Chapitre II

Le PIC 16F877A

Introduction

Dans une chaîne d'automatisation, trois parties sont envisagées : partie contrôle, partie commande et partie opérative. La partie de commande est constitué par le biais d'un microcontrôleur. Ce dernier est un circuit intégré électronique dédié à faire des calculs plus au moins complexe avec une grande précision. Pour assurer cette opération le microcontrôleur contient le plus souvent une unité centrale de traitement de données, des mémoires et des interfaces de communication (entrées/sorties, portes séries) et de multiples ressources internes. Le nombre et l'architecture de ces ressources varient d'une famille à une autre. Comme tous utilisateurs, de ce type de circuits intégrés, le connaître, un microcontrôleur est circuit intégré programmable. Il regroupe, au sein d'une même puce, les différents éléments que l'on trouve habituellement dans l'unité centrale d'un ordinateur personnel. En particulier, il contient une unité de calcul, différentes mémoires volatiles et non volatiles, ainsi qu'un ensemble « d'interfaces internes » facilitant la communication avec le monde extérieur. Il est conçu pour être programmé par un ordinateur, puis placé dans un circuit électronique dans lequel il effectua un travail plus ou moins complexe. Après programmation, il peut fonctionner indépendamment de tout ordinateur.

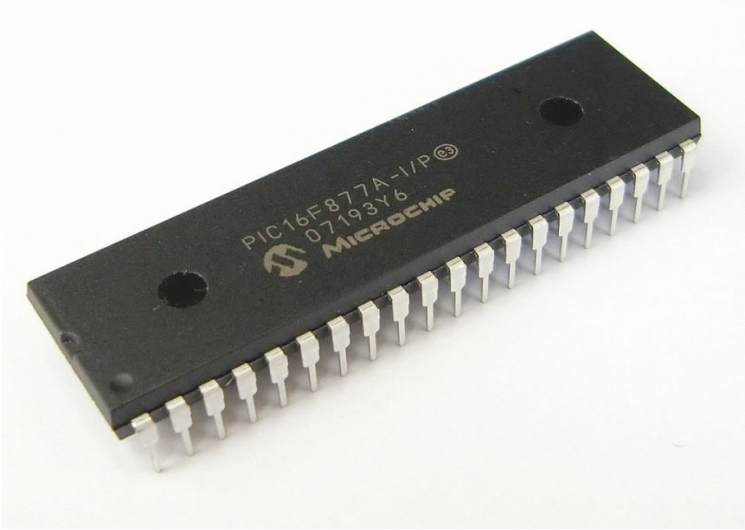

Figure II.1 : *PIC16F877A*

Leur principaux domaines d'application sont : l'industrie de fabrication de machines outils, la domotique (machines à café, appareils électroménagers, etc.), et l'électronique grand public. Malgré les architectures dédiées sont cependant parfois nécessaires pour des raisons de performances. Ce circuit intégré peut, dans beaucoup de cas de les remplacer. Différentes familles de microcontrôleurs

sont disponibles sur le marché. Parmi les plus connus, on trouve le 6800 et ses dérivés (comme le 68HC11) produits à l'origine par la société Motorola puis par Freescale Semi-conducteur ; les microcontrôleurs Philips (en particulier les séries de P87 et P89) ; Les microcontrôleur de la famille Atmel AVR (utilisée par des cartes Arduino), ainsi que les PIC de la société Microchip.[6]

Le PIC 16F877A de la société américaine Micro-chip est l'un de ces types. C'est un microcontrôleur en technologie CMOS. Ce composant encore très utilisé à l'heure actuelle, est un compromis entre simplicité d'emploi, rapidité et prix de revient. Dans ce qui suit, nous allons exposer, brièvement, l'architecture interne de ce microcontrôleur tout en discutant leurs caractéristiques.

II.1 Comment effectuer le choix entre les différentes familles de microcontrôleur ?[10]

Le choix d'un microcontrôleur est basé sur le type d'applications qu'on désire réaliser tout en assurant un meilleur compromis prix/rendement. Ce choix est le plus souvent s'appui sur l'architecture du microcontrôleur eu même et le nombre de ressource qu'il les contient (voir figure ci-dessous).

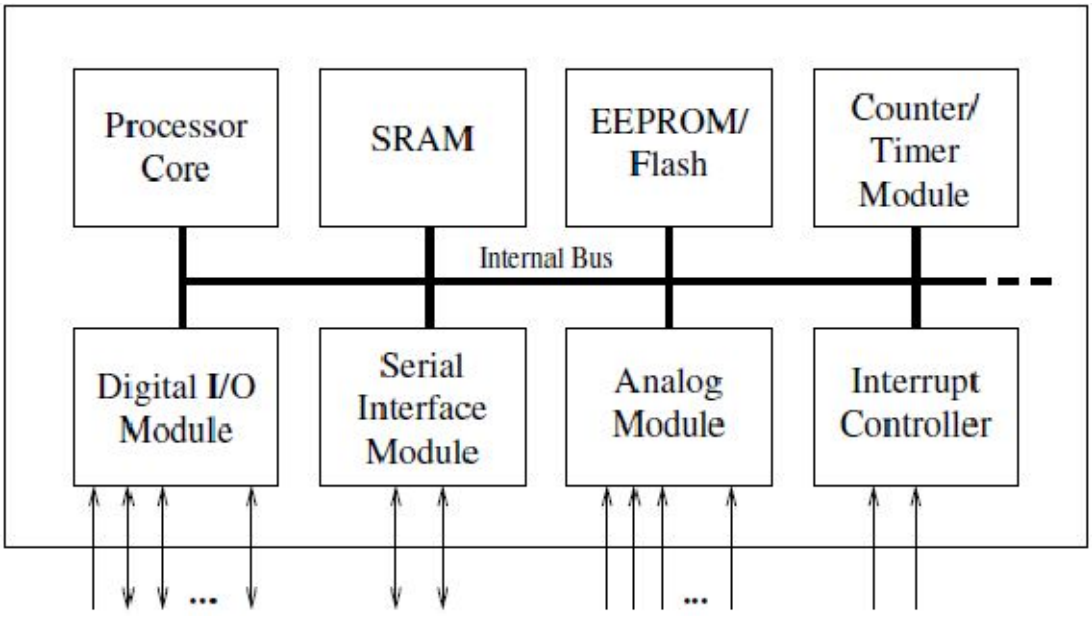

Figure II.2 : *schéma d'un microcontrôleur*

II.2 les PICs

Les PICs sont subdivisés en trois grandes familles :

- **Base line :** c'est la famille qui utilise un jeu d'instructions codé sur 12bits, exemple : l2CXXX.
- **Mid-Range :** qui utilise un jeu d'instructions codé sur 14 bits, exemple : 16F877A.
- **High Performance :** qui utilise un jeu d'instructions codé sur 16 bits, exemple : I8FXXX, 18CXXX

Un PIC est généralement identifié par une référence de la forme NN LLL XXX où :

- **NN :** désigne la famille à laquelle appartient le circuit.
- **LLL:** est un ensemble d'une, deux ou trois lettres qui indique le type de la mémoire de programme contenue dans le circuit. Aussi, il indique si la plage de tension d'alimentation est normale ou étendue.
- XXX : est un ensemble de deux ou trois chiffres constituant la référence du circuit.

Exemple : Le PIC 16F877A-20

16 : indique la famille Mid-Range.

F : indique le type de mémoire programme utilisée : F pour Flash.

877 : identité et le 20 : indique la fréquence d'horloge.

II.3 Pour quoi le PIC16F877A ?

Le choix du PIC16F877A est fait dans ce projet, en raison de sa grande partie de ses performances, sa taille, sa facilité d'utilisation et le prix du montage. Le PIC16F877A possède :

- 05 ports d'entrées sorties (A,B,C,D & E).
- Une capacité mémoire programme flash de 8K qui est nécessaire pour mettre l'application.
- Une mémoire RAM répartie sur 368 octet, ce qui permet l'utilisation de plus de variables dans un programme.
- Un convertisseur analogique-numérique 10 bits.
- Une horloge jusqu'à 20MHZ.
- Une tension de fonctionnement allant de 2V jusqu'à 5.1V.
- Un courant d'entrée-sortie des pins allant jusqu'à 25mA.
- Un module de communication USART (Universel Synchronous Asynchronous Receiver Transmitter) qui permet d'envoyer et de recevoir des données en mode série, soit de façon synchrone, soit asynchrone, ….etc.

De plus, ce PIC possède des instructions très puissantes (programme réduit) surtout lorsqu'on utilise le logiciel de programmation « mikroC PRO». Il possède un nombre important de procédures et fonctions prédéfinies dédié au PIC 16F877A. Il offre aussi la possibilité de contrôler la cohérence du code au moment de la complication, c'est-à-dire qu'il offre le moyen de vérification d'erreurs pour les corriger au moment de la programmation.

En fait ; la cause principale du choix de microcontrôleur PIC est qu'il dispose de l'option du convertisseur A/D pour satisfaire le coté d'acquisition, aussi la possibilité de l'adaptation au protocole de la liaison RS232.

II.4 Architecture externe de PIC 16F877A (Physiquement) [6]

Comme le montre la figure ci-après, le pic 16F877A est présenté sous la forme d'un circuit intégré dans un boîtier appelé « DIL 4 ». Il présente 40 broches qui sont virtuellement numérotées de 1 à 40. La 1ère broche est placée dans le coin situé à gauche de l'encoche de repérage.

Il a :

- 04 pins pour l'alimentation (VDD, VSS).
- 02 pins pour l'oscillateur (OSC1, OSC2).
- 01 pin pour le RESET (MCLR)
- 33 pins d'entrées/sorties dont chacune de ses broches assure un fonction précise.

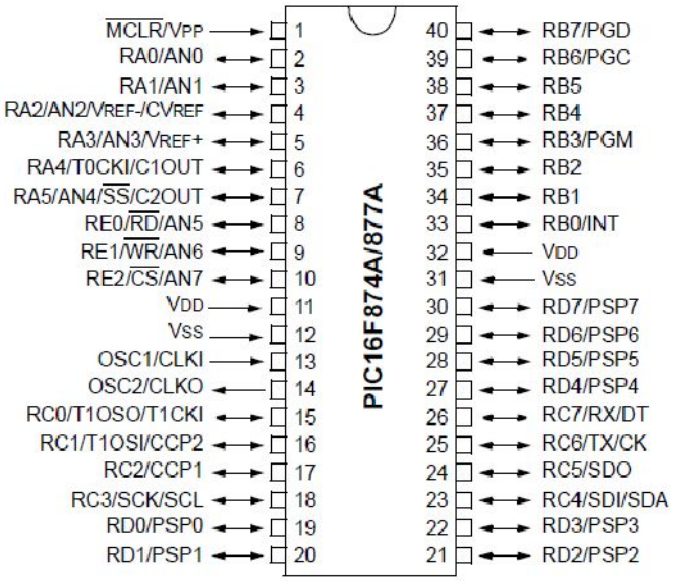

Figure II.3 : *Brochage du PIC 16F877A*

II.5 Architecture interne de PIC 16F877A [6]

L'architecture interne détaillée du PIC16F877A est exposée dans la figure ci-dessous. Par la suite nous décrivant les principaux modules à part.

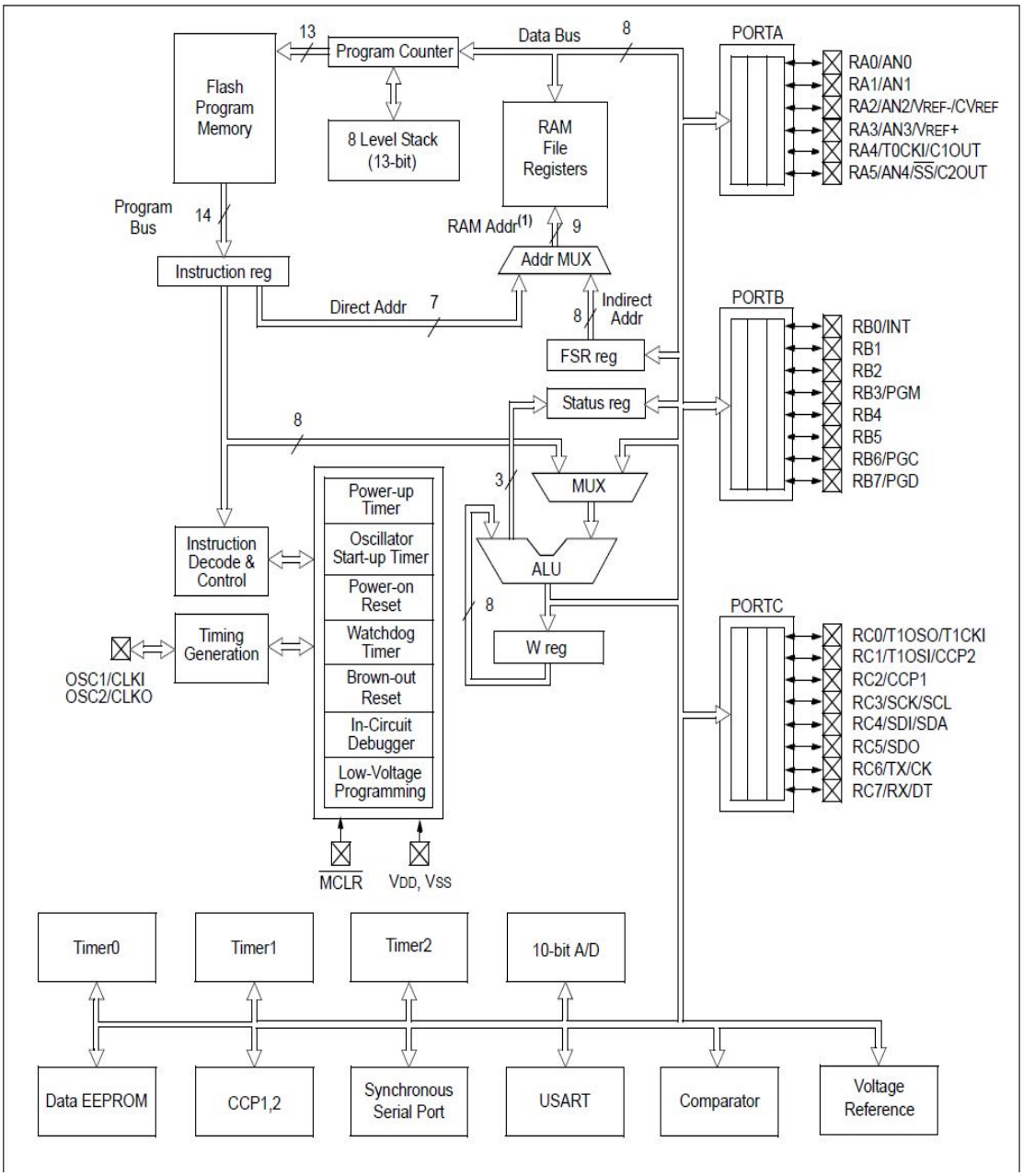

Figure II.4 : *Structure interne du PIC 16F877A*

II.5.1 Unité de traitement CPU (CENTRAL PROCESSING UNIT)[7,9]

L'architecture d'un tel microcontrôleur la plus simple, comme le cas de PIC 16F877, se compose d'un microprocesseur, d'une mémoire, et d'une entrée-sortie, le microprocesseur se compose d'une unité centrale de traitement (CPU) et d'une unité de commande (CU), l'unité centrale de traitement est le cerveau d'un microprocesseur et l'unité de commande sert à commander toutes ses fonctionnalités internes.

L'unité de traitement numérique exécute les instructions du programme (codées sur 14 bits) stockées en mémoire (mémoire FLASH).

Elle est essentiellement composée de :

- ALU (Unité Arithmétique et Logique) qui réalise toutes les opérations arithmétiques et logiques de base.
- L'Accumulateur «W» qu'est le registre de travail de 8 bits par lequel passent toutes les opérations.
- Registre STATUS,
-Registre FSR utilisé pour l'adressage indirect.
- Registre d'Instruction.
- Compteur programme «PROGRAM COUNTER» qui pointe les instructions à exécuter.
- Différents bus qui établissent la liaison entre tous ses éléments et assurent l'échange de l'information.

II.5.2 Mémoire FLASH

C'est la mémoire programme proprement dite. La mémoire FLASH est un type de mémoire stable, réinscriptible à volonté. Cette mémoire qui a fait le succès de microprocesseur PIC. Dans le cas du 16F877, elle fait 8K mots, sa vocation est de stocker le programme du PIC, mais pas les variables de votre programme. Le PIC exécute une à une les instructions logées dans la mémoire du programme.

II.5.3Mémoire RAM

C'est une mémoire de taille 368 Octets, d'accès rapide, mais labile, cette mémoire contient les registres de configuration ainsi que les différents registres de données permettant de contrôler le cœur du PIC. Elle comporte également des cases mémoires à usage générique dans lesquelles pourront être stockées les variables utilisées par le programme.

II.5.4Mémoire EEPROM

Elle est de taille de 256 Octets, électriquement effaçable, réinscriptible et stable. Ce type de mémoire est d'accès plus lent, elle est très utile pour sauver des paramètres semi-permanents, modifiés plusieurs million de fois par secondes.

II.6 Les ports d'entrées/sorties généraux

Pour communiquer avec l'extérieur le PIC dispose de 5 ports (PORT A, PORT B, PORT C, PORT D et PORT E). Les ports sont bidirectionnels, ce qui signifie qu'ils peuvent être configurés et utilisés comme des entrées ou des sorties.

Ainsi, le PORTA sert également au convertisseur Analogique/Numérique, il dispose de 5 canaux d'entrées analogiques RA0 à RA5. Nous pouvons donc échantillonner jusqu'à 5 signaux différents.

Le PORTB possède 8 pins, d'entrées/sorties classiques, numérotés de RB0 à RB7.

le PORTC est un port tout ce qu'il y a de plus classique, or il a deux pins qui servent à la communication série avec le PC à travers (TX et RX) (pin25 et pin26).

Le PORTD est très utilisé en mode parallèle esclave.

Le PORTE, contrairement aux autres ports, possède trois pins qui peuvent être utilisés comme entrées au convertisseur analogique numérique, aussi il peut contrôler le port parallèle esclave.

Il faudra faire des choix au moment de la conception du schéma électronique, on voit également que les ports B, C et D ont 8 lignes d'entrée/sortie, alors que le PORTA n'a que 6 lignes, et le PORT E que 3 lignes d'entrée/sortie. Ils sont tous connectés au bus de données (DATA BUS).

On pourra donc librement les adresser pour y lire ou écrire des données, ainsi allumer des LED, commander des moteurs pas à pas, des afficheurs LCD, lire les données envoyées par un clavier ou un bouton poussoir...etc.

On peut configurer les entrées/sorties de chaque port en entrées ou en sorties, grâce à un registre spécial dédié à chaque port, tous les ports sont pilotés par deux registres :

- Le registre de PORTX : si le PORTX où certaines lignes de PORTX sont configurées en sorties, ce registre détermine l'état logique des Sorties.
- Le registre TRISX : c'est le registre de direction. Il détermine si le PORTX ou certaines lignes de ce port sont en entrées ou en sorties. Le positionnement d'un bit à «1» place le pin en entrée, le positionnement de ce bit à « 0 » place le pin en sortie ce qui implique que chaque broche d'un port peut être configurée soit en entrée soit en sortie à l'aide des directives TRISA, TRISB, TRISC et TRISD et TRISE.

II.7 Le Port Parallèle Esclave [11]

Le Port Parallèle Esclave est un port 8 bits, permettant d'interfacer le PIC avec par exemple, un autre microprocesseur, Les données transitent via les lignes PSP0 à PSP7, qui physiquement utilisent les mêmes broches (pins) que le PORTD. Le flux de données est contrôlé par les lignes RD, WR et CS qui correspondent aux broches du PORTE. Son appellation est port «parallèle esclave», puisque c'est le microprocesseur externe qui donne les ordres, qui est le chef d'orchestre.Il Valide notre PIC par la ligne de Sélection CS (Chip Select), et indique au PIC s'il est en mode lecture ou écriture grâce aux lignes RD (READ) et WR (WRITE), notre PlC ne fait que l'exécuter.

II.8 La configuration principale de PIC [7]

II.8.1 Circuit de Reset

Ce circuit sert à remettre le PIC à ''0''. En générale, il est réalisé en connectant une résistance de 10kohm, une résistance de 470ohm parallèle avec diode de protection à partir de l'entrée MCLR à la tension d'alimentation avec un condensateur de couplage connecter a la terre (voir figure ci-dessous).

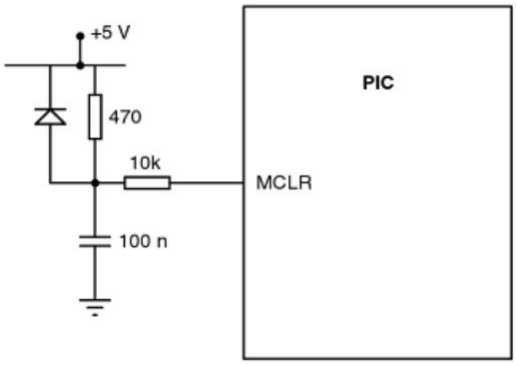

Figure II.5 : Circuit *de réinitialisation*

Dans le cas de notre application ce circuit est réalisé selon le schéma de la figure ci-après.

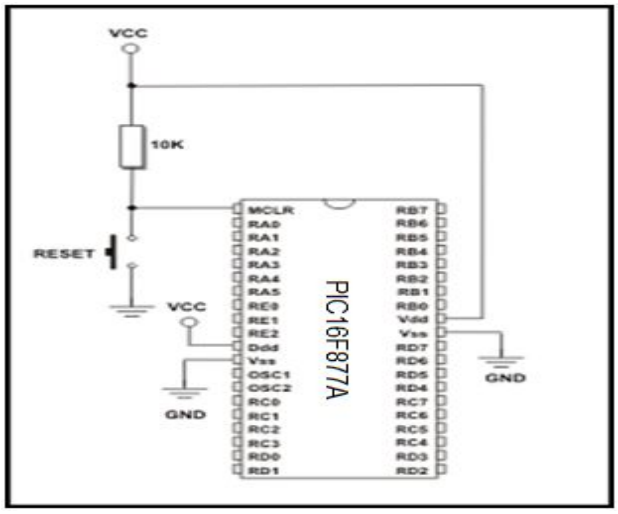

Figure II.6 : *Circuit RESET dans notre application*

II.8.2 L'oscillateur ou L'horloge

L'horloge est l'élément de base de n'importe quel PIC. Elle sert à cadencer l'exécution des instructions. Elle peut être réalisée soit avec une horloge extérieure, soit avec un circuit RC, soit avec un Quartz comme dans la figure suivant.

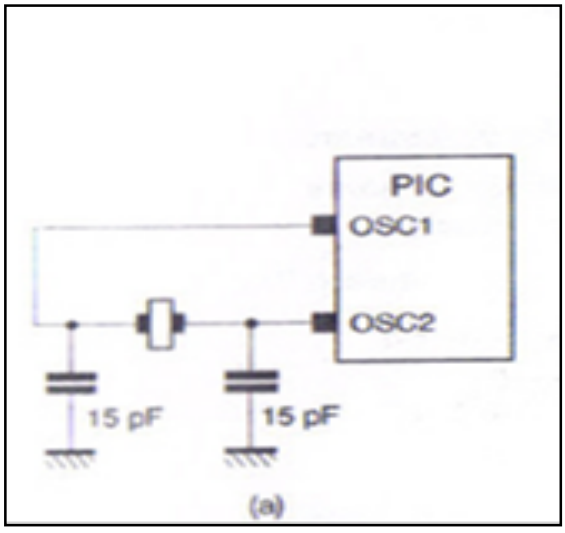

Figure II.7 : *Le circuit oscillateur.*

Le PIC peut fonctionner en 4 modes d'oscillateur :

- LP : Cristal puissance.
- XT : Cristal f résonateur.
- HS : Cristal haute vitesse f résonateur.
- RC : circuit RC résistance-condensateur.

Le mode HS est recommandé pour un fonctionnement jusqu'à 20MHz. Dans le mode de fonctionnement avec un cristal externe deux condensateurs qui sont connectés aux entrées du microcontrôleur OSC1 et OSC2. Les condensateurs doivent être choisis selon les valeurs regroupés dans le tableau ci-dessous. Par exemple, avec une fréquence de cristal de 4 MHZ, deux condensateurs 22 pF peuvent être utilisés. Les circuits résonateurs sont disponibles à partir de 4 MHZ à 8 MHZ. Ils ne sont pas aussi précis que les oscillateurs à base de cristal.

Tab. II.1 : sélection de condensateur pour le fonctionnement du cristal

Mode	Fréquence	C1, C2
LP	32 KHz	68-100 pF
LP	200 KHz	15-33 pF
XT	100 KHz	100-150 pF
XT	2 MHz	15-33 pF
XT	4 MHz	15-33 pF
HS	4 MHz	15-33 pF
HS	10 MHz	15-33 pF

Pour les applications où la précision de la synchronisation n'est pas importante, nous pouvons connecter une résistance externe et un condensateur à l'entrée OSC1 du PIC, la fréquence de l'oscillateur dépend des valeurs de la résistance et du condensateur (Voir tableau II.2), la tension d'alimentation, et de la température.

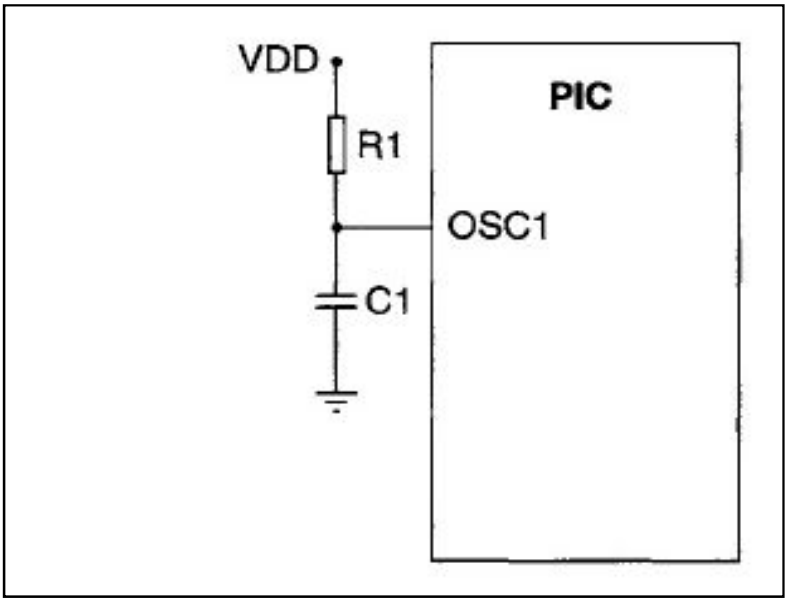

Figure II.8: *Oscillateur RC*

Tab. II.2 : Sélection des composants de l'oscillateur RC

C1	R1	Fréquence
20 pF	5K	4.61 MHz
	10K	2.66 MHz
	100K	311 kHz
100 pF	5K	1.34 MHz
	10K	756 kHz
	100K	82.8 kHz
300 pF	5K	428 kHz
	10K	243 kHz
	100K	26.3 kHz

II.9 Les TIMERS [6]

II.9.1 Timer 0

C'est le plus ancien des timers implantés dans les PICs, son ancienne appellation était RTCC, pom Real Time Clock. C'est un temporisateur /compteur 8 bits (0 à 255) simple, Il a pour rôle de gérer des événements périodiques, comme l'incrémentation des variables. Celui-ci est incrémenté soit par l'horloge interne (Fosc/4) ou par des impulsions extérieures générées par une horloge appliquée sur la broche T0CKI/ RA4.

Le mode temporisateur est sélectionné si le bit T0CS du registre(OPTION_REG) est met à 0, ce mode de fonctionnement est assuré par l'horloge interne, le timer0 est incrémenté à cycle d'instruction (sans le pré-diviseur).

Le mode compteur est sélectionné en mettant à l le bit T0CS du registre (OPTION_REG). L'horloge dans ce cas est externe, si le bit T0SE est met à 1, le Timer0 est incrémenté à chaque front montant de la patte RA4/T0CKI, et à chaque front descendant si TOSE est mis à 0.

II.9.2 Timer1

Il fonctionne sur le même principe que le timer mais il est plus moderne dans sa conception. C'est un temporisateur/compteur 16 bits. Il peut être incrémenté via le bit TMR1CS du registre (T1CON) soit par l'horloge interne, externe par des impulsions sur la broche T1CKI/RCO ou par un oscillateur (RC ou quartz) connecté sur les broches T1OSO/RC0 et T1OSI/RC1.

II.9.3 Le Timer2

Le Timer2 a un fonctionnement différent des Timer0 et Timer1. Le Timer2 est un compteur 8 bits, il sert à générer des signaux carrés, ou, en association avec le module CCP, des signaux PWM «Pulse Widlh Modulation» «Modulation de Largeur d'Impulsion (MLI) ». Il est incrémenté par l'horloge interne (Focs/4). Celle-ci peut être pré divisée. Les bits T2CKPS1 et T2KPS0 du registre (T2CON) permettent de choisir la valeur de la pré-division (1,4 ou 16).

II.10 Les Modules CCPl et CCP2 [9]

Dans quelques microcontrôleurs, la valeur d'un registre de temporisateur peut être capturée dynamiquement et être alors comparée contre une de préréglage. Quand la valeur de temporisateur est équivalente à la valeur comparée, un résultat est obligatoire.

Sur le PIC, Les deux modules CCP1 et CCP2 <<Capture, Compare, PWM >> ont une action identique, dans une opération exceptionnelle qu'est l'opération du déclenchement d'un événement spécial.

Ces derniers, et en association avec les deux timers (TIMERl et TIMER2), ont pour but de comparer l'occurrence d'un signal en entrée avec la valeur du compteur Timer1, réalisant ainsi un chronométrage de l'événement. Par exemple, si l'entrée du module CCP est reliée à un capteur qui délivre une impulsion à chaque tour de l'arbre d`un moteur, la valeur contenue dans le registre du Timer1 au moment de l'impulsion est le reflet de la vitesse de rotation, un autre exemple, si le Timer1 compte en continu, et une interruption est générée chaque fois que la valeur du compteur est égale à celle qu'on aura pré-chargée dans le registre de comparaison, on peut s'en servir pour générer un signal carré.

En interactions avec le TIMER2, ils vont nous permettre aussi de générer des signaux PWM (modulation de largeur d'impulsion).

II.11 Mécanisme d'interruptions [7,8,9]

Qu'est-ce qu'une interruption… ?

Imaginez une conversation normale, chaque interlocuteur prend la parole quand vient son tour de parler, survient alors un événement extérieur dont le traitement est urgent. Par exemple, un piano tombe du 3eme étage de l'immeuble, au pied duquel vous discutez. Vous imaginez bien que votre interlocuteur ne va pas attendre la fin de votre phrase pour vous signaler le danger. Il va donc vous interrompre durant le cours normal de votre conversation afin de pouvoir traiter immédiatement l'événement extérieur. Les interlocuteurs reprendront leur conversation où elle en était arrivée, sitôt le danger écarté.

Ainsi, pour les programmes, c`est exactement le même principe. Votre programme se déroule normalement. Survient un événement spécifique, le programme principal est interrompu (donc, subit

une INTERRUPTION), et va traiter l'événement, avant de reprendre le programme principal là où i1 avait été interrompu.

L'interruption est donc un sous-programme particulier déclenché par l'apparition d`un événement spécifique qui provoque la rupture d'une séquence de programme principal.

Les interruptions sont un concept très important dans les microcontrôleurs, une interruption fait répondre un microcontrôleur aux événements externes et internes. Quand une interruption se produit le microcontrôleur part de son écoulement normal d'exécution et saute directement à la routine de service d'interruption (ISR).

La source d'interruption peut être interne ou externe, des interruptions internes sont habituellement produites par les circuits intégrés de temporisateur quand le compte de temporisateur atteint une certaine valeur et des interruptions externes sont produites par des dispositifs de microcontrôleur, ces interruptions sont asynchrones, on ne le connaît pas quand une interruption externe sera produite. Un exemple est le convertisseur analogique-numérique (A/D), l'interruption est produite quand une conversion est accomplie. D'autres exemples d'interruption comme l'écriture en mémoire EEPRGM terminée ou le changement d'état d'une entrée de port.

Les interruptions peuvent en général être nichées tels qu'une nouvelle interruption peut suspendre l'exécution d'une autre interruption, la plupart des microcontrôleurs ont au moins un certaines Sources d'interruption. Dans des quelques microcontrôleurs, les sources d'interruption peuvent être données la priorité de sorte qu'une interruption plus élevée puisse suspendre l'exécution d'une routine de service inférieure d'une autre interruption.

II.12 Le Module USART [6]

L'USART ou (Universal Synchronous Asynchronous Receiver Transmitter) est l'un des deux modules de communication série du PIC ou SCL en anglais (Serial Communication Interface). Comme son nome l'indique, elle peut être configurée comme un système synchrone (Half Duplex) ou un Système asynchrone (Full Duplex), établir une liaison, communiquer, recevoir et transmettre des données, avec tout un autre matériel équipé d'une interface série RS232 (communication se fait sur les deux broches de PIC RC6/TX et RC7/RX doivent être configures toutes les deux), des circuits intégrés convertisseurs Numérique/Analogique ou Analogique/Numérique et des EEPROMS série…etc.

II.13 Module de conversion Analogique /Numérique (A/D) [8]

Principe de conversion A/N

La fonction conversion analogique-numérique consiste à transformer une grandeur électrique en une grandeur numérique exprimée sur N bits. Le CAN est un périphérique intégré destiné à mesurer un signal analogique (une tension électrique) et le convertir en nombre binaire équivalent qui pourra, être utilisé par un programme.

Ce convertisseur est composé de :

- Un multiplexeur analogique 8 entrées maximum permet de sélectionner l'entrée analogique à convertir, ces dernières doivent être configurées en entrée à l'aide des registres TRISA et TRISE.
- Un échantillonneur bloqueur qui est constitué d'un interrupteur d'échantillonnage et d'une capacité de blocage de 120 pF permet de mémoriser la tension analogique à convertir pendant la conversion.

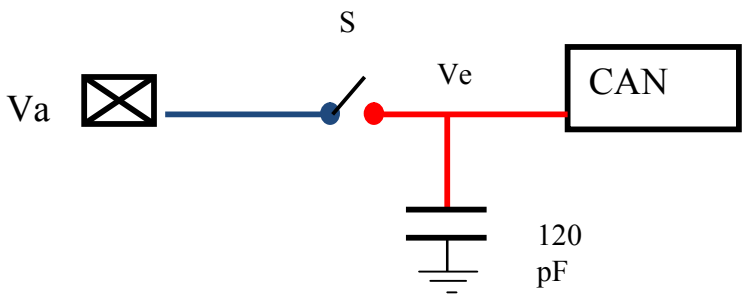

Figure II.7 : *L'échantillonneur bloqueur.*

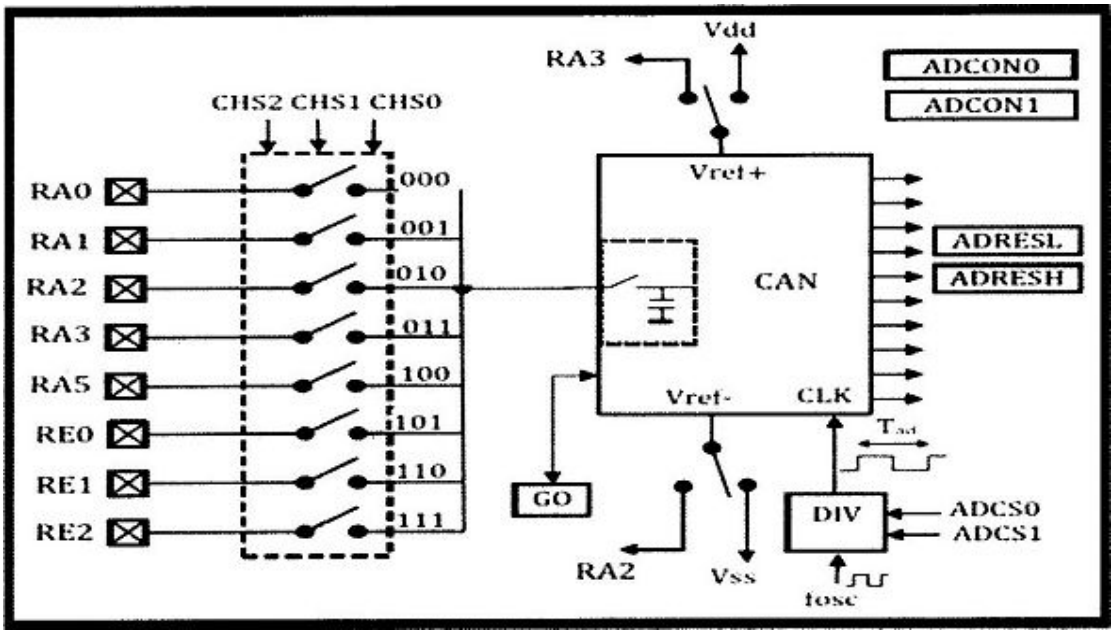

Figure II.8 : *Le module de conversion A/N*

Le control du module se fait par les deux registres ADCON0 et ADCON1

ADCON0	ADCS0	ADCS0	CHS2	CHS1	CHS0	GO/DONE	-----	ADON
ADCON1	ADFM	-----	-----	-----	PCFG3	PCFG2	PCFG1	PCFG0

ADCS1 :ADCS0 : pour le choix de l'horloge de conversion.

CHS2 :CHS0 : pour le choix de l'entrée analogique**.**

ADRESH, ADRESL : pour le changement de résultat.

II.14 Les registre ADRESL et ADRESH

Le fait que le convertisseur donne un résultat sur 10 bits, et donc celui-ci devrait obligatoirement être sauvegardé dans 2 registres. Ces derniers sont tout simplement les registres ADRESL et ADRESH.

Comme les 2 registres contiennent 16 bits, et que nous n'en utilisons que 10, Microchip nous a laissé le choix sur la façon dont est sauvegardé le résultat, nous pouvons soit justifier le résultat à gauche, soit à droite.

La justification à droite complète la partie gauche du résultat par des « 0 ». Le résultat sera donc de la forme :

ADRESH								ADRESL							
						9	8	7	6	5	4	3	2	1	0

La justification à gauche procède évidemment de la méthode inverse :

ADRESH								ADRSEL							
9	8	7	6	5	4	3	2	1	0						

La justification à droite sera principalement utilisée lorsque nous avons besoin de l'intégralité des 10 bits de résultat, tandis que la justification à gauche est très pratique lorsque 8 bits suffisent. Dans ce cas, les 2 bits de poids faibles se trouvent isolés dans ADRESL, il suffit donc de ne pas en tenir compte. Cette approche est destinée à nous épargner des décalages de résultats.

Le CAN convertit le signal analogique présent sur une de ses 8 entrées en son équivalent numérique, codé sur 10 bits. Le Signal numérique peut donc prendre l024 valeurs possibles, les pattes AN2 et AN3 peuvent être utilisées comme références de tension ou comme entrées analogiques standard. Les tensions de références étant prises sur les tensions d'alimentations du PIC (VDD et VSS) (VDD pour Vref+ et VSS pour Vref-), ils ont pour rôle de la dynamique du convertisseur.

II.15 Déroulement d'une Conversion A /N

La conversion se passe en 2 temps

- 1er temps, le signal à convertir est appliqué sur l'entrée à convertir, que le pic connecte le pin sur lequel se trouve la tension numérisée à un condensateur interne, qui va se charger via une résistance interne jusqu'a la tension appliquée. C'est le temps d'acquisition qui est défini par le temps qu'i1 faut pour que 1e condensateur interne atteigne une tension proche de la tension à convertir.

  Le signal à convertir doit être présent au moins pendant le temps Tacq temps d'acquisition est d'environ de 20µS pour 5V).
- 2éme temps, la conversion, le pin sera déconnecté du condensateur et ce dernier est connecté sur le convertisseur analogique/numérique interne pour que le pic procède à la conversion. Cette connexion prend de l'ordre de 100ns.

Le principe de conversion est basé sur la méthode d'approximation successive. Il s'agit tout simplement de couper l'intervalle de la grandeur analogique en 2 parties égales, et de déterminer dans laquelle de ces 2 parties se situe la valeur à numériser. Une fois cet intervalle déterminé, on le coupe de nouveau en 2, et ainsi de suite jusqu'à obtenir la précision demandée.

Le temps de conversion minimum est de 12 TAD (TAD est le temps de conversion d'un bit dépendant de l`horloge interne, typiquement 1.6µS).

- ✓ **TAD :** avant le début de conversion (le temps de connexion du condensateur est inclus).
- ✓ **10 * TAD** : pour la conversion des 10 bits du résultat.
- ✓ **TAD :** supplémentaire pour la fin de la conversion.

Donc, en fonction des fréquences utilisées pour le quartz du PlC, il faudra choisir le diviseur le plus approprié. Le tableau suivant reprend les valeurs de diviseur à utiliser pour quelques fréquences courantes du quartz.

Tableau II.3 : les valeurs de diviseur pour quelques fréquences courantes de quartz

diviseur	20MHz	5MHz	4 MHz	2 MHz	1.25 MHz	333.3 MHz
2	100ns	400ns	500ns	1 µs	1.6µs	6 µs
8	400ns	1.6µs	2µs	4 µs	6.4µs	24 µs
32	1.6µs	6.4µs	8µs	16 µs	25.6 µs	96 µs
OSC RC	2-6µs	2-6µs	2-6µs	2-6µs	2-6µs	2-6µs

Un temps de 12 TAD, soit dans le meilleur des cas, un temps de 12 * 1,6µs=19,2 µs.
En effet, le temps nécessaire pour effectuer l'ensemble des opérations est :

$T = T_{ACQ} + 12\ T_{AD} + 2\ T_{AD} = T_{ACQ} + 14\ T_{AD}$
T_{ACQ} : est le temps d'acquisition pour la charge du condensateur interne.
12 * T_{AD} : est le temps nécessaire pour la conversion.
2 T_{AD}: est le temps attendu avant de pouvoir recommencer une autre conversion

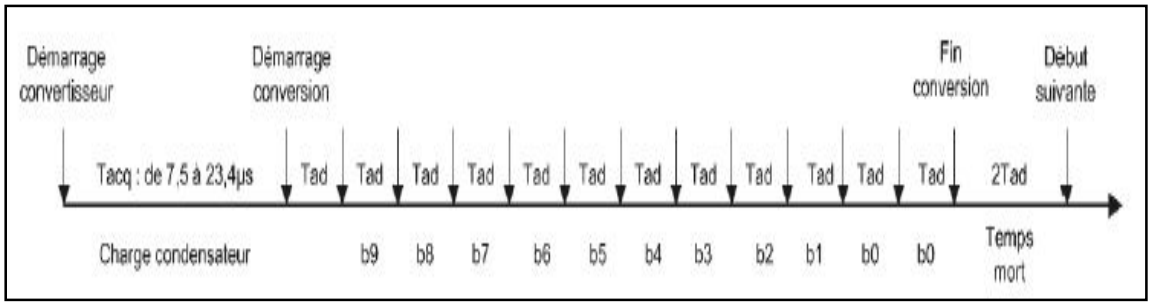

Figure II.9 : *cycle de conversion.*

II.16 Programmation du PIC

La programmation d'un microcontrôleur PIC nécessite les outils et les composants suivants:

- Un assembleur ou compilateur de langage de haut niveau. Le logiciel comprend généralement un débuguer, un simulateur, et d'autres programmes de soutien.
- Un ordinateur (généralement un PC) dans lequel est exécuté le logiciel de développement.
- Un dispositif matériel appelé un programmeur qui se connecte à l'ordinateur par l'intermédiaire du port série, port parallèle ou par USB ligne. Le commandant de bord est inséré dans le programmateur et "soufflé" en téléchargeant le code exécutable généré par le système de développement. Le programmateur de matériel comprend en général le logiciel de support.
- Un câble ou un connecteur pour connecter le programmateur à l'ordinateur.
- Un microcontrôleur PIC.

Conclusion

Dans ce chapitre on a fait une présentation générale des microcontrôleurs et en particulier le PIC16F877A vu que les performances qu'il présente sont essentielles dans cette chaine de mesure. En conclusion, nous pouvons dire que ce fameux composant, et les possibilités qu'il offre, peuvent être réunis pour faire fonctionner notre prochain système d'acquisition de température.