



**MICROCONTROLEURS**

**PIC 16F876 et 16F877**

*Documentation interne ELE*  
**D. MENESPLIER 2001**

## DESCRIPTION :

- Consommation : moins de 2mA sous 5V à 4 MHz.
- Architecture RISC : 35 instructions de durée 1 ou 2 cycles.
- Durée du cycle : Période de l'oscillateur quartz divisée par 4 soit 200 ns pour un quartz de 20 MHz.
- Deux bus distincts pour le code programme et les data.
- Code instruction : mot de 14 bits et compteur programme (PC) sur 13 bits, ce qui permet d'adresser 8 K mots ( de h'0000' à h'1FFF')
- Bus DATA sur 8 bits.
- 33 Ports Entrée-Sortie bidirectionnels pouvant produire 25 mA par sortie. PORTA = 6 bits et PORTB PORTC et PORTD = 8bits PORTE = 3 bits pour le 16F877 et 22 I/O seulement pour le 16F876.
- 4 sources d'interruption :
  - Externe par la broche partagée avec le Port B : PB<sub>0</sub>
  - Par changement d'état des bits du Port B: PB<sub>4</sub> PB<sub>5</sub> PB<sub>6</sub> ou PB<sub>7</sub>
  - Par un périphérique intégré dans le chip: écriture de Data en EEPROM terminée, conversion analogique terminée, réception USART ou I<sup>2</sup>C.
  - Par débordement du Timer.
- 2 Compteurs 8 bits et 1 compteur 16 bits avec pré diviseur programmable.
- Convertisseur analogique 10 bits à 8 entrées pour le 16F877 et 4 entrées pour le 16F876.
- UART pour transmission série synchrone ou asynchrone.
- Interface I<sup>2</sup>C.
- 2 modules pour PWM avec une résolution de 10 bits.
- Interface avec un autre micro: 8 bits + 3 bits de contrôle pour R/W et CS.
- 368 Octets de RAM
- 256 Octets d'EEPROM Data.
- 8K mots de 14 bits en EEPROM Flash pour le programme (h'000' à h'1FFF').
- 1 registre de travail : W et un registre fichier : F permettant d'accéder à la RAM ou aux registres internes du PIC. Tous les deux sont des registres 8 bits.

**PORTA** : 6 entrées -sorties . 5 entrées du CAN. Entrée CLK du Timer 0.

**PORTB** : 8 entrées-sorties. 1 entrée interruption ext. Clk et Data pour prog.

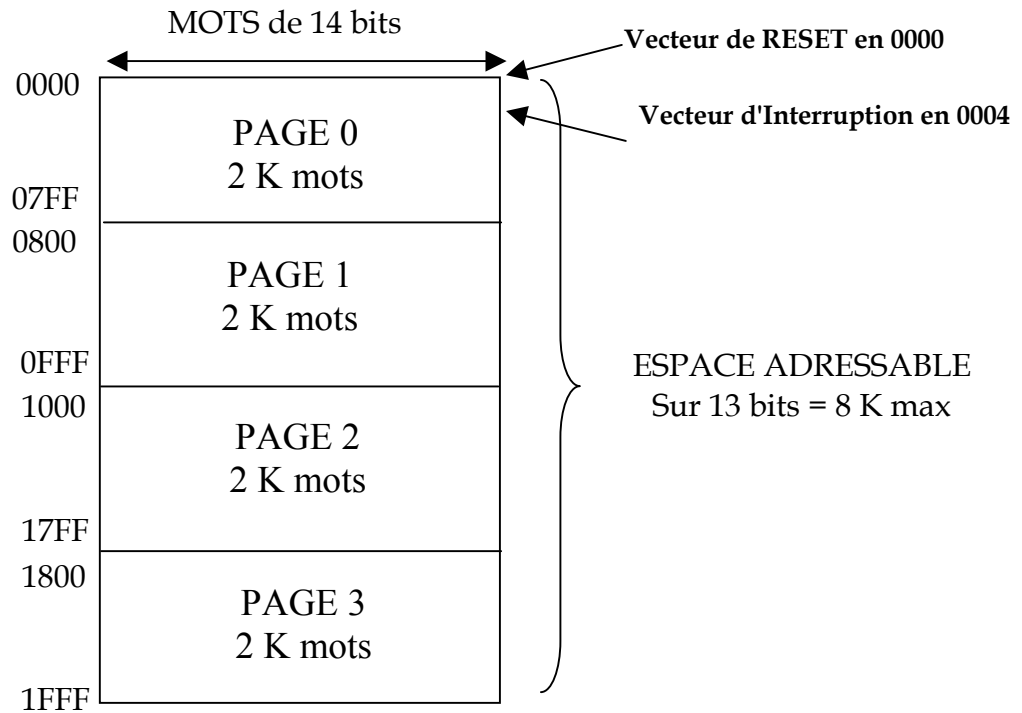
**PORTC** : 8 entrées-sorties. Clk Timer1 et PWM1. USART. I<sup>2</sup>C.

**PORTD** : 8 entrées-sorties. Port interface micro processeur (8 bits data).

**PORTE** : 3 entrées-sorties. 3 bits de contrôle interf micro. 3 entrées du CAN.

} N'existe pas sur le  
16F876 (28 broches)

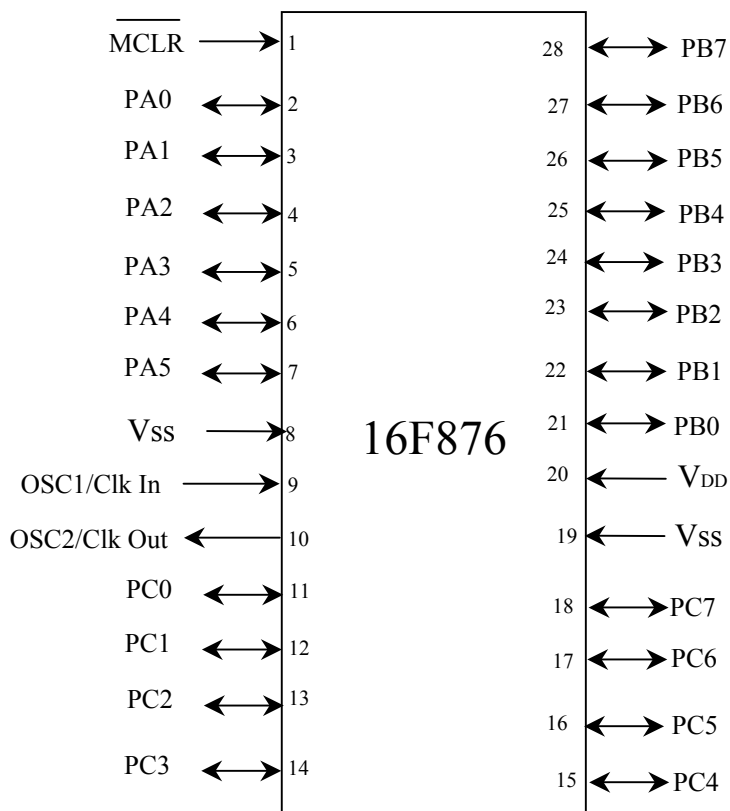
## ORGANISATION de l'ESPACE PROGRAMME :



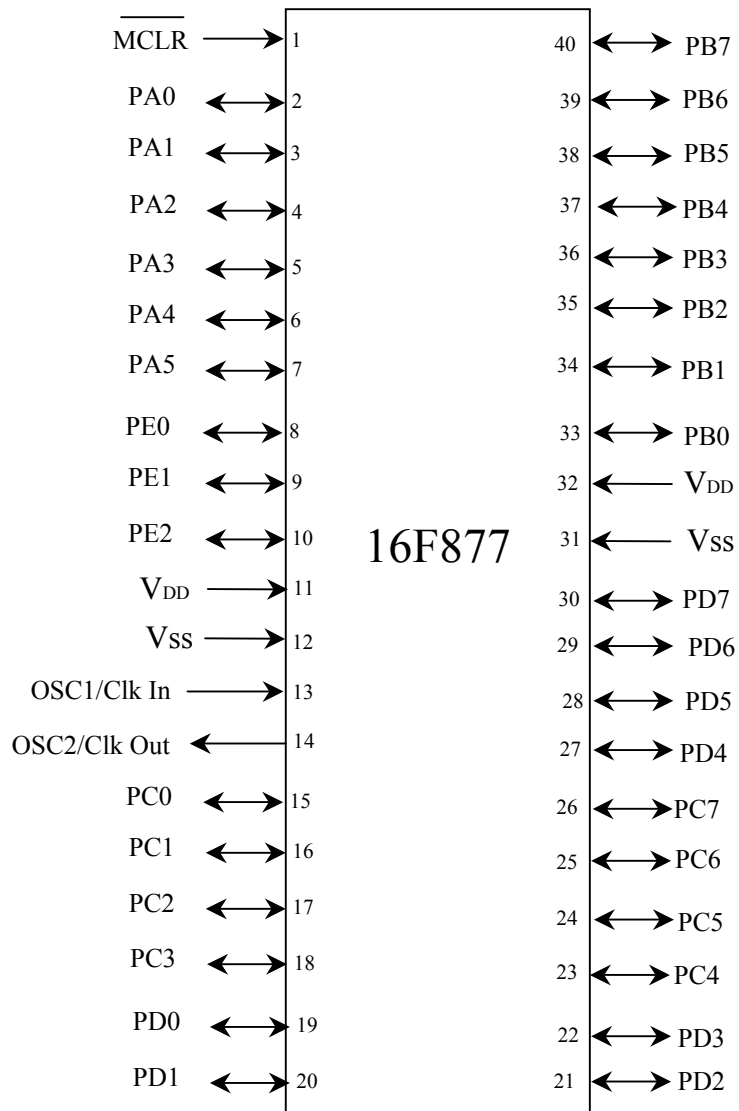
Les 2 bits MSB des 13 bits d'adresse (bits 11 et 12), viennent du registre PCLATH ( bits 3 et 4) qui est à l'adresse : h'0A'.

Il faut impérativement les positionner pour la bonne page, avant d'utiliser les instructions: CALL et GOTO.

## BROCHAGE : 16F876 ( 28 broches étroit)



## BROCHAGE : 16F877 ( 40 broches)





## LES REGISTRES INTERNES

**STATUS REGISTER** : ( h'03' ou h'83' ou h'103 ou h'183' ).

On accède indifféremment à ce registre par une quelconque de ces 4 adresses.

Bit 7

Bit 0

IRP	RP1	RP0	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	C
-----	-----	-----	------------------------	------------------------	---	----	---

Au reset : STATUS = 00011XXX

Bit 7 : **IRP** = permet la sélection des pages en adressage indirect.

Pour la PAGE 0 (de 00 à 7F) et la PAGE 1 (de 80 à FF) ce bit doit être laissé à "0". Mis à "1" il permettra d'atteindre la PAGE 3 (de 100 à 17F) et la PAGE 4 (de 180 à 1FF).

Bits 6 et 5 : **RP1 et RP0** = permettent la sélection des pages en adressage direct.

RP1	RP0	Page sélectionnée
0	0	PAGE 0 de 00 à 7F
0	1	PAGE 1 de 80 à FF
1	0	PAGE 2 de 100 à 17F
1	1	PAGE 3 de 180 à 1FF

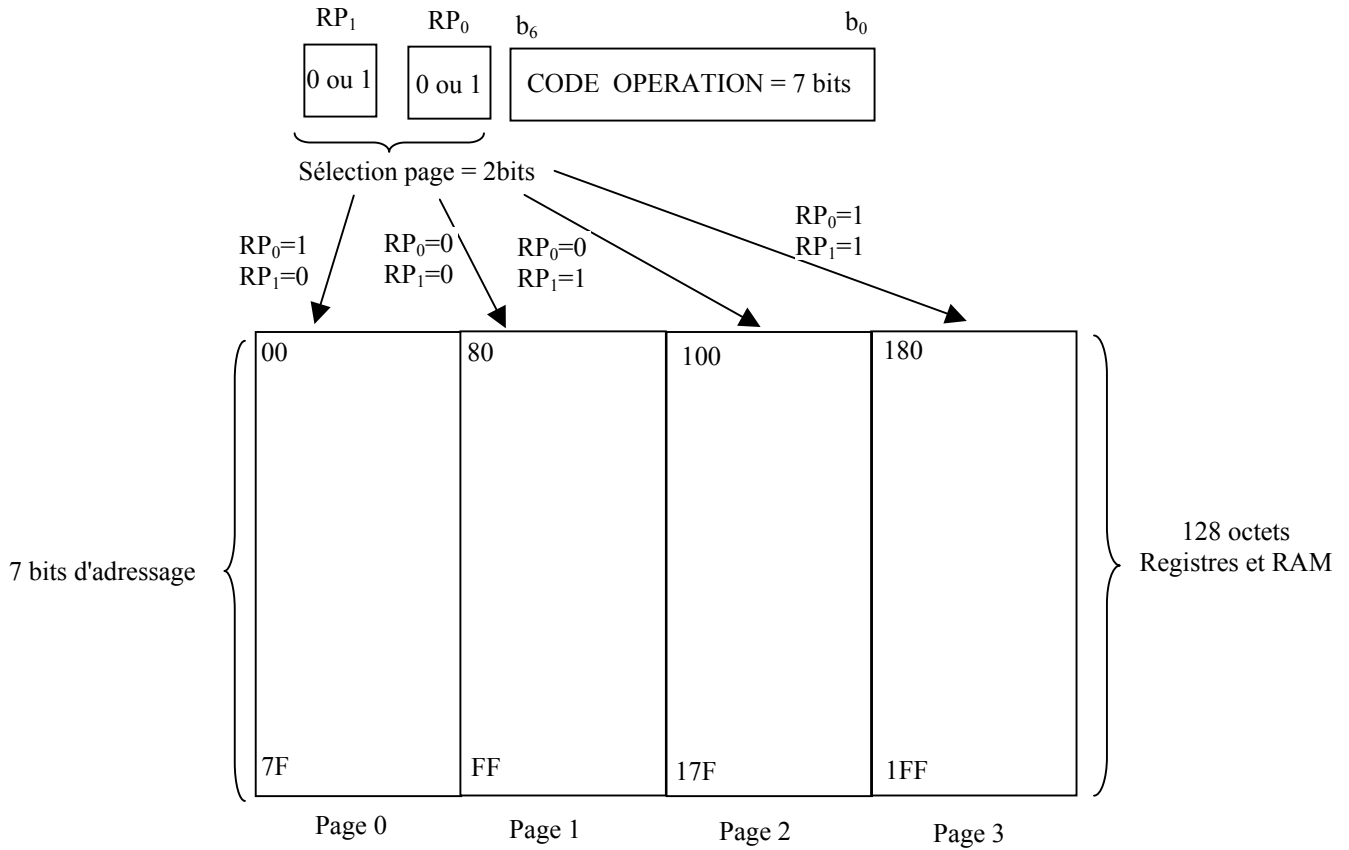
Exemple: PAGE0      BCF STATUS,5 ; RP0=0  
                           BCF STATUS,6 ; RP1=0

PAGE1      BSF STATUS,5 ; RP0=1  
                           BCF STATUS,6 ; RP1=0

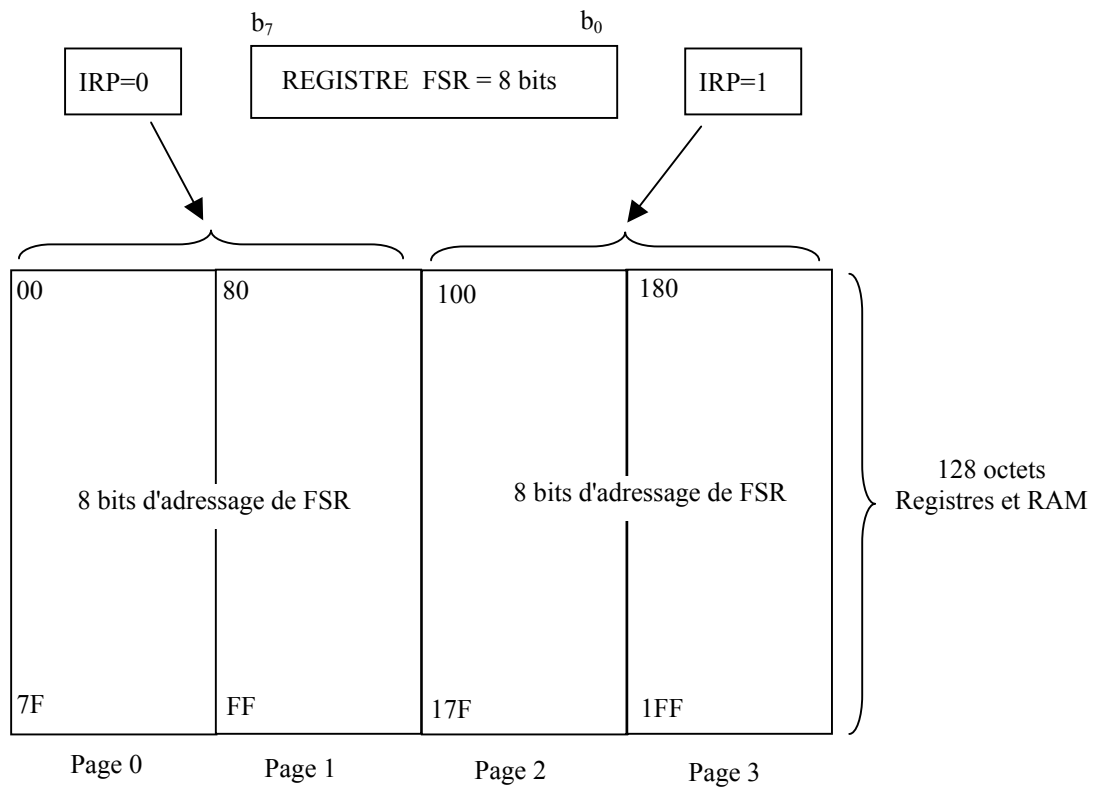
PAGE2      BCF STATUS,5 ; RP0=0  
                           BSF STATUS,6 ; RP1=1

PAGE3      BSF STATUS,5 ; RP0=1  
                           BSF STATUS,6 ; RP1=1

## Adressage DIRECT



## Adressage INDIRECT



Bit 4 :  $\overline{\text{TO}}$  = Time Out bit.

Bit en lecture seulement.

1 = Après une mise sous tension, après une RAZ du watchdog (CLRWDT) ou bien après l'instruction SLEEP.

0 = Signifie qu'un Time Out du timer de watchdog est survenu.

Bit 3 :  $\overline{\text{PD}}$  = Power Down bit.

1 = Après une mise sous tension ou bien après une RAZ du Watchdog.

0 = Après l'instruction SLEEP.

Bit 2 : **Z** = Zero bit.

1 = Le résultat d'une opération arithmétique ou logique est zéro.

0 = Le résultat d'une opération arithmétique ou logique est différent de zéro.

Bit 1 : **DC** = Digit Carry bit.

1 = Une retenue sur le 4eme bit des poids faible est survenue après les instructions : ADDWF et ADDLW.

0 = Pas de retenue sur le 4eme bit des poids faible.

Bit 0 : **C** = Carry bit.

1 = Une retenue sur le bit MSB est survenue après les instructions ADDWF et ADDLW.

0 = Pas de retenue sur le bit MSB.

### OPTION REGISTER : ( h'81' ou h'181' ).

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..

Bit 7

Bit 0

$\overline{\text{RBPU}}$	INTEDG	TOCS	TOSE	PSA	PS2	PS1	PS0
--------------------------	--------	------	------	-----	-----	-----	-----

Au reset : OPTION = 11111111

Bit 7 :  $\overline{\text{RBPU}}$  = Pull up Enable bit on Port B.

1 = Pull up désactivé sur le Port B.

0 = Pull up activé.

Bit 6 : **INTEDG** = Interrupt Edge 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 Assignment bit.  
 1 = Le prédiviseur est affecté au watchdog..  
 0 = Le prédiviseur est affecté au Timer TMR0.

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

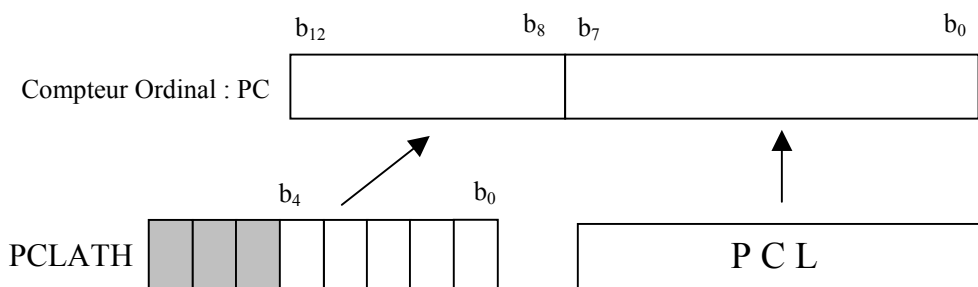
PS2	PS1	PS0	Prédiv Timer	Prédiv Watchdog
0	0	0	2	1
0	0	1	4	2
0	1	0	8	4
0	1	1	16	8
1	0	0	32	16
1	0	1	64	32
1	1	0	128	64
1	1	1	256	128

Quand le prédiviseur est affecté au Watchdog (PSA=1), TMR0 est prédivisé par 1.

**PCL REGISTER** : ( h'02' ou h'82' ou h'102' ou h'182' ).

**PCLATH REGISTER** : ( h'0A' ou h'8A ou h'10A' ou h'18A' )

Le compteur de programme est sur 13 bits. Les 8 bits de poids faible sont dans le registre PCL qui est en lecture/écriture. Les 5 bits de poids forts ne sont pas lisibles mais on peut les écrire indirectement à travers le registre PCLATH.



## INTCON REGISTER : ( h'0B' ou h'8B' ou h'10B' ou h'18B' ).

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

Bit 7

Bit 0

GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF
-----	------	------	------	------	------	------	------

Au reset : INTCON = 0000000X

Bit 7 : **GIE** = Global Interrupt Enable bit

1 = Autorise toutes les interruptions non masquées.

0 = Désactive toutes les interruptions.

Bit 6 : **PEIE** = Peripheral Interrupt Enable bit.

1 = Autorise les interruptions causées par les périphériques.

0 = Désactive les interruptions causées par les périphériques.

Bit 5 : **TOIE** = Timer TMR0 Overflow Interrupt Enable bit.

1 = Autorise les interruptions du Timer TMR0.

0 = Désactive les interruptions du Timer TMR0.

Bit 4 : **INTE** = RB0/Int Interrupt Enable 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 Interrupt Enable 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 : **TOIF** = Timer TMR0 Overflow Interrupt 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 : **INTF** = RB0/Int Interrupt 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 Interrupt Flag bit. Ce flag doit être remis à zéro par programme.

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.

## **PIE1 REGISTER : ( h'8C' : page 1).**

Ce registre contient les bits individuels d'autorisation pour les Interruptions des périphériques. Le bit 6 de INTCON (PEIE) doit être mis à "1" pour autoriser une quelconque IT de périphérique.

Bit 7

Bit 0

PSPIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE
-------	------	------	------	-------	--------	--------	--------

Au reset : PIE1 = 00000000

Bit 7 : **PSPIE** = Parallel Slave Port Interrup Enable bit .

*Ce bit n'existe pas pour un PIC 16F876 ( 28 pins). Toujours garder ce bit à "0".*

1 = Autorise les interruptions R/W du port SSP .

0 = Désactive toutes ces interruptions.

Bit 6 : **ADIE** = A/D converter Interrup Enable bit .

1 = Autorise les interruptions du convertisseur analogique/digital .

0 = Désactive cette interruption.

Bit 5 : **RCIE** = USART Receive Interrup Enable bit .

1 = Autorise les interruptions en réception de l'USART .

0 = Désactive cette interruption.

Bit 4 : **TXIE** = USART Transmit Interrup Enable bit .

1 = Autorise les interruptions en émission de l'USART .

0 = Désactive cette interruption.

Bit 3 : **SSPIE** = Synchronous Serial port Interrup Enable bit .

1 = Autorise les interruptions du module Synchrone (I2C) .

0 = Désactive cette interruption.

Bit 2 : **CCP1IE** = CCP1 Interrup Enable bit .

1 = Autorise les interruptions du CCP1 .

0 = Désactive cette interruption.

Bit 1 : **TMR2IE** = TMR2 Interrup Enable bit .

1 = Autorise les interruptions du Timer 2 TMR2 .

0 = Désactive cette interruption.

Bit 0 : **TMR1IE** = TMR1 overflow Interrup Enable bit .

1 = Autorise les interruptions de débordement du Timer 1 TMR1 .

0 = Désactive cette interruption.

## **PIR1 REGISTER : ( h'0C' : page 0).**

Ce registre contient les FLAG associés aux interruptions des périphériques. Ces Flag passent à "1" quand une IT correspondante survient et que le bit d'autorisation est bien positionné. Ces Flag doivent être remis à "0" par Soft.

Bit 7

Bit 0

PSPIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF
-------	------	------	------	-------	--------	--------	--------

Au reset : PIR1 = 00000000

Bit 7 : **PSPIE** = Parallel Slave Port Interrup Flag bit .

1 = Une opération de R/W vient d'avoir lieu sur le port SSP .

0 = Il n'y a pas eu de R/W sur le port SSP.

Bit 6 : **ADIF** = A/D converter Interrup Flag bit .

1 = Une conversion A/D est terminée .

0 = la conversion A/D n'est pas terminée.

Bit 5 : **RCIF** = USART Receive Interrup Flag bit .

1 = Le buffer de réception de l'USART est plein (donnée reçue).

0 = Le buffer de réception de l'USART est vide ( rien de reçu).

Bit 4 : **TXIF** = USART Transmit Interrup Flag bit .

1 = Le buffer de transmission de l'USART est vide ( on peut le remplir).

0 = Le buffer de réception de l'USART est plein ( on ne peut pas le charger).

Bit 3 : **SSPIF** = Synchronous Serial Port Interrup Flag bit .

1 = Une condition d'IT du module SSP est apparue .

0 = Aucune condition d'IT n'est apparue .

Bit 2 : **CCP1IF** = CCP1 Interrup Flag bit .

1 = Une condition de Capture ou de Compare du Timer1 a fait une IT.

0 = Pas d'IT de capture ou de Compare du TIMER 1.

Bit 1 : **TMR2IF** = TMR2 Interrup Flag bit .

1 = Le Timer2 a fait une IT.

0 = Pas d'IT du TIMER 2.

Bit 0 : **TMR1IF** = TMR1 Overflow Interrup Flag bit .

1 = Le débordement Timer 1 a fait une IT.

0 = Pas de débordement du TIMER 2.

## LES PORTS ENTREE / SORTIE

### PORTA (h05) et TRISA (h85) :

Ce port bidirectionnel est constitué de 6 bits. Le registre de direction correspondant est TRISA.

Quand on écrit un "1" dans TRISA , le bit correspondant du PORTA est configuré en ENTREE, et le driver de sortie est placé en haute impédance.

Si on écrit un "0", le port devient une SORTIE, et le contenu du latch correspondant est chargé sur la broche sélectionnée.

Le bit 4 du Port peut également servir pour l'entrée horloge du timer TMR0.

Les autres bits du Port sont partagés avec le CAN.

**ATTENTION:** - *après un reset le Port A est configuré en CAN. Il faut impérativement le configurer en I/O digitale pour l'utiliser comme tel. Il faut pour cela accéder au registre ADCON1 en h'9F'.*

Ce registre sera étudié dans le chapitre concernant le CAN.

On retiendra seulement que pour configurer les 5 bits du Port A en I/O digitales, il faut positionner les 4 bits PCFG à 0110.

Exemple:

```
ADCON1 EQU h'9F'  
PAGE1  
MOVLW h'06'  
MOVWF ADCON1
```

**ATTENTION:** - *Le Port PA4 qui est partagé avec l'entrée horloge du Timer 0 est un Drain ouvert. Il faut donc le tirer au +V<sub>cc</sub> par une résistance de 10 K $\Omega$  pour l'utiliser en tant que sortie.*

### PORTB (h06) et TRISB (h86) :

Il comporte 8 bits. Le registre de direction correspondant est TRISB.

Si on écrit un "1" dans le registre TRISB, le driver de sortie correspondant passe en haute impédance. Si on écrit un "0", le contenu du Latch de sortie correspondant est recopié sur la broche de sortie.

Chaque broche du PORT B est munie d'un tirage au +V<sub>DD</sub> que l'on peut mettre ou non en service en mode entrée uniquement. On active cette fonction par la mise à "0" du bit 7 dans le registre OPTION en h'81'.

Au reset, le tirage est désactivé.

Il est inactif quand le port est configuré en sortie.

Les 4 broches PB7 PB6 PB5 et PB4 provoquent une interruption sur un changement d'état si elles sont configurées en ENTREE.

On doit remettre à zéro le Flag de cette interruption (bit 0 du registre INTCON en h'0B') dans le programme d'interruption.

Cette possibilité d'interruption sur un changement d'état associé à la fonction de tirage configurable sur ces 4 broches, permet l'interfaçage facile avec un clavier. Cela rend possible le réveil du PIC en mode SLEEP par un appui sur une touche du clavier.

Le bit 0 du PORT B peut également être utilisé comme entrée d'interruption externe. Le choix du front de déclenchement se fait en configurant le bit 6 du registre OPTION.

**Au RESET:** PORT A et PORT B configurés en ENTREE (TRISA et TRISB = 1)  
PORT B : Tirage désactivé.

### **PORTC (h07) et TRISC (h87) :**

Il s'agit d'un PORT 8 bits bidirectionnel.

Il est partagé avec le module de transmission synchrone I<sup>2</sup>C et l'USART.

## USART ou SCI

Elle utilise le pins 6 et 7 du PORT C: **PC6 = T<sup>x</sup> DATA** et **PC7 = R<sup>x</sup> DATA**

Les 5 registres utilisés sont :

Registre Emission : **TXREG** en h'19' page 0.

Registre Réception: **RCREG** en h'1A' page 0.

Registre d'état Emission : **TXSTA** en h'98' page 1.

Registre d'état Réception : **RCSTA** en h'18' page 0.

Registre du choix de la vitesse : **SPBRG** en h'99 page 1.

### TXSTA REGISTER : ( h'98' : page 1).

Bit 7

Bit 0

CSRC	TX9	TXEN	SYNC		BRGH	TRMT	TX9D
------	-----	------	------	--	------	------	------

Au reset : TXSTA = 00000010

Bit 7 : **CSRC** = Clock Source en synchrone. Sans importance en asynchrone.

Bit 6 : **TX9** = Autorisation d'émission sur 9 bits.

1 = Autorisé .

0 = Non autorisé.

Bit 5 : **TXEN** = Autorisation d'émission.

1 = Autorisé .

0 = Non autorisé.

Bit 4 : **SYNC** = Sélection mode Synchrone / Asynchrone.

1 = Mode synchrone.

0 = Mode asynchrone.

Bit 3 : **Non implémenté**

Bit 2 : **BRGH** = Sélection vitesse rapide en mode asynchrone.

1 = Vitesse haute sélectionnée.

0 = Vitesse basse sélectionnée.

Bit 1 : **TRMT** = bit d'état du registre à décalage Emission.

1 = Registre vide, donc émission terminée.

0 = Registre plein, donc émission en cours.

Bit 0 : **TX9D** = 9<sup>eme</sup> bit de Data transmise .

Ce bit peut être le bit de la parité.

**SPBRG REGISTER : ( h'99' : page 1).**

Le Baud Rate Generator est un registre 8 bits qui contient le facteur de division ( N ) de l'horloge interne qui permet d'obtenir la vitesse commune d'émission et de réception.

En mode Asynchrone (bit SYNC = 0) suivant l'état du bit BRGH on aura le choix entre 2 vitesses : haute pour BRGH=1 et basse pour BRGH=0.

<b>BRGH=0 VITESSES BASSES</b>	<b>BRGH=1 VITESSES HAUTES</b>
$VITESSE = \frac{F_{oscill}}{64(N + 1)}$	$VITESSE = \frac{F_{oscill}}{16(N + 1)}$
$N = \frac{F_{oscill}}{64.Vitesse} - 1$	$N = \frac{F_{oscill}}{16.Vitesse} - 1$

Le nombre N est le nombre entier, arrondi de la valeur trouvée par les équations ci dessus.

Il est recommandé d'utiliser si possible les vitesses hautes (BRGH=1), même pour des vitesses faibles, car dans ce cas on minimise l'erreur, en obtenant un nombre N plus grand.

**Valeurs de N pour diverses vitesses avec un Quartz de 8 MHz :**

<b>VITESSES en Bits/sec ou BAUDS</b>	<b>VITESSES BASSES BRGH = 0</b>	<b>VITESSES HAUTES BRGH = 1</b>
119200	x	4
57600	x	8
38400	x	12
19200	5	25
9600	12	51
4800	25	103
2400	51	207
1200	103	x



## RCSTA REGISTER : ( h'18' : page 0 ).

Bit 7

Bit 0

SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D
------	-----	------	------	-------	------	------	------

Au reset : RCSTA = 0000000X

Bit 7 : **SPEN** = Serial Port Enable. PC7 et PC6 configurés pour le port série.

1 = Port série en service .

0 = Port série désactivé.

Bit 6 : **RX9** = Autorisation de réception sur 9 bits.

1 = Autorisé .

0 = Non autorisé.

Bit 5 : **SREN** = Single Receive Enable. Réserve pour mode Synchrone.

Non utilisé en mode Asynchrone.

Bit 4 : **CREN** = Continuous Receive Enable.

1 = Autorise la réception en continu.

0 = Désactive la réception en continu.

Bit 3 : **ADDEN** = Address Detect Enable. En mode Asynchrone 9 bits :

1 = Autorise la détection d'adresse, et charge la Data dans le registre de réception RCREG quand le 9<sup>eme</sup> bit du registre de dé sérialisation vaut "1".

0 = Deselecte la détection d'adresse. Tous les octets sont reçus et le 9<sup>eme</sup> bit peut servir de bit de parité.

Bit 2 : **FERR** = Framing Error .

1 = Une erreur de Framing est survenue.

0 = Pas d'erreur de Framing.

Bit 1 : **OERR** = Overrun Error.

Un octet est reçu alors que le registre de réception n'a pas été vidé par lecture.

1 = Erreur Overrun.

0 = Pas d'erreur Overrun.

Bit 0 : **RX9D** = 9<sup>eme</sup> bit de Data reçue .

Ce bit peut être le bit de la parité.

## **EMISSION:** Pin PC6 = T<sup>x</sup> DATA.

L'émission est autorisée par la mise à "1" du bit 5 de TXSTA: TXEN = 1.

La DATA à transmettre est mise dans le registre TXREG en h'19' page 0. Ce registre prévient qu'il est vide en mettant le flag TXIF à "1" (bit 4 de PIR1).

Ce flag passe à "0" dès que l'on charge un octet dans le registre TXREG. Il repasse à "1" par Hard quand le registre est vidé par transfert dans le registre de sérialisation : TSR. Ce registre n'est pas accessible par l'utilisateur, il n'a pas d'adresse.

Si on charge alors un 2<sup>eme</sup> octet dans le registre TXREG le flag TXIF va passer à "0" et y rester tant que le registre TSR n'aura pas complètement sérialisé l'octet précédent à transmettre. Dès que le STOP de l'octet précédent a été transmis, le registre TXREG est transféré dans TSR et le flag TXIF repasse à "1" signalant ainsi que le registre de transmission TXREG est vide et peut donc recevoir une nouvel octet à transmettre.

Le bit TRMT ( bit 1 de TXSTA) informe sur l'état du registre TSR. Quand le registre TSR n'a pas fini de sérialisé, TRMT=0. Ce flag repasse à "1" quand le registre est vide, c'est à dire quand le stop a été émis.

Le flag TXIF permet aussi de générer une interruption, à condition qu'elle soit autorisée par mise à "1" du bit 4 de PIE1 : TXIE = 1. Il faut dans ce cas autoriser les interruptions des périphériques par mise à "1" du bit 6 de INTCON: PEIE = 1, et par la mise à "1" du bit 7 : GIE = 1.

## **PROCEDURE POUR EMETTRE :**

- Initialiser SPBRG pour la vitesse désirée et choix pour BRGH.
- Autoriser mode Asynchrone : SYN = 0 et SPEN = 1.
- Eventuellement faire TX9 = 1 si une émission sur 9 bits est désirée.
- Autoriser l'émission par TXEN = 1.
- Si une transmission 9 bits a été choisie, mettre le 9<sup>eme</sup> bit dans TX9D.
- Mettre l'octet à transmettre dans TXREG.
- Avant de remettre l'octet suivant à transmettre dans TXREG, il faut tester le flag TXIF qui est à "0" si le registre n'est pas disponible. Dès que le registre est vide ce flag passe à "1" et on peut alors charger TXREG par l'octet à transmettre.
- Pour savoir si le dernier octet a été émis, il suffit de tester le flag TRMT qui signale par son passage à "1" que le dernier bit du dernier octet et son STOP ont bien été sérialisés.
- On peut alors stopper le module émission de l'USART par TXEN=0.

## **RECEPTION:** Pin PC7 = R<sup>x</sup> DATA.

La réception est autorisée par la mise à "1" du bit 4 de RCSTA: CREN = 1.

La DATA reçue est mise dans le registre RCREG en h'1A' page 0. Ce registre prévient qu'il est plein en mettant le flag RCIF à "1" (bit 5 de PIR1). On peut autoriser la génération d'une interruption quand RCIF = 1 , c'est à dire quand une donnée valide est disponible dans RCREG par mise à "1" du bit 5 de PIE1 : RCIE = 1. Le flag RCIF repasse à "0" par hard quand on vide le registre RCREG par sa lecture.

Si le STOP d'un 2eme octet survient alors que le registre RCREG n'a pas été vidé, une erreur OVERRUN se produit. Elle est signalée par le passage à "1" du bit 1 de RCSTA : OERR=1. L'octet dans le registre de dé sérialisation est alors perdu.

Le bit d'erreur OERR doit être remis à zéro par soft. Pour cela il faut stopper la réception par CREN=0 puis remettre en service la réception par CREN=1.

En fait le registre RCREG est un double registre FIFO. On peut donc recevoir 2 octets et ne pas les lire avant qu'un 3<sup>eme</sup> octet ne fasse un OVERRUN.

On doit alors lire deux fois RCREG pour le vider les 2 octets reçus qui sont dans le FIFO.

Si un STOP est trouvé à "0" alors une ERROR FRAMING est générée par mis à "1" du bit 2 de RCSTA: FERR=1.

## **PROCEDURE POUR RECEVOIR :**

- Initialiser SPBRG pour la vitesse désirée et choix pour BRGH.
- Autoriser mode Asynchrone : SYN = 0 et SPEN = 1.
- Eventuellement faire RX9 = 1 si une réception sur 9 bits est désirée.
- Eventuellement faire RCIE = 1 si une réception par interruption est désirée.
- Autoriser l'émission par RCEN = 1.
- Test du Flag RCIF (ou attente IT) pour savoir si un octet a été reçu.
- Lire éventuellement le 9<sup>eme</sup> bit de Data dans RCSTA pour tester la parité.
- Lire les bits FERR et OERR pour déterminer les erreurs éventuelles.
- Si une erreur est survenue il faut faire CREN=0 puis CREN=1 pour RAZ.
- Lecture du registre RCREG pour récupérer l'octet reçu.

## **MODE 9 bits avec DETECTION D'ADRESSE :**

- Faire toutes les initialisations précédentes: SPBRG, BRGH, SYN, SPEN.
- Faire RX9=1 ADDEN=1 et CREN=1
- Dès que registre RCREG est signalé plein, il s'agit de l'adresse ( 9<sup>eme</sup> bit =1).
- Le lire et comparer avec l'adresse déterminée. Si OK faire ADDEN=0 pour recevoir maintenant tous les octets, même si le 9<sup>eme</sup> bit n'est pas à "1". Si pas OK rester avec ADDEN=1 pour ne recevoir que les octets filtrés d'adresse.

## MODULE MSSP pour I<sup>2</sup>C

Le module MSSP (Master Synchronous Serial Port) permet de faire des transmissions au format I<sup>2</sup>C (Inter Integrated Circuits) en tant que Maître ou Esclave.

Il utilise les pins 3 et 4 du PORT C: **PC3 = SCL** et **PC4 = SDA**.

Les 5 registres utilisés sont :

Registre Buffer **SSPBUF** en h'13' page 0.

Registre d'adresse **SSPADD** en h'93' page 1.

Registre d'état **SSPSTAT** en h'94' page 1.

Registre de contrôle **SSPCON** en h'14' page 0.

Registre de contrôle n°2 **SSPCON2** en h'91' page 1.

### SSPSTAT REGISTER : ( h'94' : page 1).

Bit 7

Bit 0

SMP	CKE	D/ $\bar{A}$	P	S	R/ $\bar{W}$	UA	BF
-----	-----	--------------	---	---	--------------	----	----

Au reset : SSPSTAT = 00000000

Bit 7 : **SMP** = bit d'échantillonnage.

Bit 6 : **CKE** = Clock Edge Select. Ce bit à "0" : conforme aux spécifications I<sup>2</sup>C.

Bit 5 : **D/ $\bar{A}$**  = Data / Adresse bit.

1 = indique que le dernier octet reçu est une Data.

0 = indique que le dernier octet reçu est une adresse.

Bit 4 : **P** = STOP bit. Ce bit est remis à "0" quand le module SSP est désactivé.

1 = indique qu'un Stop a été détecté.

0 = indique qu'il n'a pas été détecté de Stop.

Bit 3 : **S** = START bit. Ce bit est remis à "0" quand le module SSP est désactivé.

1 = indique qu'un Start a été détecté.

0 = indique qu'il n'a pas été détecté de Start.

Bit 2 : **R/ $\bar{W}$**  = Information sur le bit R/W. Donne la valeur de ce bit qui suit la dernière adresse. Ce bit est valable jusqu'à la réception du STOP.

- *Mode ESCLAVE* :

1 = R.

0 = W.

- *Mode MAITRE* :

1 = Transmission en cours.

0 = Pas de transmission en cours.

Bit 1 : **UA** = mise à jour adresse. Utilisé en mode 10 bits seulement.

Bit 0 : **BF** = Buffer plein.

- Mode Réception I<sup>2</sup>C:

1 = Registre plein : réception terminée.

0 = Registre vide : donc réception non terminée.

- Mode Emission I<sup>2</sup>C:

1 = Registre plein : donc émission en cours.

0 = Registre vide : donc émission terminée.

### SSPCON REGISTER : ( h'14' : page 0).

Bit 7

Bit 0

WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0
------	-------	-------	-----	-------	-------	-------	-------

Au reset : SSPCON = 00000000

Bit 7 : **WCOL**= détection de collision à l'écriture.

1 = Collision à l'écriture.

0 = Pas de collision.

Bit 6 : **SSPOV** = Overflow indicator en réception. Ne sert pas en émission

Doit être remis à "0" par soft

1 = Un octet vient d'être reçu dans le buffer qui n'était pas vidé : Overflow.

0 = Pas d'Overflow.

Bit 5 : **SSPEN** = Enable Module SSP.

1 = Module I<sup>2</sup>C activé et pin SCL et SDA sur PC3 et PC4 configurées.

0 = Module I<sup>2</sup>C désactivé. PC3 et PC4 libres pour usage I/O.

Bit 4 : **CKP** = Polarité du Clock. Inutilisé en mode maître.

En mode Esclave :

1 = Autorise horloge.

0 = Maintient CLK = 0.

Bit 3 à bit 0 : **SSPM3 / SSPM0** = Select Mode.

SSPM3	SSPM2	SSPM1	SSPM0	MODE
0	1	1	0	I <sup>2</sup> C esclave 7 bits.
0	1	1	1	I <sup>2</sup> C esclave 10 bits.
1	0	0	0	I <sup>2</sup> C maître CLK= F <sub>osc</sub> /4(SSPADD+1).

## SSPCON2 REGISTER : ( h'91' : page 1).

Bit 7

Bit 0

GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN
------	---------	-------	-------	------	-----	------	-----

Au reset : SSPCON2 = 00000000

Bit 7 : **GCEN** = Appel général en mode I<sup>2</sup>C esclave seulement.

1 = Autorise IT quand une adresse appel général (h'0000) est reçue.

0 = Adresse appel général désactivée.

Bit 6 : **ACKSTAT** = Bit de ACK en mode I<sup>2</sup>C maître seulement.

1 = Pas de ACK reçu de l'esclave.

0 = ACK de l'esclave a été reçu.

Bit 5 : **ACKDT** = ACK Data bit. En mode maître et réception : valeur qui sera transmise quand on va lancer une séquence de ACK en fin de réception.

1 = NO ACK.

0 = ACK.

Bit 4 : **ACKEN** = ACK Sequence Enable. En mode maître et réception, lance une séquence de ACK ou de NOACK suivant la valeur dans ACKDT.

1 = Lance la séquence de ACK et transmet ACKDT. Remis à "0" par hard.

0 = Pas de séquence de ACK.

Bit 3 : **RCEN** = Receive Enable Bit. En mode I<sup>2</sup>C maître seulement.

1 = Autorise mode réception.

0 = Pas de réception autorisée.

Bit 2 : **PEN** = Stop Condition Enable. En mode I<sup>2</sup>C maître seulement.

1 = Lance la séquence de STOP sur SDA et SCL. Remis à "0" par hard.

0 = Pas de séquence de STOP.

Bit 1 : **RSEN** = Repeated Start Condition . En mode I<sup>2</sup>C maître seulement.

1 = Séquence de répétition de START sur SDA et SCL. Remis à "0" par hard.

0 = Pas de séquence de répétition de START.

Bit 0 : **SEN** = Start Condition Enable. En mode I<sup>2</sup>C maître seulement.

1 = Lance la séquence de START sur SDA et SCL. Remis à "0" par hard.

0 = Pas de séquence de START.

## **MODE MAITRE :**

Les pins SDA et SCL sont manipulées par le Hard. Les événements qui causent le passage à "1" du flag SSPIF ( bit 3 de PIR1 en h'0C') et éventuellement une IT si elles sont autorisées, sont :

- Condition de START.
- Condition de STOP.
- Répétition d'un START.
- ACK après un transfert.
- Transfert d'octet en Emission ou réception.

***Attention:*** Le flag SSPIF doit être remis à "0" par soft.

## **Lancement d'un START:**

- L'utilisateur doit mettre le bit SEN (de SSPCON2) à "1".
- Les pins SDA et SCL étant toutes les deux à "1", le module fait passer la pin SDA de "1" à "0" ce qui génère une condition de START.
- Le bit S (de SSPSTAT) passe à "1" pour signaler le START.
- A la fin du START, le bit SEN est remis à "0" par le hard.
- Dès que le flag SSPIF (de PIR1) passe à "1" pour signaler la fin du START, on peut charger le registre de transmission SSBUF avec l'octet à transmettre. Ne pas oublier de remettre ce flag à "0" par le soft.

## **Transmission :**

- Dès que le registre SSBUF est chargé, le bit BF (de SSPSTAT) passe à "1" pour signaler que la transmission est en cours.
- Au 8<sup>eme</sup> coup de CLK, la transmission est terminée et le bit BF repasse à "0".

## **Acquittement :**

- Quand l'esclave répond le ACK, le bit ACKSTAT passe à "0".
- La fin du ACK est signalé par le Flag SSPIF qui passe à "1". On doit alors remettre ce flag à "0" par soft.

## **Lancement d'un STOP :**

- L'utilisateur doit mettre le bit PEN (de SSPCON2) à "1". Le module fait alors passer SDA à "0" puis force SCL à "1". Quand SCL est à "1" il fait passer SDA de "0" à "1", ce qui génère une condition de STOP.
- Le bit P (de SSPSTAT) passe à "1" pour signaler le STOP.
- A la fin du STOP, le bit PEN est remis à "0" par le hard.
- Le flag SSPIF (de PIR1) passe à "1" pour signaler la fin du STOP.

## Réponse du maître à l'esclave :

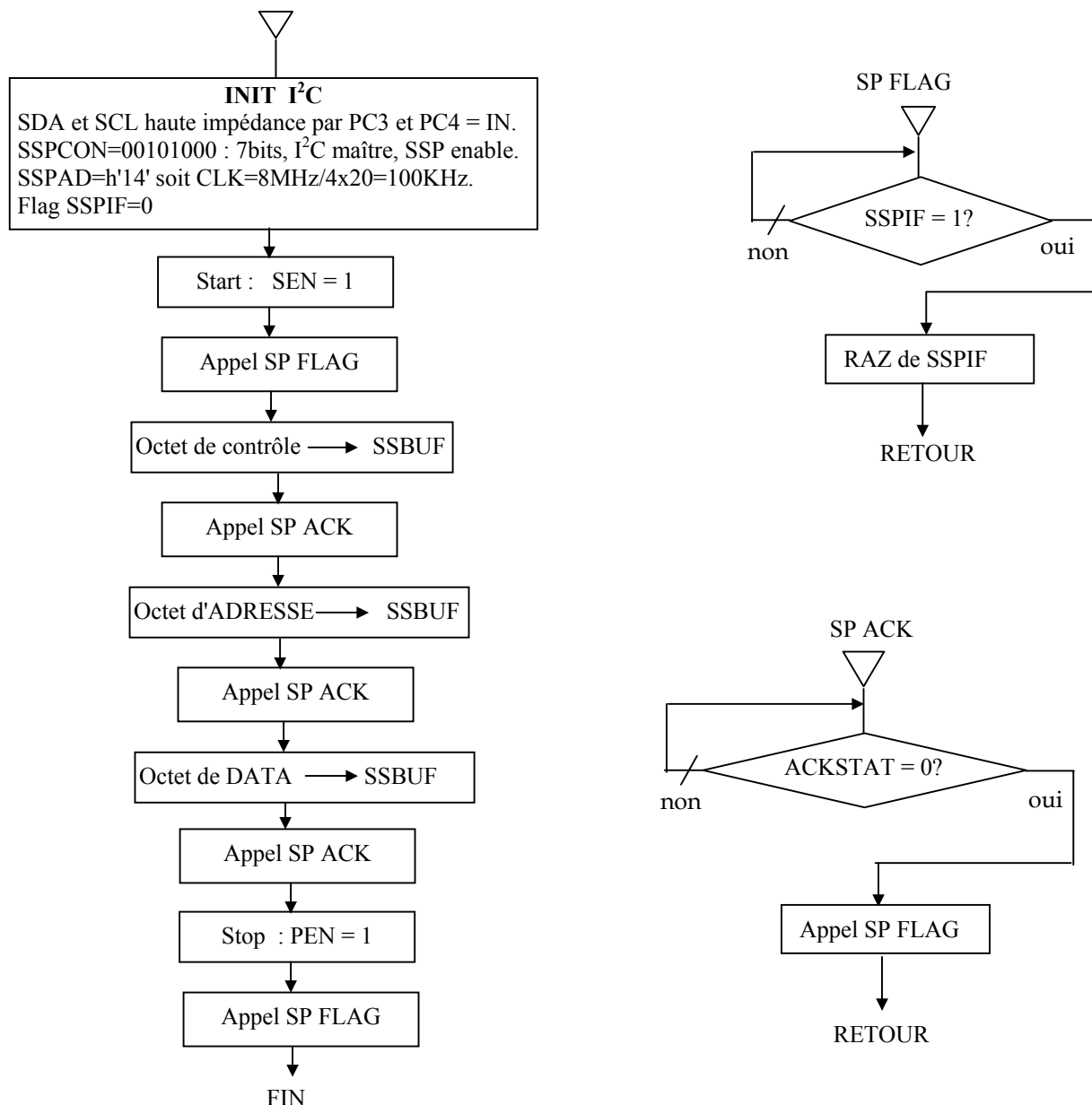
Il s'agit soit d'un ACK soit d'un NOACK :

- Positionner le bit ACKDT (de SSPCON2) en fonction de la réponse à faire.
- Mettre le bit ACKEN (de SSPCON2) à "1".
- La fin de transmission du ACK ou du non ACK est signalé par le flag SSPIF qui passe à "1". Ne pas oublier de le remettre à "0" par soft.

## Réception d'un octet par le maître :

- Pour cela il faut mettre le module en réception en portant le bit RCEN à "1".
- La fin de réception est signalé par le passage à "1" du flag SSPIF, que l'on remettra à "0" par soft.
- On peut alors lire l'octet reçu dans SSBUF.

**EXEMPLE de PROGRAMME** : Ecriture dans une EEPROM I<sup>2</sup>C type 24C16.





## CONVERTISSEUR A/D

Il est constitué d'un module convertisseur à 5 entrées pour le boîtier 28 broches ( 16F876) et à 8 entrées pour le boîtier 40 broches (16F877).

Les 5 premières entrées sont sur le Port A en PA<sub>0</sub>, PA<sub>1</sub>, PA<sub>2</sub>, PA<sub>3</sub> et PA<sub>5</sub>.

Les 3 entrées supplémentaires du boîtier 40 pins sont en PE<sub>0</sub>, PE<sub>1</sub> et PE<sub>2</sub>.

Le résultat de la conversion est codé sur 10 bits. C'est une valeur comprise entre h'000' et h'3FF'.

Les tensions de référence haute et basse peuvent être choisies par programmation parmi: V<sub>DD</sub> ou la broche PA<sub>3</sub> pour V<sub>REF</sub><sup>+</sup> et V<sub>SS</sub> ou la broche PA<sub>2</sub> pour V<sub>REF</sub><sup>-</sup>.

Les 4 registres utilisés par le module convertisseur A/D sont :

- ADRESH en h'1E' page 0 : MSB des 10 bits du résultat.
- ADRESL en h'9E' page 1 : LSB des 10 bits du résultat.
- ADCON0 en h'1F' page 0 : registre de contrôle n°0 du convertisseur.
- ADCON1 en h'9F' page 1 : registre de contrôle n°1 du convertisseur.

### **ADCON0 : ( h'1F' : page 0).**

Bit 7

Bit 0

ADSC1	ADSC0	CHS2	CHS1	CHS0	GO/Don <sup>e</sup>		ADON
-------	-------	------	------	------	---------------------	--	------

Au reset : ADCON0 = 00000000

Bit 7 et bit 6 : **ADSC1 et ADSC0** = Clock Select bits.

Ces 2 bits permettent de choisir la vitesse de conversion :

00= F<sub>osc</sub>/2.

01= F<sub>osc</sub>/8.

10= F<sub>osc</sub>/32.

11= Oscillateur RC interne.

Le temps de conversion d'un bit est T<sub>AD</sub>. Pour une conversion totale des 10 bits il faut : 12.T<sub>AD</sub>.

Pour que la conversion soit correcte il faut que T<sub>AD</sub> soit au minimum de **1,6µs**.

Avec l'oscillateur interne RC on a : T<sub>AD</sub> = 4 µs typique ( entre 2 et 6 µs ).

Bit 5 bit4 et bit 3 : **CHS2 CHS1 et CHS0** = Channel Select bits.

Ces 3 bits permettent de choisir l'entrée qui va être convertie.

Canal	CHS2	CHS1	CHS0	PORT
0	0	0	0	PA <sub>0</sub>
1	0	0	1	PA <sub>1</sub>
2	0	1	0	PA <sub>2</sub>
3	0	1	1	PA <sub>3</sub>
4	1	0	0	PA <sub>5</sub>
5	1	0	1	PE <sub>0</sub>
6	1	1	0	PE <sub>1</sub>
7	1	1	1	PE <sub>2</sub>

} Non disponible sur le 16F876 ( 28 pins).

Bit 2 : **GO/DONE** : Status bit si ADON=1.

1 = Démarre la conversion A/D. Ce bit est remis à "0" par hard.

0 = La conversion A/D est terminée.

Bit 1 : **Bit non implanté.**

Bit 0 : **ADON** : A/D on bit.

1= Convertisseur A/D en service.

0 = Convertisseur A/D à l'arrêt.

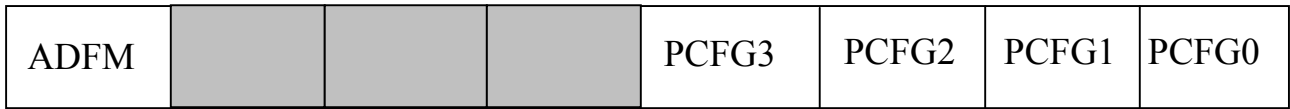
Temps de conversion  $T_{AD}$  en fonction du Quartz et des bits du Clock select :

QUARTZ	CLOCK	$T_{AD}$	$12.T_{AD}$	Ne convient pas si $T_{AD} < 1,6\mu s$
4 MHz	$F_{osc}/2 = 2 \text{ MHz}$	0,5 $\mu s$	6 $\mu s$	Ne convient pas
	$F_{osc}/8 = 500 \text{ KHz}$	2 $\mu s$	24 $\mu s$	OK
	$F_{osc}/32 = 125 \text{ KHz}$	8 $\mu s$	96 $\mu s$	OK
8 MHz	$F_{osc}/2 = 4 \text{ MHz}$	0,25 $\mu s$	3 $\mu s$	Ne convient pas
	$F_{osc}/8 = 1 \text{ MHz}$	1 $\mu s$	12 $\mu s$	Ne convient pas
	$F_{osc}/32 = 250 \text{ KHz}$	4 $\mu s$	48 $\mu s$	OK
12 MHz	$F_{osc}/2 = 6 \text{ MHz}$	0,16 $\mu s$	1,92 $\mu s$	Ne convient pas
	$F_{osc}/8 = 1,5 \text{ MHz}$	0,66 $\mu s$	8 $\mu s$	Ne convient pas
	$F_{osc}/32 = 375 \text{ KHz}$	2,6 $\mu s$	32 $\mu s$	OK
16 MHz	$F_{osc}/2 = 8 \text{ MHz}$	0,125 $\mu s$	1,5 $\mu s$	Ne convient pas
	$F_{osc}/8 = 2 \text{ MHz}$	0,5 $\mu s$	6 $\mu s$	Ne convient pas
	$F_{osc}/32 = 500 \text{ KHz}$	2 $\mu s$	24 $\mu s$	OK

**ADCON1 : ( h'9F' : page 1).**

Bit 7

Bit 0

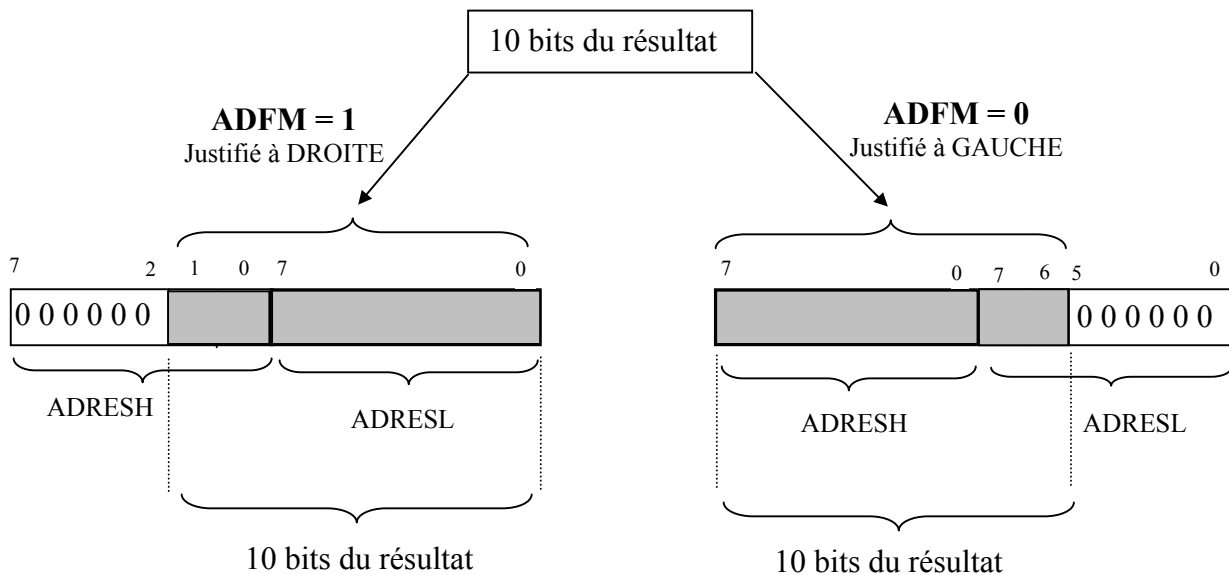


Au reset : ADCON1 = 00000000

Bit 7 : **ADFM** = A/D Result format.

1 = Justifié à droite. ADRESH ne contient que les 2 MSB du résultat. Les 6 MSB de ce registre sont lus comme des "0".

0 = Justifié à gauche. ADRESL ne contient que les 2 LSB du résultat. Les 6 LSB de ce registre sont lus comme des "0".



Bit 6 bit 5 et bit 4 : **Bits non implémentés.**

Bit 3 bit 2 bit 1 et bit 0 : **PCFG3 PCFG2 PCFG1 et PCFG0**

Bits de contrôle de la configuration des Ports.

Ces bits permettent de choisir le partage entre entrées analogiques et digitales sur les PORTS A et E.

Ils permettent également de choisir pour  $V_{REF}^+$  entre  $V_{DD}$  et  $PA_3$  et pour  $V_{REF}^-$  entre  $V_{SS}$  et  $PA_2$ .

## Configuration des PORTS en fonctions des 4 bits PCFG:

A = Entrée Analogique.

D = I/O Digitale.

4 Bits PCFG	N°7 PE <sub>2</sub>	N°6 PE <sub>1</sub>	N°5 PE <sub>0</sub>	N°4 PA <sub>5</sub>	N°3 PA <sub>3</sub>	N°2 PA <sub>2</sub>	N°1 PA <sub>1</sub>	N°0 PA <sub>0</sub>	V <sub>REF</sub> <sup>+</sup>	V <sub>REF</sub> <sup>-</sup>
0000	A	A	A	A	A	A	A	A	V <sub>DD</sub>	V <sub>SS</sub>
0001	A	A	A	A	V <sub>REF</sub> <sup>+</sup>	A	A	A	PA <sub>3</sub>	V <sub>SS</sub>
0010	D	D	D	A	A	A	A	A	V <sub>DD</sub>	V <sub>SS</sub>
0011	D	D	D	A	V <sub>REF</sub> <sup>+</sup>	A	A	A	PA <sub>3</sub>	V <sub>SS</sub>
0100	D	D	D	D	A	D	A	A	V <sub>DD</sub>	V <sub>SS</sub>
0101	D	D	D	D	V <sub>REF</sub> <sup>+</sup>	D	A	A	PA <sub>3</sub>	V <sub>SS</sub>
011x	D	D	D	D	D	D	D	D	V <sub>DD</sub>	V <sub>SS</sub>
1000	A	A	A	A	V <sub>REF</sub> <sup>+</sup>	V <sub>REF</sub> <sup>-</sup>	A	A	PA <sub>3</sub>	PA <sub>2</sub>
1001	D	D	A	A	A	A	A	A	V <sub>DD</sub>	V <sub>SS</sub>
1010	D	D	A	A	V <sub>REF</sub> <sup>+</sup>	A	A	A	PA <sub>3</sub>	V <sub>SS</sub>
1011	D	D	A	A	V <sub>REF</sub> <sup>+</sup>	V <sub>REF</sub> <sup>-</sup>	A	A	PA <sub>3</sub>	PA <sub>2</sub>
1100	D	D	D	A	V <sub>REF</sub> <sup>+</sup>	V <sub>REF</sub> <sup>-</sup>	A	A	PA <sub>3</sub>	PA <sub>2</sub>
1101	D	D	D	D	V <sub>REF</sub> <sup>+</sup>	V <sub>REF</sub> <sup>-</sup>	A	A	PA <sub>3</sub>	PA <sub>2</sub>
1110	D	D	D	D	D	D	D	A	V <sub>DD</sub>	V <sub>SS</sub>
1111	D	D	D	D	V <sub>REF</sub> <sup>+</sup>	V <sub>REF</sub> <sup>-</sup>	D	A	PA <sub>3</sub>	PA <sub>2</sub>



Ces entrées n'existent pas sur le 16F876 ( boîtier 28 pins).

### ATTENTION :

Au reset le registre ADCON1 est initialisé à h'00'. Cela signifie que les 5 bits du Port A et les 3 bits du Port E sont configurés en entrées analogiques.

Pour récupérer le 5 bits du Port A et les 3 bits de Port E en tant que I/O digitales il faut écrire la valeur h'06' dans ADCON1.

## LA MEMOIRE EEPROM

Il y a 2 zones mémoire EEPROM accessible par l'utilisateur. La zone DATA et la mémoire Flash de Programme.

- La zone DATA est organisée en mot de 8 bits. Les 256 octets se programment et se lisent comme avec un PIC 16F84. Elle est positionnée dans le chip entre les adresses h'2100' et h'21FF'.

On l'adressera (de h'00' à h'FF') par le registre spécifique: EEADR en h'10D' (page 2). Les données transiteront par le registre: EEDATA en h'10C' (page 2).

- La zone EEPROM de programme qui est organisée en mots de 14 bits, est également accessible par l'utilisateur. Pour adresser les 8 K mots il faut 13 bits d'adresse. Un registre supplémentaire est disponible: EEADRH en h'10F' (page 2). Les 8 bits LSB de l'adresse seront passés par EEADR et les 5 bits MSB par EEADRH.

Les données sur 14 bits seront échangées à travers le registre EEDATA pour les 8 bits LSB et grâce à un registre supplémentaire : EEDATH en h'10E' (page 2) pour les 6 bits MSB.

### Les registres utilisés par l'EEPROM sont :

**EEDATA** : en h'10C' (page 2).

**EEADR** : en h'10D' (page 2).

**EEDATH** : en h'10E' (page 2).

**EEADRH** : en h'10F' (page 2).

**EECON1** : en h'18C' (page 3).

**EECON2** : en h'18D' (page 3).

### EECON1 : ( h'18C' : page 3).

Bit 7

Bit 0

EEPGD				WRERR	WREN	WR	RD
-------	--	--	--	-------	------	----	----

Au reset : EECON1 = X000X000

Bit 7 : **EEPGD** = Program/Data Eeprom select bit : choix de la zone EEPROM.

1= Access à la zone EEPROM de programme.

0= Accès à la zone EEPROM de DATA.

Ce bit ne doit pas être changé pendant une opération de lecture ou d'écriture.

Bit 3 : **WRERR** =EEPROM error flag.

1= Opération d'écriture prématurément terminée ( causée par reset ).

0= Opération d'écriture achevée normalement.

Bit 2 : **WREN** =EEPROM write enable bit.

1= Autorise cycle d'écriture en EEPROM.

0= Interdit l'écriture en EEPROM.

Bit 1 : **WR** = Write Control bit.

1= Lance le cycle écriture de l'EEPROM. Ce bit est remis à "0" par hard.

0= Le cycle d'écriture est terminé.

Bit 0 : **RD** = Read Control bit.

1= Lance un cycle de lecture de l'EEPROM. Ce bit est remis à "0" par hard.

0= Pas de cycle lecture.

### EECON2 : ( h'18D' : page 3).

Ce registre n'est pas un registre de configuration physique. Il n'est utilisé que pendant les séquences d'écriture en EEPROM.

**ATTENTION:** *Pour pouvoir écrire en EEPROM programme, il faut configurer correctement certains bits du REGISTRE de CONFIG en h'2007'.*

*Le bit WRT (Flash Program Memory enable) doit être forcé à "1". Les bits CP0 et CP1 doivent être tous les deux à "1" ( Code Protect OFF), ou bien ne protéger qu'une zone particulière dans laquelle on ne voudra pas écrire.*

Les procédures de lecture et d'écriture en zone DATA sont identiques à celles décrites pour le PIC 16F84.

Pour la zone flash du programme, elles sont particulières pour les PIC 16F876 et 16F877.

### LECTURE EN ZONE EEPROM PROGRAMME :

- 1 - Mettre le MSB de l'adresse dans EEADRH.
- 2 - Mettre le LSB de l'adresse dans EEADR
- 3 - Mettre le bit EEPGD à "1" pour pointer la zone programme.
- 4 - Mettre le bit RD à "1" pour lancer le cycle de lecture.
- 5 - Attendre 2 cycles par deux NOP.
- 6 - Lire le MSB dans EEDATH et le LSB dans EEDATA.

## ECRITURE EN ZONE EEPROM PROGRAMME :

- 1 - Mettre le MSB de l'adresse dans EEADRH.
- 2 - Mettre le LSB de l'adresse dans EEADR.
- 3 - Mettre le MSB de la DATA dans EEDATH.
- 4 - Mettre le LSB de la DATA dans EEDATA.
- 5 - Mettre le bit EEPGD à "1" pour pointer la zone programme.
- 6 - Mettre le bit WREN à "1" pour autoriser l'écriture.
- 7 - Inhiber les interruptions par mise à "0" du bit GIE de INTCON.
- 8 - Lancer la séquence spécifique suivante :
  - Ecrire h'55' dans EECON2.
  - Ecrire h'AA' dans EECON2.
  - Lancer le cycle d'écriture par WR=1.
  - On attend 2 cycles par deux NOP.
  - Le PIC ignore maintenant toutes les instructions.
  - A la fin de l'écriture , le PIC exécute l'instruction suivante.
- 9 - On autorise les interruptions par GIE=1.
- 10 - On interdit l'écriture en EEPROM par WREN = 0.

## Exemple de programme :

```
.*****  
,  
;*** Ecriture en h'03FF' de la Data h'3439' (2 caractères ASCII : "4" et "9") ***  
;*****  
,
```

```
PAGE2          ;accès page 2  
MOVLW h'03'  
MOVWF EEADRH  ;MSB de l'adresse=03  
MOVLW h'FF'  
MOVWF EEADR   ;LSB de l'adresse=FF  
MOVLW h'34'  
MOVWF EEDATH  ;MSB de la DATA=34  
MOVLW h'39'  
MOVWF EEDATA  ;LSB de la DATA=39  
PAGE3          ;accès page 3  
BSF  EECON1,7 ;bit EEPGD=1 : accès mémoire programme  
BSF  EECON1,2 ;bit WREN=1 : autorise écriture en EEPROM  
BCF  INTCON,7 ;bit GIE=0 : masque les IT
```

**Séquence  
spécifique**

```
{  
MOVLW h'55'  
MOVWF EECON2  
MOVLW h'AA'  
MOVWF EECON2  
BSF  EECON1,1 ;bit W=1 : démarre cycle écriture  
NOP  ;2 cycles d'attente  
NOP  ;Le micro attend la fin de l'écriture  
BSF  INTCON,7 ;bit GIE=1 : autorise les IT  
BCF  EECON1,2 ;bit WREN=0 : interdit écriture en EEPROM
```

## MODULE TIMER 0 :

Ce module est le même que dans le PIC 16F84.

Le compteur/Timer TMR0 a les caractéristiques suivantes :

- Compteur sur 8 bits.
- Lecture / écriture de TMR0.
- Prédiviseur 8 bits programmable.
- Choix de l'horloge : interne en Timer et externe en compteur.
- Interruption au débordement ( passage de FF à 00).
- Choix du front de l'horloge en mode horloge externe.

Tous les bits de configuration de TMR0 sont dans le registre OPTION en h'81' page 1 ou en h'181' page 3.

Le registre TMR0 est à l'adresse h'01' page 0 ou en h'101' page 2.

**Mode TIMER :** Le choix de ce mode se fait par : TOCS = 0 (b5 de OPTION). TMR0 est incrémenté à chaque cycle instruction (  $F_{osc}/4$ ), en considérant le prédiviseur avec un rapport de 1.

**Mode COMPTEUR :** Ce mode est sélectionné si TOCS = 1. TMR0 est alors incrémenté à chaque front montant ou descendant sur la broche PA4/CLK (pin3). Le choix du front est fait par le bit TOSE ( b4 de OPTION). Si TOSE = 0 le compteur s'incrémente à chaque front montant. Si TOSE = 1 c'est le front descendant qui incrémente le compteur.

**LE PREDIVISEUR :** Il est partagé entre le Watchdog et TMR0. L'affectation se fait par le bit PSA (b3 de OPTION). Si PSA = 0 le prédiviseur est affecté à TMR0. Le choix du rapport de division se fait avec les bits PS2, PS1 et PS0 ( b2, b1 et b0 de OPTION). Si PSA = 1 le prédiviseur est affecté au Watchdog et le rapport de division pour TMR0 est fixé à 1.

**INTERRUPTION :** Elle est générée quand TMR0 passe de la valeur FF à 00. Le Flag TOIF (b2 de INTCON) passe à "1". On peut masquer la génération de l'interruption en mettant le bit TOIE ( b5 de INTCON) à "0". Le Flag TOIF dit être remis à zéro par soft dans le sous programme d'interruption, avant de re-autoriser cette interruption.



## MODULE TIMER 1 :

Le Timer 1 est un compteur sur 16 bits constitué de 2 registres 8 bits TMR1H en h'0F' page 0 et TMR1L en h'0E' page 0 également, que l'on peut lire ou écrire.

Le registre TMR1 ( constitué de TMR1H et TMR1L) s'incrémente de h'0000' jusqu'à h'FFFF' et repasse ensuite à h'0000' pour continuer le comptage. Quand il y a débordement, une interruption peut être générée si on la autorisée par TMR1IE =1 ( bit 0 de PIE1) et le Flag TMR1IF (bit 0 de PIR1) passe à "1".

Ce module peut fonctionner en mode TIMER, quand il s'incrémente à chaque cycle instruction (  $F_{osc}/4$  avec le pré diviseur considéré à "1") ou en mode compteur, quand il s'incrémente à chaque front montant de l'horloge externe appliquée sur le Port C<sub>0</sub>.

L'horloge externe peut également être l'oscillateur interne, dont la fréquence est fixée par un quartz externe branché entre la broche Port C<sub>0</sub> et la broche Port C<sub>1</sub>.

Le contrôle du TIMER 1 se fait par le registre T1CON en h'10' page 0.

### T1CON : ( h'10' : page 0).

Bit 7

Bit 0

		T1CKPS1	T1CKPS0	T1OSCEN	$\overline{T1SYNC}$	TMR1CS	TMR1ON
--	--	---------	---------	---------	---------------------	--------	--------

Au reset : T1CON = 00000000

Bit 7 et bit 6 : **bits non implémentés.**

Bit 5 et bit 4 : **T1CKPS** = Sélection du pré diviseur placé avant le TIMER.

T1CKPS1	T1CKPS0	PRE DIV
0	0	<b>1</b>
0	1	<b>2</b>
1	0	<b>4</b>
1	1	<b>8</b>

Bit 3 : **T1OSCEN** : Bit d'autorisation de l'oscillateur du Timer 1.

1 = oscillateur autorisé

0 = oscillateur stoppé.

Bit 2 : **T1SYNC** : Bit de contrôle de la synchronisation du CLK externe.

1 = Pas de synchronisation de l'horloge externe.

0 = Synchronisation de l'horloge externe.

Bit 1 : **TMR1CS** : Bit de sélection de la source horloge.

1 = Mode Compteur: Clk externe sur la broche PC<sub>0</sub> ou Quartz entre PC<sub>0</sub> et PC<sub>1</sub>

0 = Mode Timer: Clk interne =  $F_{osc}/4$ .

Bit 0 : **TMR1ON** : Bit d'autorisation du Timer 1.

1 = Timer 1 en service.

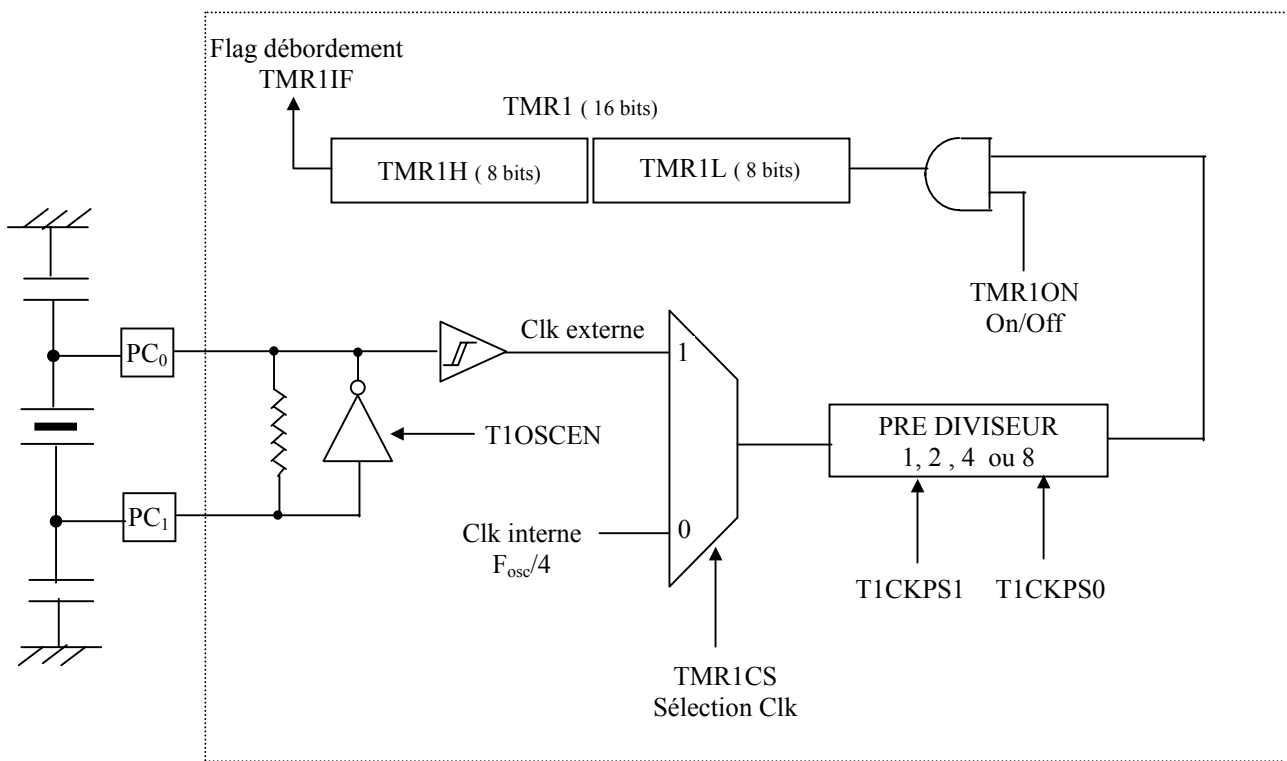
0 = Timer 1 stoppé.

### OSCILLATEUR INTERNE du TIMER 1 :

Un oscillateur à quartz a été embarqué sur le chip. Il est branché entre les broches PC<sub>0</sub> (oscillateur out) et PC<sub>1</sub> (oscillateur in). Il est mis en service par la mise à "1" du bit T1OSCEN. Cet oscillateur à faible consommation est limité à 200 KHz. Il continue à osciller en mode SLEEP du PIC.

Il est principalement destiné pour générer un événement temps réel toutes les secondes par utilisation d'un quartz 32,768 KHz.

### Synoptique du TIMER 1 :



## MODULE TIMER 2 :

Le module Timer 2 est un compteur 8 bits avec pré diviseur et post diviseur . Ce compteur TMR2 en h'11 ' page 0 est un registre en lecture ou écriture. Il possède un registre 8 bits pour la période : PR2 en h'92' page 1. Le compteur s'incrémente de h'00' jusqu'à la valeur contenue par PR2 et repasse ensuite à "0" pour continuer le comptage. Au reset PR2 est initialisé à "FF".

L'entrée du compteur est l'horloge cycle interne :  $F_{osc}/4$  qui passe à travers un pré diviseur programmable par 1, 4 ou 16.

La sortie du compteur passe dans un post diviseur programmable sur 4 bits entre 1 et 16.

Quand la sortie du compteur passe par la valeur programmée dans PR2, il y a génération d'une interruption (si elle a été autorisée par TMR2IE=1) et le flag TMR2IF est positionné à "1". Ceci bien entendu en considérant le post diviseur programmé à "1".

Le contrôle du Timer 2 se fait par le registre T2CON en h'12' page 0.

### T2CON : ( h'12' : page 0).

Bit 7

Bit 0

	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0
--	---------	---------	---------	---------	--------	---------	---------

Au reset : T2CON = 00000000

bit 7 : **bit non implémenté.**

bit 6 à bit 3 : **TOUTPS** : Programmation du Post diviseur.

0 0 0 0 = post divise par 1.

0 0 0 1 = post divise par 2.

0 0 1 0 = post divise par 3.

.....

.....

1 1 1 1 = post divise par 16.

bit 2 : **TMR2ON** : mise en service du Timer 2.

1= Timer 2 : On.

0= Timer 2 : Off.

bit 1 et bit0 : **T2CKPS** : Programmation du pré diviseur.

0 0= pré divise par 1.

0 1= pré divise par 4.

1 X= pré divise par 16.

## MODULE CCP : CAPTURE COMPARE et PWM

Il y a deux modules identiques CCP1 et CCP2 composés chacun d'un registre 16 bits. Ils peuvent opérer soit comme un registre 16 bits de capture, soit comme un registre 16 bits de comparaison, soit enfin comme un registre 8 bits pour générer du PWM.

Le module CCP1 est constitué de deux registres de 8 bits : CCPR1L en h'15' page 0 et CCPR1H en h'16' page 0. Ce module est contrôlé par le registre CCP1CON en h'17' page 0. La sortie en mode COMPARE ou mode PWM et l'entrée en mode CAPTURE se font par la broche PC<sub>2</sub>.

Le module CCP2 est constitué de deux registres de 8 bits : CCPR2L en h'1B' page 0 et CCPR2H en h'1C' page 0. Ce module est contrôlé par le registre CCP2CON en h'1D' page 0. La sortie en mode COMPARE ou mode PWM et l'entrée en mode CAPTURE se font par la broche PC<sub>1</sub>.

En mode COMPARE ou CAPTURE, les modules utilisent le TIMER 1. En mode PWM, ils utilisent le TIMER 2.

Les registres de contrôles CCP1CON et CCP2CON sont identiques. On ne décrira que CCP1CON.

### CCP1CON : ( h'17' : page 0). (et CCP2CON en h'1D' : page 0.)

Bit 7

Bit 0

		CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0
--	--	-------	-------	--------	--------	--------	--------

Au reset : CCP1CON = 00000000

bit 7 et bit 6 : bits non implémentés.

bit 5 et bit 4 : CCP1X et CCP1Y :

Bits non utilisés en modes Compare et Capture.

Ce sont les 2 bits LSB pour le Duty cycle en mode PWM. Les 8 bits MSB sont dans le registre CCPR1L en h'15' page 0.

bit 3 à bit 0 : CCP1M3 à CCP1M0 : bits de sélection du mode.

0 0 0 0 = Module CCP stoppé.

0 1 0 0 = Mode Capture à chaque front descendant.

0 1 0 1 = Mode Capture à chaque front montant.

0 1 1 0 = Mode Capture tous les 4 fronts montants.

0 1 1 1 = Mode Capture tous les 16 fronts montants.

1 0 0 0 = Mode Compare. Pin de sortie mise à "1" et Flag CCP1IF = 1 à l'égalité.

1 0 0 1 = Mode Compare. Pin de sortie mise à "0" et Flag CCP1IF = 1 à l'égalité.

1 0 1 0 = Mode Compare. Génération d'une Interrup. et Flag CCP1IF = 1 à l'égalité. } Pin de sortie

1 0 1 1 = Mode Compare. Evénement spécial généré et Flag CCP1IF = 1 à l'égalité. } non affectée

1 1 x x = Mode PWM.

### MODE COMPARE :

Les deux modules CCP étant identiques on ne décrira que le module 1.

Les 16 bits des registres CCPR1 (CCPR1H et CCPR1L) sont constamment comparés avec la valeur sur 16 bits des registres du Timer 1 ( TMR1H et TMR1L). Quand il y a égalité, la broche préalablement programmée en sortie PC2, passe soit à "1" soit à "0" suivant la configuration des 4 bits CCP1M du registre CCP1CON. Au même instant le Flag CCP1IF est mis à "1".

En mode Compare, les événements spéciaux générés quand il y a égalité sont:

- Pour CCP1: reset du Timer 1.
- Pour CCP2 : reset du Timer 1 et démarrage d'une conversion A/D.

Dans ce cas la broche de sortie n'est pas affectée, mais le Flag CCP1IF est mis à "1". Il est rappelé que ce Flag doit être remis à "0" par soft.

### MODE CAPTURE :

Quand un événement extérieur apparaît sur la broche préalablement programmée en entrée PC2, la valeur des 16 bits des registres du Timer 1 (TMR1L et TMR1H) est recopiée dans les registres CCPR1 ( CCPR1H et CCPR1L). Cet événement est programmable par les 4 bits CCP1M du registre CCP1CON. La capture peut avoir lieu à chaque front descendant, à chaque front montant, tous les 4 ou tous les 16 fronts montants.

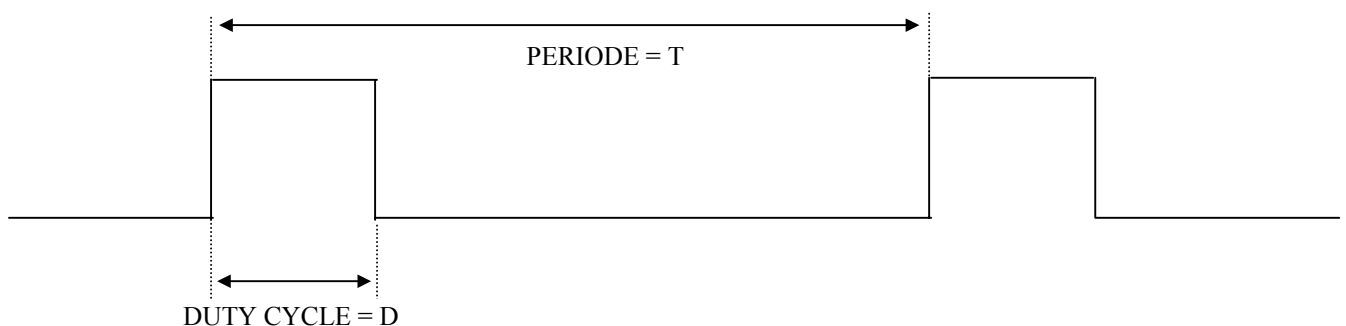
Quand la capture a eu lieu, le flag CCP1IF est mis à "1". Ce bit doit être remis à "0" par soft.

Si une nouvelle capture survient alors que la valeur dans CCPR1 n'a pas été lue, l'ancienne valeur est perdue.

Les fonctions de Capture et de Compare sur le Timer 1 par les modules CCP1 et CCP2 peuvent générer une interruption quand le Flag CCP1IF passe à "1" si le bit d'autorisation CCP1IE du registre PIE1 est mis à "1".

### MODE PWM :

Un signal PWM est caractérisé par une période, et un temps de travail ou le signal est à "1". Ce temps est appelé DUTY CYCLE.



La broche PC2 doit être configurée comme une sortie en mettant le bit 2 de TRISC à "0".

Le signal PWM est fabriqué à partir du Timer 2. Le signal interne  $F_{osc}/4$  passe à travers le pré diviseur programmable par 1, 4 ou 16 et fait compter TMR2.

Quand ce registre atteint la valeur écrite dans le registre PR2, trois événements se produisent :

- RAZ du registre TMR2.
- La broche de sortie PC2 est mise à "1", sauf si le Duty cycle vaut 0.
- Chargement de la valeur du registre CCPR1L dans le registre de Duty.

Quand le registre TMR2 atteint la valeur inscrite dans le registre interne de Duty, c'est à dire la valeur qui avait été inscrite dans CCPR1L, la broche de sortie PC2 est remise à "0".

La période est déterminée par la relation suivante :

$$T = [PR_2 + 1] \cdot 4 \cdot T_{osc} \cdot \text{valeur du pré diviseur}$$

La durée du Duty cycle est la valeur écrite dans le registre 8 bits : CCPR1L.

La durée du signal au niveau "1" est donnée par la relation :

$$D = [CCPR1L] \cdot 4 \cdot T_{osc} \cdot \text{valeur du pré diviseur}$$

On peut avoir une meilleure résolution sur le Duty cycle en utilisant les 2 bits de LSB réservés à cet effet dans le registre CCP1CON.

Il s'agit des bits 4 et 5 : CCP1X et CCP1Y. La résolution est dans ce cas divisée par 4. Si on appelle X le registre sur 10 bits, constitué de ces 2 bits en LSB et des 8 bits de CCPR1L comme MSB, la durée à "1" est donnée par la relation :

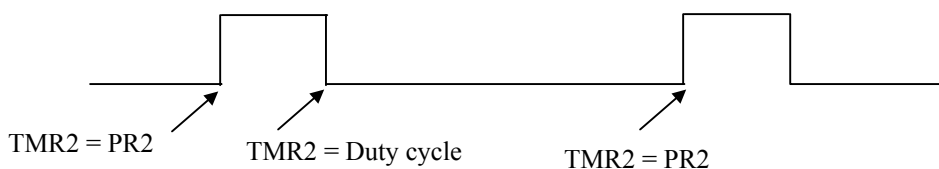
$$D = X \cdot T_{osc} \cdot \text{valeur du pré diviseur}$$

Dans les 2 cas, quand on incrémente le registre CCPR1L de une unité, le Duty cycle augmente d'une durée égale à :  $4 \cdot T_{osc} \cdot \text{valeur du pré diviseur}$ .

Quand on passe à "1" le bit 5 la durée augmente de  $2 \cdot T_{osc} \cdot \text{pré div}$ .

Quand on passe à "1" le bit 4 de CCP1CON, la durée augmente de  $T_{osc} \cdot \text{pré div}$

Si la durée du Duty cycle est supérieure à la période, le signal n'est pas remis à "0".



Marche à suivre pour faire du PWM :

- Ecrire dans PR2 la valeur permettant d'obtenir la Période désirée.
- Ecrire dans CCPR1L la valeur donnant la durée du Duty cycle.
- Configurer la broche d'utilisation en sortie ( bit = 0 dans TRISC).
- Initialiser le Timer 2 par T2CON ( prédiv, TMR2 on).
- Initialiser le CCP par CCP1CON (mode PWM, 2 bits LSB Duty cycle).

RECAPITULATIF sur les TIMER et les modules CCP

TIMER 0	TIMER 1	TIMER 2
Broche utilisée : PA4	Broches utilisées : PC0 et PC1	Broches utilisées : néant
<ul style="list-style-type: none"> <li>- 8 bits</li> <li>- Pré div de 2<sup>0</sup> à 2<sup>8</sup></li> <li>- IT et Flag au débordement</li> <li>- Clk int (F<sub>osc</sub>/4) ou ext (PA4)</li> </ul>	<ul style="list-style-type: none"> <li>- 16 bits</li> <li>- Pré div de 2<sup>0</sup> à 2<sup>3</sup></li> <li>- IT et Flag au débordement</li> <li>- Clk int (F<sub>osc</sub>/4) ou ext (PC0)</li> <li>- Oscil quartz ext PC<sub>0</sub>/PC<sub>1</sub></li> <li>- Sert en Capture et Compare</li> </ul>	<ul style="list-style-type: none"> <li>- 8 bits</li> <li>- Période programmable dans PR2</li> <li>- Pré div par 1, 4 ou 16</li> <li>- Post div de 1 à 16.</li> <li>- IT et Flag quand Timer 2= PR2</li> <li>- Clk int (F<sub>osc</sub>/4)</li> <li>- Sert en PWM</li> </ul>

CCP1	CCP2
Broche utilisée : PC2	Broche utilisée : PC1
<p><b>- Capture :</b> entrée sur broche PC2.  <i>Recopie des 16 bits du timer 1 dans les 2 registres CCPR1H et CCPR1L quand survient un événement extérieur sur la broche PC2 tel que :</i></p> <p style="text-align: center;"> </p> <p><b>- Compare :</b> sortie sur broche PC2.  <i>Comparaison des 16 bits du timer 1 et du contenu des 2 registres CCPR1H et CCPR1L et passage soit à "1" soit à "0" de la broche PC2.</i></p> <p><b>- PWM :</b> sortie sur broche PC2.  <i>Période du signal donnée par Timer 2 et valeur dans PR2. Durée du Duty cycle dans CCPR1L.</i></p>	<p><b>- Capture :</b> entrée sur broche PC1.  <i>Recopie des 16 bits du timer 1 dans les 2 registres CCPR2H et CCPR2L quand survient un événement extérieur sur la broche PC1 tel que :</i></p> <p style="text-align: center;"> </p> <p><b>- Compare :</b> sortie sur broche PC1.  <i>Comparaison des 16 bits du timer 1 et du contenu des 2 registres CCPR2H et CCPR2L et passage soit à "1" soit à "0" de la broche PC1.</i></p> <p><b>- PWM :</b> sortie sur broche PC1.  <i>Période du signal donnée par Timer 2 et valeur dans PR2. Durée du Duty cycle dans CCPR2L.</i></p>