Microcontrôleurs PIC

(Cas du 16f628)

(Version 1.00 de février 2002)
Par

Jacques WEISS

SUPELEC Campus de Rennes

Table des illustrations :

1 Introduction

Ce document se veut être une aide à la compréhension de l'architecture de microcontrôleurs PIC de MicroChip ; il ne présente que quelques aspects d'un modèle de composant (PIC16f628) ; il ne dispense ABSOLUMENT PAS d'aller consulter la documentation technique du fabricant, même si elle fait 160 pages écrites en anglais !

Ce composant intègre un microcontrôleur 8 bits, c'est à dire un processeur et des périphériques, dans un boîtier « *Dual in Line* » de 18 broches. Il est réalisé en technologie CMOS et peut cadencé par une horloge allant de 0 à 20 MHz ; il doit être alimenté par une tension allant de 3 à 5,5 Volts.

Les broches du composant possèdent plusieurs affectations entre les ports d'E/S, les périphériques et les fonctions système.

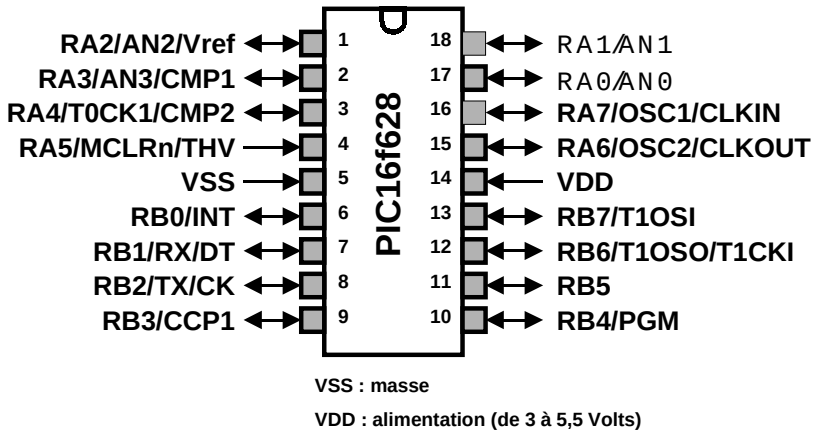

Fig. 1.0 : Brochage du PIC16f628

Note : la notation des signaux actifs à l'état bas utilise la lettre minuscule n en fin de symbole ; exemple : le signal « ENTREE BARRE » sera écrit : ENTREEn.

Il est ainsi possible, dans certaines configurations, de profiter de 16 broches d'E/S (en fait 16 sorties et 15 entrées) sur le composant.

En périphérie de l'unité centrale, on peut recenser les ressources suivantes sur le composant :

Mémoire Flash Programme : 2048 instructions
Mémoire RAM Données : 224 Octets
Mémoire EEPROM Données : 128 Octets
Ports d'E/S : 2 ports 8 bits
Périphériques :
- 3 Timers (8 et 16 bits)
- 1 module *Capture/Compare/PWM*
- 2 comparateurs analogiques
- 1 référence de tension
- 1 USART (émission/réception série synchrone et asynchrone)

2 Architecture

Le microcontrôleur est composé d'une unité centrale et de périphériques ; le fonctionnement est géré par un séquenceur qui, en fonction des modes opératoires, fournit les signaux de contrôle à chaque module.

Le fonctionnement de l'unité centrale est de type RISC (*Reduced Instruction Set Computer*), le jeu d'instructions est réduit à 35.

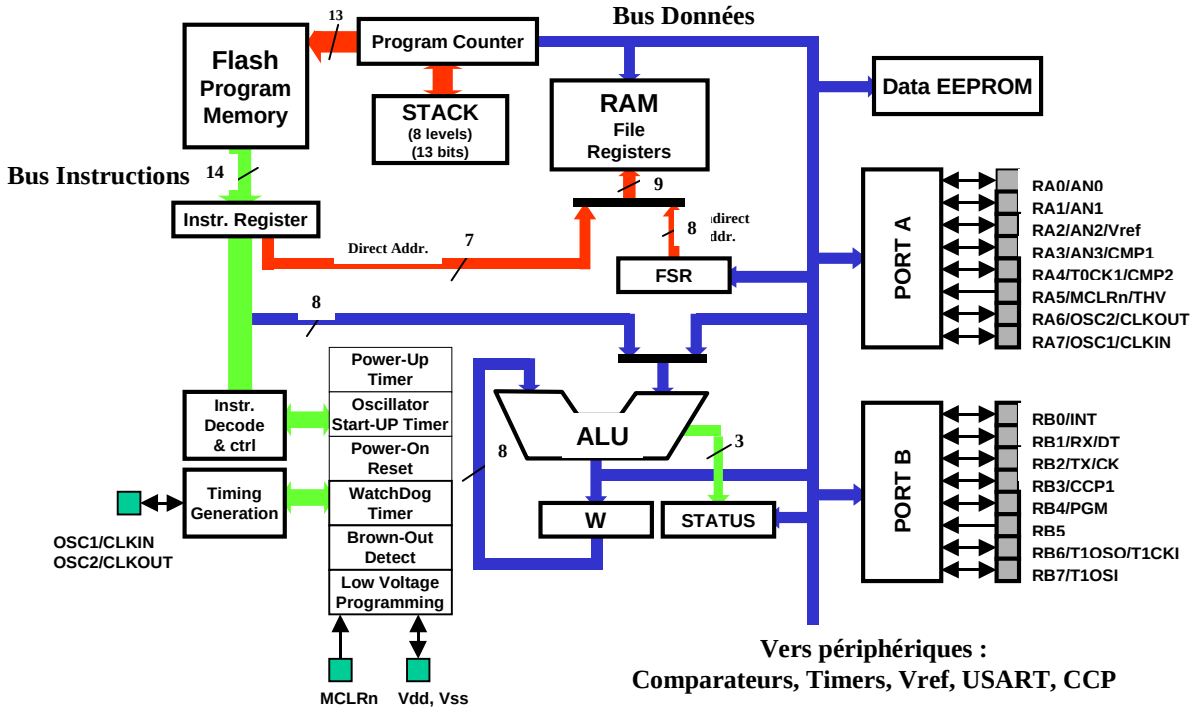

Fig. 2.1 : Architecture PIC (16f628)

2.1 Unité centrale

Le cœur du processeur est composé de 3 entités :

L'ALU (opérations arithmétiques et logiques sur 8 bits) et son registre d'état (STATUS)
Le registre de travail : W (*Working Register*)
Le registres d'usage général (*File Registers*)

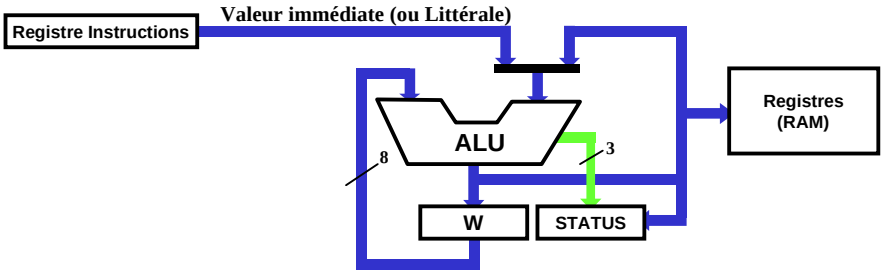

Fig. 2.2 : Cœur d'un processeur PIC

Les opérations possibles sont alors les suivantes (avec un exemple) :

- W ← W (op) valeur immédiate — ADDLW 0xFF
- Dest. ← W (op) Registre — ADDWF REGISTRE, dest.
- Registre ← W — MOVWF REGISTRE
- Dest. ← Registre — MOVF REGISTRE,dest.
- Dest. ← Registre ± 1 — INCF REGISTRE,dest.
- Dest. ← Registre ± 1 plus test — INCFSZ REGISTRE,dest.
- Manipulation et test individuel de bits sur les registres

Où :

(op) représente une opération arithmétique ou logique
Dest. Représente la destination des données (W ou le REGISTRE concerné)

2.1.1 Instructions

2.1.1.1 Format des instructions

Les instructions sont codées sur 14 bits ; leur format, figure suivante, dépend des variables employées (octet, bit ou valeur immédiate).

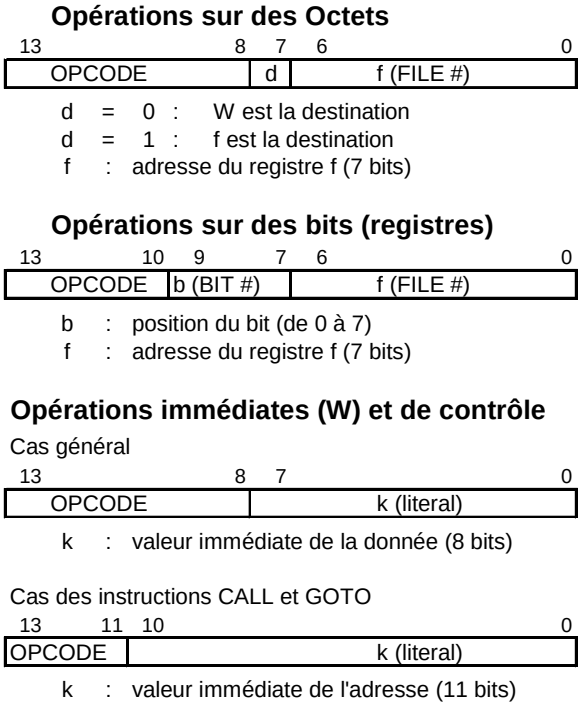

Fig. 2.3 : Format des instructions

2.1.1.2 Séquencement des instructions

Le processeur est piloté par un générateur de signaux qui divise l'horloge système par 4 ; ainsi le cycle machine est 4 fois plus long que la période d'horloge.

Le processeur fonctionne en mode pipe-line (x2), ce qui lui permet d'exécuter potentiellement une instruction par cycle ; cette cadence est brisée lors de sauts de programme, il faut dans ce cas là vider le pipe-line (*flush*) avant d'exécuter le sous-programme.

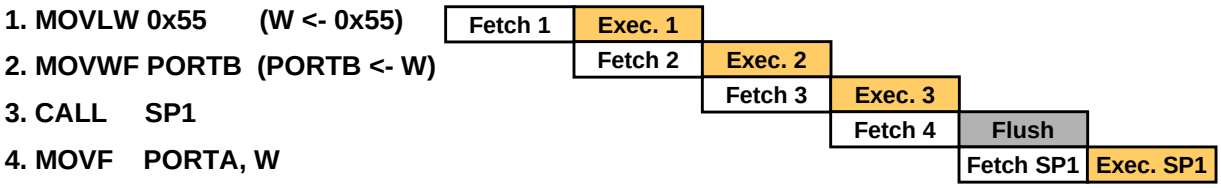

Fig. 2.4 : Séquencement du processeur

2.1.1.3 Jeu d'instructions

La figure suivante présente l'ensemble des 35 instructions d'un processeur PIC ; ce jeu est commun à la plupart de microcontrôleurs du fabricant. Pour avoir le détail de chaque instruction, il faut se reporter à la documentation du constructeur.

Mnémonique, opérande		Description	Code Opératoire (14 bits) MSb LSb				Drapeaux (STATUS)
Opérations sur des Octets							
ADDWF	f, d	Add W and f	00	0111	dfff	ffff	C, DC, Z
ANDWF	f, d	AND W with f	00	0101	dfff	ffff	Z
CLRF	f, d	Clear f	00	0001	1fff	ffff	Z
CLRW	-	Clear W	00	0001	0xxx	xxxx	Z
COMF	f, d	Complement f	00	1001	dfff	ffff	Z
DECF	f, d	Decrement f	00	0011	dfff	ffff	Z
DECFSZ	f, d	Decrement f Skip if 0	00	1011	dfff	ffff	
INCF	f, d	Increment f	00	1010	dfff	ffff	Z
INCFSZ	f, d	increment f Skip if 0	00	1111	dfff	ffff	
IORWF	f, d	Inclusive OR W with f	00	0100	dfff	ffff	Z
MOVF	f, d	Move f	00	1000	dfff	ffff	Z
MOVWF	f	Move W to f	00	0000	1fff	ffff	
NOP	-	No Operation	00	0000	0xx0	0000	
RLF	f, d	Rotate Left f through Carry	00	1101	dfff	ffff	C
RRF	f, d	Rotate Right f throuch Carry	00	1100	dfff	ffff	C
SUBWF	f, d	Substract W from f	00	0010	dfff	ffff	C, DC, Z
SWAPF	f, d	Swap nybbles with f	00	1110	dfff	ffff	
XORWF	f, d	Exclusive OR W with f	00	0110	dfff	ffff	Z
Opérations sur des bits (registres)							
BCF	f, b	Bit Clear f	01	00bb	bfff	ffff	
BSF	f, b	Bit Set f	01	01bb	bfff	ffff	
BTFSC	f, b	Bit Test f, Skip if Clear	01	10bb	bfff	ffff	
BTFSS	f, b	Bit Test f, Skip if Set	01	11bb	bfff	ffff	
Opérations immédiates (W) et de contrôle							
ADDLW	k	Add Literal and W	11	111x	kkkk	kkkk	C, DC, Z
ANDLW	k	AND Literal and W	11	1001	kkkk	kkkk	Z
CALL	k	Call subroutine	10	0kkk	kkkk	kkkk	
CLRWDT	-	Clear Watchdog Timer	00	0000	0110	0100	T0n, PDn
GOTO	k	Go To address	10	1kkk	kkkk	kkkk	
IORLW	k	Inclusive OR literal with W	11	1000	kkkk	kkkk	Z
MOVLW	k	Move Literal to W	11	00xx	kkkk	kkkk	
RETFIE	-	Return from interrupt	00	0000	0000	1001	
RETLW	k	Return with literal in W	11	01xx	kkkk	kkkk	
RETURN	-	Return from Subroutine	00	0000	0000	1000	
SLEEP	-	Go into standby mode	00	0000	0110	0011	T0n, PDn
SUBLW	k	Substract W from literal	11	110x	kkkk	kkkk	C, DC, Z
XORLW	k	Exclusive OR literal with W	11	1010	kkkk	kkkk	Z

Notations

f	Registre (*File Register*)
d	Destination de l'opération (W("0") ou f ("1"))
k	Valeur immédiate (*Literal value*)
b	Position du bit dans l'octet

Fig. 2.5 : Jeu d'instructions

2.2 Mémoire

Ce microcontrôleur est basé sur une architecture de processeur de type *Harvard*, c'est à dire qu'il y a séparation des bus d'instructions et de données ainsi que de l'espace d'adressage.

2.2.1 Mémoire programme

Les instructions du programme sont stockées sur 14 bits dans une zone mémoire dont l'adresse s'étend de 000h à 7FFh (2048 lignes de programme pour le 16f628).

Il y a 2 adresses réservées pour les vecteurs d'initialisation (*Reset* et interruptions) ; le processeur possède une pile de 8 niveaux pour gérer les adresses de retour de sous programmes ; il n'y a aucun contrôle sur l'état de la pile par le processeur.

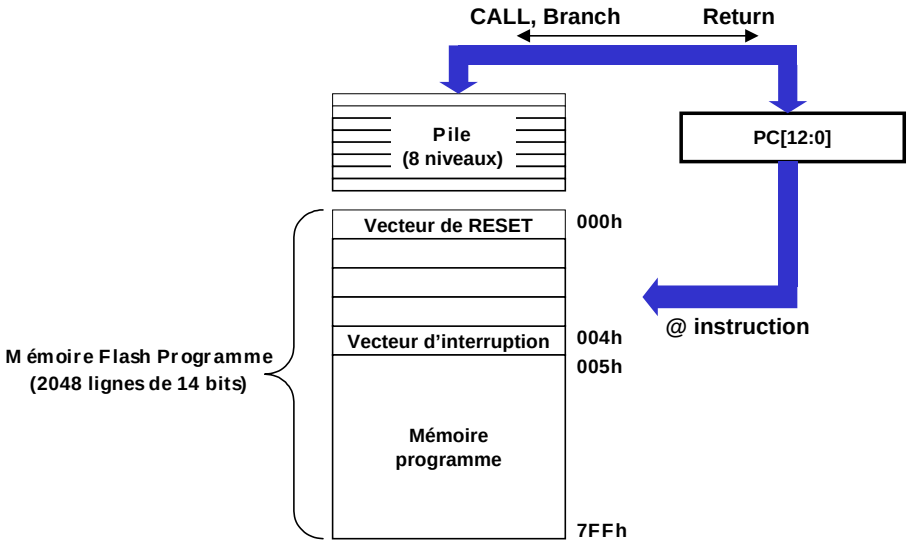

Fig. 2.6 : Mémoire programme

2.2.2 Mémoire données (Registres)

La mémoire données correspond aux registres (*File Registers*) vus par l'unité centrale ; ces registres sont de 2 types :

- Registres « Système » : ils permettent la configuration et la surveillance de l'état du processeur et de ses périphériques ; ces registres sont accessibles en lecture ou en écriture suivant leur fonction ; il y a 34 registres de ce type pour le 16f628.

- Registres d'usage général : ils permettent le stockage de variables; ils sont accessibles en lecture et en écriture par le programme ; il y a 224 registres de ce type pour le 16f628.

Le jeu d'instructions du processeur ne permet l'adressage des registres que sur 7 bits (128 registre), l'espace mémoire est alors décomposé en 4 bancs de 128 registres ; la cartographie de la mémoire données est donnée par la figure suivante ; les zones grisées ne sont pas implémentées.

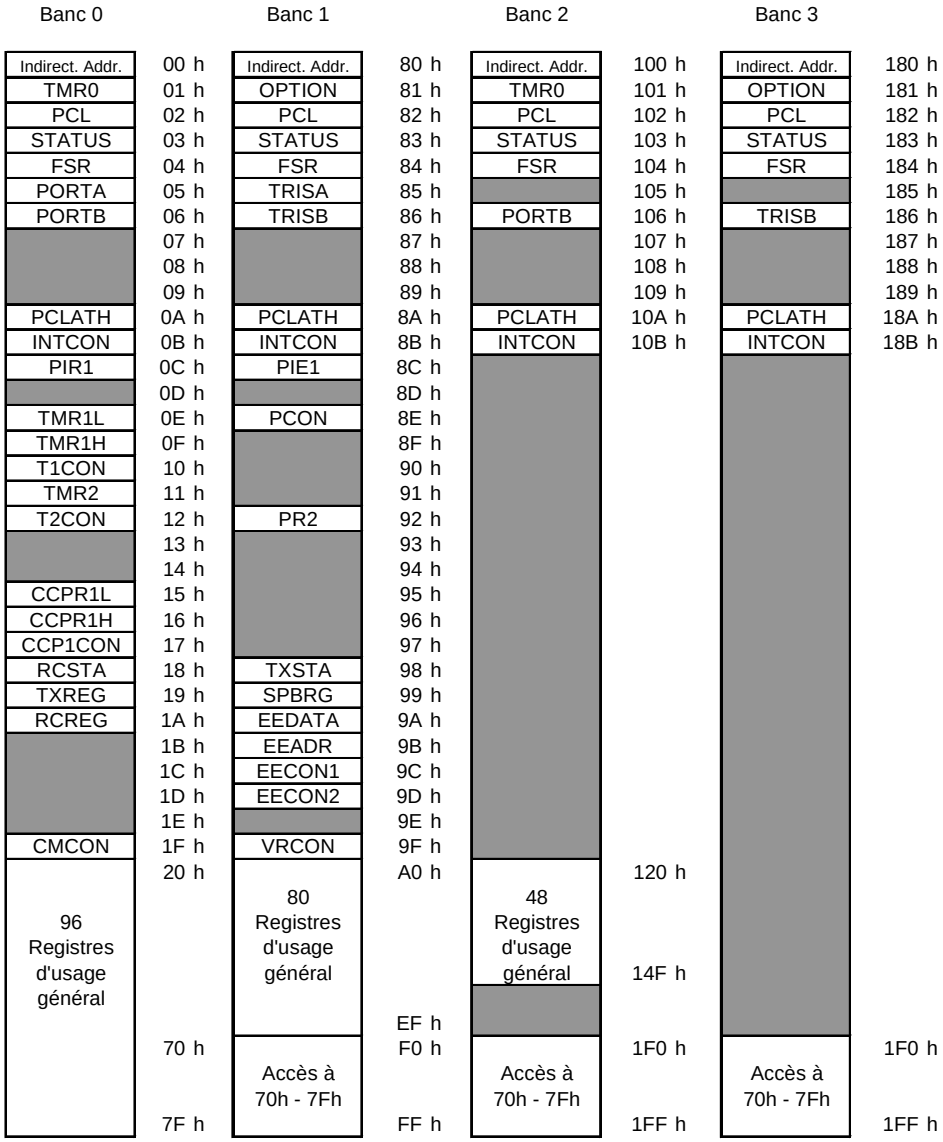

Fig. 2.7 : Mémoire données (Registres)

Nous allons maintenant nous intéresser à quelques registres « système » ; la description qui suit se veut sommaire ; les tableaux suivants utilisent la syntaxe suivante :

R/W-x pour *Read/Write* – valeur (x) à la mise sous tension.

2.2.2.1 STATUS

Ce registre permet de choisir le banc auquel on veut accéder et de lire les drapeaux (*Flags*) de l'ALU et du timer.

	@	R/W-0	R/W-0	R/W-0	R-1	R-1	R/W-x	R/W-x	R/W-x
STATUS	03h ou 83h	IRP	RP1	RP0	TOn	PDn	Z	DC	C
		bit 7							bit 0

IRP : sélection des bancs 0, 1 ou 2, 3 en adressage indirect
RP0, RP1 : choix du banc (de 0 à 3) en adressage direct
Z, DC, C : drapeaux de l'ALU

2.2.2.2 OPTION

Ce registre permet de configurer cetains paramètres du Port B et le Timer 0.

	@	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
OPTION	81h	RBPUn	INTDEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
		bit 7							bit 0

RBPUn : Validation de résistances de *Pull-Up* sur le Port B
INTEDG : sélection du front actif de l'interruption à partir de RB0/INT
Les autres bits permettent de configurer la source du Timer 0.

2.2.2.3 INTCON

Ce registre contient les bits de validation et les drapeaux des sources principales d'interruptions.

	@	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
INTCON	0Bh ou 8Bh	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF
		bit 7							bit 0

Validation/masquage des interruptions :

GIE : (*Glogal Interrupt Enable*) : toutes les interruptions
PEIE : (*Peripheral Int. Enable*) : interruptions des périphériques
T0IE : (*timer 0 Int. Enable*) : interruption de RB0/INT
INTE : (*External Int. Enable*) : interruption de RB0/INT
RBIE : (*Port B Change Int. Enable*) : interruption de RB7-RB4

Drapeaux d'interruptions :

T0IF : Timer 0
INTF : RB0/INT
RBIF : changement de RB7-RB4

2.2.2.4 PIE1 et PIR1

Ces registres permettent de gérer la validation des interruptions (PIE1) et de scruter les drapeaux (PIR1) des organes périphériques.

	@	R/W-0	R/W-0	R/W-0	R/W-0		R/W-0	R/W-0	R/W-0
PIE1	8Ch	EEIE	CMIE	RCIE	TXIE		CCPIE	TMR2IE	TMR1IE
		bit 7							bit 0

	@	R/W-0	R/W-0	R/W-0	R/W-0		R/W-0	R/W-0	R/W-0
PIR1	0Ch	EEIF	CMIF	RCIF	TXIF		CCPIF	TMR2IF	TMR1IF
		bit 7							bit 0

EEI*x* : EEPROM Données
CMI*x* : Comparateurs analogiques
RCI*x* : Réception USART
TXI*x* : Emission USART
CCPI*x* : *Capture/Compare/PWM*
TMR*x*I*y* : Timers 1 et 2

2.2.2.5 PCON

Ce registre sert à gérer les conditions de réveil du processeur (*Resets, Sleep, Brown-Out*)

	@				R/W-1		R/W-q	R/W-q
PCON	8Eh				OSCF		PORn	BODn
					bit 7			bit 0

OSCF : choix de la fréquence de l'oscillateur interne (modes INTRC/ER)

2.2.3 Compteur Ordinal (PC, PCL et PCLATH)

Le compteur ordinal a une longueur de 13 bits (on en utilise 11 pour les 2048 lignes de code du 16f628) alors que le processeur ne gère les données que sur 8 bits et les adresses que sur 11 bits. Ainsi, le compteur est considéré en 2 morceaux, la partie basse (8 bits LSB) est présente dans le registre PCL, alors que la partie haute n'est accessible qu'indirectement (PCLATH)

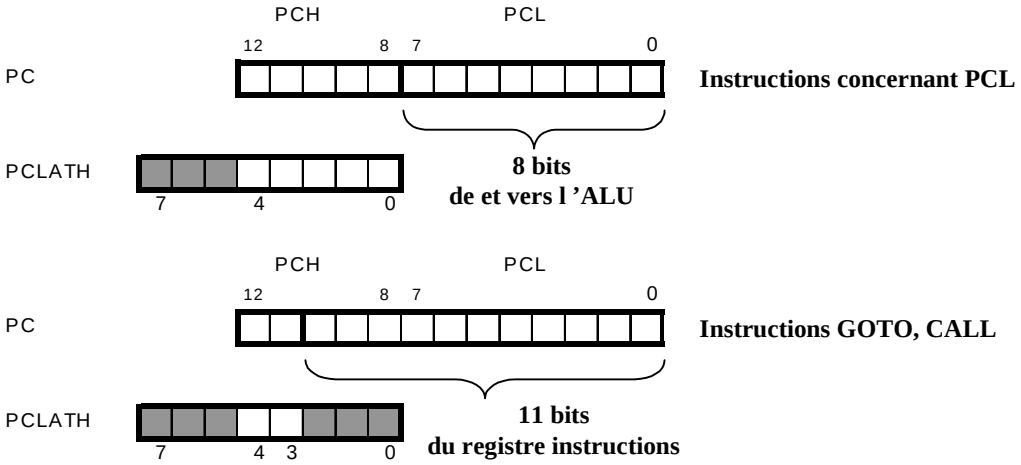

Fig. 2.8 : Compteur ordinal (PC)

Il est important de noter qu'en cas d'instruction de saut calculé en cours d'exécution du programme (offset dans une table, par exemple), on ne peut modifier que les 8 bits de poids faible (PCL) du compteur ; c'est pour cela qu'une table ne pourra pas excéder 256 cases, en supposant qu'elle démarre en début de page.

2.2.3.1 Adressage indirect (INDF et FSR)

L'adressage indirect est possible en passant par un registre virtuel (INDF) dont l'adresse est positionnée par le registre FSR (*File Select Register*) et le bit IRP du registre STATUS.

La figure suivante présente les 2 contextes d'adressage, direct et indirect.

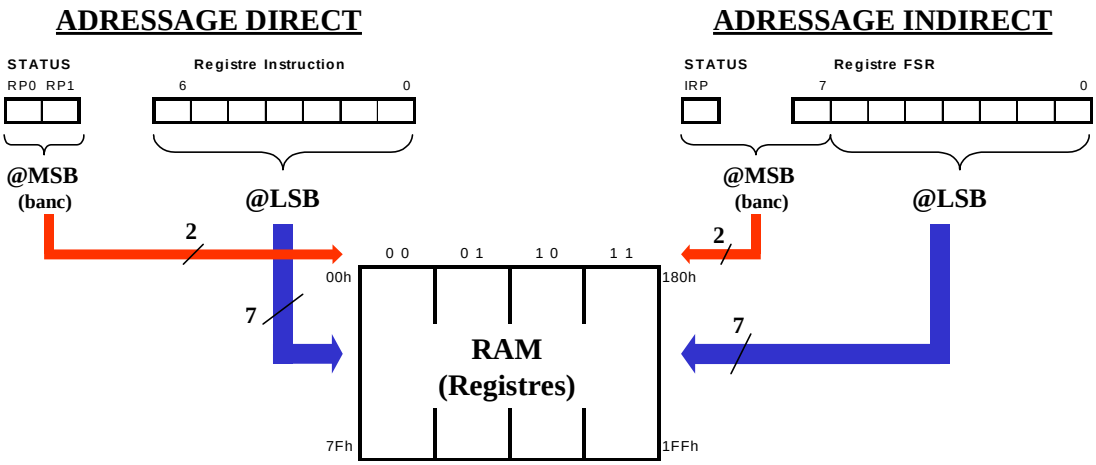

Fig. 2.9 : Adressage Direct/Indirect

Exemple d'adressage indirect, on envoie 4 octets à partir de l'adresse DEPART sur le Port B :

```
    MOVLW    DEPART            ; début du bloc (adresse DEPART)
    MOVWF    FSR               ; enregistrement dans le pointeur (FSR)
    MOVLW    d'4'
    MOVWF    COMPTEUR
Boucle
    MOVF     INDF,w            ; Lecture de la valeur (adresse FSR)
    MOVWF    PORTB             ; envoi sur le port B
    INCF     FSR,f             ; incrémentation du pointeur
    DECFSZ   COMPTEUR, f       : décomptage
    GOTO     Boucle
```

2.3 Ports d'E/S

Le composant dispose de 2 ports d'E/S : A et B ; ceux-ci peuvent servir d'E/S numériques standards ou d'E/S de périphériques internes.

Ces ports sont bi-directionnels, leur configuration se fait par des registres spécifiques (TRISx) ; par exemple : TRISA configure le Port A : chaque bit placé à 1 place le bit du port en entrée. En cas d'utilisation de périphériques internes, ceux-ci sont prioritaires pour l'affectation des broche d'E/S.

Description sommaire des ports :

Note : sauf spécification particulière, les broches sont configurables en E/S numériques bi-directionnelles ; la liste ci-dessous ne présente alors que les configurations spécifiques à chaque broche.

2.3.1 Port A

A7 (RA7/OSC1/CLKIN) : broche d'entrée pour le signal l'horloge
A6 (RA6/OSC2/CLKOUT) : broche de sortie pour le signal d'horloge
A5 (RA5/MCLRn/THV) : broche de «Reset » et de programmation « haute tension »
A4 (RA4/T0CKI) : E/S « Drain Ouvert, entrée d'horloge pour le Timer 0
A3 (RA3/AN3) : E/S pour les comparateurs analogiques
A2 (RA2/AN2/Vref) : entrée pour les comparateurs analogiques et sortie de la tension de référence interne)
A1 (RA1/AN1) : entrée pour les comparateurs analogiques
A0 (RA0/AN0) : entrée pour les comparateurs analogiques

Les registres associés au port A sont : PORTA, TRISA, CMCON et VRCON

2.3.2 Port B

B7 (RB7/T1OSI) : entrée d'horloge pour le Timer 1
B6 (RB6/T1OSO/T1CKI) : E/S du Timer 1 et ligne d'horloge de la programmation du composant.
B5 (RB5) :
B4 (RB4/PGM) : contrôle du mode de programmation « basse-tension »
B3 (RB3/CCP1) : E/S du module *Capture/Compare/PWM*
B2 (RB2/TX/CK) : sortie ou horloge de l'USART
B1 (RB1/RX/DT) : : entrée de l'USART
B0 (RB0/INT) : : entrée d'interruption externe

Les registres associés au port B sont : PORTB, TRISB, OPTION et VRCON

Exemple d'initialisation de ports d'E/S

```
    CLRF     PORTA
    CLRF     PORTB
    BSF      STATUS, RP0       ; passage au banc 1
    BSF      OPTION_REG,7      ; Pas de Pull-Up sur PORT B,
    CLRF     TRISA             ; PORTA en sortie
    MOVLW    0x02
    MOVWF    TRISB             ; PORT B en SORTIE sauf RB1 (RxD)
    BCF      STATUS, RP0       ; retour au banc 1
```

2.4 Périphériques

2.4.1 Timer 0

Le module Timer 0 est un compteur 8 bits (TMR0), accessible en lecture et en écriture, dont la retenue alimente un drapeau d'interruption ; l'horloge de ce compteur peut être interne ou externe par l'intermédiaire d'un prédiviseur (3 bits).

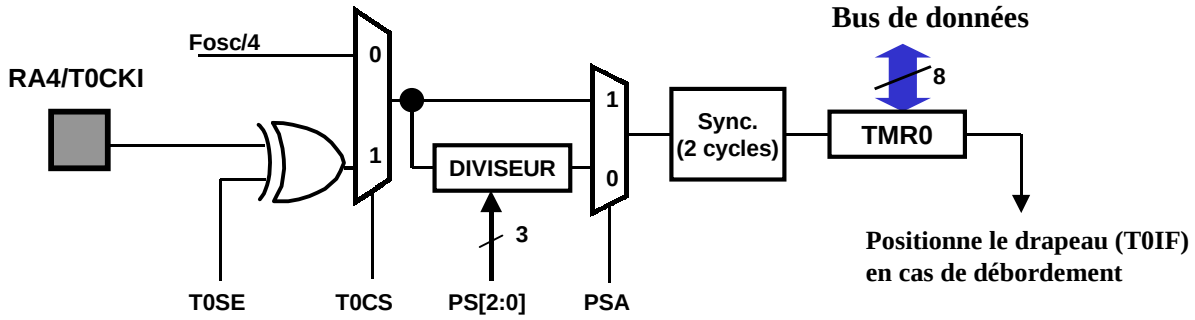

Fig. 2.10 : Timer 0

Les registres associés au Timer 0 sont : TMR0, INTCON, OPTION,TRISA(4)

2.4.2 Comparateurs analogiques

Le module est composé de 2 comparateurs analogiques, alimentés par le port A et la tension de référence interne.

	@	R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CMCON	1Fh	C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0
		bit 7							bit 0

C*x*OUT : sorties des comparateurs (peuvent être placées sur le Port A)
C*x*INV : polarité des sorties
CIS : multiplexage des entrées : RA0/RA3 et RA1/RA2
CM[2 :0] : choix du mode :

- 000 : comparateurs inactifs (entrées en mode analogique)
- 001 : 2 comparateurs avec une entrée commune (1 comp. Avec entrées multiplexées)
- 010 : 4 entrées (Port A) multiplexées vers 2 comparateurs (avec Vref)
- 011 : 2 comparateurs avec une entrée commune
- 100 : 2 comparateurs indépendants (entrées sur le port A)
- 101 : 1 comparateur
- 110 : 2 comparateurs avec une entrée commune et sortie sur le port A
- 111 : comparateurs inactifs (entrées en mode numérique)

Les registres associés au module comparateurs sont : CMCON, VRCON, INTCON, PIR1, PIE1 et TRISA.

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR	Value on All Other Resets
1Fh	CMCON	C2OUT	C1OUT	C2INV	C1NV	CIS	CM2	CM1	CM0	0000 0000	0000 0000
9Fh	VRCON	VREN	VROE	VRR	—	VR3	VR2	VR1	VR0	000- 0000	000- 0000
0Bh/8Bh/ 10Bh/18Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	EEIF	CMIF	RCIF	TXIF	—	CCP1IF	TMR2IF	TMR1IF	0000 -000	0000 -000
8Ch	PIE1	EEIE	CMIE	RCIE	TXIE	—	CCP1IE	TMR2IE	TMR1IE	0000 -000	0000 -000
85h	TRISA	TRISA7	TRISA6	—	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	11-1 1111	11-1 1111

Fig. 2.11 : registres associés au module comparateuts

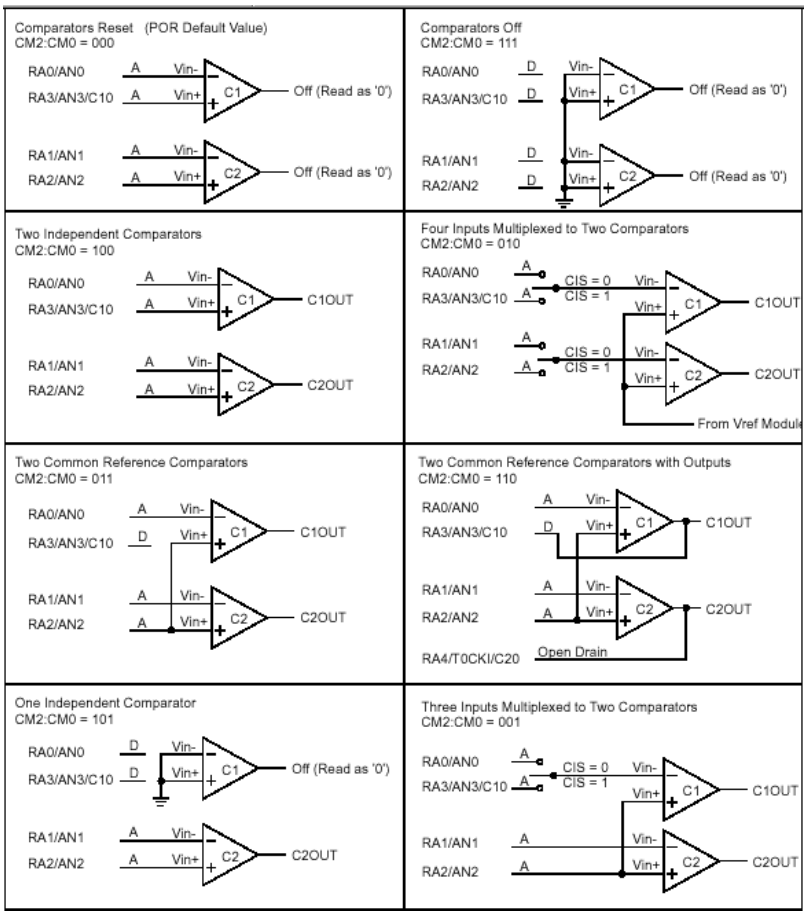

Fig. 2.12 : Modes de configuration des comparateurs

2.4.3 Tension de référence

Le module de référence est constitué d'un réseau de 16 résistances permettant d'alimenter les comparateurs avec une tension de référence.

	@	R/W-0	R/W-0	R/W-0		R/W-0	R/W-0	R/W-0	R/W-0
VRCON	9Fh	VREN	VROE	VRR		VR3	VR2	VR1	VR0
		bit 7							bit 0

VREN (*VREF Enable*) : Validation du module
VROE (*VREF Output Enable*) : sortie sur RA2
VRR (*VREF Range Selection*) : choix de la plage
VR[3 :0] : choix de la valeur (dans la plage) :
Si VRR = 1 : VREF = (VR[3 :0]/24)*Vdd
Si VRR = 0 : VREF =Vdd/4 (VR[3 :0]/32)*Vdd

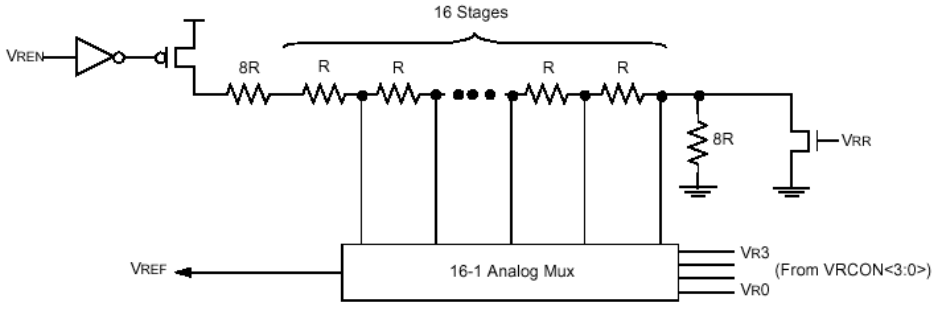

Fig. 2.13 : Module de Tension de référence

2.4.4 EEPROM Données

Le composant dispose d'une zone mémoire non-volatile (EEPROM) de 128 Octets ; cette mémoire n'est accessible que par adressage indirect via un registre propre au module.

Il faut savoir qu'une mémoire EEPROM offre un fonctionnement dissymétrique :

Lecture : le fonctionnement est comparable à celui d'une RAM

Ecriture : l'opération demande un certain temps avant d'être réalisée (mise en route d'une circuiterie d'élévation de tension et effacement de la cellule) dépendant des conditions (alimentation, température, technologie) ; les composants présentent un vieillissement à chaque écriture qui se traduit par un ralentissement de l'écriture.

Effacement : il permet de placer toutes les cellules mémoire à 1.

Pour limiter l'usure de l'EEPROM, il peut être intéressant d'analyser le contenu de la cellule mémoire pour savoir s'il n'y a que des 0 à inscrire (pas besoin d'effacer la cellule dans ce cas là).

Il y a 4 registres pour régir le module :

EECON1 et EECON2 : registres de configuration
EEADR : registre (7 bits) pointant sur l'adresse concernée
EEDATA : registre d'indirection

Le registre EECON1 ne contient que 4 bits pour le contrôle des opérations

	@	R/W	R/W	R/W	R/W-x	R/W-0	R/S-0	R/S-x
EECON1	9Bh				WRERR	WREN	WR	RD
		bit 7						bit 0

WRERR (*EEPROM Error Flag*) : drapeau indiquant une erreur d'écriture
WREN : autorisation d'écriture en mémoire
WR (ne peut qu'être forcé à 1) : Ordre d'écriture, réinitialisé automatiquement en fin d'opération (durée de plusieurs cycles machines suivant conditions)
RD (ne peut qu'être forcé à 1) : Ordre de lecture, réinitialisé automatiquement en fin d'opération (durée 1 cycle machine)

La manipulation de la mémoire EEPROM ne doit se faire que suivant un protocole imposé, voici des exemples :

La lecture est relativement facile à mettre en œuvre, elle se fait par adressage indirect et lancement d'un ordre de lecture ; l'opération se faisant en 1 cycle, il n'y a pas à se soucier de la disponibilité de la mémoire.

```
;*****************************************************************
     Lecture de la case EEPROM pointée par EEADR ; retour dans EEDATA
        Bsf    STATUS,RP0        ; Passage au banc mémoire 1
        bsf    EECON1,RD         ; Ordre de lecture EEPROM
        bcf    STATUS,RP0        ; Retour au banc mémoire 0
```

L'écriture est plus complexe à manipuler du fait de l'obligation d'écrire dans le registre virtuel EECON2 la séquence 55h puis AAh ; l'opération durant plusieurs cycles, il faut attendre la fin de l'écriture (par boucle (polling) ou par interruption).

```
;*****************************************************************
      Ecriture la case EEPROM pointée par EEADR (contenu dans EEDATA)
         bcf    INTCON,GIE        ; inhibition des interruptions
         bsf    STATUS,RP0        ; Passage au banc mémoire 1
         bsf    EECON1,WREN       ; Ordre d'écriture EEPROM
         movlw  H'55'           ; Séquence imposée
         movwf  EECON2          ; par le matériel
         movlw  H'AA'           ; de gestion de
         movwf  EECON2          ; la mémoire
         bsf    EECON1,WR         ; Ordre d'écriture EEPROM
;*****************************************************************
      Boucle d'attente de l'écriture effective (peut aussi être géré sous forme d'interruption)
 WAIT_EE
         btfsc  EECON1,WR
         goto   WAIT_EE
;*****************************************************************
         bcf    EECON1,EEIF
         bcf    EECON1,WREN       ; Protection d'écriture EEPROM
         bcf    STATUS,RP0        ; Retour au banc mémoire 0
         bsf    INTCON,GIE        ; validation des interruptions
```

2.5 Fonctions Spéciales

Le composant dispose de plusieurs configurations possibles qui vont définir le mode opératoire du microcontrôleur :

- le brochage : utilisation pour des fonctions « système » ou pour l'utilisateur,
- l'horloge : interne, externe, lente ou rapide,
- la temporisation à la mise en route (*Power-Up Timer*),
- la surveillance du bon déroulement du programme (*Watchdog Timer*),
- la surveillance de l'alimentation (*Brown-Out Detect*),
- la programmation ICSP,
- la protection du code.

Le choix de ces configurations se fait au moment du chargement du programme dans la mémoire Flash :

2.5.1 Mot de configuration

Aussi dénommé « *Device Fuses* », ce mot est stocké dans un registre spécial de 14 bits placé à l'adresse 2007h ; ce registre n'est pas accessible par le programme.

CONFIG 2007h	CP1	CP0	CP1	CP0	-	CPD	LVP	BODEN	MCLRE	FOSC2	PWRTEn	WDTE	FOSC1	FOSC0
	bit 13													bit 0

CP0 et CP1 : protection partielle ou totale du code programme
CPD : protection de la mémoire EEPROM
LVP : validation de la programmation « basse tension »
BODEN (*Brown-Out Detect Enable*) : supervision de l'alimentation
MCLRE : attribution de la broche RA5 en E/S ou pour le reset
PWRTEn (*Power-Up Timer Enable*) : temporisation à la mise sous tension
WDTE (*Watchdog Enable*) : chien de garde
FOSC[2 :0] : configuration de l'horloge (8 modes)

Voici un exemple de déclaration de configuration ; les mots clés sont fournis par le fichier p16f628.inc de MPLAB :

```
list      p=16f628
#include <p16f628.inc>
__CONFIG _CP_OFF & _WDT_OFF & _BODEN_OFF & _PWRTE_OFF & _EXTCLK_OSC &  _LVP_OFF
   &_MCLRE_ON
;      Code Protection Off
;      WatchDog désactivé
;      Brown-Out Detect inhibé
;      Power-Up Timer désactivé
;      External Clock (sans CLKOUT sur RA6)
;      Low Voltage Programming inhibé
;      Reset sur RA5/MCLRn
```

2.5.2 Configuration de l'horloge

Le composant dispose de 8 modes de configuration de l'horloge :

- ER (*External Resistor*) : l'horloge est réalisée par un oscillateur de type RC, la capacité est interne au composant et la résistance, externe, permet de fixer la fréquence désirée ; ce mode existe en 2 variantes suivant que l'on utilise la broche RA6 en E/S ou en sortie d'horloge (CLKOUT)

- INTRC (*Internal RC*) : même principe que précédemment mais avec une résistance interne.

- EC : on doit fournir un signal d'horloge au composant (broche RA7)

- HS, XT, LP : modes « classiques » pour un microcontrôleur correspondant à placer un quartz à un inverseur interne au composant (figure suivante) ; le choix du mode dépend de la fréquence de l'horloge.

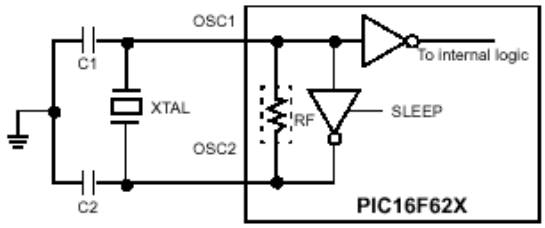

Fig. 2.14 : configuration d'oscillateur HS, XT et LP

2.5.3 Interruptions

Le processeur dispose de 10 sources d'interruption :

Externe : à partir de la broche RB0/INT
Débordement du Timer 0
Changement de l'état des broches RB[7 :4]
Module comparateurs analogiques
USART
Module CCP (*Capture/Compare/PWM*)
Débordement du Timer 1
Reconnaissance d'occurrence (*Match*) du Timer 2

Ces interruptions sont masquables individuellement ou par groupe ou encore en totalité ; la gestion est réalisée dans les registres de contrôle tels que INTCON, PIE1, PIR1, CMCON ….

Chaque source d'interruption est contrôlée par un *Enable* et génère un drapeau (*Flag*) ; il faut noter que le drapeau est indépendant de la validation de l'interruption concernée. La figure suivante présente la logique de gestion des interruptions.

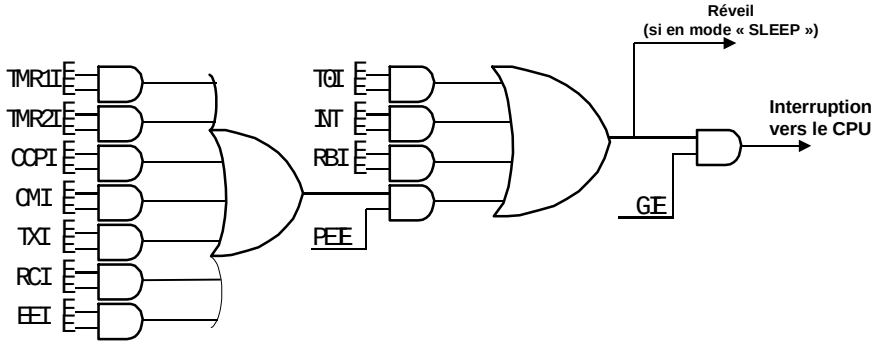

Fig. 2.15 : Logique de gestion des interruptions

Voici un exemple de sous-programme d'interruption dans le quel on sauvegarde le contexte (W et STATUS), on détermine la source de l'interruption, on la traite et on restaure le contexte avant de retourner au programme principal :

```
ORG      0x04              ; vecteur de départ des interruptions
MOVWF    W_TEMP            ; instructions "Pseudo PUSH"
SWAPF    STATUS,W
MOVWF    STATUS_TEMP       ; STATUS_TEMP <- STATUS (swapped)
BCF      STATUS,RP0        ; banc 0
BCF      INTCON,GIE        ; inhibition de toutes les interruptions
```

Recherche de l'origine de l'interruption

```
BTFSC    INTCON,T0IF
CALL     SPInt_Timer       ; Débordement du timer (TMR0)
BTFSC    PIR1,CMIF
CALL     SPInt_Compare     ; comparateur
BTFSC    INTCON,RBIF
CALL     SPInt_RB          ; Changement sur le Port B[7:4]
```

Fin de la gestion des interruptions, restauration du contexte et sortie.

```
BCF      STATUS,RP0        ; on se place sur le banc 0
SWAPF    STATUS_TEMP,W     ; instructions "Pseudo POP"
MOVWF    STATUS            ; STATUS <- STATUS_TEMP(swapped)
MOVF     W_TEMP, W         ; W <- W_TEMP
RETFIE   ; retour au programme et revalidation de INTCON, GIE
```

Il ne faut pas oublier que les drapeaux sont toujours actionnés, indépendamment de la validation de l'interruption concernée.

2.5.4 Programmation ICSP

La programmation consiste à charger le code du programme dans la mémoire Flash et des données dans la mémoire EEPROM Données . La technologie de ces mémoires requiert une forte tension (environ 12 Volts) pour polariser les cellules ; le 16f628 dispose d'une circuiterie interne (montage à pompage de charge) capable de générer cette tension, on parlera alors de programmation « basse tension » (LVP : *Low Voltage Programming*) ; il est aussi possible de fournir cette haute tension au composant, on se place alors dans le mode de programmation « haute tension » (THV) ; le choix du mode est fait par le mot de configuration.

La programmation se fait par liaison série synchrone ; il faut alors 3 fils pour réaliser l'opération :

1. Ligne pour forcer le composant en mode de programmation (RA5 en mode THV ou RB4 pour le mode LVP)
2. Ligne d'horloge (RB6)
3. Ligne de données (RB7).

Ce principe, dénommé, ICSP (*In-Circuit Serial Programming*), permet de programmer le composant In-Situ c'est à dire sur la carte de l'application, par l'intermédiaire d'un simple connecteur ; il faut toutefois prendre certaines précautions en ce qui concerne les branchements au composant :

- RA5/MCLRn : cette broche est portée à +12 Volts lors de la programmation,
- RB6 et RB7 : les composants périphériques ne doivent pas perturber le programmateur,
- RB4 doit se trouver à l'état bas si l'on veut programmer en mode THV.

3 Bibliographie

Documents techniques MICROCHIP

DS40300B : PIC16f62x *Data Sheet*
DS30277b : *In-Circuit Serial Programming*
DS39025e : EEPROM *Memory Programming Specification*

Livres :

Microcontrôleur PIC C. Tavernier, Editions Dunod
Apprendre la programmation des PIC P. Mayeux, Editions Dunod

Sites WEB (en français) :

http://www.supelec-rennes.fr/ren/fi/elec : pages Supélec
http://www.aurelienr.com/electronique.htm : réalisations à base de PIC
http://perso.wanadoo.fr/yves.heilig/ : électronique et robotique (autour de PIC)