MICROCONTROLEURS PIC (exemple 16F84)

Mme I. LACHKAR

Qu'est-ce qu'un PIC ?

- Un PIC est un microcontrôleur de chez Microchip. Ses caractéristiques principales sont :
- Séparation des mémoires de programme et de données (architecture Harvard) : On obtient ainsi une meilleure bande passante et des instructions et des données pas forcément codées sur le même nombre de bits.
- Communication avec l'extérieur seulement par des ports : il ne possède pas de bus d'adresses, de bus de données et de bus de contrôle comme la plupart des microprocesseurs.

Qu'est-ce qu'un PIC ?

- Utilisation d'un jeu d'instructions réduit, d'où le nom de son architecture : RISC (Reduced Instructions Set Construction). Les instructions sont ainsi codées sur un nombre réduit de bits, ce qui accélère l'exécution (1 cycle machine par instruction sauf pour les sauts qui requirent 2 cycles). En revanche, leur nombre limité oblige à se restreindre à des instructions basiques, contrairement aux systèmes d'architecture CISC (Complex Instructions Set Construction) qui proposent plus d'instructions donc codées sur plus de bits mais réalisant des traitements plus complexes.

Les trois familles de PIC :

Base-Line : Les instructions sont codées sur 12 bits
Mid-Line : Les instructions sont codées sur 14 bits
High-End : Les instructions sont codées sur 16 bits
Un PIC est identifié par un numéro de la forme suivante : xx(L)XXyy –zz
xx : Famille du composant (12, 14, 16, 17, 18)
L : Tolérance plus importante de la plage de tension
XX : Type de mémoire de programme C - EPROM ou EEPROM
CR - PROM F - FLASH
yy : Identification
zz : Vitesse maximum du quartz Nous utiliserons un PIC 16F84 –10, soit :
16 : Mid-Line
F : FLASH
84 : Type
10 : Quartz à 10MHz au maximum

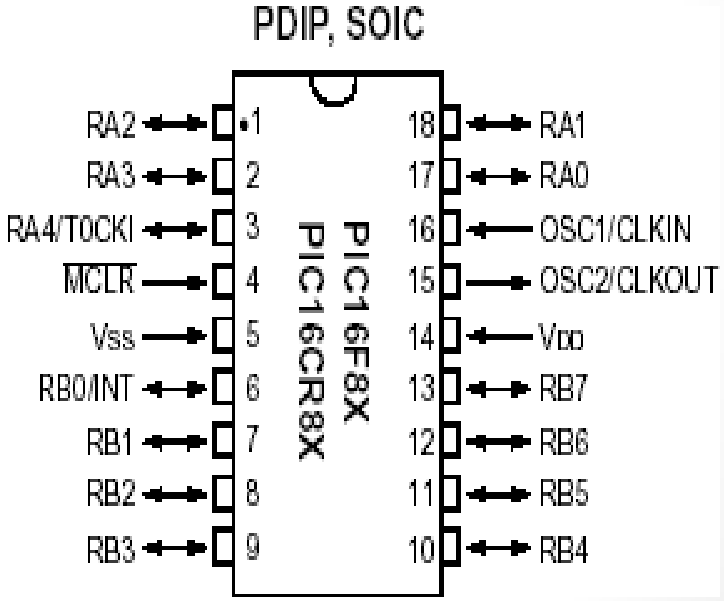
PIC 16F84

- Il s'agit d'un microcontrôleur 8 bits à 18 pattes.
- Principales caractéristiques :
  - 35 instructions
  - Instructions codées sur 14 bits
  - Données sur 8 bits
  - 1 cycle machine par instruction, sauf pour les sauts (2 cycles machine)
  - Vitesse maximum 10 MHz soit une instruction en 400 ns (1 cycle machine = 4 cycles d'horloge)
  - 4 sources d'interruption
  - 1000 cycles d'effacement/écriture pour la mémoire flash, 10.000.000 pour la mémoire de donnée EEPROM

Brochage et fonction des pattes

- Ci-contre le brochage du circuit. Les fonctions des pattes sont les suivantes :

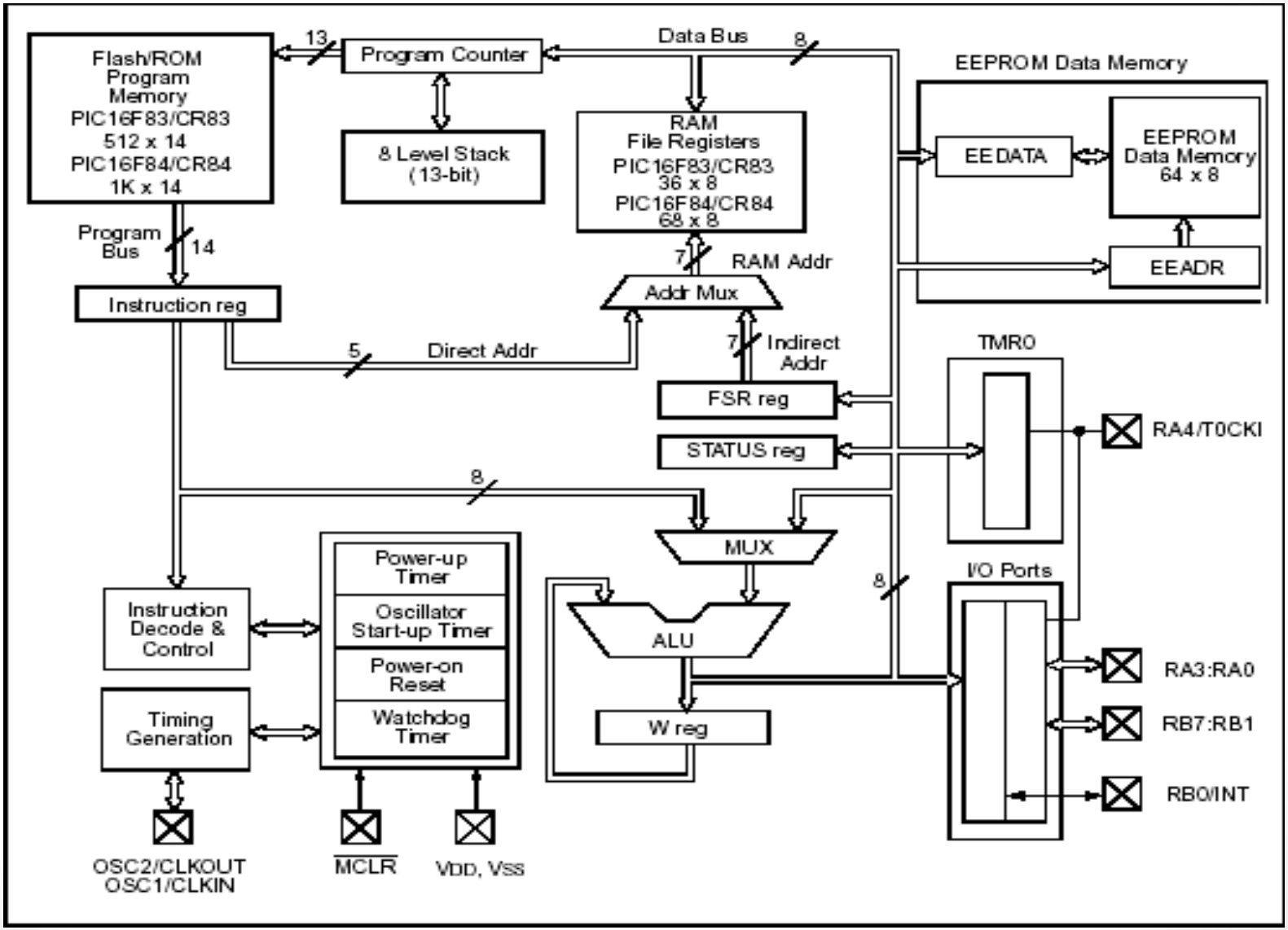
  - V_{SS}, V_{DD} :Alimentation
  - OSC1,2 : Horloge
  - RA0-4 : Port A
  - RB0-7 : Port B
  - T0CKL : Entrée de comptage
  - INT : Entrée d'interruption
  - MCLR : Reset : 0V



Architecture générale

- Il est constitué des éléments suivants :
  - un système d'initialisation à la mise sous tension (power-up timer, …)
  - un système de génération d'horloge à partir du quartz externe (timing génération)
  - une unité arithmétique et logique (ALU)
  - une mémoire flash de programme de 1k "mots" de 14 bits
  - un compteur de programme (program counter) et une pile (stack)
  - un bus spécifique pour le programme (program bus)
  - un registre contenant le code de l'instruction à exécuter
  - un bus spécifique pour les données (data bus)
  - une mémoire RAM contenant
    - les SFR
    - 68 octets de données
  - une mémoire EEPROM de 64 octets de données
  - 2 ports d'entrées/sorties
  - un compteur (timer)
  - un chien de garde (watchdog)

Architecture générale..

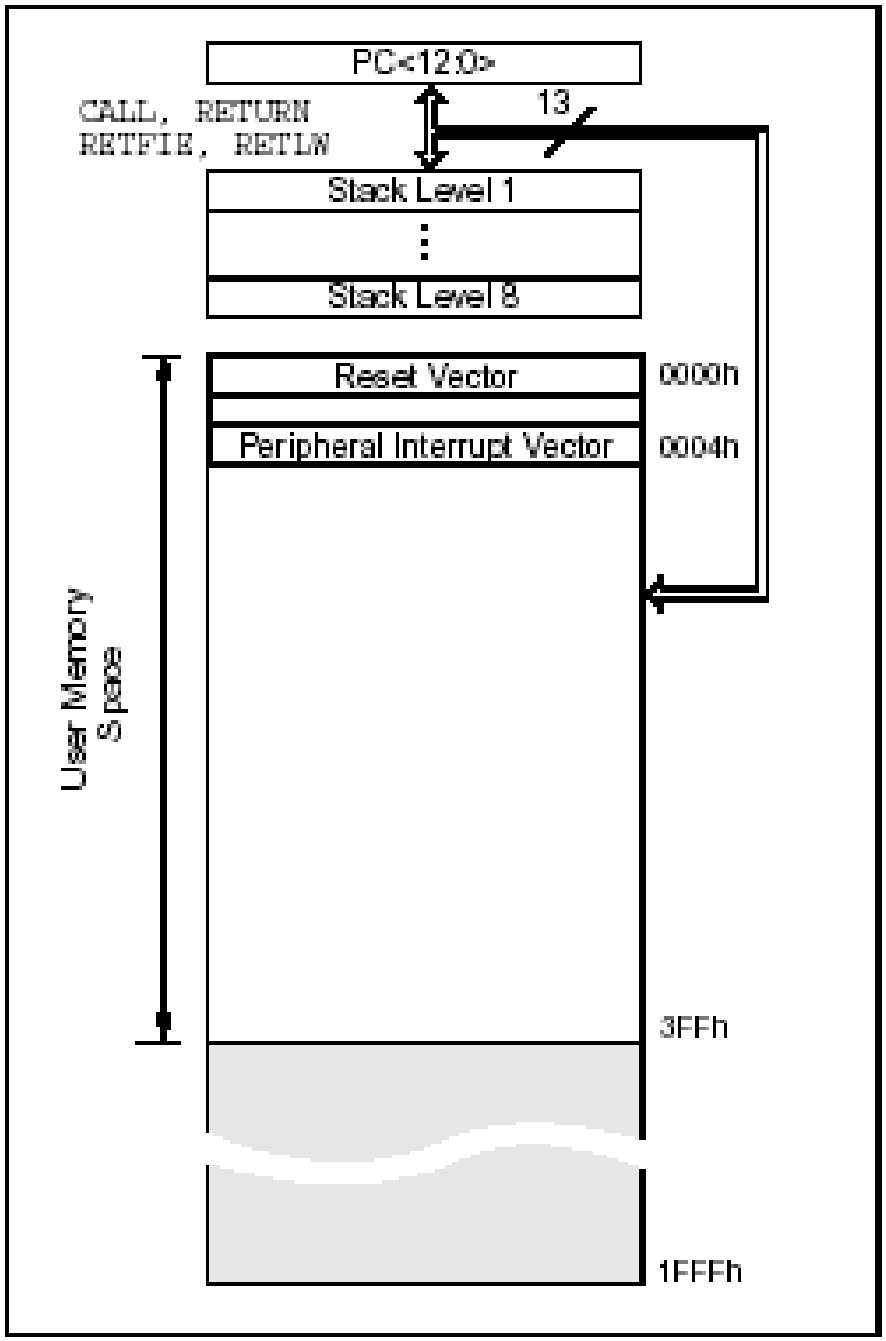


Organisation de la mémoire

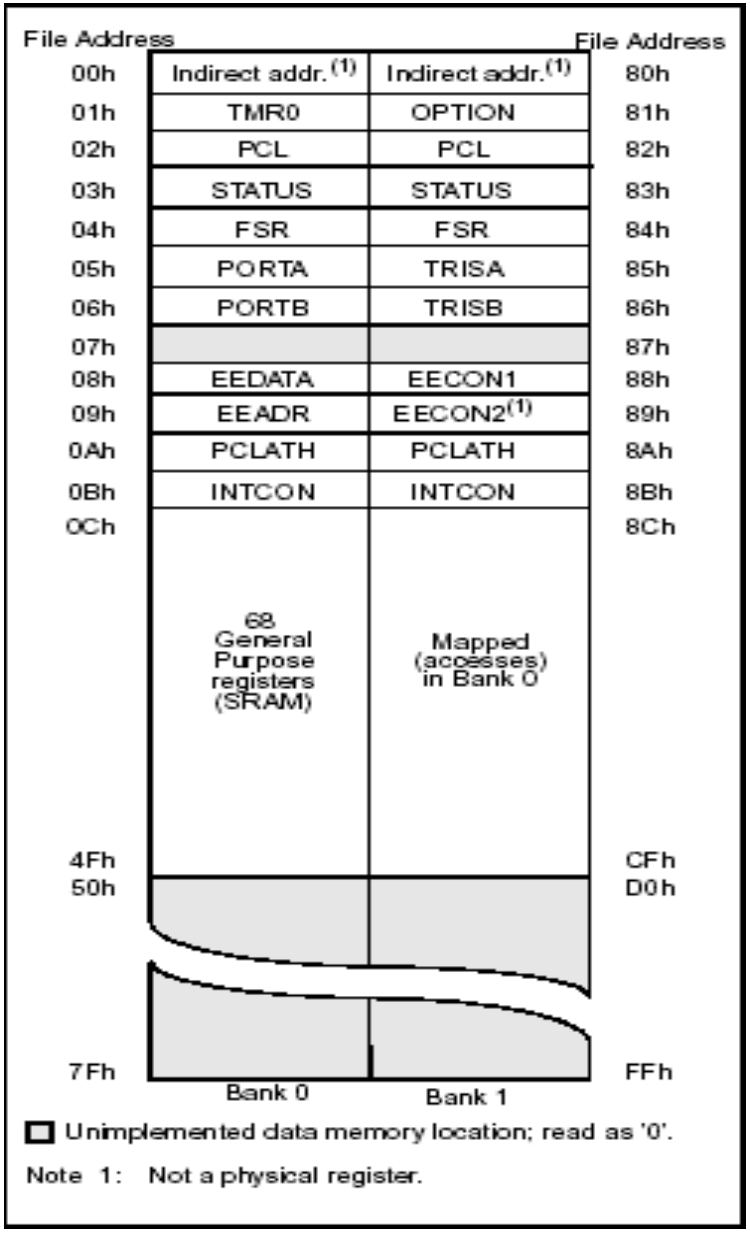
- Le PIC contient une mémoire de programme et une mémoire de données. La structure Harvard des PICs fournit un accès séparé à chacune. Ainsi, un accès aux deux est possible pendant le même cycle machine.

Mémoire de programme

- C'est elle qui contient le programme à exécuter. Ce dernier est téléchargé par liaison série.
- Elle contient 1k "mots" de 14 bits dans le cas du PIC 16F84, même si le compteur de programme (PC) de 13 bits peut en adresser 8k.
- Il faut se méfier des adresses images ! L'adresse 0000h contient le vecteur du reset, l'adresse 0004h l'unique vecteur d'interruption du PIC.
- La pile contient 8 valeurs. Comme le compteur de programme, elle n'a pas d'adresse dans la plage de mémoire. Ce sont des zones réservées par le système.


PC<12:0>
CALL, RETURN
RETFIE, RETLW
13
Stack Level 1
Stack Level 8
Reset Vector
0000h
Peripheral Interrupt Vector
0004h
User Memory Space
3FFh
1FFFh

Mémoire de données



Mémoire de données..

- Elle se décompose en deux parties de RAM et une zone EEPROM. La première contient les SFRs (Special Function Registers) qui permettent de contrôler les opérations sur le circuit. La seconde contient des registres généraux, libres pour l'utilisateur. Cette dernière contient 64 octets.
- le constructeur a défini deux banques. Le bit RP0 du registre d'état (STATUS.5) permet de choisir entre les deux. Ainsi, une adresse sur 8 bits est composée de RP0 en poids fort et des 7 bits provenant de l'instruction à exécuter.

Registres spéciaux - SFRs

- Ils permettent la gestion du circuit. Certains ont une fonction générale, d'autres un fonction spécifique attachée à un périphérique donné. La Figure donne la fonction de chacun des bits de ces registres. Ils sont situés de l'adresse 00h à l'adresse 0Bh dans la banque 0 et de l'adresse 80h à l'adresse 8Bh dans la banque 1. Les registres 07h et 87h n'existent pas.

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-on Reset	Value on all other resets (Note 3)
Bank 0											
00h	INDF	Uses contents of FSR to address data memory (not a physical register)								---- ----	---- ----
01h	TMR0	8-bit real-time clock/counter								xxxx xxxx	uuuu uuuu
02h	PCL	Low order 8 bits of the Program Counter (PC)								0000 0000	0000 0000
03h	STATUS (2)	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	000q quuu
04h	FSR	Indirect data memory address pointer 0								xxxx xxxx	uuuu uuuu
05h	PORTA	—	—	—	RA4/T0CKI	RA3	RA2	RA1	RA0	---x xxxx	---u uuuu
06h	PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0/INT	xxxx xxxx	uuuu uuuu
07h		Unimplemented location, read as '0'								---- ----	---- ----
08h	EEDATA	EEPROM data register								xxxx xxxx	uuuu uuuu
09h	EEADR	EEPROM address register								xxxx xxxx	uuuu uuuu
0Ah	PCLATH	—	—	—	Write buffer for upper 5 bits of the PC (1)					---0 0000	---0 0000
0Bh	INTCON	GIE	EEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
Bank 1											
80h	INDF	Uses contents of FSR to address data memory (not a physical register)								---- ----	---- ----
81h	OPTION_REG	$\overline{RBPU}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
82h	PCL	Low order 8 bits of Program Counter (PC)								0000 0000	0000 0000
83h	STATUS (2)	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	000q quuu
84h	FSR	Indirect data memory address pointer 0								xxxx xxxx	uuuu uuuu
85h	TRISA	—	—	—	PORTA data direction register					---1 1111	---1 1111
86h	TRISB	PORTB data direction register								1111 1111	1111 1111
87h		Unimplemented location, read as '0'								---- ----	---- ----
88h	EECON1	—	—	—	EEIF	WRERR	WREN	WR	RD	---0 x000	---0 q000
89h	EECON2	EEPROM control register 2 (not a physical register)								---- ----	---- ----
0Ah	PCLATH	—	—	—	Write buffer for upper 5 bits of the PC (1)					---0 0000	---0 0000
0Bh	INTCON	GIE	EEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u

Legend: x = unknown, u = unchanged, - = unimplemented read as '0', q = value depends on condition.

Note 1: The upper byte of the program counter is not directly accessible. PCLATH is a slave register for PC<12:8>. The contents of PCLATH can be transferred to the upper byte of the program counter, but the contents of PC<12:8> is never transferred to PCLATH.

2: The $\overline{TO}$ and $\overline{PD}$ status bits in the STATUS register are not affected by a $\overline{MCLR}$ reset.

3: Other (non power-up) resets include: external reset through $\overline{MCLR}$ and the Watchdog Timer Reset.

- INDF (00h - 80h) : Utilise le contenu de FSR pour l'accès indirect à la mémoire.
- TMR0 (01h) : Registre lié au compteur.
- PCL (02h - 82h) : Contient les poids faibles du compteur de programmes (PC). Le registre PCLATH (0Ah-8Ah) contient les poids forts.
- STATUS (03h - 83h) : Il contient l'état de l'unité arithmétique et logique ainsi que les bits de sélection des banques .
- FSR (04h - 84h) : Permet l'adressage indirect.
- PORTA (05h) : Donne accès en lecture ou écriture au port A, 5 bits. Les sorties sont à drain ouvert. Le bit 4 peut être utilisé en entrée de comptage.
- PORTB (06h) : Donne accès en lecture ou écriture au port B. Les sorties sont à drain ouvert.
- Le bit 0 peut être utilisé en entrée d'interruption.

- EEDATA (08h) : Permet l'accès aux données dans la mémoire EEPROM. EEADR (09h) : Permet l'accès aux adresses de la mémoire EEPROM.
- PCLATCH (0Ah - 8Ah) : Donne accès en écriture aux bits de poids forts du compteur de programme.
- INTCON (0Bh - 8Bh) : Masque d'interruptions.
- OPTION_REG (81h) : Contient des bits de configuration pour divers périphériques.
- TRISA (85h) : Indique la direction (entrée ou sortie) du port A.
- TRISB (86h) : Indique la direction (entrée ou sortie) du port B.
- EECON1 (88h) : Permet le contrôle d'accès à la mémoire
- EEPROM. EECON2 (89h) : Permet le contrôle d'accès à la mémoire EEPROM

Registre d'état (Status)

bit 7	**IRP**	Bit non utilisé (à laisser à 0)
bit 6	**RP1**	Bit non utilisé (à laisser à 0)
bit 5	**RP0**	Bit de sélection de la banque. •Il faut mettre ce bit à 0 pour accéder à la banque 0 (**bcf STATUS , RP0**), et à 1 pour accéder à la banque 1 (**bsf STATUS , RP0**).
bit 4	**/TO** **NOT_TO**	Bit "Time-out" (en lecture uniquement) •bit mis à 1 après : • une mise sous tension • une instruction SLEEP • une instruction CLRWDT • un réveil (sortie du mode SLEEP) dû à une interruption •bit mis à 0 quand la temporisation du watchdog est dépassée **plus d'informations**

Registre d'état (Status)…

bit 3	**/PD** **NOT_PD**	Bit "Power-down" (en lecture uniquement) •bit mis à 1 après : • une mise sous tension • une instruction CLRWDT •bit mis à 0 après une instruction SLEEP
bit 2	**Z**	Bit "Zero" •Cela ne concerne que les instructions qui affectent le bit Z (addwf, andlw, movf ...) •Ce bit est mis à 1 quand le résultat d'une opération est 0. •Ce bit est mis à 0 quand le résultat d'une opération est différent de 0.
bit 1	**DC**	Bit "Digit Carry" •Cela concerne les opérations arithmétiques (instructions : addwf, addlw, subwf et sublw) en système de numération DCB (binary coded decimal). •Ce bit est mis à 1 quand le résultat d'une opération génère une retenue. •Ce bit est mis à 0 quand le résultat d'une opération ne génère pas de retenue.
bit 0	**C**	Bit "Carry" •Cela concerne les opérations arithmétiques (instructions : addwf, addlw, subwf et sublw) en système de numération binaire en complément à 2. •Ce bit est mis à 1 quand le résultat d'une opération génère une retenue. •Ce bit est mis à 0 quand le résultat d'une opération ne génère pas de retenue

Registre de configuration de périphériques - OPTION_REG

bit 7	**NOT_RBPU**	"Port B Pull-up Enable" •Ce bit doit être mis à 0 pour activer les **résistances de pull-up** du port B •Ce bit doit être mis à 1 pour désactiver les résistances de pull-up du port B
bit 6	**INTEDG**	"Interrupt edge select" •Ce bit doit être mis à 0 pour que l'interruption de la broche RB0/INT soit active sur un front descendant •Ce bit doit être mis à 1 pour que l'interruption de la broche RB0/INT soit active sur un front montant

Registre de configuration de périphériques - OPTION_REG..

bit 5	**T0CS**	"TMR0 Clock Source Select" •Ce bit doit être mis à 0 pour que l'horloge du module TMR0 soit l'horloge interne (un quart de la fréquence du signal OSC1/CLKIN) •Ce bit doit être mis à 1 pour que l'horloge du module TMR0 soit le signal de la broche RA4/T0CKI
bit 4	**T0SE**	"TMR0 Source edge select" Dans le cas où T0CS = 1, le signal d'horloge de la broche RA4/T0CKI est actif : •sur front montant quand T0SE = 0 •sur front descendant quand T0SE = 1
bit 3	**PSA**	"Prescaler assignement" Le prédiviseur est attribué : •au module TMR0 quand PSA = 0 •au Watchdog quand PSA = 1
bits 2, 1, 0	**PS2, PS1, PS0**	"Prescaler rate select" (Cf. tableau ci- après)

Registre de configuration de périphériques - OPTION_REG…

Paramètres du prescaler :

PSA	**PS2, PS1, PS0**	Taux de prédivision du module TMR0	Taux de prédivision du Watchdog
0	000	2	**1**
0	001	4	**1**
0	010	8	**1**
0	011	16	**1**
0	100	32	**1**
0	101	64	**1**
0	110	128	**1**
0	111	256	**1**
1	000	**1**	1
1	001	**1**	2
1	010	**1**	4
1	011	**1**	8
1	100	**1**	16
1	101	**1**	32
1	110	**1**	64
1	111	**1**	128

Mémoire EEPROM

- Le PIC possède une zone EEPROM de 64 octets accessibles en lecture et en écriture par le programme. On peut y sauvegarder des valeurs, qui seront conservées même si l'alimentation est éteinte, et les récupérer lors de la mise sous tension. Leur accès est spécifique et requiert l'utilisation de registres dédiés. La lecture et l'écriture ne peut s'exécuter que selon des séquences particulières décrite plus tard.
- L'EEPROM du PIC 16F84A est indirectement accessible à travers les registres EEADR, EEDATA, EECON1 et EECON2.

EEPROM..

1- Registre EEADR

- Il s'agit d'un registre spécial situé à l'adresse 0x09 (banque 0) de la mémoire des données (Data RAM).
- Ce registre contient l'adresse de l'emplacement mémoire que l'on veut manipuler (pour une lecture ou une écriture).
- Les 64 octets sont situés aux adresses **0x00 à 0x3F**.

2- Registre EEDATA

- Il s'agit d'un registre spécial situé à l'adresse 0x08 (banque 0) de la mémoire des données (Data RAM).
- Ce registre contient la valeur (8 bits) de l'emplacement mémoire que l'on veut manipuler (pour une lecture ou une écriture).

EEPROM…

3- Registre EECON1

- Il s'agit d'un registre spécial situé à l'adresse 0x88 (banque 1) de la mémoire des données (Data RAM).

bit 7	-	-
bit 6	-	-
bit 5	-	-
bit 4	**EEIF**	"EEPROM Write operation interrupt flag" •drapeau (flag) mis à 1 en fin d'opération d'écriture •ce drapeau ne peut être effacé que de façon logicielle (**bcf EECON1, EEIF**)
bit 3	**WRERR**	"EEPROM Error flag bit" •drapeau (flag) mis à 1 en cas de problème durant l'opération d'écriture (Reset externe, Reset dû au Watchdog) •drapeau mis à 0 en fin d'opération d'écriture

EEPROM…

3- Registre EECON1..

bit 2	**WREN**	"EEPROM Write Enable bit" •1 : autorise l'écriture •0 : interdit l'écriture
bit 1	**WR**	"Write Control bit" •1 : lance une opération d'écriture •bit mis à 0 par hardware en fin d'opération d'écriture • Remarques : • on ne peut pas mettre à 0 ce bit par logiciel • une opération d'écriture dure plusieurs millisecondes
bit 0	**RD**	"Read Control bit" •1 : lance une opération de lecture •bit mis à 0 par hardware • Remarques : • on ne peut pas mettre à 0 ce bit par logiciel • une opération de lecture dure 1 cycle d'horloge

4- Registre EECON2

Il s'agit d'un registre spécial situé à l'adresse 0x89 (banque 1) de la mémoire des données (Data RAM). Ce registre n'est pas un registre physique ...

Lecture d'une donnée en mémoire EEPROM

Exemple : lecture de l'emplacement mémoire situé à l'adresse 0x10

bcf STATUS , RP0 ; passage en banque 0
movlw 0x10
movwf EEADR ; 0x10 est l'adresse de l'emplacement mémoire
bsf STATUS , RP0 ; passage en banque 1
bsf EECON1 , RD ; lecture de l'EEPROM
bcf STATUS , RP0 ; passage en banque 0
movf EEDATA, W ; la valeur lue dans l'EEPROM est placée dans l'accumulateur

Ecriture d'une donnée en mémoire EEPROM

Exemple : écriture de la donnée 0xE3 dans l'emplacement mémoire situé à l'adresse 0x10

bcf STATUS , RP0 ; passage en banque 0
movlw 0x10
movwf EEADR ; 0x10 est l'adresse de l'emplacement mémoire
movlw 0xE3
movwf EEDATA ; 0xE3 est la donnée à écrire dans l'emplacement mémoire

bsf STATUS , RP0 ; passage en banque 1
bcf INTCON , GIE ; désactivation de toutes les interruptions (recommandation de Microchip)
bsf EECON1 , WREN ; autorisation de l'écriture

Jeu d'instructions

- Les PICs sont conçus selon une architecture RISC. Programmer avec un nombre d'instructions réduit permet de limiter la taille de leur codage et donc de la place mémoire et du temps d'exécution.
- Toutes les instructions sont codées sur 14 bits. Elles sont regroupées en trois grands types :
  - Instructions orientées octets
  - Instructions orientées bits
  - Instructions de contrôle
- Le registre de travail W joue un rôle particulier dans un grand nombre d'instructions.
- Une instruction nécessite 1 cycle, ou bien 2 cycles dans le cas d'une instruction de branchement (GOTO, CALL ...).
- Avec une horloge à quartz de 20 MHz, un cycle correspond à $4/(20.10^6) = 200$ nanosecondes.
- Le microcontrôleur peut donc exécuter jusqu'à 5 millions d'instructions par seconde (5 MIPS) !

Mnémonique , opérande	Description	bit du registre STATUS affecté	nombre de cycles
MOVLW k	k (8 bits) est chargé dans (W)	-	1
ADDLW k	Additionne k (8 bits) et (W) et place le résultat dans (W)	C, DC , Z	1
SUBLW k	Soustrait W de k (8 bits) et place le résultat dans (W) k - (W) -> (W)	C, DC , Z	1
ANDLW k	Réalise un ET logique entre k (8 bits) et (W), et place le résultat dans (W)	Z	1
IORLW k	Réalise un OU logique (inclusif) entre k (8 bits) et (W), et place le résultat dans (W)	Z	1
XORLW k	Réalise un OU exclusif entre k (8 bits) et (W), et place le résultat dans (W)	Z	1

W : registre de travail (accumulateur), taille 8 bits
k : valeur littérale, taille 8 bits

Mnémonique , opérande	Description	bit du registre STATUS affecté	nombre de cycles
GOTO L	Branchement à l'adresse L	-	2
CALL L	Appelle un sous-programme (subroutine) situé à l'adresse L	-	2
RETURN	Retour de sous-programme	-	2
RETLW k	Retour de sous-programme, avec chargement de la valeur littérale k (8 bits) dans (W)	-	2
RETFIE	Retour de sous-programme d'interruption	-	2
CLRWDT	Efface le Watchdog	/TO, /PD	1
SLEEP	Place le microcontrôleur en mode sommeil	/TO, /PD	1

L : label (étiquette)

Mnémonique , opérande	Description	bit du registre STATUS affecté	nombre de cycles
BCF f , b	Mise à 0 du b ème bit du registre f	-	1
BSF f , b	Mise à 1 du b ème bit du registre f	-	1
BTFSC f , b	Si le b ème bit du registre f est égal à 0, alors l'instruction suivante est ignorée, et une instruction NOP est exécutée à la place (soit 2 cycles)	-	1 ou 2
BTFSS f , b	Si le b^{ème} bit du registre f est égal à 1, alors l'instruction suivante est ignorée, et une instruction NOP est exécutée à la place (soit 2 cycles)	-	1 ou 2

f : registre (spécial ou d'usage général)
b : position du bit (0 à 7)

Mnémonique , opérande	Description	bit du registre STATUS affecté	nombre de cycles
MOVWF f	(W) est chargé dans (f)	-	1
MOVF f , d	(f) (8 bits) est chargé dans (destination)	Z	1
ADDWF f , d	Additionne le contenu du registre f (8 bits) et (W), et place le résultat dans (destination)	C, DC , Z	1
SUBWF f , d	Soustrait (W) de (f) (8 bits) et place le résultat dans (destination). (f) - (W) ->(destination)	C, DC , Z	1
ANDWF f , d	Réalise un ET logique entre (f) (8 bits) et (W), et place le résultat dans (destination)	Z	1

f : registre (spécial ou d'usage général)
d : registre de destination (on peut choisir entre le *registre de travail* **W** et le *registre* **f**).

IORWF f , d	Réalise un OU logique (inclusif) entre (f) (8 bits) et (W), et place le résultat dans (destination)	Z	1
XORWF f , d	Réalise un OU exclusif entre (f) (8 bits) et (W), et place le résultat dans (destination)	Z	1
COMF f , d	Réalise le complément logique de (f) (8 bits), et place le résultat dans (destination)	Z	1
DECF f , d	Décrémente (f) et place le résultat dans (destination). (f) - 1 -> (destination)	Z	1
DECFSZ f , d	Décrémente (f) et place le résultat dans (destination). Si le résultat est 0, alors l'instruction suivante est ignorée, et une instruction NOP est exécutée à la place (soit 2 cycles)	-	1 ou 2
INCF f , d	Incrémente (f) et place le résultat dans (destination). (f) + 1 -> (destination)	Z	1

INCFSZ f , d	Incrémente (f) et place le résultat dans (destination). Si le résultat est 0, alors l'instruction suivante est ignorée, et une instruction NOP est exécutée à la place (soit 2 cycles)	-	1 ou 2
CLRF f	Efface le contenu du registre (f). Remarque : le bit Z est donc mis à 1.	Z	1
CLRW	Efface le contenu de l'accumulateur (W). Remarque : le bit Z est donc mis à 1.	Z	1
RLF f , d	Réalise une rotation circulaire à gauche : Le résultat est placé dans (destination). 	C	1
RRF f , d	Réalise une rotation circulaire à droite : Le résultat est placé dans (destination). 	C	1
SWAPF f , d	Les 4 bits de poids forts et les 4 bits de poids faibles de (f) sont échangés. Le résultat est placé dans (destination).	-	1
NOP	Cette instruction ne fait rien (durée 1 cycle).	-	1

Modes d'adressages

On ne peut pas concevoir un programme qui ne manipule pas de données. Il existe trois grands types d'accès à une donnée ou modes d'adressage :

- Adressage immédiat : La donnée est contenue dans l'instruction.
- Exemple : movlw 0xC4 ; **Transfert la valeur 0xC4** dans W
- Adressage direct : La donnée est contenue dans un registre.
- Ce dernier peut être par un nom (par exemple W) ou une adresse mémoire.
- Exemple : movf 0x2B, 0 ; **Transfert dans W** la valeur **contenue à l'adresse 0x2B.**
- Adressage indirect : L'adresse de la donnée est contenue dans un pointeur.

```
Exemple : movlw   0x1A      ; Charge 1Ah dans W

          movwf   FSR       ; Charge W, contenant 1Ah, dans FSR

          movf    INDF, 0   ; Charge la valeur contenue à l'adresse 1Ah dans W
```

Ports d'entrées/Sorties

Port A

- Il comporte 5 pattes d'entrée/sortie bidirectionnelles, notées RAx avec x={0,1,2,3,4} sur le brochage du circuit. Le registre PORTA, d'adresse 05h dans la banque 0, permet d'y accéder en lecture ou en écriture. Le registre TRISA, d'adresse 85h dans la banque 1, permet de choisir le sens de chaque patte (entrée ou sortie) : un bit à 1 positionne le port en entrée, un bit à 0 positionne le port en sortie.
- La patte RA4 peut aussi servir d'entrée de comptage pour le timer0.

Ports d'entrées/Sorties..

Port B

- Il comporte 8 pattes d'entrée/sortie bidirectionnelles, notées RBx avec x={0,1,2,3,4,5,6,7} sur le brochage du circuit. Le registre PORTB, d'adresse 06h dans la banque 0, permet d'y accéder en lecture ou en écriture. Le registre TRISB, d'adresse 86h dans la banque 1, permet de choisir le sens de chaque patte (entrée ou sortie) : un bit à 1 positionne le port en entrée, un bit à 0 positionne le port en sortie.
- On peut noter la fonction particulière pilotée par le bit RBPU (OPTION_REG.7) qui permet d'alimenter (RBPU=0) ou non (RBPU=1) les sorties.
- Les quatre bits de poids fort (RB7-RB4) peuvent être utilisés pour déclencher une interruption sur changement d'état.
- RB0 peut aussi servir d'entrée d'interruption externe.

Registre TMR0 (Timer 0)

- Il s'agit d'un registre spécial situé à l'adresse 0x01 (banque 0) de la mémoire des données (Data RAM).
- **Ce registre contient un nombre de 8 bits (0 à 255 en numération décimale).**
- Le module TMR0 (Timer 0) possède deux modes de fonctionnement :

1- Le mode timer

2- Le mode compteur

Le mode timer

- Pour configurer le module TMR0 en mode timer, il faut au préalable que : **T0CS = 0** (bit 5 du registre **OPTION_REG**)
- Le contenu du registre TMR0 est alors incrémenté à chaque cycle de l'horloge interne. 1 cycle correspond à une durée de 1 µs pour un oscillateur à quartz de 4 MHz (1 cycle = 4 / F OSC).
- En toute rigueur, cela est vrai si le taux de prédivision (**prescaler**) est réglé à 1 (bit 3 du registre OPTION_REG = PSA = 1).
- Notez que si le contenu du registre TMR0 est 255 (0xFF), il passera à 0 (0x00) à la prochaine incrémentation.
- En conclusion, le mode timer est utilisé pour **mesurer des durées.**

Le mode compteur

- Pour configurer le module TMR0 en mode compteur, il faut au préalable que : **T0CS = 1** (bit 5 du registre **OPTION_REG**)
- Le contenu du registre TMR0 est alors incrémenté à chaque front du signal présent sur la broche RA4/T0CKI :
- front montant si T0SE = 0 (bit 4 du registre OPTION_REG)
- front descendant si T0SE = 1
- En toute rigueur, cela est vrai si le taux de prédivision (**prescaler**) est réglé à 1 (bit 3 du registre OPTION_REG = PSA = 1).
- En conclusion, le mode compteur est utilisé pour **faire du comptage.**

Interruption

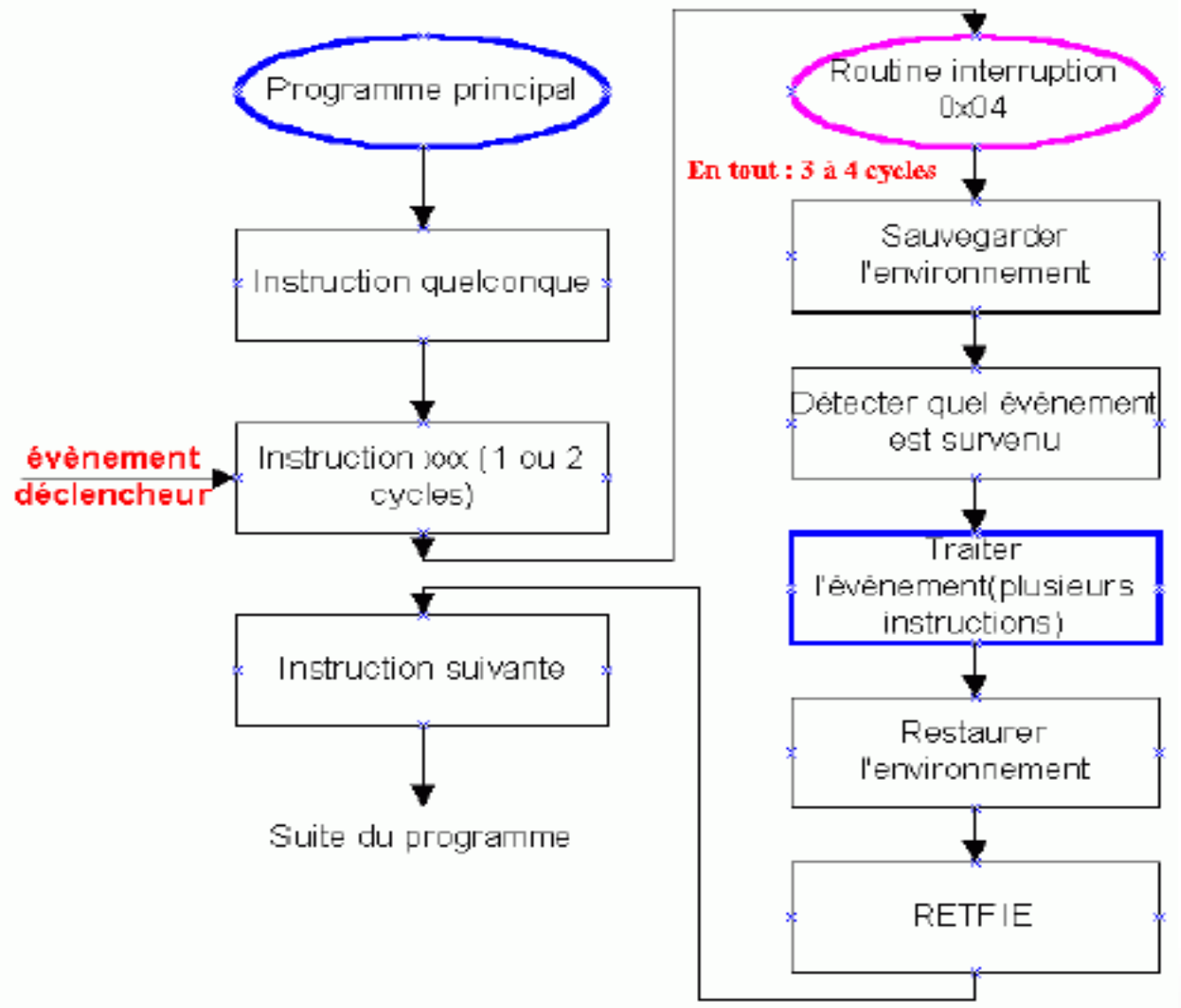
On peut activer une interruption quand le registre TMR0 déborde (passage de 0xFF à 0x00).

Les interruptions

Le PIC 16F84A dispose de 4 sources d'interruptions :

- Interruption sur la broche RB0/INT
- Interruption "RB" : sur changement du niveau logique d'au moins une de ces 4 broches : RB4, RB5, RB6 ou RB7 (port B)
- Interruption de débordement du registre TMR0 (H'FF' -> H'00')
- Interruption de fin d'écriture de l'EEPROM

Mécanisme d'interruption



Registre INTCON

Il s'agit d'un registre spécial situé à l'adresse H'0B' (banque 0) de la mémoire des données (Data RAM). Ce registre est également accessible en banque 1 (adresse H'8B').

	Nom	Description
bit 7	**GIE**	"Global Interrupt Enable" •bit à mettre à 0 pour désactiver **toutes** les interruptions •bit à mettre à 1 pour autoriser **toutes** les interruptions
bit 6	**EEIE**	"EEPROM write complete interrupt enable" •1 : **si GIE = 1**, autorise l'interruption de fin d'écriture de l'EEPROM •0 : désactive l'interruption de fin d'écriture de l'EEPROM
bit 5	**T0IE**	"TMR0 overflow interrupt enable" •1 : **si GIE = 1**, autorise l'interruption de débordement du registre TMR0 (H'FF' -> H'00') •0 : désactive l'interruption de débordement du registre TMR0

bit 4	**INTE**	"RB0/INT external interrupt enable" •1 : **si GIE = 1**, autorise l'interruption sur la broche RB0/INT •0 : désactive l'interruption sur la broche RB0/INT
bit 3	**RBIE**	"RB port change interrupt enable" •1 : **si GIE = 1**, autorise l'interruption sur les broches **RB4, RB5, RB6, RB**7 du port B (sur changement du niveau logique d'au moins une de ces broches)Attention, seules les broches configurées en entrée sont concernées •0 : désactive l'interruption "RB"
bit 2	**T0IF**	"TMR0 overflow interrupt flag" •drapeau (flag) mis à 1 lors du débordement du registre TMR0 (H'FF' -> H'00') •ce drapeau ne peut être effacé que de façon logicielle (**bcf INTCON, T0IF**)

bit 1	**INTF**	"RB0/INT external interrupt flag" •drapeau (flag) mis à 1 lors d'un front (montant ou descendant selon l'état du bit INTEDG du registre **OPTION_REG**) sur la broche RB0/INT •ce drapeau ne peut être effacé que de façon logicielle (**bcf INTCON, INTF**)
bit 0	**RBIF**	"RB port change interrupt flag" •drapeau (flag) mis à 1 lors d'un changement de niveau logique d'au moins une des broches : RB4, RB5, RB6 ou RB7 (cela ne concerne que les broches configurées en entrée) •ce drapeau ne peut être effacé que de façon logicielle (**bcf INTCON, RBIF**)

Remarque : le drapeau **EEIF** se trouve dans le registre spécial EECON1 (bit 4).

Bits (ou "fusibles") de configuration

On se place dans le cadre de l'outil de développement MPLAB de Microchip. La syntaxe utilisée est alors :

CP (Code Protection bit)	ON	Rend impossible la lecture de la mémoire de programme Flash et de l'EEPROM (à travers un programmateur). C'est une protection contre le piratage industriel.
	OFF	Lecture possible
PWRTE (Power-up Timer Enable bit)	ON	A la mise sous tension du µC, lance une temporisation d'environ 72 ms durant laquelle est effectué un RESET interne. Il est conseillé d'utiliser cette configuration.
	OFF	Temporisation désactivée
WDT (Watchdog Timer Enable bit)	ON	Active le watchdog (chien de garde)
	OFF	Désactive le watchdog

Bits (ou "fusibles") de configuration…

OSC (Oscillator Selection bits)	RC	Oscillateur de type Résistance / Condensateur. Remarques : économique, réservé aux applications où la précision de la base de temps n'est pas critique.
	HS	Oscillateur à quartz haute fréquence (4 MHz, 20 MHz ...).
	XT	Oscillateur à quartz ou à résonateur céramique
	LP	Oscillateur à quartz de faible puissance (32,768 kHz ...)

Exemple de configuration

Dans le code source (fichier avec extension .asm), les fusibles de configuration sont indiqués au compilateur avec la directive suivante :

__config _CP_OFF & _WDT_OFF & _PWRTE_ON & _HS_OSC

Remarque

Les bits de configuration ne sont pas modifiables.

Dans l'exemple ci-dessus, pour activer le watchdog et utiliser un oscillateur de type RC, il faut modifier le code source, recompiler et reprogrammer le µC ...

__config _CP_OFF & _WDT_ON & _PWRTE_ON & _RC_OSC