

2023

**Architecture des microcontrôleurs**

MICROCONTROLEUE | 3B10

Projet : Système

de Contrôle automobile

Mohamed iheb tbessi

Soltan Mohamed

Anis bouchagraoui

Mohamed sayari

Khalil ahmed

Malek Salmen

**TABLE DES MATIÈRES**

[I. **Introduction :** 4](#_Toc151286029)

[II. **Généralités sur les PICs:** 4](#_Toc151286030)

[1. Définition : 4](#_Toc151286031)

[2. Les avantage du microcontrôleur : 4](#_Toc151286032)

[3. Les différentes familles des PIC: 5](#_Toc151286033)

[4. Identification du PIC (6) : 5](#_Toc151286034)

[B. Le microcontrôleur PIC 16F877 : 5](#_Toc151286035)

[C. Le microcontrôleur PIC 16F84 : 10](#_Toc151286036)

[III. **Le registre.** 13](#_Toc151286037)

[1. Le registre « Option » : 13](#_Toc151286038)

[2. Le registre « INTCON » : 14](#_Toc151286039)

[3. Les registres « PORTx et TRISx » : 15](#_Toc151286040)

[IV. **Des différences entre les microcontrôleurs PIC 16F84 et PIC 16F877.** 15](#_Toc151286041)

[1. Nombre de Broches d'Entrée/Sortie : 15](#_Toc151286042)

[2. Diversité des Sources d'Interruptions : 15](#_Toc151286043)

[3. Exposition des Différents Cahiers des Charges : 15](#_Toc151286044)

[4. Attribution des Projets aux Groupes : 16](#_Toc151286045)

[5. Objectif Spécifique : 16](#_Toc151286046)

[V. **Conclusion :** 16](#_Toc151286047)

INTRODUCTION GENERALE

**Le but de ce projet est la mise au point d’un fréquencemètre numérique à base de PIC16F877/ PIC16F84A possédant deux fonctions de mesure, et qui permettra de mesurer la fréquence d’un signal périodique en hertz ainsi que sa valeur moyenne en volts.**

**La mesure de la fréquence possède des applicatio/ PIC16F84A ns vastes en électronique analogique et numérique ainsi que dans le domaine des télécommunications. Aujourd'hui beaucoup de méthodes de mesure permettent cela en faisant appel aux différentes méthodes de mesures électrique et électronique en utilisant par exemple différents types de circuits intégrés spécialisés.**

**Dans ce chapitre de décrire le microcontrôleur PIC16F877 / PIC16F84A utilisé pour notre appareil de mesure, ainsi que son mode de fonctionnement et de différence.**

CHAPITRE : DESCRIPTION GENERALE DU FONCTIONNEMENT DES PIC :

PIC16F877/ PIC16F84A

# **Introduction :**

Un objet technique, intégrant de l’électronique, fait souvent apparaître des fonctions ayant pour rôle le traitement d’information : opérations arithmétiques (Addition, multiplication…) ou

Logiques (ET, OU…) entre plusieurs signaux d’entrée permettant de générer des signaux de sortie. Ces fonctions peuvent être réalisées par des circuits analogiques ou logiques.

Mais, lorsque l’objet technique devient complexe, et qu’il est alors nécessaire de réaliser un ensemble important de traitements d’informations, il devient plus simple de faire appel à une structure à base de microcontrôleur PIC.

Au niveau de ce chapitre on va essayer de mieux connaître le PIC16F877 (PIC choisie pour ce projet), de savoir manipuler ces instructions internes.

# **Généralités sur les PICs:**

Dans les années 90 une société Américaine nommée Microchip Technologie a fait ses premiers microcontrôleurs CMOS, qu'ils ont appelé PIC (Peripheral Interface Controller), sont disponibles à partir de micro puce, chaque série de PIC possède son propre langage assembleur.

Les programmes écrits pour une série ne sont en général pas compatibles pour les autres, cependant il est souvent facile de porter de manière systématique les programmes d’une série de numéro faible vers une série de numéro plus grand.



### Définition :

Un PIC n’est rien d’autre qu’un microcontrôleur, c’est à dire une unité de traitement de l’information de type microprocesseur à laquelle on a ajouté des périphériques internes permettant de réaliser des montages sans nécessiter l’ajout de composants externes.

La dénomination PIC est sous copyright de Microship, donc les autres fabricants ont été dans

l’impossibilité d’utiliser ce terme pour leurs propres microcontrôleurs.

Les PICs sont des composants dits RISC (Réduc Instructions Computing Set), ou encore

composant à jeu d’instructions réduit.

### Les avantage du microcontrôleur :

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.

ous 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

-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

-Moins cher que les autres composants qu’il remplace.

-Diminuer les coûts de main d’œuvre.

### Les différentes familles des PIC:

Il y en a trois grandes familles de **PICs** :

-La famille **Base Line**, qui utilise des mots d’instructions de 12 bits.

-La famille **Mid-Range**, qui utilise des mots de 14 bits.

-La famille **High-End**, qui utilise des mots de 16 bits.

Toutes les PICsMid-Range ont un jeu de 35 instructions, stockent chaque instruction dans un seul mot de programme, et exécutent chaque instruction (sauf les sauts) en un cycle. On atteint donc des très grandes vitesses, et les instructions sont de plus très rapidement assimilées.

### Identification du PIC (6) :

Pour identifier une PIC, on utilise simplement son numéro. Les 2 premiers chiffres indiquent la catégorie de la PIC, **16 indique une PIC Mid-Range.**

Vient ensuite parfois une lettre L : Celle-ci indique que la **PIC peut** fonctionner avec une plage de tension beaucoup plus tolérante. Ensuite, on trouve :

-C indique que la mémoire programme est une **EPROM** ou plus rarement une **EEPROM CR**

Pour indiquer une mémoire de **type ROM.**

- F pour indiquer une mémoire de **type F**LASH.

## Le microcontrôleur PIC 16F877 :

Le pic 16F877 est un circuit intègre contenu dans un boîtier nommer «DIL 40», il présente 40 broches, 20 de chaque côté. Les broches sont virtuellement numérotées de 1 à 40. La 1ere broche est placé dans le coin situé à gauche de l’encoche de repérage (Figure.1).



**Figure. 1 PIC 16F877A**

#### Caractéristiques générales :

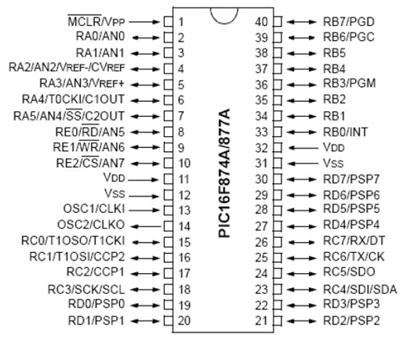
Voici un tableau récapitulatif des principales caractéristiques du PIC16F877.

|  |  |
| --- | --- |
| Caractéristiques | 16F877 |
| Broches | 40 |
| E/S max | 33 |
| µy flash | 8 KO |
| µy E²PROM | 256 O |
| CAN | 7 |
| PWM | 2 DE 10 BITS |
| TIMER r | 3 |
| Comparateurs | 2 |
| Interruption | 13 |
| oscillateur | 20 MHz MAX |
| Port série | USART/SSP |

**Tableau. Caractéristiques du PIC 16F877**

#### Architecture externe :

La figure ci-dessous montre l'architecture externe d'un PIC 16F877.



**Figure. Brochage du PIC 16F877.**

-Le boitier du PIC 16F877 décrit par la figure.2 comprend 40 pins : 33 pins D’entrées/sorties, 4 pins pour l'alimentation, 2 pins pour l'oscillateur et une pin pour le **Reset (MCLR).**

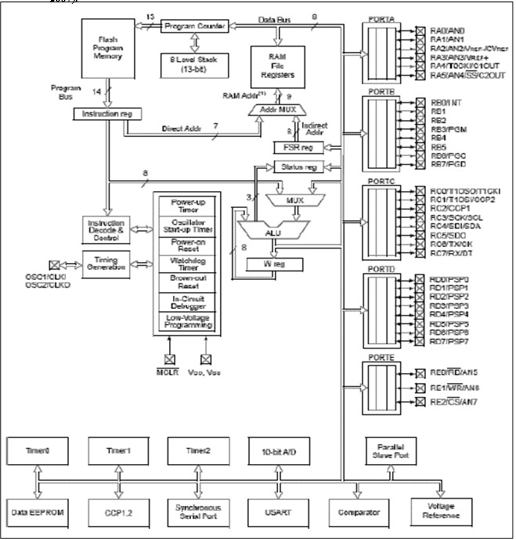
-La broche MCLR sert à initialiser le μC en cas de la mise sous tension, de remise à zéro Externe, de chien de garde et en cas de la baisse de tension d'alimentation.

-Les broches VDD et VSS servent à alimenter le PIC.

On remarque qu'on a 2 connections "**VDD**" et 2 connections "**VSS**". La présence de ces 2 pins s'explique pour une raison de dissipation thermique. Les courants véhicules dans le pic sont loin d'être négligeables du fait des nombreuses entrées/sorties disponibles.

#### Architecture interne):

La figure présente l'architecture interne de PIC16F877 :



**Figure. Architecture Interne du PIC16F877**

#### Les ports entrée/sortie :

Le PIC 16F877 comporte cinq ports entrées / sorties :

##### Port A :

c’est un port d’entrée sortie, il contient 6pin d’entrées /sorties de RA0 à RA5 repartie sur deux registres : le registre port A et le registre tris A. Le bit 6 et7 ne sont pas implémente, ils seront lus comme 0.

Au moment de reset on doit forcer une valeur dans le registre ADCON1, pour pouvoir utiliser ce port en entrée/sortie de type générale.

##### Port B

le pin **RB0** qui, en configuration d’entrée, est de type «Trigger de Schmitt» quand elle est utilisée en mode interruption «**INT**».La lecture simple de RB0 se fait, elle, de façon tout à fait classique, en entrée de type **TTL**.

##### Port C

c’est un port qui n’existait pas sur le 16F84. Voyons donc, toujours au niveau de son utilisation classique, quel que soit les caractéristiques.

La mise sous tension du PIC, ou reset, force tous les bits utiles de TRIS x à 1, ce qui place toutes les pins en entrée.

##### Port D

ce port n’est présent que sur le 16F877.il fonctionne d’une façon identique aux

autres, dans son mode de fonctionnement général.

Le registre TRISD comporte donc les 8 bits de direction.

Les 8 pins I/O, en mode entrée, sont de type «Trigger de Schmitt».Le fonctionnement de ce port dépend de la valeur placée dans TRISE, qui concerne, a première vue, le port E. Mais au moment de mise sous tension, la valeur placée dans TRISE configure le PORTD en port I/O de type général.

##### Port E

Il ne comporte que 3 pins, RE0, RE1 et RE2 contrairement aux autres ports, les bits non concernés de

TRISE sont implémentées pour d’autres fonctions.

On remarque que les pins Rex peuvent être utilisés comme pins d’entrées analogiques. C’est le registre ADCON1 qui détermine si ce port sera utilisé comme port I/O ou comme port analogique.

#### Les mémoires du PIC 16F877 :

Les mémoires sont de trois types différents :

##### La mémoire FLASH :

C’est une mémoire programme de taille 8K octets. Chaque case mémoire unitaire est de taille 13 bits. Cette mémoire est de type mémoires stable, c'est-à-dire qu’on peut réécrire dessus à volonté, car le 16F877 est caractérisé par la possibilité d’écrire des données.

La zone mémoire est caractérisée par une adresse de 13 bits, alors ceci nous impose donc pour l’adressage les registres EEAR et EEADRH. De même, nous aurons pour les données, les registres EEDATA et EEDATH.

##### La mémoire RAM :

Cette mémoire de taille 368 octets est une mémoire d’accès rapide et elle est volatile (les données seront perdus lorsque elle n’est plus sous tentions). Elle contient tous les registres de configuration du PIC ainsi que les différents registres de données. Elle contient également les variables utilisées par le programme. La RAM est la mémoire la plus utilisée. Toutes les données qui y sont stockées sont perdues lors d’une coupure de courant.

##### TIMERS (5,6,9) :

* TIMER 0 :

Le Timer1 fonctionne sur le même principe que le Timer0, mais avec un registre de comptage plus gros : 16 bits au lieu de 8, ce qui étend notablement ces capacités de comptage. De plus, il possède un mode de fonctionnement particulier : on peut l’utiliser en association avec un des modules **CCP** (modules de capture et de comparaison, voir plus loin). Voyons rapidement le mode « capture » : lorsqu’un événement survient sur l’entrée du module CCP, la valeur du Timer1 est lue.

* TIMER 1 :

Le Timer1 fonctionne sur le même principe que le Timer0, mais avec un registre de comptage plus gros : 16 bits au lieu de 8, ce qui étend notablement ces capacités de comptage. De plus, il possède un mode de fonctionnement particulier : on peut l’utiliser en association avec un des modules **CCP** (modules de capture et de comparaison, voir plus loin). Voyons rapidement le mode « capture » : lorsqu’un événement survient sur l’entrée du module CCP, la valeur du Timer1 est lue.

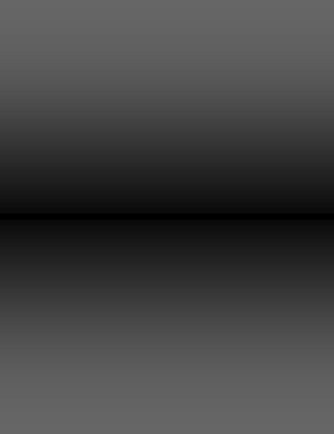
* TIMER 2 :

Le **Timer2** a un fonctionnement différent des **Timer0** et **Timer1**. C’est un compteur 8 bits avec pré-diviseur et post-diviseur. On s’en sert pour générer des signaux carrés, ou, en association avec le module **CCP**, des signaux **PWM**.

PWM étant l’acronyme de «Pulse Width Modulation» ou, en français, Modulation de Largeur d’Impulsion (**MLI**)

## Le microcontrôleur PIC 16F84 :

Le PIC 16F84A, dont le brochage est indiqué figure ci-dessous, est présenté en boîtier 18 pattes, dont nous allons examiner tout à tour le rôle afin de comprendre son schéma de mise en œuvre. (Figure.2).



**RA2**

**RA3 RA4/T0CKI**

**MCLR**

**RA1**

**RA0 OSC1/CLKIN**

**OSC2/CLKOUT**

**VSS**

**RB0/INT RB1 RB2**

**RB3**

**PIC 16F84A -04**

**VDD**

**RB7 RB6 RB5**

**RB4**

**Figure. 2 PIC 16F84A**

#### Caractéristiques générales

Le tableau ci-dessous présente les facteurs qui doivent être pris en compte lors de la migration de la PIC16F84 au PIC16F84A.

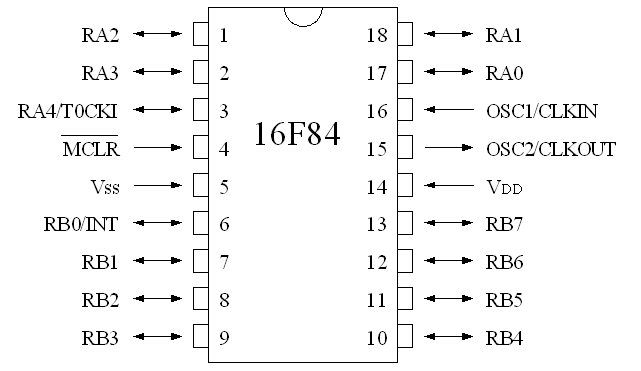
|  |  |  |
| --- | --- | --- |
| **N °** | **Module** | **Les différences de PIC16F84** |
| **1** | Oscillateur | L'oscillateur PIC16F84 peut fonctionner jusqu'à 10 MHz.  L'oscillateur PIC16F84A peut fonctionner jusqu'à 20 MHz. |

**Tableau .Les différences fonctionnelles PIC16F84 avec PIC16F84A**

Le PIC16F84A peut utiliser des cristaux jusqu'à 20 MHz, ce qui double la vitesse d'exécution.

#### Architecture externe :

La figure ci-dessous montre l'architecture externe d'un PIC 16F84A.



**Figure. Brochage du PIC 16F84A.**

#### Architecture interne :

La figure présente l'architecture interne de PIC16F84A :

**Figure. Architecture Interne du PIC16F84A**



13

BUS DE DONNÉES

MÉMOIRE DE DONNÉES

ADRESSE

MÉMOIRE DE DONNÉES

8

14

BUS PROGRAMME

REGISTRE

D’INSTRUCTIONS

ADRESSAGE

DIRECT

MPX

5

7 INDIRECT ADRISSAGE

RA4/T0CK1

8

STATUS

DECODEUR

D’INSTRUCTIONS

ET UNITÉ DE COMMANDE

PORTS E/S

MPX

8

PORTA

UAL

PORTB

W

RA0

RA1 RA2 RA3 RB0

RB1 RB2 RB3 RB4 RB5 RB6

RB7

OSC1/CLKIN OSC2/CLIKOUT MCLR VDD

VSS

HORLOGE SYSTÉME

CHIEN

DE GARDE

RESET CONNEXION ALIMENTATION

TEMPORISATEUR DÉMARRAGE OSCILLATEUR

TEMPORISATEUR MISE EN ROUTE

FSR

TMR0

POINTEUR DE PILE (8 NIVEAUX) ,13BITS

RAM

REGISTRES 68\*8

EEADR

EEPROM 64\*8

EEDATA

COMPTEUR DE PROGRAMME

FLASH

MÉMOIRE DE PROGRAMME 1K\*14

#### Les ports entrée/sortie (6) :

Le PIC16F84A est équipé de 13 lignes d’entrées/sorties réparties en deux ports :

##### Port A :

Particularité du **portA** : les bits b7 à b5 des registres **PORTA** ne correspondent à rien car il n‟y a que 5 lignes (b0 à b4), RA4 est une ligne à collecteur ouvert, cela veut dire que configurée en sortie cette broche assure 0Volt à l’état bas, mais qu’à l’état haut, il est nécessaire de fixer la valeur de la tension grâce à une résistance de tirage (pull up)

* **le port A : RA0 à RA4**

##### Port B

Particularité du **portB** : il est possible de connecter de façon interne sur chaque ligne une résistance de tirage (pull up) dont le rôle consiste à fixer la tension de la patte (configuré en entrée) à un niveau haut lorsque qu’aucun signal n’est appliqué sur la patte, pour connecter ces résistances, il suffit de placer le bit RBPU/ du registre OPTION à 0.

* **le port B : RB0 à RB7**

#### Les mémoires du PIC 16F884 (:

Cette mémoire contient toutes les instructions que doit effectuer le μC, pour le PIC 16F84A chaque instruction est codée sur 14 bits et la taille mémoire est de 1 Kox 14 bits (soit 1024 instructions), cette taille reste suffisante pour les petites applications qui nous intéressent.

##### La mémoire FLASH :

La mémoire programme peut être de différente nature selon le microcontrôleur utilisé, dans notre cas la mémoire est de type **FLASH/ROM**, c’est-à-dire réinscriptible à volonté, mais il existe aussi des mémoires EPROM qui ne peuvent être programmées qu’une seule fois.

##### La mémoire RAM :

Pour fonctionner, un programme doit généralement pouvoir stocker temporairement des données, une zone est spécialement prévue à cet effet : c‟est la RAM (Ramdom Acces Memory), contrairement à la MÉMOIRE DE PROGRAMME, cette dernière s‟efface lorsque l‟on coupe l‟alimentation, on s‟aperçoit que la taille de cette mémoire est de 68 x 8 bits, comme chaque donnée est codée sur 8 bits il est donc possible de mémoriser 68 données à la file, de plus la mémoire RAM contient deux autres zones (appelées Bank 0 et Bank 1) réservées aux registres de configuration du système.

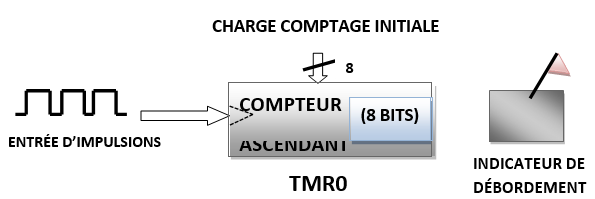
##### TIMERS (5,6,9) :

Il sert au fonctionnement du TIMER0 :

* TIMER 0 :

Le PIC 16F84A dispose de deux timers, un à usage général (le TMR0) et un autre utilisé pour le chien de garde WDG (Watch dog).

Le TMR0 est un compteur ascendant (qui compte) de 8 bits qui peut être chargé avec une valeur initiale quelconque, il est ensuite incrémenté à chaque coup d'horloge jusqu'à ce que le débordement ait lieu (passage de FF à 00).



**Figure. Le principe de fonctionnement TIME**

# **Le registre.**

### Le registre « Option » :

Ce registre en lecture écriture permet de configurer les prédiviseurs du **Timer** et du

**Watchdog**, la source du Timer, le front des interruptions et le choix du **Pull up** sur le **Port B**.



**Tableau. Les différents bits de registre Option**

**Au Reset : OPTION = 11111111**

**Bit 7 : RBPU = Pull up Enable bit on Port B.** 1 =Pull up désactivé sur le **Port B. 0 = Pull up** **activé.**

**Bit 6 : INTEDG** = **InterruptEdge select bit. 1** = Interruption si front montant sur la broche

**PB0/IRQ (pin 6). 0** = **Interruption si front descendant sur PB0/IRQ.**

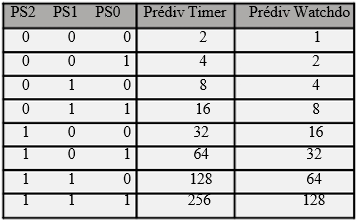
**Bit 5 : TOCS** = **Timer TMR0 Clock Source select bit. 1** = L'horloge du Timer est l'entrée

**PA4/Clk(pin 3). 0** = Le Timer utilise l'horloge interne du **PIC.**

**Bit 4 : TOSE** = **Timer TMR0 Source Edge select bit. 1** = Le Timer s'incrémente à chaque front montant de la broche **PA4/Clk.** 0 = Le Timer s'incrémente à chaque front descendant de la broche **PA4/Clk.**

**Bit 3 : PSA** = **Prescaler Assignement bit. 1** = Le prédiviseur est affecté au Watchdog.. 0 = Le prédiviseur est affecté au **TimerTMR0.**

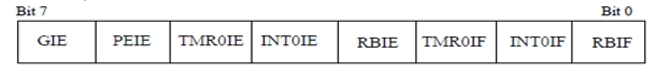
**Bits 2 à 0 : PS2 PS1 PS0** = Prescaler Rate Select bits.



**Tableau. Configuration des bits (PS0 :PS2) de registre option pour effectuer Les prés diviseurs**

### Le registre « INTCON » :

Ce registre en lecture écriture permet de configurer les différentes sources d'interruption.



**Tableau. Les différents bits de registre INTCON Bit**

**AuReset : INTCON = 0000000X**

**Bit 7** : **GIE = Global InterrupEnable bit** 1 = Autorise toutes les interruptions non masquées. 0 = Désactive toutes les interruptions.

**Bit 6** : **PEIE = PeripheralInterruptEnable bit**. 1 = Autorise les interruptions causées par les périphériques. 0 = Désactive les interruptions causées par le périphérique.

**Bit 5** : **TMR0IE = Timer TMR0 OverflowInterrupEnable bit**. 1 = Autorise les interruptions du Timer**TMR0**. 0 = Désactive les interruptions du Timer TMR0.

**Bit 4** : **INT0IE = RB0/Int InterrupEnable bit**. 1 = Autorise les interruptions sur la broche : PB0/IRQ (pin6). 0 = Désactive les interruptions sur la broche : **PB0/IRQ** (**pin6**).

**Bit 3** : **RBIE = RB Port Change InterrupEnable bit**. 1 = Autorise les interruptions par changement d'état du Port B (PB4 à PB7). 0 = Désactive les interruptions par changement d'état du Port B (**PB4** à **PB7**).

**Bit 2** : **TMR0IF = Timer TMR0 OverflowInterrup Flag bit**. 1 = Le Timer à débordé. Ce flag doit être remis à zéro par programme. 0 = Le Timer n'a pas débordé.

**Bit 1** : **INT0IF = RB0/Int Interrup Flag bit**. 1 = Une interruption sur la broche **PB0/IRQ** (**pin 6**) est survenue. 0 = Pas d’interruption sur la broche PB0/IRQ (pin 6).

**Bit 0** : **RBIF = RB Port Change Interrup Flag bit**. 1 = Quand au moins une entrée du port B (de PB4 à PB7) a changé d'état. 0 = Aucune entrée de **PB4** à **PB7** n'a changé d'état.

**NB** : Ce flag doit être remis à zéro par programme. Ceci n'est possible qu'après une lecture du Port B.

### Les registres « PORTx et TRISx » :

Tous les ports sont pilotés par deux registres :

-Le registre de **PORTx**, si le PORTx ou certaines lignes de PORTx sont configurées en sortie, 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 port sont en entrée ou en sortie. L'écriture d'une 1 logique correspond à une entrée (1 comme Input) et l'écriture d'une 0 logique correspond à une sortie (0 comme Output). Les registres TRISx appartiennent à la banque 1 des **SFR**.

# **Des différences entre les microcontrôleurs PIC 16F84 et PIC 16F877.**

Voici les points spécifiques qui seront examinés :

### Nombre de Broches d'Entrée/Sortie :

Le PIC 16F877 a un nombre de broches d'entrée/sortie plus élevé par rapport au PIC 16F84. Cela offre une plus grande flexibilité dans la connectivité des composants.

### Diversité des Sources d'Interruptions :

Le PIC 16F877 offre une diversité plus grande en termes de sources d'interruptions par rapport au PIC 16F84. Cela permet une gestion plus sophistiquée des événements extérieurs.

### Exposition des Différents Cahiers des Charges :

Il sera probablement discuté comment les caractéristiques spécifiques de chaque microcontrôleur influent sur la conception du système, en mettant en avant les avantages et les limitations de chaque modèle.

### Attribution des Projets aux Groupes :

La séance implique également l'attribution de projets aux groupes. Chaque groupe pourrait se voir confier des tâches spécifiques en fonction de la compréhension des différences entre les deux microcontrôleurs.

### Objectif Spécifique :

Identifier les entrées/sorties du système en tenant compte des spécificités du PIC 16F877.

Explorer des solutions lorsque le nombre de broches disponibles n'est pas suffisant, en se basant sur les capacités du PIC 16F877.

Réaliser un schéma conceptuel du système en prenant en considération les caractéristiques du microcontrôleur.

Réaliser du code en langage C qui assure l'affichage du niveau de carburant, de l'état des portes et de l'état du frein à main, en exploitant les fonctionnalités du PIC 16F877.

# **Conclusion :**

Dans ce second chapitre, nous avons présenté le mode de fonctionnement du Microcontrôleur PIC16F788 avec ses différents circuits internes qu’il contient (CAN, Compteur, Ports, Registres, Watchlog, Oscillateur…), et avec ses différents modes d’Interruptionet cela dans le but de coordonner et de contrôlerles différentes étapes de la mesure de la fréquenceet la valeur moyenne du signal avec une grande précision.

En conclusion, nous pouvons dire que le microcontrôleur 16F84A peut bien jouer le rôle d'une unité de commande pour notre système, il contient tous les modules nécessaires pour, réalise notre carte.