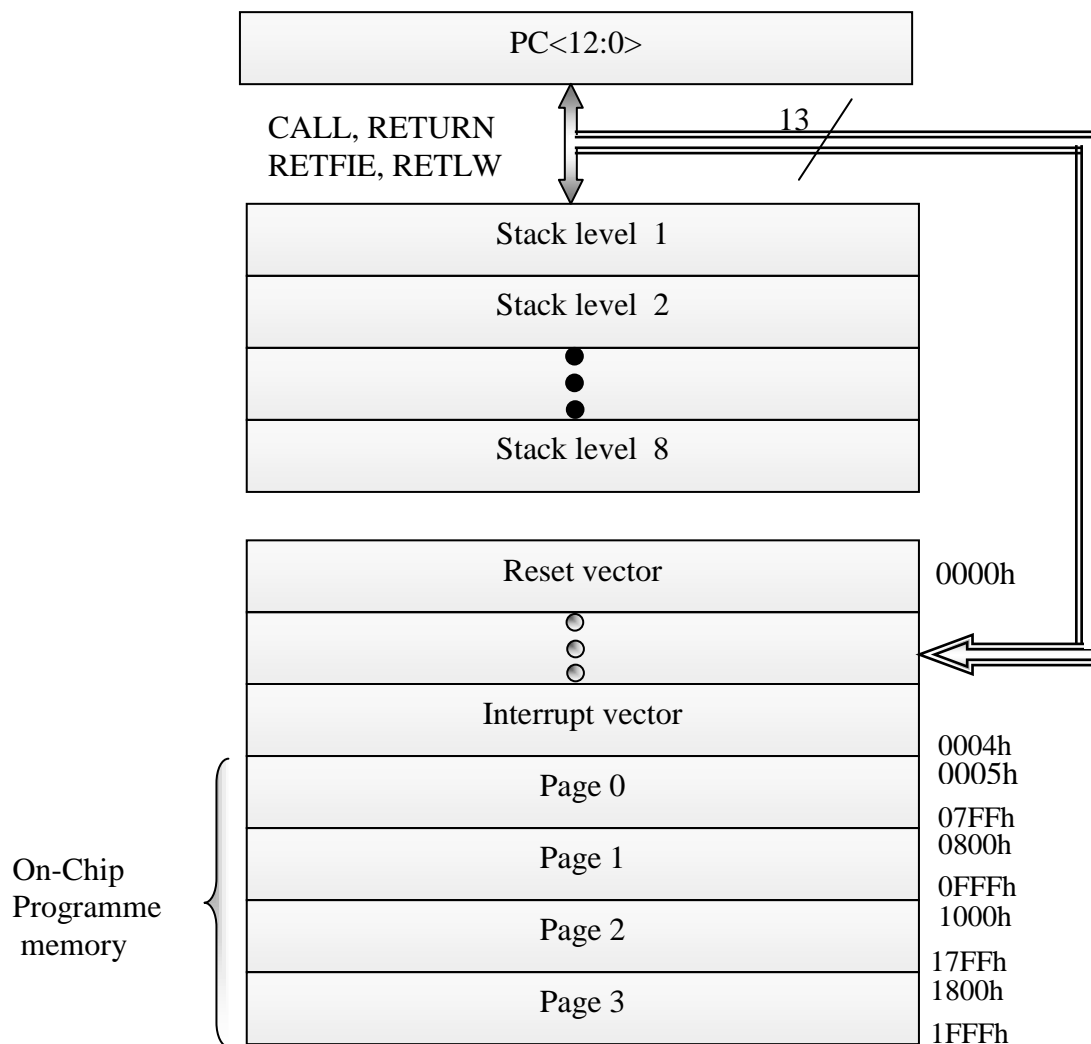


Figure Ch.2-2 Plan mémoire.

La pile utilisée par les sous programmes n'est pas implantée en mémoire de données comme avec les microcontrôleurs classiques, mais dans la mémoire programme. Elle est utilisée lors d'appels de sous programmes, on ne peut pas imbriquer plus de 8 sous programmes.

VI-4- Convertisseur Analogique-Numérique

Le convertisseur Analogique-Numérique est à approximations successives et il possède une résolution de 10 bits. Il est composé de :

- Multiplexeur analogique 5 voies (PIC16F876).
- Echantillonneur bloqueur.
- Convertisseur Analogique-Numérique de 10 bits.

VI-4- 1- Fonctionnement du convertisseur [8]

La conversion se passe en deux temps :

- 1^{er} temps : le signal à convertir est appliqué sur l'entrée à convertir, ce signal doit être présent au moins pendant le temps Tacq (temps d'acquisition environ 20µS pour 5V).
- 2^{ème} temps : la conversion à approximations successives.

Le temps de conversion minimum est de 12 TAD (TAD c'est le temps de conversion dépendant de l'horloge interne, typiquement 1.6 µs). La conversion est terminée, GO/DONE repasse à 0. Donc pour pouvoir lire le résultat dans les registres ADRESL et ADRESH il suffit d'attendre que le bit GO/DONE passe à 0.

La valeur résultante N de la conversion ADRESH: ADRESL est égale à :

$$N (\text{valeurs numérisées}) = ((V_{IN} - V_{REF-}) / (V_{REF+} - V_{REF-})) * 1023$$

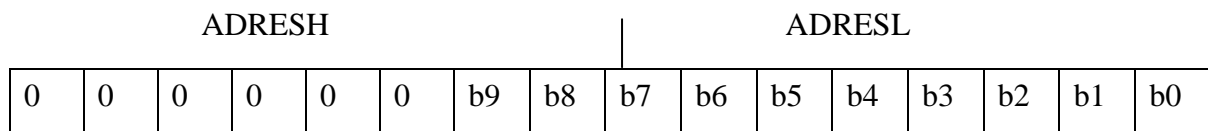
Si $V_{REF+} = V_{DD} = 5V$ et $V_{REF-} = V_{SS} = 0V$

Alors :

$$N (\text{valeurs numérisées}) = 1023 * (V_{IN} / 5)$$

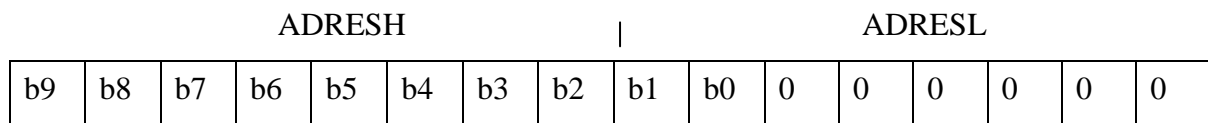
VI-4- 2- Justification à droite [8],[7]

Complète la partie gauche du résultat par des « 0 ». Le résultat sera donc de la forme :



VI-4- 3- justification à gauche [7]

Procède bien évidemment de la méthode inverse.



La justification à droite sera principalement utilisée lorsque nous avons besoin de l'intégralité des 10 bits de résultat.

VI-4- 4- Registre ADCON0 (1Fh) [7]

Ce registre est le dernier utilisé par le convertisseur Analogique-Numérique. Il contient les bits que nous allons manipuler lors de notre conversion. Sur les 8 bits de ce registre, 7 seront utilisés.

ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/DONE	—	ADON
-------	-------	------	------	------	---------	---	------

Bit 7-6 ADCS1, ADCS0 : (A/D Conversion Clock Select bits).

00 = $F_{OSC}/2$, pour fréquence maximale 1,250 MHz.

01 = $F_{OSC}/8$, pour fréquence maximale 5,000 MHz.

10 = $F_{OSC}/32$, pour fréquence maximale 20,000 MHz.

11 = Osc RC, Si > 1MHz.

Ces configurent le facteur du diviseur, afin de déterminer l'horloge du convertisseur en fonction de la fréquence du quartz utilisé (F_{osc}).

Bit 5-3 CHS2, CHS0: (Analog Channel Select bits).

000 = canal analogique 0, (RA0/AN0).

001 = canal analogique 1, (RA1/AN1).

010 = canal analogique 2, (RA2/AN2).

011 = canal analogique 3, (RA3/AN3).

100 = canal analogique 4, (RA5/AN4).

A partir de ces bits, on peut sélectionner les canaux utilisés comme des entrées analogiques.

Bit 2 $\overline{\text{GO/DONE}}$ (A/D Conversion Status bit, DONE: fait la conversion).

1 = Conversion A/N en marche (commence la conversion A/N).

0 = Conversion A/N non en marche (ce bit est automatiquement libéré par le matériel quand la conversion A/N est complète).

Bit 1 Inutilisé (=0).

Bit 0 ADON : A/D On bit.

1 = Le module convertisseur A/N est actif.

0 = Le module convertisseur A/N non actif.

VI-4- 5- Registre ADCON1 (9Fh) [7]

Il permet de déterminer le rôle de chacune des pins AN0 à AN4. Il permet donc de choisir si une broche sera utilisée comme entrée analogique, comme entrée/sortie standard (numérique), ou comme tension de référence. Il permet également de décider de la justification du résultat.

ADFM	—	—	—	PCFG3	PCFG2	PCFG1	PCFG0
------	---	---	---	-------	-------	-------	-------

Bit 7 ADFM (A/D Result **F**or**M**at select bit).

1 : Résultat de la conversion sera justifié à droite.

0 : Résultat de la conversion sera justifié à gauche.

Bit 4-6 Non utilisées (= 0).

Bit 0 à Bit 3 PCFG (**P**ort **C**on**F**i**G**uration control bit).

PCFG Bit3 à Bit0	AN4 RA5	AN3 RA3	AN2 RA2	AN1 RA1	AN0 RA0	V _{ref+}	V _{ref-}	A/D/R ^(*)
0000	A	A	A	A	A	V _{DD}	V _{SS}	5/0/0
0001	A	V _{ref+}	A	A	A	RA3	V _{SS}	4/0/1

0010	A	A	A	A	A	V _{DD}	V _{SS}	5/0/0
0011	A	V _{ref+}	A	A	A	RA3	V _{SS}	4/0/1
0100	D	A	D	A	A	V _{DD}	V _{SS}	3/2/0
0101	D	V _{ref+}	D	A	A	RA3	V _{SS}	2/2/1
0110	D	D	D	D	D	V _{DD}	V _{SS}	0/5/0
0111	D	D	D	D	D	V _{DD}	V _{SS}	0/5/0
1000	A	V _{ref+}	V _{ref-}	A	A	RA3	RA2	3/0/2
1001	A	A	A	A	A	V _{DD}	V _{SS}	5/0/0
1010	A	V _{ref+}	A	A	A	RA3	V _{SS}	4/0/1
1011	A	V _{ref+}	V _{ref-}	A	A	RA3	RA2	3/0/2
1100	A	V _{ref+}	V _{ref-}	A	A	RA3	RA2	3/0/2
1101	D	V _{ref+}	V _{ref-}	A	A	RA3	RA2	2/1/2
1110	D	D	D	D	A	V _{DD}	V _{SS}	1/4/0
1111	D	V _{ref+}	V _{ref-}	D	A	RA3	RA2	1/2/2

(*) : A : nombre des entrées analogiques, D: le nombre des entrées/sorties numériques,
R : nombre d'application des tensions de référence.

IV-4-6- Temps d'acquisition (Tacq) [8],[9]

C'est le temps qu'il faut pour que le condensateur interne atteigne une tension proche de la tension à convertir. Cette charge s'effectue à travers une résistance interne (R_{IC}) et la résistance de la source (R_s) connectée au pin. Ce temps est incrémenté du temps de réaction des circuits internes et d'un temps qui dépend de la température (coefficient de température). Il faut savoir en effet que les résistances augmentent avec la température, donc les temps de réaction des circuits également.

On note :

R_s: Résistance de source (<10kΩ).

C_{PIN}: Condensateur de pin d'entrée.

R_{IC}: Résistance interne.

SS: Interrupteur d'échantillonneur.

R_{SS}: Résistance de l'échantillonneur.

C_{hold}: Capacité de blocage.

Donc, si on pose :

Tacq : temps d'acquisition total.

Tamp : temps de réaction des circuits.

Tc : temps de charge du condensateur.

Tcoff : temps qui dépend du coefficient de température.

La formule est donc :

$$T_{acq} = T_{amp} + T_c + T_{coff}$$

Le temps de réaction Tamp est typiquement de 2µs, pas donc de problème à ce niveau :

Donc, ce temps Tcoff sera au minimum de 0 (à moins de 25°C) et au maximum de $[(50(t^{\circ}_{max}) - 25^{\circ}C) \cdot 0,05\mu s/^{\circ}C]$, soit 1,25µs. La température du PIC ne pouvant pas, en effet, excéder 50°C

$$0 < T_{coff_{max}} < 1,25\mu s$$

Sachant que le condensateur interne (C_{hold}) à une valeur de 120pF pour les versions actuelles du PIC16F876. La formule du temps de charge du condensateur est donnée par la relation suivante :

$$T_c = - C_{hold} \cdot (R_{interne} + R_{source}) \cdot \ln (1/2047)$$

Et sachant que : $R_{source} \approx 1\Omega$

$$R_{interne} = (R_{IC} + R_{SS}) = 1\text{ k}\Omega + 7\text{ k}\Omega = 8\text{ k}\Omega \text{ sous } (5V, 4MHz)$$

Donc :

$$T_c = 0,914895 \cdot 10^{-9} \cdot (1 + 8 \cdot 10^3).$$

$$T_c = 7,32\text{ }\mu s$$

A partir des ces valeurs, on peut calculer le temps d'acquisition (Tacq) :

$$T_{acq} = 2\mu s + 1,25\mu s + 7,32\mu s = 10,57\mu s.$$

Maintenant, pour nos utilisations classiques, sous 5V, nous aurons dans le pire des cas ($R_{source} = 10\text{ k}\Omega$) :

$$T_c (\text{typique}) = 16,46\text{ }\mu s.$$

Donc :

$$T_{acq} \approx 20\text{ }\mu s$$

Donc, nous prendrons un T_{acq} de $20\mu s$ pour un PIC alimenté sous 5V. Si on travaille sous une tension différente, il nous faudra adapter ces valeurs.

IV-4-7- Temps de conversion [8]

Concernant notre PIC, il faut savoir qu'il nécessite, pour la conversion d'un bit, un temps qu'on va nommer T_{AD} . Ce temps est dérivé par division de l'horloge principale. Le diviseur peut prendre une valeur de 2, 8 ou 32.

V- Conclusion

Ce chapitre a été totalement consacré à l'étude de la structure du microcontrôleur 16F876A, tout en illustrant ces différentes caractéristiques afin de mieux l'exploiter. On a montré durant ce chapitre que ce CI :

- Dispose d'une mémoire flash pour stocker le programme (donc il est possible de le reprogrammer en cas de bug ou d'évolution). La mémoire flash peut être programmée par le programme lui-même.
- Facile de fabriquer un programmeur et de nombreux outils sont disponibles.
- La mémoire RAM est de taille suffisante.
- Dispose d'un port série, d'un convertisseur analogique numérique et de timers.
- Ce microcontrôleur se trouve partout et est peu coûteux.