



MICROCONTRÔLEUR

PIC 16F88

*Documentation interne ELE
D. MENESPLIER 2005*

DESCRIPTION :

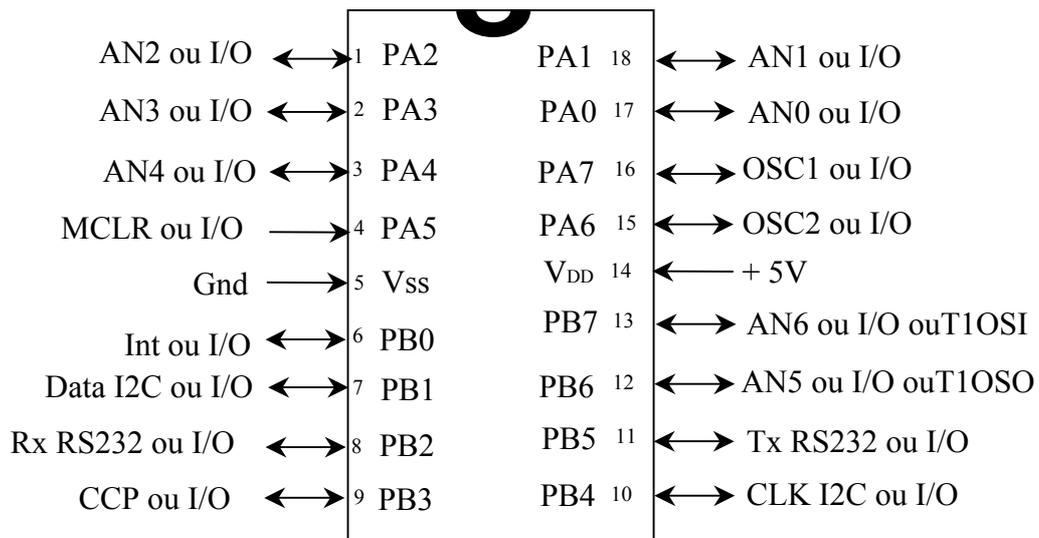
Le 16F 88 est compatible broche à broche avec le 16F84. Mais il possède plus de périphériques intégrés (USART, I2C, CAN) et une mémoire programme plus grande. De plus il peut fonctionner sur son oscillateur interne, ce qui libère 2 broches supplémentaires pour des I/O.

- 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.
- 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.
- 15 Ports Entrée-Sortie bidirectionnels pouvant produire 25 mA par sortie.
- 4 sources d'interruption :
 - Externe par la broche partagée avec le Port B : PB₀
 - Par changement d'état des bits du Port B: PB4 PB5 PB6 ou PB7
 - Par un périphérique intégré dans le chip: écriture de Data en EEPROM terminée, conversion analogique terminée, réception USART ou I2C.
 - Par débordement du Timer.
- 2 Compteurs 8 bits et 1 compteur 16 bits avec prédiviseur programmable.
- Convertisseur analogique 10 bits à 7 entrées
- UART pour transmission série synchrone ou asynchrone.
- Interface I²C pour un esclave seulement.
- 1 module pour PWM avec une résolution de 10 bits.

MEMOIRE:

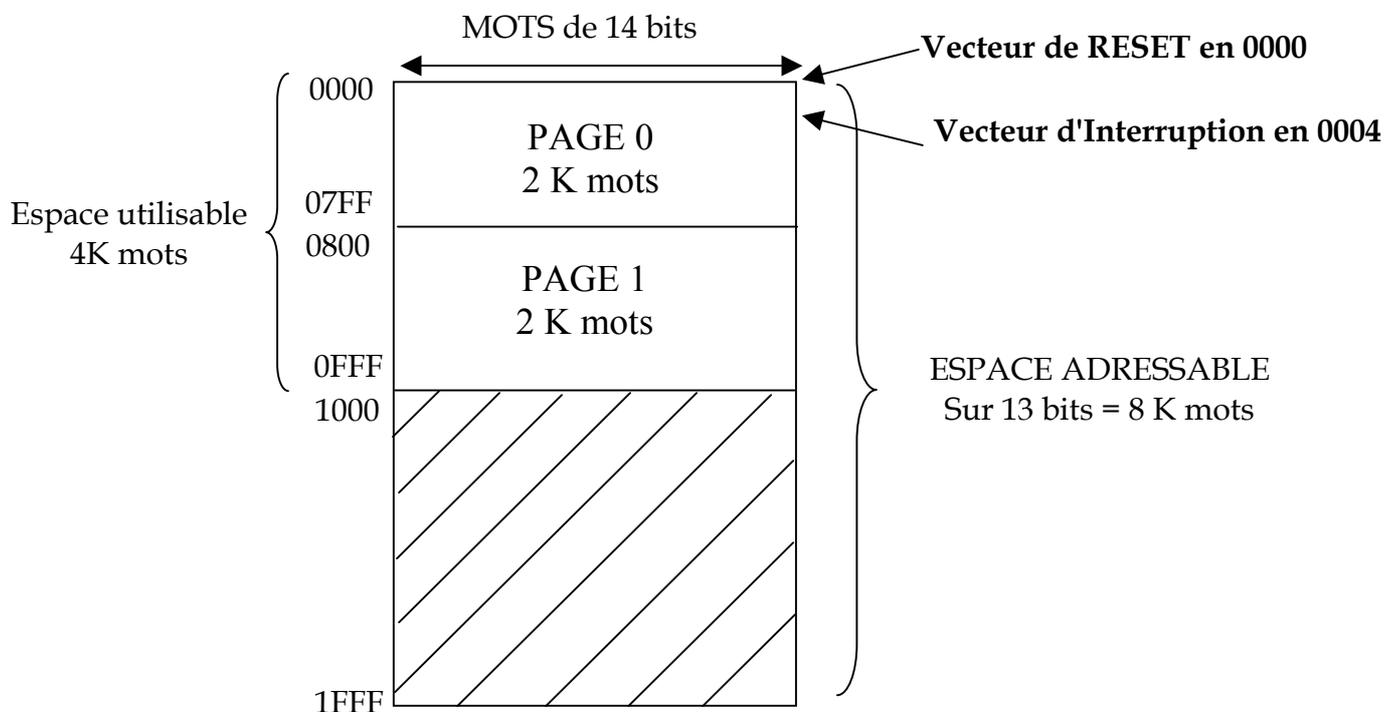
- 368 Octets de RAM
- 256 Octets d'EEPROM Data.
- 4K mots de 14 bits en EEPROM Flash pour le programme (h'000' à h'FFF').
- 1 registre de travail : W et un registre fichier : F permettant d'accéder à la RAM ou aux registres internes du PIC. Tous les deux sont des registres 8 bits.

Brochage du 16F88 :



Organisation de la mémoire programme:

Il s'agit de mot de 14 bits.



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 |

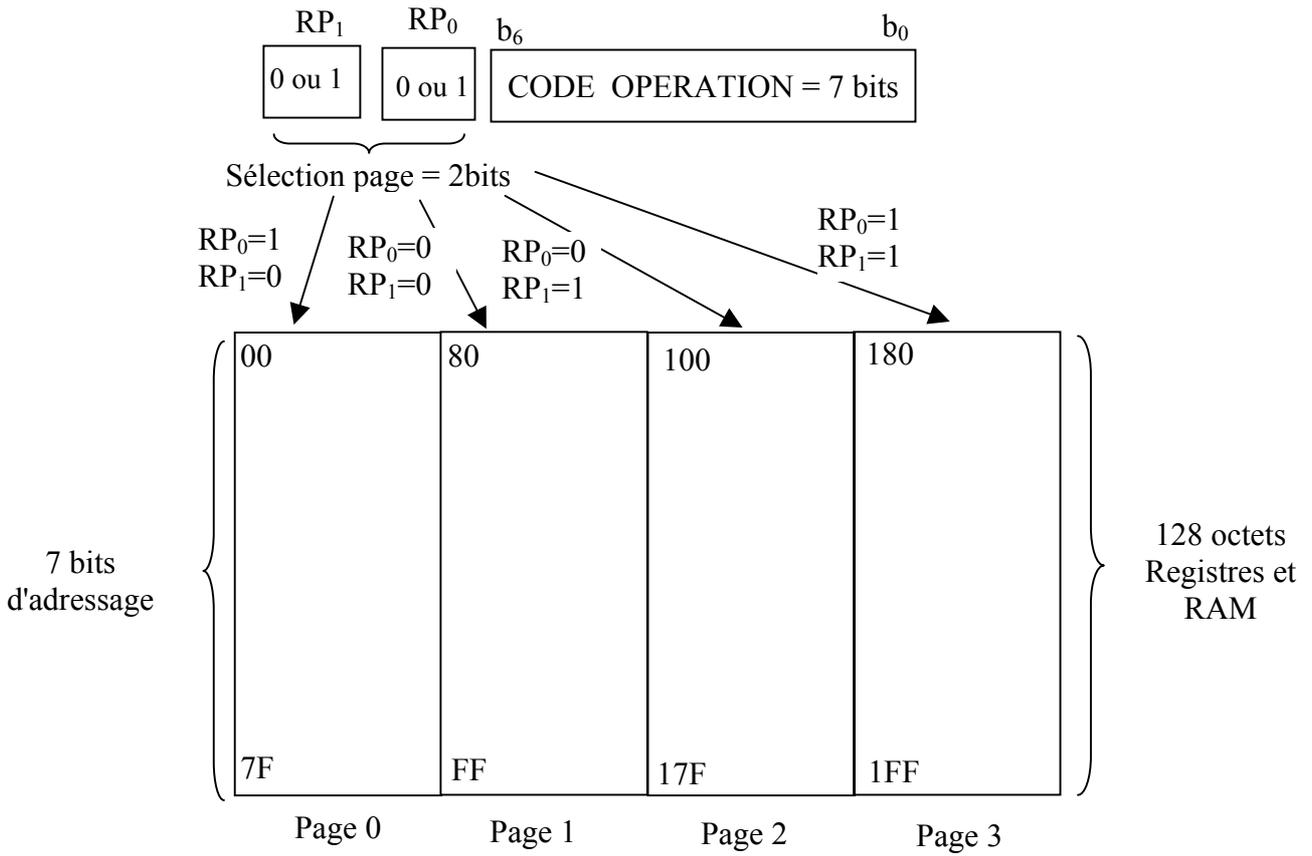
Exemple: PAGE0 BCF STATUS,5 ; RP0=0
 BCF STATUS,6 ; RP1=0

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

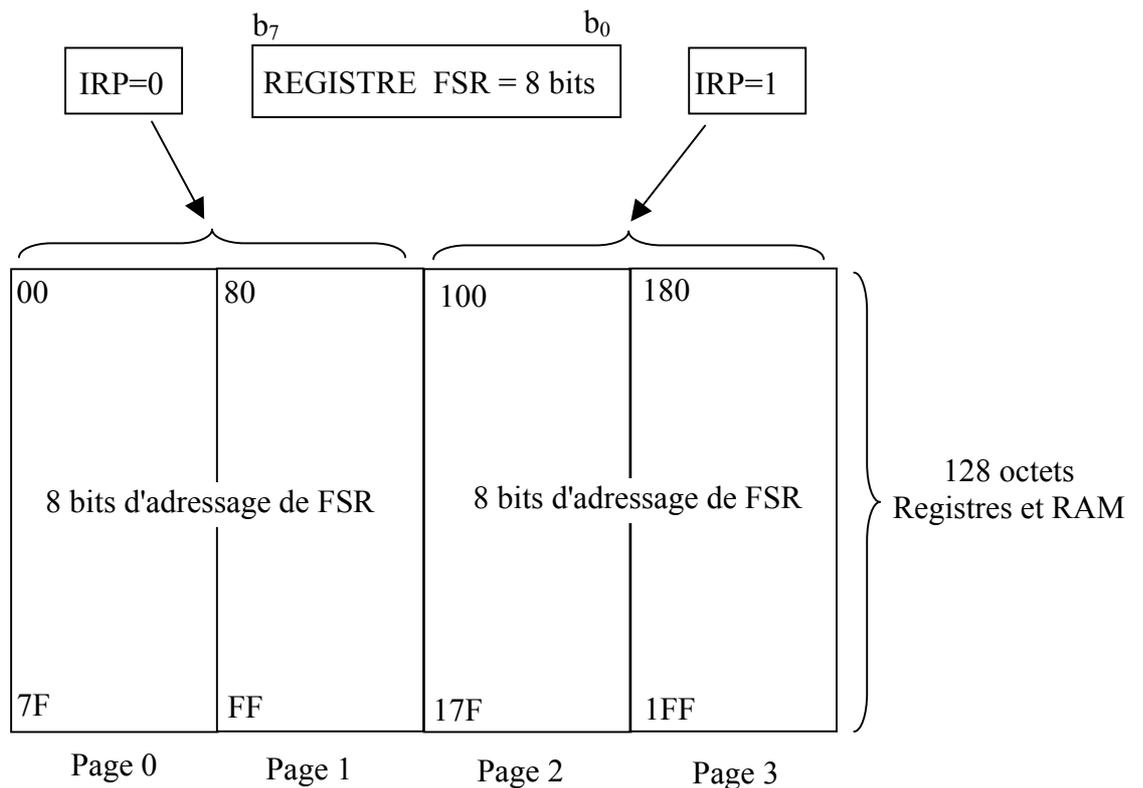
PAGE2 BCF STATUS,5 ; RP0=0
 BSF STATUS,6 ; RP1=1

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

Adressage DIRECT



Adressage INDIRECT



Bit 4: **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: **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 / Borrow.

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

0 = Pas de retenue sur le bit MSB.

Pour une Borrow, l'information est inversée (après SUBLW et SUBWF)

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

| | | | | | | | |
|-------------|--------|------|------|-----|-----|-----|-----|
| <u>RBPU</u> | INTEDG | TOCS | TOSE | PSA | PS2 | PS1 | PS0 |
|-------------|--------|------|------|-----|-----|-----|-----|

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 = Interrupt Edge select bit.

1 = Interruption si front montant sur la broche PBO/IRQ (pin 6).

0 = Interruption si front descendant sur PBO/IRQ.

Bit 5 : TOCS = Timer TMRO 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 TMRO 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 TMRO.

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 pré diviseur est affecté au Watchdog (PSA=1), TMRO est pré divisé par 1.

La période du Watchdog est de 16,38 ms au reset. Donc avec le pré diviseur maximum on aura environ un Watchdog maximum toutes les 2 secondes environ.

On verra que la période de base du Watchdog peut être programmée entre 1 ms et 2 secondes. Elle est de 16,38ms au reset.

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 | TMROIE | INTOIE | RBIE | TMROIF | INTOIF | 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 le périphériques.

Bit 5 : **TMROIE** = Timer TMRO Overflow Interrupt Enable bit.

1 = Autorise les interruptions du Timer TMRO.

0 = Désactive les interruptions du Timer TMRO.

Bit 4 : **INTOIE** = RBO/Int Interrupt Enable bit.

1 = Autorise les interruptions sur la broche : PBO/IRQ (pin6).

0 = Désactive les interruptions sur la broche : PBO/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 : **TMROIF** = Timer TMRO 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 : **INTOIF** = RBO/Int Interrupt Flag bit.

1 = Une interruption sur la broche PBO/IRQ (pin 6) est survenue.

0 = Pas d' interruption sur la broche PBO/IRQ (pin 6).

Bit 0 : **RBIF** = RB Port Change Interrupt 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.

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

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

Au reset : PIE1 = 00000000

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

1 = Autorise les interruptions du convertisseur analogique/digital .

0 = Désactive cette interruption.

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

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

0 = Désactive cette interruption.

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

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

0 = Désactive cette interruption.

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

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

0 = Désactive cette interruption.

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

1 = Autorise les interruptions du CCP1 .

0 = Désactive cette interruption.

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

1 = Autorise les interruptions du Timer 2 TMR2 .

0 = Désactive cette interruption.

Bit 0 : **TMR1IE** = TMR1 overflow Interrupt 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

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

Au reset : PIR1 = 00000000

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

1 = Une conversion A/D est terminée .

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

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

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

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

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

1 = Le Timer2 a fait une IT.

0 = Pas d'IT du TIMER 2.

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

1 = Le débordement Timer 1 a fait une IT.

0 = Pas de débordement du TIMER 2.

PIE2 REGISTER : (h'8D' : page 1).

Ce registre contient les bits individuels d'autorisation pour les Interruptions de l'EEPROM en écriture.

Bit 7

Bit 0

| | | | | | | | |
|-------|------|--|------|--|--|--|--|
| OSFIE | CMIE | | EEIE | | | | |
|-------|------|--|------|--|--|--|--|

Au reset : PIE1 = 00000000

Bit 7: **OSFIE** = Oscillator Fail Interrup Enable

1 = Autorise cette interruption si l'oscillateur stoppe

0 = Désactive cette interruption.

Bit 6: **CMIE** = Comparator Interrup bit .

1 = Autorise cette interruption du comparateur.

0 = Désactive cette interruption.

Bit 4: **EEIE** = EEPROM write Interrup Enable.

1 = Autorise cette interruption de fin d'écriture de l'EEPROM.

0 = Désactive cette interruption.

PIR2 REGISTER : (h'0D' : page 0).

Ce registre contient les flag pour les Interruptions de l'EEPROM en écriture.

Bit 7

Bit 0

| | | | | | | | |
|-------|------|--|------|--|--|--|--|
| OSFIF | CMIF | | EEIF | | | | |
|-------|------|--|------|--|--|--|--|

Au reset : PIE1 = 00000000

Bit 7: **OSFIF** = Oscillator Fail Interrup Flag

1 = oscillateur est HS

0 = oscillateur est OK

Bit 6: **CMIF** = Comparator Interrup Flag .

1 = l'entrée comparateur a changé

0 = l'entrée comparateur n'a pas changé

Bit 4: **EEIF** = EEPROM write Interrup Flag.

1 = Opération d'écriture en EEPROM terminée.

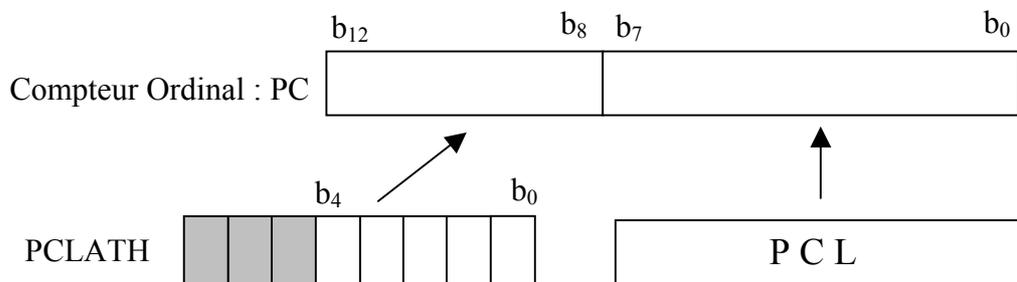
0 = Opération d'écriture en EEPROM non terminée.

NB: Ces flag doivent être remis à zéro par le programme

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.



LES PORTS ENTREE / SORTIE

PORTA (h05 en page 0) et TRISA (h85 en page 1) :

Ce port bidirectionnel est constitué de 8 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.

Les bits 0 à 4 du Port peuvent servir d'I/O digitales ou bien être les 5 premières entrées du convertisseur analogique/numérique. C'est par le registre ANSEL en h'9B' page 1 que sera fait le choix. On étudiera ce registre avec le CAN.

Le bit 5 peut être soit une entrée digitale soit la commande MCLR qui est le reset. C'est par le mot de configuration que le choix de cette broche sera fait.

Les bits 6 et 7 peuvent être soit des I/O digitales soit les broches servant à relier le quartz de l'oscillateur. Quand on veut récupérer ces broches, il faudra faire fonctionner le PIC sur son oscillateur interne. C'est par le mot de configuration, comme pour PA5 que le choix de ces broches sera fait.

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 ANSEL en h'9B'.

On retiendra seulement que pour configurer les bits en I/O digitales, il faudra mettre à "0" les bits correspondants de ANSEL. (5 sur le Port A et 2 sur le Port B)

Exemple:

PAGE1

CLRF ANSEL ; toutes entrées du CAN désactivées.

PORTB (h06 page 0) et TRISB (h86 en page 1) :

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 $+V_{DD}$ 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.

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

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

Attention: Pour pouvoir remettre ce flag à "0", il faut au préalable avoir lu ou écrit sur le Port B pour mémoriser l'état et voir ensuite si il y a un changement par rapport à cet état.

Une lecture du Port B (par MOVFW PORTB ou BTFSS PORTB,i par exemple) permet une RAZ du flag en suivant. Mais une écriture du Port B (par MOVWF PORTB ou CLRF PORTB par exemple) doit être suivi d'un NOP (1 cycle) avant de faire la RAZ du flag. Sinon la RAZ ne sera pas effective et on repart aussitôt en IT.

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.

Le bit 1 peut également servir pour le périphérique I2C embarqué. Ce sera la broche pour les data I2C.

Le bit 2 peut servir pour l'UART RS232. Ce sera la broche Rx data.

Le bit 3 est partagé avec le module CCP.

Le bit 4 sera la broche CLK I2C si le module I2C est activé.

Le bit 5 sera la broche Tx DATA si l'UART pour RS232 est activée.

Les bits 6 et 7 du port B seront soit des I/O soit les entrées 6 et 7 du CAN.

ATTENTION: *après un reset les bits 6 et 7 du Port B sont configurés en CAN. Il faut pour les configurer en I/O digitales, accéder au registre ANSEL en h'9B'et mettre à "0" les bits correspondants.*

LES MODES OSCILLATEUR

Le 16F88 peut fonctionner sous les mêmes modes classiques que les 16F84 ou 16F877, c'est à dire:

- Mode LP : Avec des quartz de fréquence basse (200 KHz max).
- Mode XT: Avec des quartz de fréquence max 4 MHz.
- Mode HS: Avec des quartz de fréquence max 20 MHz.
- Mode RC: Avec un condensateur et une résistance extérieure.

Il possède de nouveaux modes:

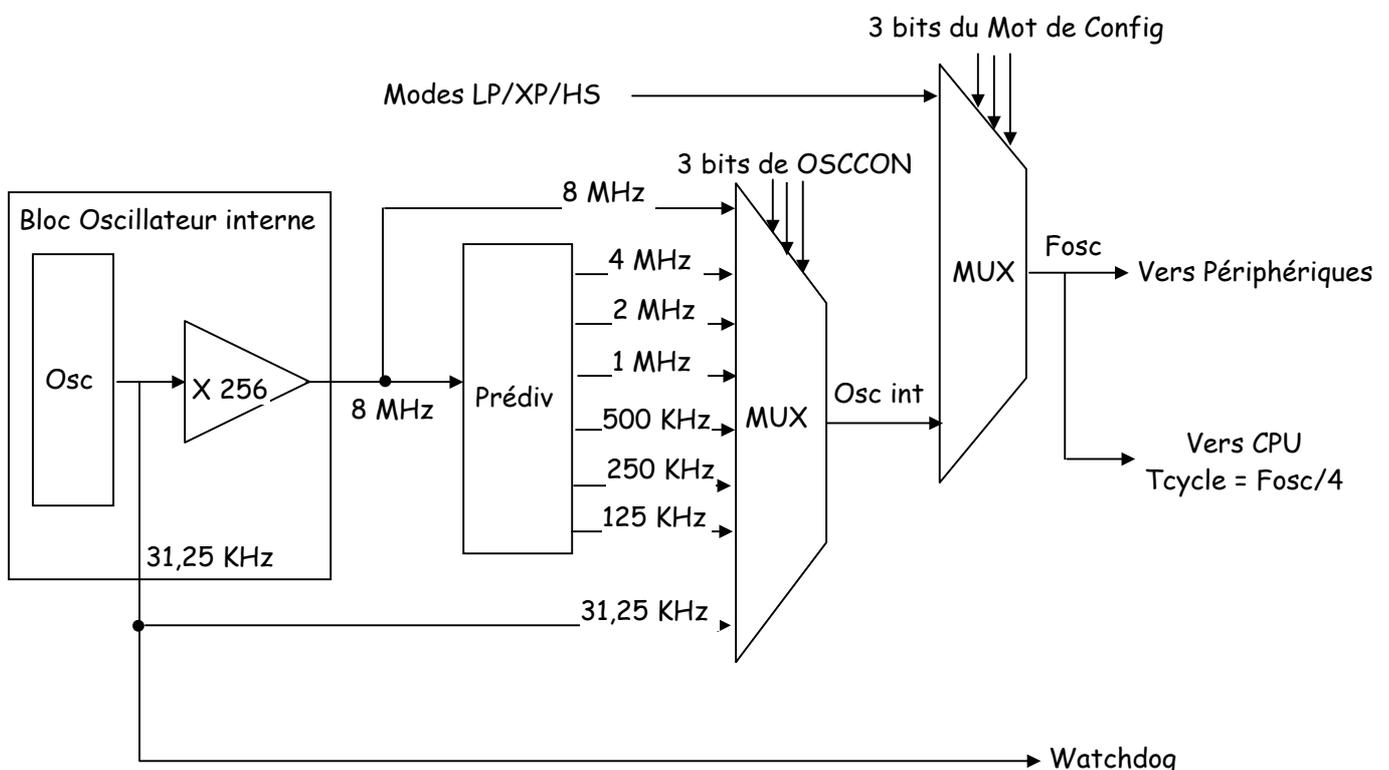
- Mode ECIO: Avec une horloge externe entrant sur PA6.
- Mode INTIO1 : Oscillateur interne. Fosc/4 sur PA6 et PA7 libre pour I/O.
- Mode INTIO2 : Oscillateur interne. PA6 et PA7 libres pour des I/O.

Le choix du mode sera fait par les 3 bits Fosc0 Fosc1 et Fosc2 du mot de configuration en h'2007', et que l'on étudiera plus tard.

LE BLOC OSCILLATEUR INTERNE:

Le 16F88 possède un bloc oscillateur qui produit un signal de 31,25 KHz et un signal de 8 MHz. Le premier sert à piloter le "watchdog". Le signal à 8 MHz qui passe à travers un pré diviseur, peut servir d'horloge système. L'oscillateur étant divisé par 4, on aura une horloge cycle maximum de 2 MHz soit un temps de cycle de 500 ns.

Le choix du prédiviseur pour le signal de 8 MHz est fait par les 3 bits IRCF0 IRCF1 et IRCF2 du registre OSCCON en h'8F' page 1.



OSCCON REGISTER : (h'8F' : page 1).

Ce registre permet de configurer l'oscillateur interne.

Bit 7

Bit 0

| | | | | | | | |
|--|-------|-------|-------|------|------|------|------|
| | IRCF2 | IRCF1 | IRCF0 | OSTS | IOFS | SCS1 | SCS0 |
|--|-------|-------|-------|------|------|------|------|

Au reset : OSCCON = 00000000

Bit 0 et 1 : **SCS0 et SCS1** = Oscillator Mode.

Ces 2 bits doivent être à "0" pour que le mode oscillateur soit défini par les 3 bits du mot de configuration.

Bit 2 et 3 : **OSTS et IOFS**

Ces 2 bits indiquent l'état de l'oscillateur

Bits 4 à 6 : **IRCF** = Internal RC Oscillator Frequency.

Ces bits permettent de choisir la valeur du pré diviseur de l'horloge du bloc oscillateur interne à 8 MHz.

| IRCF2 | IRCF1 | IRCF0 | Fréquence |
|-------|-------|-------|-----------|
| 0 | 0 | 0 | 31,25 KHz |
| 0 | 0 | 1 | 125 KHz |
| 0 | 1 | 0 | 250 KHz |
| 0 | 1 | 1 | 500 KHz |
| 1 | 0 | 0 | 1 MHz |
| 1 | 0 | 1 | 2 MHz |
| 1 | 1 | 0 | 4 MHz |
| 1 | 1 | 1 | 8 MHz |

Pour avoir 8 MHz, il faudra donc mettre la valeur: 0 1 1 1 0 0 0 0 soit h'70' dans le registre OSCCON. Et pour 4 MHz on prendra h'60'.

Exemple:

```
MOVLW    h'70'  
PAGE1  
MOVWF    OSCCON  
PAGE0
```

OSCTUNE REGISTER : (h'90' : page 1).

Ce registre permet de faire varier la fréquence de l'oscillateur interne, de part et d'autre de sa valeur calibrée en usine.

Bit 7

Bit 0

| | | | | | | | |
|--|--|------|------|------|------|------|------|
| | | TUN5 | TUN4 | TUN3 | TUN2 | TUN1 | TUN0 |
|--|--|------|------|------|------|------|------|

Au reset : OSCTUNE = 00000000

Bit 0 et 5 : TUN0 à TUN5 = Frequency tuning.

0 1 1 1 1 1 = Fréquence mAX.

0 1 1 1 1 0

.

.

.

0 0 0 0 0 1

0 0 0 0 0 0 = Fréquence centrale calibrée en usine.

1 1 1 1 1 1

.

.

.

1 0 0 0 0 0 = Fréquence minimum.

NB : La variation de fréquence est de $\pm 12,5\%$. Ce qui donne $\pm 3,9$ KHz autour des 31,25 KHz de base. Soit un pas de variation de 120 Hz environ.

USART ou SCI

C'est le même périphérique que pour le PIC 16F877.

Elle utilise le pins 2 et 5 du PORT B: **PB2 = R^x DATA** et **PB5 = T^x 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^{eme} 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.

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

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 :

| VITESSES en Bits/sec ou BAUDS | Fosc = 8 MHz | | Fosc = 20 MHz | |
|-------------------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|
| | Vitesses basses BRGH = 0 | Vitesses hautes BRGH = 1 | Vitesses basses BRGH = 0 | Vitesses hautes BRGH = 1 |
| 115200 | x | 3 | 2 | 10 |
| 57600 | x | 8 | 4 | 20 |
| 38400 | x | 12 | 7 | 31 |
| 19200 | 5 | 25 | 15 | 64 |
| 9600 | 12 | 51 | 31 | 129 |
| 4800 | 25 | 103 | 64 | x |
| 2400 | 51 | 207 | 129 | x |
| 1200 | 103 | x | x | 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. PB2 (Rx) et PB5 (Tx) 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^{eme} bit du registre de dé sérialisation vaut "1".

0 = Désactive la détection d'adresse. Tous les octets sont reçus et le 9^{eme} 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^{eme} bit de Data reçue .

Ce bit peut être le bit de la parité.

EMISSION: Pin PB5 = T^X 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^{ème} 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^{ème} 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 PB2 = R^X 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 a 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 srialisation 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^{eme} 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^{eme} 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^{eme} 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^{eme} bit n'est pas à "1". Si pas OK rester avec ADDEN=1 pour ne recevoir que les octets filtrés d'adresse.

Un MAITRE et Multi ESCLAVES en RS232 :

L'adressage se fait mode 9 bits MARK, le 9eme bit étant à "1".

Le maître pour adresser un esclave doit passer en mode 9 bits, avec le 9eme bit à "1". Les esclaves sont tous à l'écoute de la ligne réception, en mode 9 bit avec RX9 = 1 et ADDEN = 1.

Quand un mot de 9 bit avec 9eme bit à "1" est reçu, le PIC le signale par le Flag RCREG. On doit alors lire la data sur 8 bits et comparer avec son adresse. Si on ne reconnaît pas son adresse, l'esclave doit rester en mode 9 bit avec 9eme bit à "1".

Si l'esclave s'est reconnu, il doit répondre "ACK" mais d'abord passer en mode 8 bits par ADDEN =0.

Le dialogue avec le maître se fait alors en mode 8 bits avec le seul esclave sélectionné.

Le maître peut dialoguer soit en 8 bits soit en 9 bits mais avec le 9eme bit à "0". En effet un esclave qui a fait ADDEN=0 ne tiens plus compte du 9eme bit.

Exemple d'utilisation de l'USART : Réception d'un octet et renvoi en écho.

```

ORG          h'00'
             GOTO          PROG
#include "c:\mplab\include\menes88C.inc" ; osc interne à 8MHz et récup des 2 pins osc sur Port A.
PROG
PAGE1
CLRF        ANSEL          ; Port A en I/O digital
CLRF        TRISA          ; Port A en sortie, sauf PA5
BCF         TRISB,5
MOVLW      h'70'
MOVWF      OSCCON          ; prog osc int à 8 MHz
MOVLW      d'51'
MOVWF      SPBRG           ; baud rate = 9600 bits /sec
BSF        BRGH            ; bit BRGH mis à "1" vitesse haute
BCF        SYNC           ; bit SYNC=0 mode asynchrone
PAGE0
BSF        SPEN            ; bit SPEN à "1" UART en service
BSF        CREN            ; bit CREN à "1" réception autorisée
CLRF        PORTA
BOU
BTFSS      RCIF            ; teste si RCIF=1 donc si réception
GOTO       BOU             ; non
MOVWF      RCREG           ; oui lecture registre réception
CALL      EMISS            ; renvoi en écho
MOVWF      PORTA          ; affichage sur Port A
GOTO       BOU

;*****
;**          SP Emission USART          **
;**  L'octet dans W est émis par l'USART  **
;*****
EMISS      PAGE1
BSF        TXEN            ; TXEN=1 mise en service du TX
ATT
PAGE0
BTFSS      TXIF            ; teste TXIF: registre emission vide ?
GOTO       ATT             ; TXIF = 0 donc Registre plein
MOVWF      TXREG           ; octet à emettre mis dans USART
PAGE1
PAFINI
BTFSS      TRMT            ; teste TRMT: si emission terminée ?
GOTO       PAFINI          ; non car TRMT = 0
BCF        TXEN            ;oui donc arrêt du TX:TXEN=0
PAGE0
RETURN
END

```

CONVERTISSEUR A/D

Il est constitué d'un module convertisseur à 7 entrées.

Les 5 premières entrées sont sur le Port A en PA₀, PA₁, PA₂, PA₃ et PA₄, et les 2 dernières sur le Port B en PB₆ et PB₇.

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_{DD} ou la broche PA₃ pour V_{REF}⁺ et V_{SS} ou la broche PA₂ pour V_{REF}⁻.

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/Done | | 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 :

- Si le bit ADSC2 = 0 (b6 du registre ADCON1) = diviseur par 2 inactif.
 - 00= $F_{osc}/2$.
 - 01= $F_{osc}/8$.
 - 10= $F_{osc}/32$.
 - 11= Oscillateur RC interne.
- Si le bit ADSC2 = 1 (b6 du registre ADCON1) = diviseur par 2 activé.
 - 00= $F_{osc}/4$.
 - 01= $F_{osc}/16$.
 - 10= $F_{osc}/64$.
 - 11= Oscillateur RC interne.

Le temps de conversion d'un bit est T_{AD}.

Pour une conversion totale des 10 bits il faut : 12.T_{AD}.

Pour que la conversion soit correcte il faut que T_{AD} soit au minimum de **1,6µs**.

Avec l'oscillateur interne RC on a : T_{AD} = 4 µs typique (entre 2 et 6 µs).

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 $T_{AD} < 1,6\mu s$ |
|--------|----------------------------------|--------------|--------------|----------------------------------------|
| 8 MHz | $F_{osc}/2 = 4 \text{ MHz}$ | $0,25 \mu s$ | $3 \mu s$ | Ne convient pas |
| | $F_{osc}/4 = 2 \text{ MHz}$ | $0,50 \mu s$ | $6 \mu s$ | Ne convient pas |
| | $F_{osc}/8 = 1 \text{ MHz}$ | $1 \mu s$ | $12 \mu s$ | Ne convient pas |
| | $F_{osc}/16 = 500 \text{ KHz}$ | $2 \mu s$ | $24 \mu s$ | OK |
| | $F_{osc}/32 = 250 \text{ KHz}$ | $4 \mu s$ | $48 \mu s$ | OK |
| | $F_{osc}/64 = 125 \text{ KHz}$ | $8 \mu s$ | $96 \mu s$ | OK |
| 20 MHz | $F_{osc}/2 = 10 \text{ MHz}$ | $0,1 \mu s$ | $1,2 \mu s$ | Ne convient pas |
| | $F_{osc}/4 = 5 \text{ MHz}$ | $0,2 \mu s$ | $2,4 \mu s$ | Ne convient pas |
| | $F_{osc}/8 = 2,5 \text{ MHz}$ | $0,4 \mu s$ | $4,8 \mu s$ | Ne convient pas |
| | $F_{osc}/16 = 1,25 \text{ MHz}$ | $0,8 \mu s$ | $9,6 \mu s$ | Ne convient pas |
| | $F_{osc}/32 = 625 \text{ KHz}$ | $1,6 \mu s$ | $19,2 \mu s$ | OK |
| | $F_{osc}/64 = 312,5 \text{ KHz}$ | $3,2 \mu s$ | $38,4 \mu s$ | OK |

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_0 |
| 1 | 0 | 0 | 1 | PA_1 |
| 2 | 0 | 1 | 0 | PA_2 |
| 3 | 0 | 1 | 1 | PA_3 |
| 4 | 1 | 0 | 0 | PA_4 |
| 5 | 1 | 0 | 1 | PB_6 |
| 6 | 1 | 1 | 0 | PB_7 |

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.

ATTENTION: Le bit **GO/DONE** ne doit pas être mis à "1" avec la même instruction qui met **ADON** à "1".

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

Bit 7

Bit 0

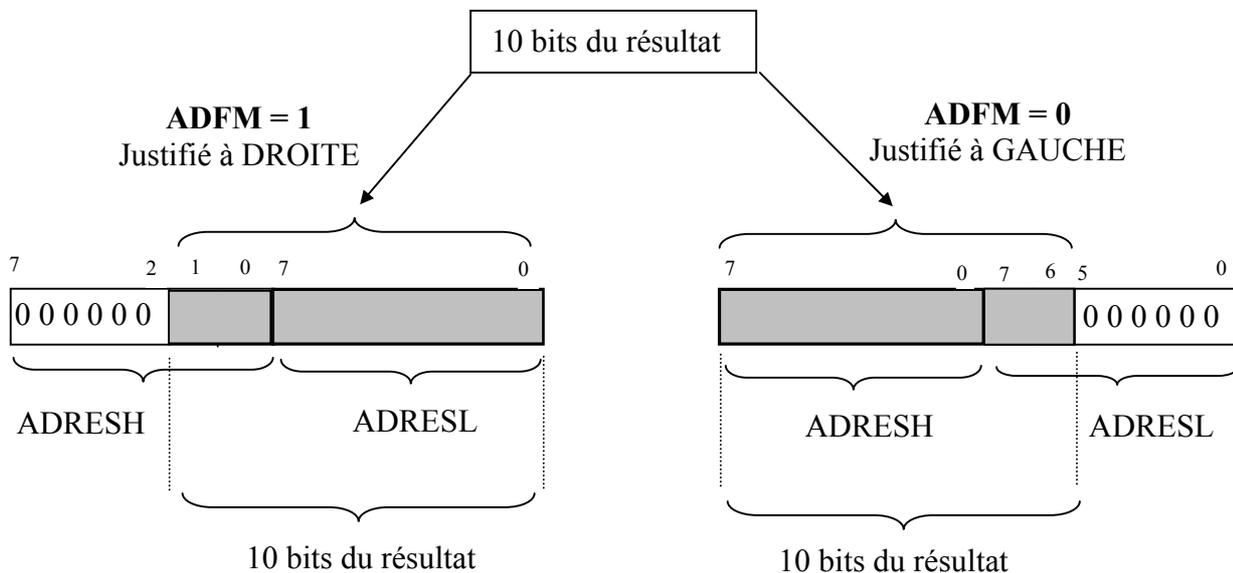


Au reset : ADCON1 = 00000000

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: **ADCS2** = A/D Clock Divide by 2

1 = CLK divisée par 2

0 = diviseur par 2 inactif.

Bit 4 et 5: **VCFG0 et VCFG1** = Voltage Reference Configuration

| VCFG1 | VCFG0 | V Ref+ | V Ref- |
|-------|-------|--------|--------|
| 0 | 0 | +Vcc | masse |
| 0 | 1 | + Vcc | PA2 |
| 1 | 0 | PA3 | masse |
| 1 | 1 | PA3 | PA2 |

ANSEL : (h'9B' : page 1).

Ce registre permet la sélection des entrées analogiques.

Bit 7

Bit 0

| | | | | | | | |
|--|------|------|------|------|------|------|------|
| | ANS6 | ANS5 | ANS4 | ANS3 | ANS2 | ANS1 | ANS0 |
|--|------|------|------|------|------|------|------|

Au reset : ADCON1 = 01111111

Bit 7 : Bit non implanté.

Bit 0 à 6 : ANS0 à ANS6 = analog input select.

1 = Entrée configurée en analogique.

0 = Entrée configurée en I/O digitale.

ATTENTION :

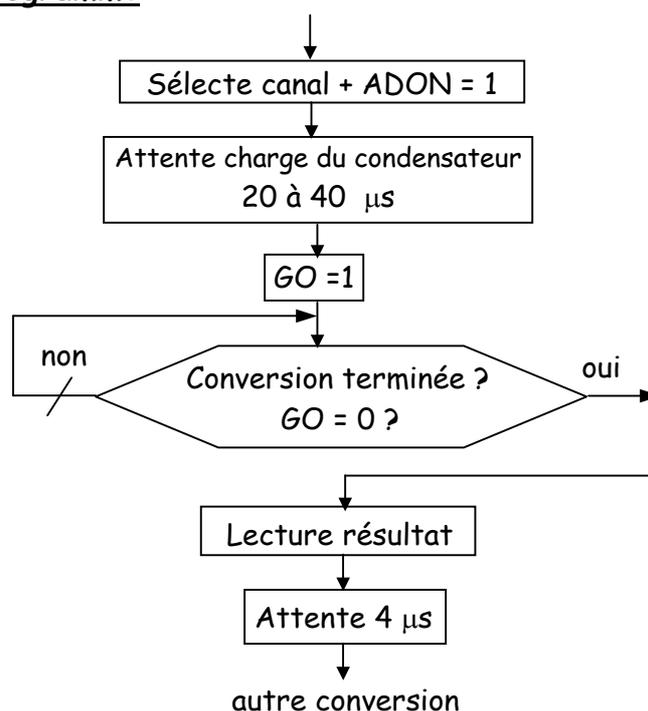
- 1 - Au reset le registre ANSEL est initialisé à h'7F'. Cela signifie que les 5 bits du Port A et les 2 bits du Port B sont configurés en entrées analogiques.

Pour récupérer ces broches en tant que I/O digitales il faut écrire la valeur h'00' dans ANSEL

- 2 - Il faut attendre que le condensateur du CAN soit chargé avant de lancer la conversion. Ce temps d'acquisition est de l'ordre de 20 μ s à 40 μ s.

Il faut également attendre $2 \cdot T_{AD}$ soit environ 4 μ s entre deux conversions.

Exemple de programme:



MODULE COMPAREUR

Le module compareur est constitué de 2 comparateurs analogique qui utilisent les broches PA0 PA1 PA2 et PA3 comme entrées. Le résultat positionne les bits C1OUT et C2OUT ou bien les broches PA3 ou PA4.

CMCON REGISTER: adresse h'9C' en PAGE 1

bit 7

bit 0

| | | | | | | | |
|-------|-------|-------|-------|-----|-----|-----|-----|
| C2OUT | C1OUT | C2INV | C1INV | CIS | CM2 | CM1 | CM0 |
|-------|-------|-------|-------|-----|-----|-----|-----|

Au reset : CMCON = 00000111

bit 7: C2OUT

Sortie du comparateur 2. Ce bit est fonction du bit C2INV.

Pour C2INV=0 : si $V_{in+} > V_{in-}$ alors C2OUT = 1
Si $V_{in+} < V_{in-}$ alors C2OUT = 0

Pour C2INV=1 : si $V_{in+} > V_{in-}$ alors C2OUT = 0
Si $V_{in+} < V_{in-}$ alors C2OUT = 1

bit 6: C1OUT

Sortie du comparateur 1. Ce bit est fonction du bit C1INV.

Pour C1INV=0 : si $V_{in+} > V_{in-}$ alors C1OUT = 1
Si $V_{in+} < V_{in-}$ alors C1OUT = 0

Pour C1INV=1 : si $V_{in+} > V_{in-}$ alors C1OUT = 0
Si $V_{in+} < V_{in-}$ alors C1OUT = 1

bit 5: C2INV

Bit d'inversion de la sortie de comparaison n°2.

C2INV = 0 : sortie non inversée.

C2INV = 1 : sortie inversée.

bit 4: C1INV

Bit d'inversion de la sortie de comparaison n°1.

C1INV = 0 : sortie non inversée.

C1INV = 1 : sortie inversée.

bit 3: CIS

Bit de sélection de l'entrée, qui sera connectée à l'entrée V_{in-} du comparateur, quand il y a comparaison à la tension de référence interne, qui est connectée à l'entrée V_{in+} du comparateur. Soit PA0 ou PA3 pour le comparateur 1 et PA1 ou PA2 pour le comparateur 2.

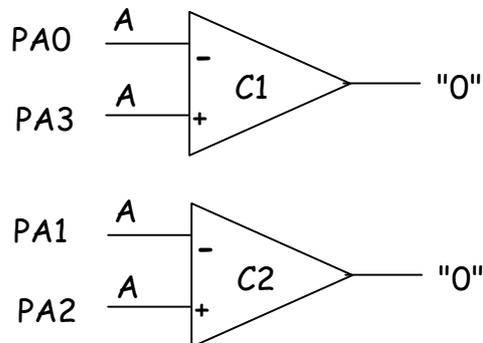
bit 0 à 2: CM2 CM1 et CM0

Choix du mode de comparaison.

Configuration des modes de comparaison :

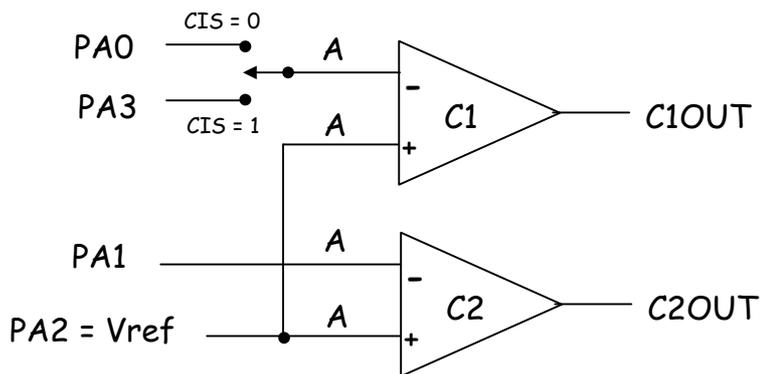
Il y en a 8 qui sont déterminés par les 3 bits CM2 CM1 et CM0

MODE 0: CM2=0 CM1=0 CM0=0 : RESET des Comparateurs



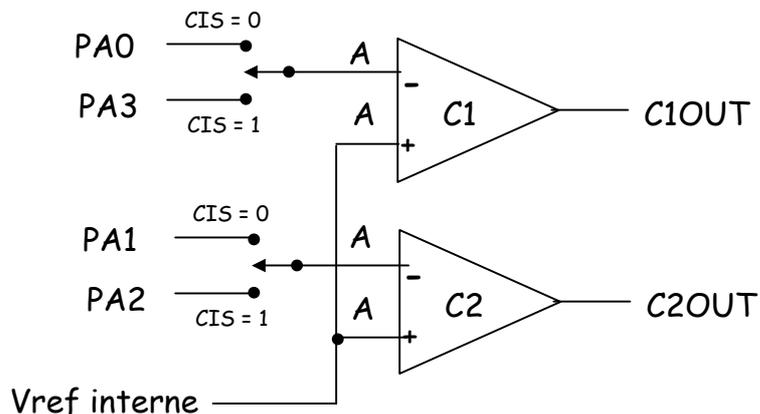
La sortie des comparateurs est toujours à "0".

MODE 1: CM2=0 CM1=0 CM0=1 : 2 Comparateurs et 3 entrées multiplexées.



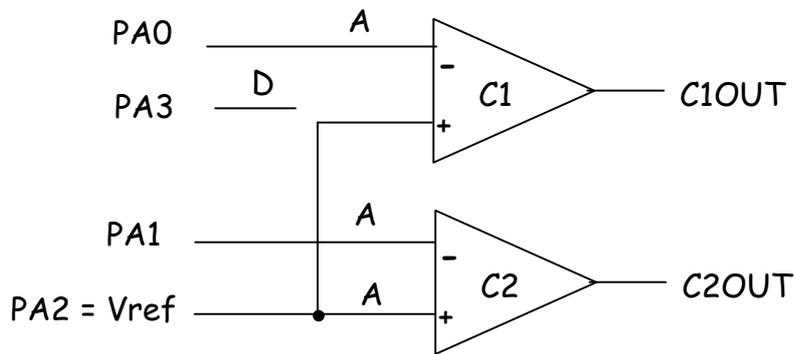
La tension de référence sur l'entrée "+" des 2 comparateurs est la tension extérieure appliquée sur la broche PA2. Le comparateur 1 reçoit sur son entrée "-", soit la tension sur PA0 soit celle sur PA3. Le comparateur 2 reçoit lui la tension sur PA1.

MODE 2: CM2=0 CM1=1 CM0=0 : 2 Comparateurs et 4 entrées multiplexées.

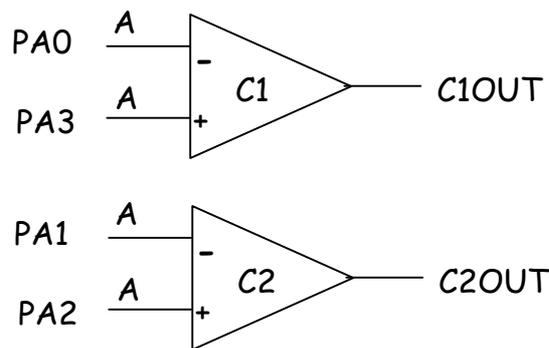


MODE 3: $CM2=0$ $CM1=1$ $CM0=1$: 2 Comparateurs avec référence commune.

La broche PA3 peut être utilisée comme une I/O digitale.

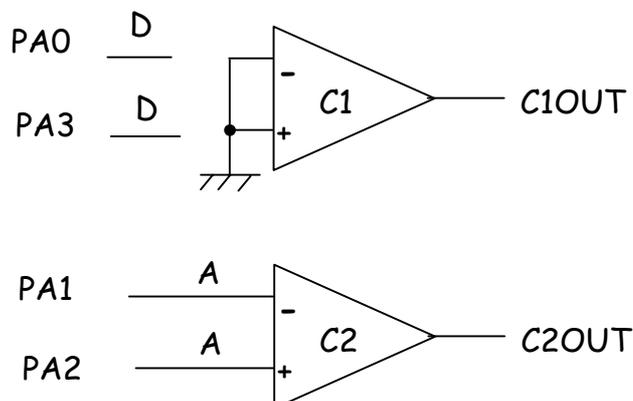


MODE 4: $CM2=1$ $CM1=0$ $CM0=0$: 2 Comparateurs indépendants.



MODE 5: $CM2=1$ $CM1=0$ $CM0=1$: 1 seul Comparateur indépendant.

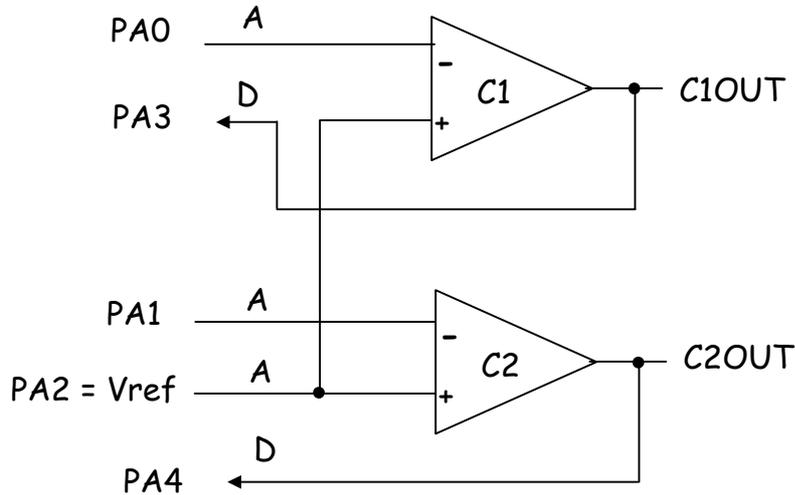
Le comparateur 1 n'est pas utilisé et on peut utiliser les broches PA0 et PA3 comme des I/O digitales.



MODE 6: CM2=1 CM1=1 CM0=0 : 2 Comparateurs avec référence commune et sorties sur les broches PA3 et PA4.

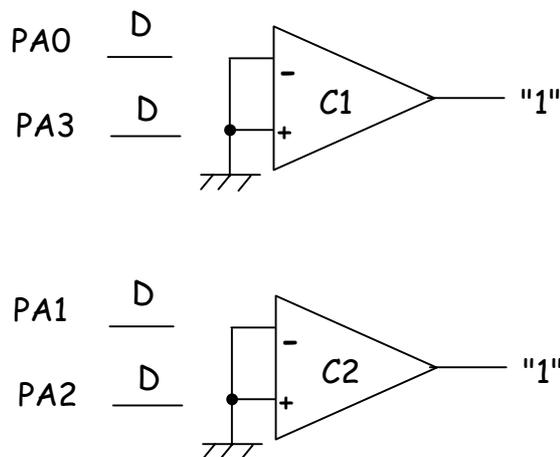
Les résultats des comparaisons sont disponibles sur les broches PA3 ou PA4 et par lecture des bits C1OUT ou C2OUT.

La tension de référence commune aux 2 comparateurs est appliquée sur la broche PA2.



MODE 7: CM2=1 CM1=1 CM0=1 : 2 Comparateurs OFF.

C'est le mode de fonctionnement au reset. Les entrées PA0 PA1 PA2 et PA3 peuvent être utilisées en I/O digitales.



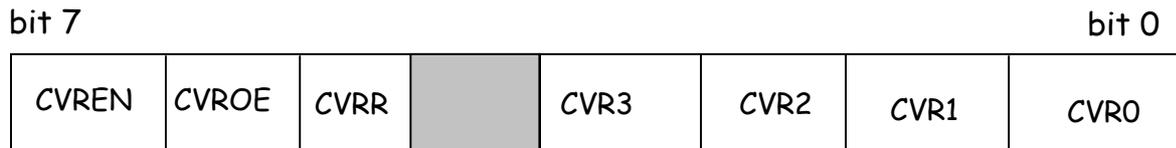
La tension de référence interne:

La tension de référence interne est programmable par le registre CVRCON. Il y a deux gammes de tension possible, en fonction du bit 5 (CVRR) de CVRCON.

Dans chacune des gammes on peut choisir parmi 16 valeurs de tension différentes par programmation de 4 bits.

Cette tension de référence peut être récupérée sur la broche PA2. On obtient ainsi une sortie de convertisseur digital analogique.

CVRCON REGISTER: *adresse h'9D' en PAGE 1*



Au reset : VRCON = 0000 0000

bit 7: CVREN = Comparator Voltage Reference Enable.

Mise en service de la tension de référence interne si ce bit est à "1".

Quand ce bit est à "0", la référence interne n'est pas en service.

bit 6: CVROE = Comparator Vref Output Enable

La tension de sortie Vref est connectée à la broche PA2 si ce bit est à "1".

Quand ce bit est à "0" elle n'est pas connectée à PA2.

bit 5: CVRR = Comparator Vref Range Selection.

Sélection de la gamme. Si CVRR est à "1" c'est la gamme basse et si CVRR est à "0" c'est la gamme haute.

bit 0 à 3: CVRO CVR1 CVR2 CVR3

Programmation de la valeur de la tension de référence interne dans la gamme haute ou basse sélectionnée par CVRR.

En gamme basse (bit CVRR=1) :

$$V_{ref} = 5V \times (VR0 \text{ à } VR3)/24$$

Ce qui donne 16 valeurs de pas 0,208 V entre 0 V et 3,125 V.

En gamme haute (bit CVRR=0) :

$$V_{ref} = 5V/4 + 5V \times (VR0 \text{ à } VR3)/32$$

Ce qui donne 16 valeurs de pas 0,156 V entre 1,25 V et 3,59 V.

LE TIMER 0

Le compteur/Timer TMRO a les caractéristiques suivantes :

- Compteur sur 8 bits.
- Lecture / écriture de TMRO.
- 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 sont dans le registre OPTION en h'81' en page 1.

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

Mode TIMER : Le choix de ce mode se fait par : TOCS = 0 (b5 de OPTION).

TMRO 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. TMRO 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 TMRO.

L'affectation se fait par le bit PSA (b3 de OPTION).

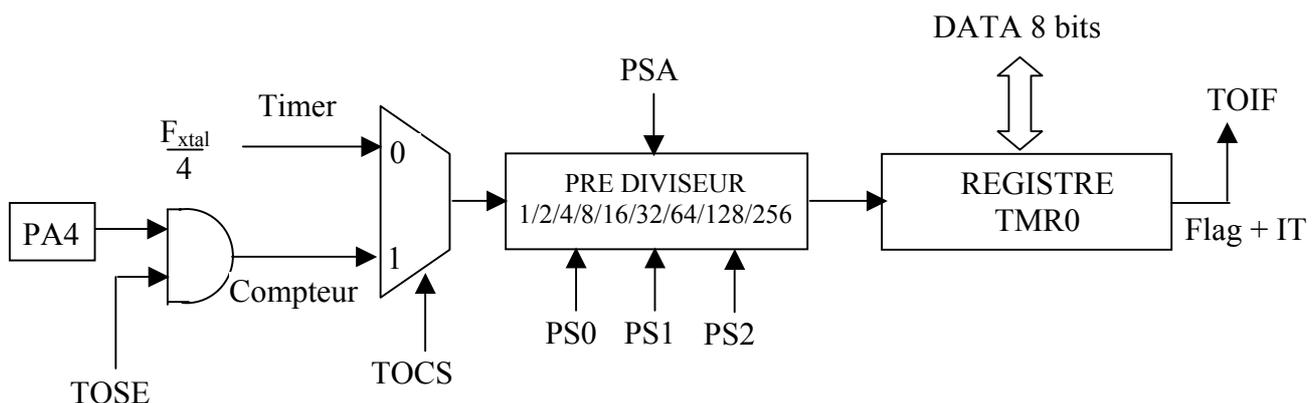
Si PSA = 0 le prédiviseur est affecté à TMRO. Le choix du rapport de division se fait avec les bits PS2, PS1 et PS0 (b2, b1 et b0 de OPTION).

Si PSA = 1 le prédiviseur est affecté au Watchdog et le rapport de division pour TMRO est fixé à 1.

INTERRUPTION : Elle est générée quand TMRO 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 doit être remis à zéro par soft dans le sous programme d'interruption, avant de re-autoriser cette interruption.



LE 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 C₀.

L'horloge externe peut également être l'oscillateur interne, dont la fréquence est fixée par un quartz externe branché entre la broche Port C₀ et la broche Port C₁.

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 | <u>T1SYNC</u> | 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 | 1 |
| 0 | 1 | 2 |
| 1 | 0 | 4 |
| 1 | 1 | 8 |

Bit 3 : **T1OSCEN** : Bit d'autorisation de l'oscillateur externe 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 PC₀ ou Quartz entre PC₀ et PC₁

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

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

1 = Timer 1 en service.

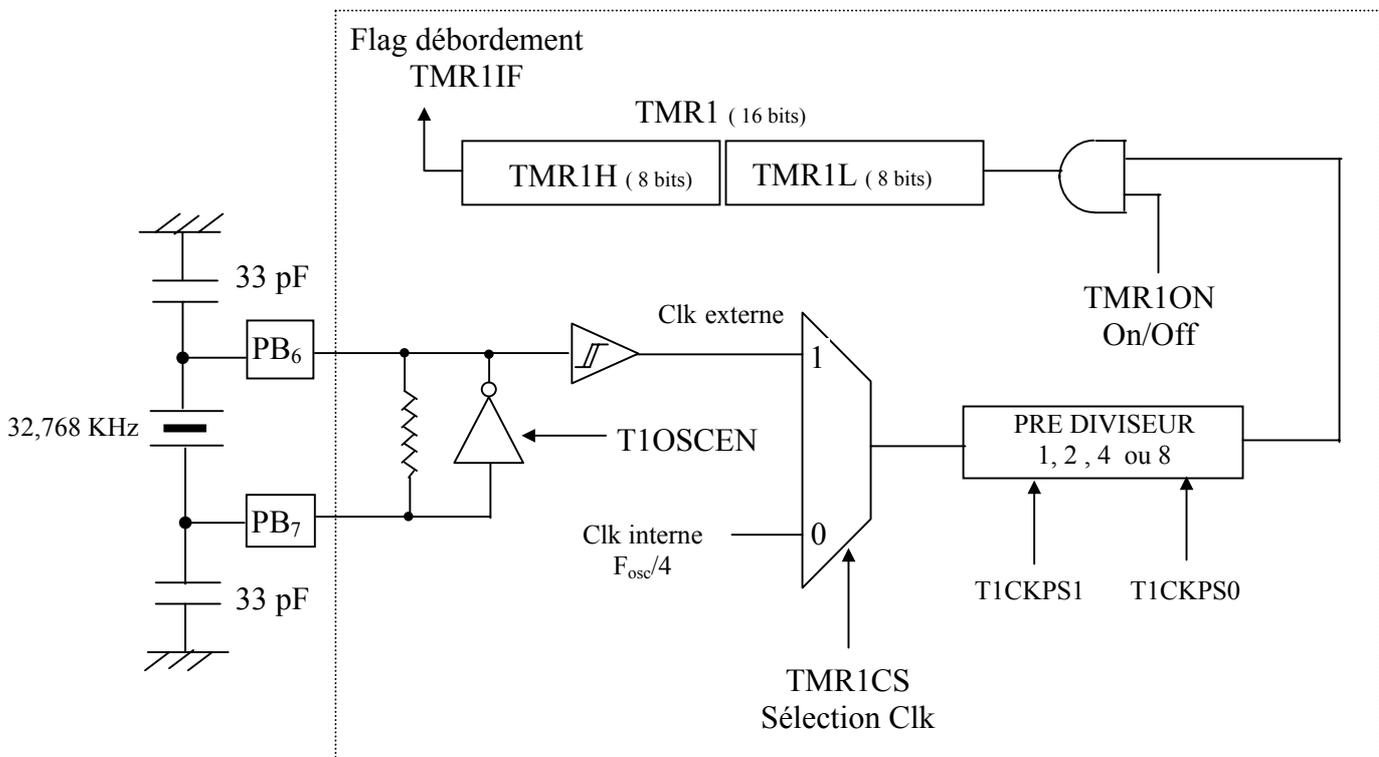
0 = Timer 1 stoppé.

OSCILLATEUR INTERNE du TIMER 1 :

Un oscillateur pour quartz a été embarqué sur le chip. Le quartz est à brancher entre les broches PB₆ (oscillateur out) et PB₇ (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 :



LE 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 bit 0: **T2CKPS** : Programmation du pré diviseur.

0 0= pré divise par 1.

0 1= pré divise par 4.

1 X= pré divise par 16.

MOTS de CONFIGURATION

Le 16F88 possède 2 mots de configuration de 14 bits:
CONFIG1 en h'2007' et CONFIG2 en h'2008'

REGISTRE CONFIG1 en h'2007' :

bit 13

bit 0

| | | | | | | | | | | | | | |
|----|-------|-------|------|------|-----|-----|-------|-------|-------|--------|-------|-------|-------|
| CP | CCPMX | DEBUG | WRT1 | WRT0 | CPD | LVP | BOREN | MCLRE | FOSC2 | PWRTEN | WDTEN | FOSC1 | FOSC0 |
|----|-------|-------|------|------|-----|-----|-------|-------|-------|--------|-------|-------|-------|

bit 13: CP = Program Memory Code Protection.

1 = Pas de protection de la mémoire programme.

0 = Toute la zone programme de h'0000' à h'0FFF' est protégée.

bit 12: CCPMX = CCP1 pin selection.

1 = La fonction CCP1 est sur la broche PB0.

0 = La fonction CCP1 est sur la broche PB3.

bit 11: DEBUG = In Circuit Debugger Mode.

1 = La fonction Debugger est inactive. Les broches PB6 et PB7 sont des I/O.

0 = La fonction Debugger est active et utilise les broches PB6 et PB7.

bits 10 et 9: WRT1 et WRT0 = Flash Program Memory Write Enable.

1 1 = Pas de protection pour l'écriture en mémoire programme.

1 0 = Seuls les 256 premiers octets de h'0000' à h'00FF' sont protégés en écriture.

0 1 = Seuls les 2K premiers octets de h'0000' à h'07FF' sont protégés en écriture.

0 0 = Toute la zone de 4 K octets de h'0000' à h'0FFF' est protégée en écriture.

bit 8: CPD = Data EEPROM Memory Protection

1 = Pas de protection de la zone EEPROM Data.

0 = La zone EEPROM Data est protégée.

bit 7: LVP = Low Voltage Programming Enable.

1 = La broche PB3 est utilisée pour la programmation. Le LVP est activé.

0 = LVP inactif. Il faut utiliser la broche MCLR pour programmer. PB3 est une I/O.

bit 6: BOREN = Brown Out Reset Enable.

1 = BOR en service.

0 = BOR non activé.

bit 5: MCLRE = PA5/MCLR Pin Function.

1 = La broche PA5 est le MCLR.

0 = La broche PA5 est une I/O utilisable en entrée seulement. Le MCLR est interne.

16F88 en remplacement d'un 16F84 :

- Changer le début de la RAM dans la directive CBLOCK. Pour un 16F88 mettre h'20'.
- Passer les broches des PORTA et B qui servent au CAN en I/O digitales par RAZ des bits correspondants dans le registre ANSEL.

Exemple:

```
PAGE1
CLRF    ANSEL           ; toutes les broches du CAN en I/O digitales.
```

- Choisir le type d'oscillateur dans le mot de configuration, par le fichier include: MENES88A.inc pour un quartz HS ou MENES88C.inc pour l'oscillateur interne.
- Dans le cas du fonctionnement avec le bloc oscillateur interne, on devra configurer la fréquence par programmation des 3 bits IRCF du registre OSCCON.

Exemple: pour avoir 8 MHz soit une fréquence cycle de 2 MHz et donc un temps de cycle de 500 ns, il faudra mettre h'70' dans OSCCON.

```
MOVLW   h'70'
PAGE1
MOVWF   OSCCON         ; fréquence oscillateur interne 8 MHz
```

Exemple de début de programme avec un 16F88:

```
.*****
;
;**          TEST16F88.asm          **
;*****
;
          CBLOCK   h'20'           ; réservation de 2 cases en RAM
SAUV
CPT
          ENDC
          ORG     h'0000'         ; vecteur de Reset
          GOTO   PROG
#INCLUDE "MENES88C.inc"         ; oscillateur interne, PA6 et PA7 en I/O
PROG
          PAGE1
          CLRF   ANSEL           ; broches du CAN en I/O
          CLRF   TRISA          ; PORTA en sortie
          MOVLW  h'70'
          MOVWF  OSCCON         ; fréquence 8 MHz
          PAGE0
MAIN                                           ; Programme principal
```