

2012

TOA_01

Document technique des options avancés

IFT592 - Projet informatique - Microcontrôleur programmable PIC18F4550

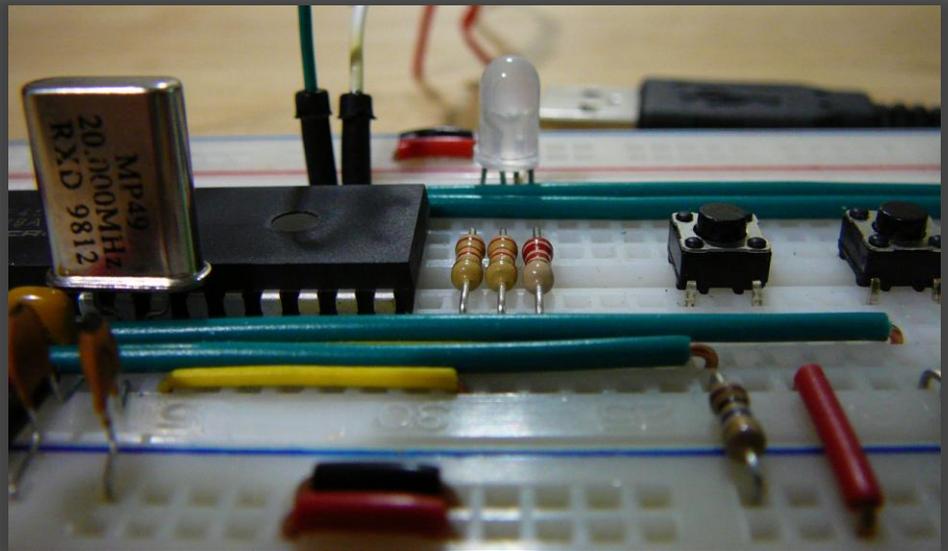


Table des matières

Données de publication.....	iv
Historique des révisions	iv
Sommaire	iv
Caractéristiques du microcontrôleur	1
Architecture de la mémoire	2
Mémoire d'instructions.....	2
Compteur ordinal	2
Pile d'adresses	3
Mémoire de données RAM	4
Les cycles d'instructions	6
Les ports d'entrée/sortie.....	7
Diagramme des ports	8
Description détaillée des ports.....	9
Port A.....	9
Port B.....	10
Port C.....	11
Port D.....	12
Port E	13
Trois broches particulières	14
Langage d'assemblage.....	15
La syntaxe en mémoire (mot d'instruction).....	15
Syntaxe des opérations « Byte-oriented »	15
Syntaxe des opérations « Bit-oriented »	16
Syntaxe des opérations « Literal »	16
Syntaxe des opérations « Control ».....	16
Jeu complet d'instructions standard	17
Lexique	17
Les opérations « Byte-oriented »	18
Les opérations « Bit-oriented »	18

Les opérations « Literal »	19
Les opérations « Control ».....	19
Configuration.....	20
Fréquence d'horloge	20
Sélection de l'horloge USB (seulement avec le mode USB « Full-Speed »)	22
Sélection du type d'oscillateur	22
Fail-Safe Clock Monitor	23
Le « Démarrage à deux vitesses »	23
Power-up Timer (PWRT).....	23
Brown-out Reset (BOR)	23
Régulateur de voltage USB.....	24
Watchdog Timer (WDT).....	24
« Postscale » du Watchdog Timer	24
Configuration A/D de « PORTB ».....	25
Low-Power Timer 1 Oscillator Enable bit:	25
Activation de la broche « MCLR »	25
Pile des retours d'appel.....	25
Extended Instruction Set Enable bit	Erreur ! Signet non défini.
Protection du code	25
Protection du code pour le « Boot Block ».....	26
Protection du code pour la mémoire programme	26
Accès en lecture pour le « Boot Block »	26
Accès en lecture pour la mémoire programme.....	27
Accès en écriture pour le « Boot Block ».....	27
Accès en écriture pour la mémoire programme	27
Protection du registre de configuration.....	27
La réinitialisation (reset).....	28
Les registres RCON et STKPTR	28
RCON (Reset Control register).....	28
STKPTR	28
Tableau des valeurs après réinitialisation	28
Les registres du SFR (Special Function Registers).....	29

La gestion des événements et des interruptions	34
INTCON	35
INTCON2	36
INTCON3	36
PEI	37
PEI2	37
IPR	38
IPR2	38
PIR	39
PIR2	39
Les entrées analogiques	40
ADCON0	40
ADCON1	41
ADCON2	42

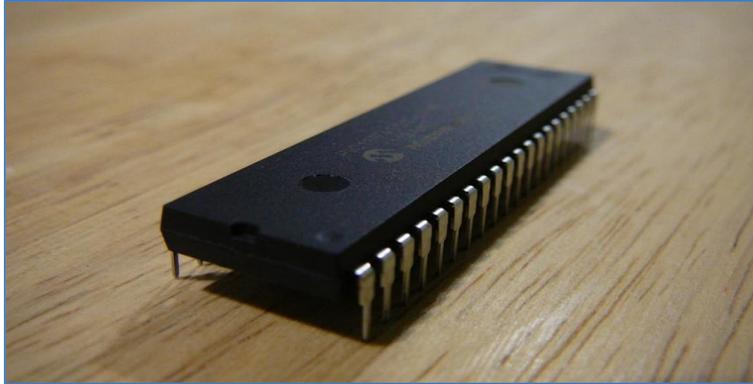
Données de publication

Historique des révisions

version	date	auteur	description
0.1.0	2012-02-06	PB	Caractéristiques du microcontrôleur Architecture de la mémoire Les cycles d'instructions Les ports d'entrée/sortie Langage d'assemblage
0.1.1	2012-02-12	PB	Modifications mineures des quelques phrases.
0.2.0	2012-02-20	PB	Configuration
0.3.0	2012-03-05	PB	La réinitialisation
0.3.1	2012-03-20	PB	Corrections suite aux recommandations de Richard St-Denis.
0.4.0	2012-04-22	PB	Les entrées analogiques

Sommaire

Ce document apporte les détails techniques de toutes les options du microcontrôleur PIC18F4550. Les informations qu'il contient couvrent également les sujets traités dans le document tutoriel, mais sans vulgarisation, ni explication pour les débutants. Il s'adresse à des développeurs expérimentés.



Caractéristiques du microcontrôleur

Fréquence d'opération	DC – 48 MHz
Taille de la mémoire d'instructions (octets)	32768
Taille de la mémoire d'instructions (instructions)	16384
Taille de la mémoire des données (octets)	2048
Taille de la mémoire des données EEPROM (octets)	256
Sources d'interruption	20
Ports E/S (entrée/sortie)	Ports A, B, C, D, E
Nombre de chronomètres (timers)	4
Nombre de modules CCP ¹	1
Nombre de modules ECCP ²	1
Communication série	MSSP, EUSART
Nombre de modules USB	1
Streaming Parallel Port (SPP)	Oui
Module de conversion 10-Bit (Analog-to-Digital)	13 canaux d'entrée
Nombre de modules comparateur	2
Réinitialisation (et délais)	POR, BOR, RESET, Stack Full, Stack Underflow (PWRT, OST), MCLR (optional), WDT
Détection programmable du bas voltage	Oui
Réinitialisation de type « Brown-out » programmable	Oui
Jeu d'instructions	75 instructions; 83 si le jeu d'instructions étendu est activé.
Types de boîtier disponible pour le microcontrôleur.	40-Pin PDIP 44-Pin QFN 44-Pin TQFP

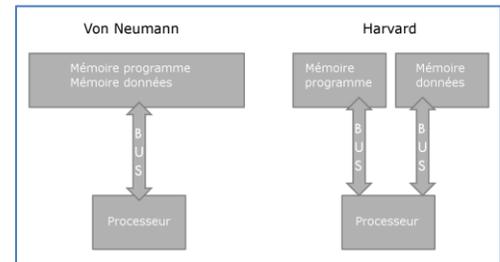
¹ CCP : Capture/Compare/PWM (Pulse-Width Modulation)

² ECCP : Enhanced Capture/Compare/PWM (Pulse-Width Modulation)

Architecture de la mémoire

Il y a trois types de mémoire dans le PIC18F4550.

- Mémoire d'instructions
- Mémoire de données RAM
- Mémoire de données EEPROM



Étant donné que le périphérique possède une architecture Harvard, les mémoires d'instructions et de données utilisent des bus séparés. Ceci permet un accès concurrent aux deux espaces de mémoire. La mémoire EEPROM quant à elle est adressée, et accessible, via un ensemble de registres de contrôle. Elle ne sera pas vue tout de suite.

Mémoire d'instructions

Le PIC18F4550 implémente un compteur ordinal (PC <20:0> dans le diagramme ci-contre) sur 21 bits, lequel est en mesure d'adresser sur 2 Mo. Cela dit, le PIC18 n'implémente que 32 Ko de mémoire pour le programme, soit de 0000h à 7FFFh. Au-delà de cette limite physique (de 8000h à 1FFFFFFh), tout accès à la mémoire retourne la valeur '0' (une instruction NOP).

Le PIC18F4550 possède également un vecteur de réinitialisation, à l'adresse 0000h, puis deux vecteurs d'interruption, soit le vecteur de haute priorité à l'adresse 0008h et le vecteur de basse priorité à l'adresse 0018h.

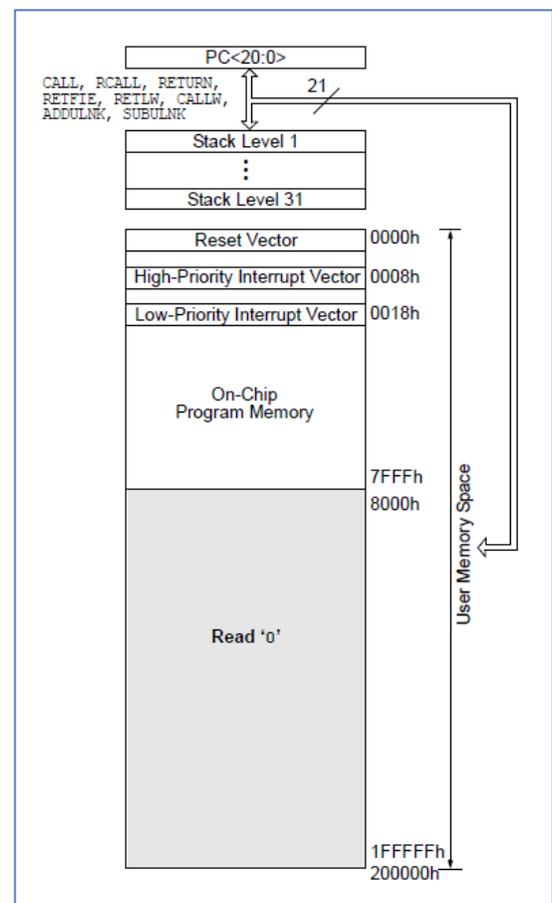
Compteur ordinal

Le compteur ordinal spécifie l'adresse de la prochaine instruction à exécuter. Il est formé de 21 bits qui se divisent sur trois registres séparés de 8 bits chacun.

Le premier octet (bits <7:0>) est dit « octet bas » et est connu sous le nom de registre PCL (Program Counter Low). Ce dernier est accessible en lecture et en écriture.

Le second octet (bits <15:8>) est dit « octet haut » et est connu sous le nom de registre PCH (Program Counter High). Ce dernier n'est pas accessible directement en lecture, ni en écriture. Les modifications à apporter à ce registre sont effectuées via le registre PCLATH.

Le troisième octet (bits <20:16>) est dit « octet supérieur » et est connu sous le nom de registre PCU (Program Counter Upper). Ce dernier n'est également pas accessible directement en lecture, ni en écriture. Les modifications à apporter à ce registre sont effectuées via le registre PCLATU.



Les contenus de PCLATH et PCLATU sont transférés automatiquement au compteur ordinal par toute instruction qui écrit dans le registre PCL. De la même manière, les contenus de PCH et PCU sont transférés automatiquement dans les registres PCLATH et PCLATU par toute instruction qui lit dans le registre PCL.

Note : Les instructions de branchement CALL, RCALL et GOTO écrivent directement dans le compteur ordinal. Pour ces instructions, les contenus de PCLATH et PCLATU ne sont pas transférés dans le compteur ordinal.

Les instructions sur lesquelles pointe le compteur ordinal sont formées de 16 bits, soit deux octets (un mot). Afin que le compteur ordinal soit toujours aligné sur une frontière d'instruction, la valeur du bit le moins significatif du registre PCL est fixée à '0'. Ainsi, le compteur ordinal incrémente toujours de 2 pour séquencer les instructions dans la mémoire d'instructions.

Pile d'adresses

Lorsque les instructions CALL ou RCALL sont exécutées, ou lorsqu'une interruption est lancée, l'adresse contenue dans le compteur ordinal est mise (PUSH) sur la pile d'adresses. Cette dernière permet 31 combinaisons de branchements et d'interruptions. La valeur du compteur ordinal est récupérée (POP) de la pile d'adresses lors de l'exécution des instructions RETURN, RETLW ou RETFIE. Les registres PCLATH et PCLATU ne sont aucunement affectés par les instructions RETURN et CALL.

La pile d'adresses est constituée de 31 niveaux de 3 octets chacun, pour enregistrer des adresses de 21 bits, ainsi que du registre STKPTR, accessible en lecture et en écriture. Ce registre contient un pointeur de pile de 5 bits (bits <4:0>), un drapeau (flag) pour le débordement vers le bas (underflow) et un autre drapeau pour indiquer que la pile est pleine. Ces deux drapeaux sont respectivement les bits 6 et 7. Le bit 5 de ce registre n'est pas implémenté. Les bits 6 et 7 sont automatiquement mis à '1' lorsque la pile atteint ses limites, mais ne sont pas remis à '0' automatiquement. Seuls un POR (Power-on Reset) ou le programme de l'utilisateur peuvent les remettre à '0'.

La pile d'adresses ne fait partie, ni de l'espace mémoire d'instructions, ni de l'espace mémoire des données. Elle possède son propre espace mémoire. Elle est accessible en lecture et en écriture, et ce, directement via trois registres du SFR (Special Function Registers), soit :

- TOSL → Top Of Stack Low (bits <7:0>)
- TOSH → Top Of Stack High (bits <15:8>)
- TOSU → Top Of Stack Upper (bits <20:16>)

Seul le dessus de la pile est accessible en lecture et en écriture. Il est conseillé de désactiver les interruptions lors des accès à la pile d'adresses afin de prévenir toute corruption de la pile. Pour mettre une valeur ailleurs que sur le dessus de la pile, il faut d'abord modifier le pointeur de pile dans le registre STKPTR, puis utiliser les registres TOS. Ne pas oublier de remettre le pointeur de pile dans son état initial. Lors d'une réinitialisation, le pointeur de pile prend la valeur '00000'. Lorsqu'on ajoute sur la pile, le pointeur est d'abord incrémenté, ensuite les données sont écrites sur la pile. Lorsqu'on retire de la pile, les données sont d'abord lues, ensuite le pointeur est décrémenté.

Mémoire de données RAM

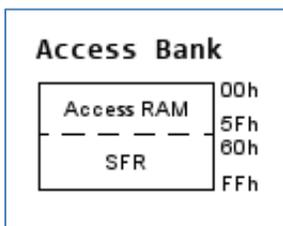
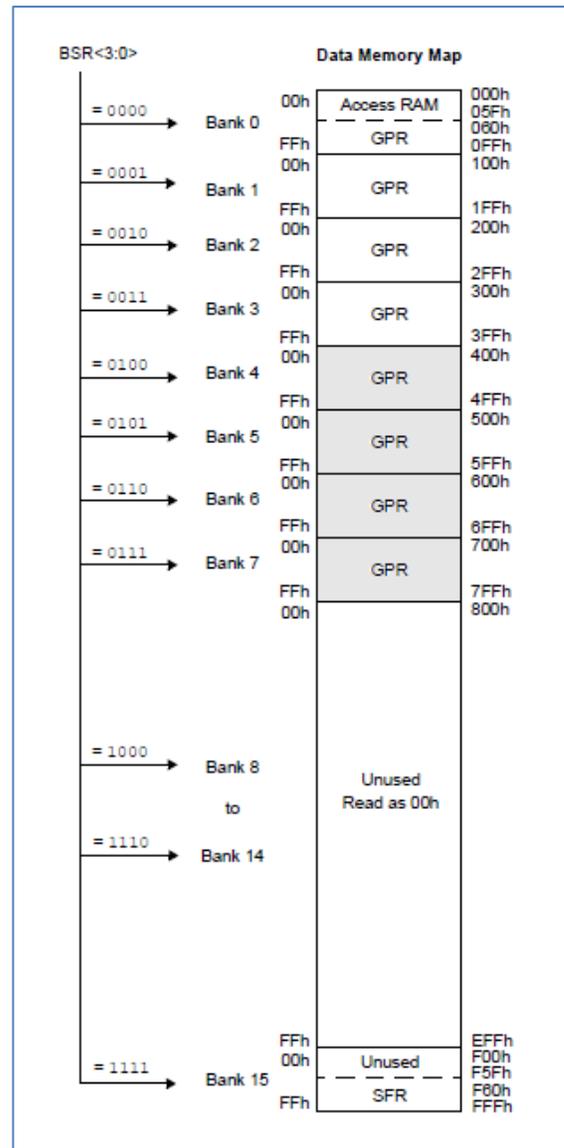
La mémoire de données du PIC18 est implémentée en tant que RAM statique. Chaque registre dans la mémoire de données est adressé sur 12 bits, ce qui offre une possibilité de 4096 octets de mémoire (000h à FFFh). Ces 4096 octets sont divisés en 16 banques, contenant chacune 256 octets (00h à FFh). Cela dit, le PIC18 n'implémente que 8 de ces 16 banques. Ce qui nous donne accès à un total de 2048 octets pour la mémoire des données.

La division de la mémoire en 16 banques permet de contourner le problème que posent les instructions qui n'acceptent que 8 bits pour une adresse de mémoire. Les quatre bits les plus significatifs de l'adresse à laquelle on veut accéder seront donc écrits dans le registre BSR (Bank Select Register), qui correspond au choix de la banque.

Note : Suite à une réinitialisation, le registre BSR prend la valeur « ----0000 ». Suite à un réveil après une mise en veille, que ce soit par le WDT (WatchDog Timer) ou par une interruption, le registre BSR conserve la dernière valeur enregistrée.

La mémoire de données contient des registres de type SFR (Special Function Registers) et de type GPR (General Purpose Registers). Les SFR sont utilisés pour le contrôle et le statut du contrôleur et des fonctions périphériques, tandis que les GPR sont utilisés pour le stockage de données et les opérations de l'application de l'utilisateur.

Dans le diagramme ci-contre, on remarque qu'il y a des banques GPR qui sont grisées et d'autres qui sont blanches. Les blanches sont disponibles en tout temps pour l'utilisateur. Les grisées sont disponibles uniquement si le module USB est désactivé. Lorsque ce module est activé, les banques grisées deviennent une zone tampon pour les communications USB.



Il existe une autre banque qui s'appelle *Access Bank*. Cette banque permet un accès direct par les instructions aux espaces mémoire *Access RAM* (000h à 05Fh) et SFR (F60h à FFFh), et ce, sans utiliser le BSR. Ceci permet d'économiser au moins un cycle par accès à la mémoire, étant donné qu'on n'a pas à écrire, ni à lire la valeur du BSR. L'utilisation de l'*Access Bank* est idéale pour un accès rapide aux données, ou encore pour les variables communes du programme qui sont fréquemment utilisées, tel un

compteur dans une boucle qui effectue un grand nombre d'itérations. Bien que les deux espaces mémoire concernés soient disjoints dans la mémoire physique, ils sont contigus dans l'*Access Bank* (00h à FFh).

Pour définir si on accède à un espace de mémoire avec une adresse sur 12 bits (BSR) ou sur 8 bits (*Access bank*), il faut se reporter à l'utilisation du paramètre optionnel « a » des instructions qui accèdent directement à la mémoire. Pour une valeur $a = 0$, l'instruction utilisera l'*Access Bank* et on ignorera le BSR. Pour une valeur $a = 1$, l'instruction utilisera le BSR. La valeur par défaut est $a = 1$.

Voici un exemple avec l'instruction ADDWF, donc la syntaxe est : « ADDWF f {,d {,a}} ». Cette instruction additionne la valeur située dans le registre de travail (appelé « w » pour Work directory) et une valeur située dans un espace mémoire (paramètre « f »). Les autres paramètres pour ADDWF sont optionnels . Il y a d'abord « a » que l'on a vu précédemment, puis il y a « d », le paramètre qui identifie la destination du résultat. Avec $d = 0$, le résultat est mis dans le registre de travail (« w »), et avec $d = 1$, le résultat est mis dans l'espace mémoire (« f »). Par défaut, $d = 1$.

Donc, pour additionner le contenu de « w » avec le contenu de l'espace mémoire situé à l'adresse « 03Ah » et mettre le résultat dans « w ». Il faudrait écrire « ADDWF H'3A', 0, 0 » pour utiliser l'*Access bank*. Pour utiliser le BSR, il faudrait d'abord mettre la valeur « 0h » dans le BSR, puis utiliser « ADDWF H'3A', 0, 1 » pour effectuer l'addition. Ce qui nous ferait un total de trois cycles, soit 1 pour écrire dans le BSR, 1 pour que ADDWF lise le BSR, puis 1 pour effectuer l'addition, alors qu'avec l'utilisation de l'*Access bank*, un seul cycle a été nécessaire.

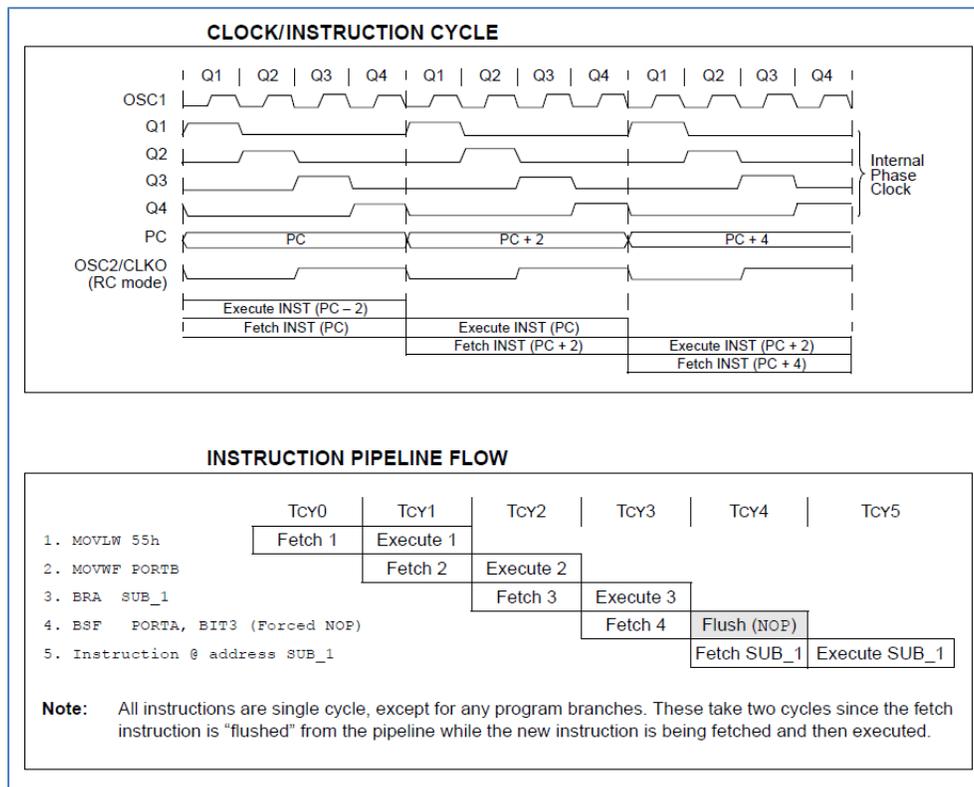
Rappel : La nomenclature utilisée pour définir la base d'un littéral est d'écrire d'abord la lettre correspondante à la base en majuscule, puis la valeur littérale entre apostrophes.

Les cycles d'instructions

La fréquence de l'horloge à l'entrée du microcontrôleur, que ce soit une horloge interne ou externe, est divisée par 4, afin de générer quatre quarts de cycle qui ne se chevauchent pas (nommés Q1, Q2, Q3 et Q4 dans le diagramme ci-dessous). Le compteur ordinal est incrémenté à chaque Q1. Une instruction est récupérée depuis la mémoire d'instructions et verrouillée dans le registre d'instruction durant Q4. Une instruction est décodée et exécutée entre Q1 et Q4 du cycle suivant. La mémoire de données est lue lors de Q2 (lecture de l'opérande) et écrite lors de Q4 (écriture dans la destination). Cette séquence peut être observée dans le diagramme au bas de la page.

Bien qu'on dise qu'une instruction nécessite un cycle d'horloge (4 tics, soit de Q1 à Q4 inclusivement), ce n'est pas tout à fait exact dans la réalité. Pour exécuter une instruction, il faut d'abord prendre un cycle pour la lire, puis un cycle pour l'exécuter. Pour pallier cette perte de temps, le PIC18F4550 utilise un pipeline. Ce qui signifie que pendant qu'une instruction s'exécute, la suivante est en train de se charger. Il faut le voir comme s'il s'agissait de deux acheminements en parallèle vers le processeur. Ainsi, au tout premier cycle d'instruction, lors du démarrage du programme contenu dans le PIC18, seul le chargement de la première instruction sera exécutée. Pour les cycles suivants, une exécution et un chargement seront exécutés en même temps. C'est de cette manière qu'on s'assure qu'il y a une instruction qui s'exécute à chaque cycle.

Dans le diagramme ci-dessous, PC représente le compteur ordinal, TcyX représente le compteur de tics, OSC1 et OSC2 représentent les deux broches de l'oscillateur et FETCH est la lecture d'une instruction.



Les ports d'entrée/sortie

Le diagramme ci-dessous montre les différents rôles que peuvent prendre chacune des broches du PIC18F4550. Nous pouvons voir que les broches appartenant à un port ne sont pas nécessairement regroupées. Voyez les broches appartenant au port D par exemple (RD0 à RD7).

Remarquez la demi-lune en haut du microcontrôleur, ainsi que le petit point au dessus du « 1 ». Ceci permet d'identifier physiquement la broche numéro 1 sur votre PIC. Les flèches quant à elles indiquent si une broche est bidirectionnelle, ou le sens de la communication dans le cas d'une broche unidirectionnelle. V_{DD} et V_{SS} indiquent respectivement les bornes positives et négatives de l'alimentation. OSC1 et OSC2 servent pour l'oscillateur et MCLR sert à la réinitialisation du microcontrôleur. Notez que chacun des ports, de même que les rôles que peuvent prendre ses broches, sera détaillé dans les sections suivantes.

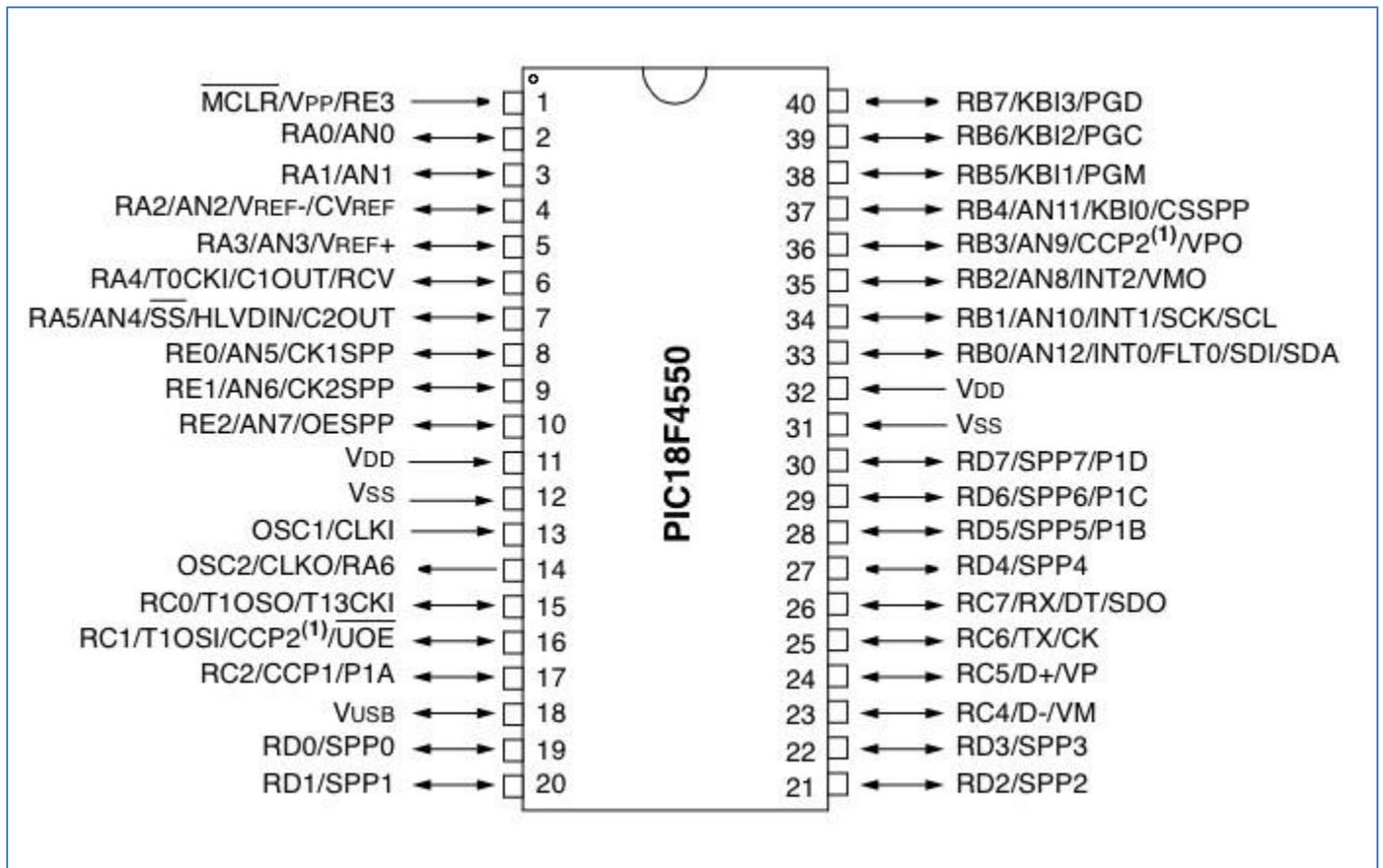
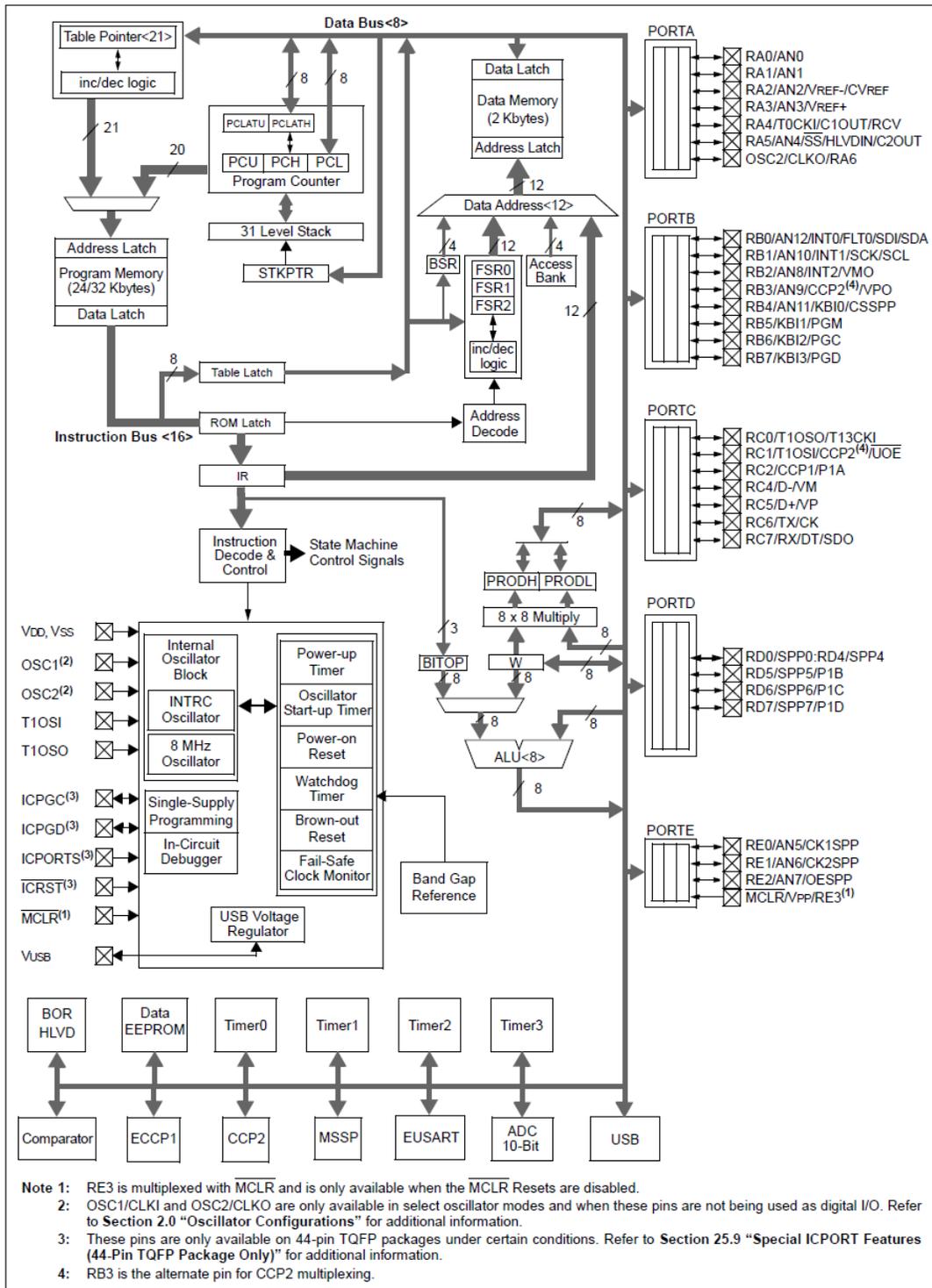


Diagramme des ports

Le diagramme ci-dessous montre les relations entre les différents ports et les composants internes du microcontrôleur.



Langage d'assemblage

Le coeur du PIC18F4550 incorpore un jeu de 75 instructions standards, mais aussi un jeu de 8 instructions étendues pour l'optimisation du code récursif ou l'utilisation d'une pile dans le programme. Nous parlerons plus tard de l'ensemble d'instructions étendues.

La plupart des instructions sont codées sur 16 bits, soit un mot. Elles nécessitent donc deux octets en mémoire. Ce sont les instructions simples. Par contre, il y a quatre instructions qui nécessitent deux mots en mémoire. Ces instructions sont MOVFF, CALL, GOTO et LFSR. Elles sont les instructions doubles.

Les instructions doubles, nécessitent deux cycles d'instruction pour s'exécuter. À noter que les quatre bits les plus significatifs du deuxième mot ont toujours la valeur '1111'. Ainsi, si on modifie le compteur ordinal pour passer par dessus le premier mot et accéder directement au deuxième mot, un NOP sera exécuté.

La syntaxe en mémoire (mot d'instruction)

Chaque instruction est divisée en un opcode (code d'opération), qui spécifie le type d'instruction, et d'un ou plusieurs opérandes, qui précisent le fonctionnement de l'instruction.

Le jeu d'instructions est divisé en quatre catégories de base, soit :

- Les opérations orientées octet (Byte-oriented)
- Les opérations orientées bit (Bit-oriented)
- Les opérations littérales
- Les opérations de contrôle

Syntaxe des opérations « Byte-oriented »

Les opérations « orientées octets » ont trois opérandes, soit :

1. Le registre de fichier (spécifié par la lettre 'f')
2. La destination du résultat (spécifiée par la lettre 'd')
3. Le type d'accès mémoire (spécifié par la lettre 'a')

Le registre de fichier, spécifié par la lettre 'f', désigne un espace mémoire qui contient la valeur sur laquelle s'appliquera l'instruction. La destination du résultat, spécifiée par la lettre 'd', détermine l'endroit où sera enregistré le résultat de l'instruction. Si 'd' est égale à 0, le résultat sera mis dans le registre de travail WREG (aussi appelé 'w'). Si 'd' est égale à 1, le résultat sera mis dans le registre de fichier d'origine, spécifié précédemment par la lettre 'f'.

Opération sur un registre de fichier (1 mot)	Opération de déplacement (2 mots)																				
<table border="1"><tr><td>15</td><td>10</td><td>9</td><td>8</td><td>7</td><td>0</td></tr><tr><td colspan="2">opcode</td><td>d</td><td>a</td><td colspan="2">f</td></tr></table>	15	10	9	8	7	0	opcode		d	a	f		<table border="1"><tr><td>15</td><td>12</td><td>11</td><td>0</td></tr><tr><td colspan="2">opcode</td><td colspan="2">f (source)</td></tr></table>	15	12	11	0	opcode		f (source)	
15	10	9	8	7	0																
opcode		d	a	f																	
15	12	11	0																		
opcode		f (source)																			
<p>d = 0, pour mettre le résultat dans le registre WREG. d = 1, pour mettre le résultat dans le registre 'f'. a = 0, pour forcer l'utilisation de l'Access bank. a = 1, pour choisir la banque via le BSR. f = l'adresse d'un registre de fichier sur 8 bits.</p>	<table border="1"><tr><td>15</td><td>12</td><td>11</td><td>0</td></tr><tr><td colspan="2">1111</td><td colspan="2">f (destination)</td></tr></table> <p>f = l'adresse d'un registre de fichier sur 12 bits.</p>	15	12	11	0	1111		f (destination)													
15	12	11	0																		
1111		f (destination)																			

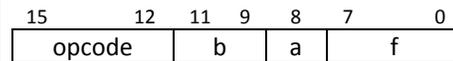
Syntaxe des opérations « Bit-oriented »

Les opérations « orientées bit » ont trois opérandes, soit :

1. Le registre de fichier (spécifié par la lettre 'f')
2. Le désignateur de bit (spécifié par la lettre 'b')
3. Le type d'accès mémoire (spécifié par la lettre 'a')

Le désignateur de bit, spécifié par la lettre 'b', définit le numéro du bit affecté par l'opération, alors que le registre de fichier, spécifié par la lettre 'f', contient l'adresse de l'octet qui contient le bit 'b'.

Opération sur un registre de fichier (1 mot)



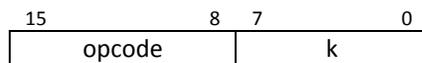
b = position du bit (sur 3 bits) dans registre de fichier 'f'.
a = 0, pour forcer l'utilisation de l'Access bank.
a = 1, pour choisir la banque via le BSR.
f = l'adresse d'un registre de fichier sur 8 bits.

Syntaxe des opérations « Literal »

Les opérations littérales peuvent utiliser un des opérandes suivants;

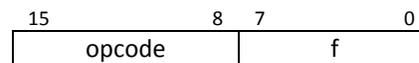
1. Une valeur littérale à charger dans un registre de fichier (spécifié par la lettre 'k')
2. Un registre FSR dans lequel charger une valeur littérale (spécifié par la lettre 'f')
3. Sans opérande (spécifié par '—')

Opération sur une valeur immédiate (1 mot)



k = une valeur immédiate sur 8 bits.

Opération sur un registre de fichier (1 mot)



f = l'adresse d'un registre de fichier sur 8 bits.

Syntaxe des opérations « Control »

Les opérations de contrôle peuvent utiliser un des opérandes suivants;

1. Une adresse dans la mémoire d'instructions (spécifiée par la lettre 'n')
2. Le mode des instructions CALL et RETURN (spécifié par la lettre 's')
3. Le mode des instructions de lecture et écriture de table (spécifié par la lettre 'm')
4. Sans opérande (spécifié par '—')

<p>Instruction BRA (1 mot)</p> <p>15 11 10 0</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">opcode</td> <td style="width: 50%;">n<10:0></td> </tr> </table> <p>n = valeur immédiate sur 10 bits</p>	opcode	n<10:0>	<p>Instruction GOTO (2 mots)</p> <p>15 8 7 0</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">opcode</td> <td style="width: 50%;">n<7:0></td> </tr> </table> <p>15 12 11 0</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">1111</td> <td style="width: 50%;">n<19:8></td> </tr> </table> <p>n = valeur immédiate sur 20 bits (séparés en deux)</p>	opcode	n<7:0>	1111	n<19:8>	
opcode	n<10:0>							
opcode	n<7:0>							
1111	n<19:8>							
<p>Instruction BC (1 mot)</p> <p>15 8 7 0</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">opcode</td> <td style="width: 50%;">n<7:0></td> </tr> </table> <p>n = valeur immédiate sur 7 bits</p>	opcode	n<7:0>	<p>Instruction CALL (2 mots)</p> <p>15 9 8 7 0</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 33%;">opcode</td> <td style="width: 33%;">S</td> <td style="width: 34%;">n<7:0></td> </tr> </table> <p>15 12 11 0</p> <table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">1111</td> <td style="width: 50%;">n<19:8></td> </tr> </table> <p>n = valeur immédiate sur 20 bits (séparés en deux) S = bit rapide</p>	opcode	S	n<7:0>	1111	n<19:8>
opcode	n<7:0>							
opcode	S	n<7:0>						
1111	n<19:8>							

Jeu complet d'instructions standard

Lexique

Champ	Description
a	Bit d'accès à la RAM. a = 0 : La RAM est accédée via l' <i>Acess bank</i> (le BSR est ignoré). a = 1 : La banque de la RAM est accédée par le registre BSR.
BSR	Bank Select Register. Utilisé pour sélectionner la banque courante de la RAM.
b	Adresse du bit à l'intérieur d'un registre de fichier (0 à 7).
C	Les bits d'état du registre d'état. C arry ---> Report
DC	D igit C arry ---> Bit de retenue
Z	Z ero ---> Zéro
OV	O verflow ---> Débordement
N	N egative ---> Négatif
d	Bit de sélection de destination. d = 0 : Le résultat est enregistré dans le registre de travail WREG. d = 1 : Le résultat est enregistré dans le registre de fichier 'f'.
f	Adresse du registre de fichier sur 8 bits (00h à FFh)
f _s	Adresse du registre de fichier sur 8 bits (000h à FFFh). s = source.
f _d	Adresse du registre de fichier sur 8 bits (000h à FFFh). d = destination.
k	Champ littéral, donnée constante ou étiquette (valeur sur 8, 12 ou 20 bits).
TO	Bit du « Time-out »
PD	Bit du « Power-down »

Les opérations « Byte-oriented »

Mnémoniques, Opérandes	Description	Cycles	Mot d'instruction				États affectés
ADDWF	f, d, a	1	0010	01da	ffff	ffff	C, DC, Z, OV, N
ADDWFC	f, d, a	1	0010	00da	ffff	ffff	C, DC, Z, OV, N
ANDWF	f, d, a	1	0001	01da	ffff	ffff	Z, N
CLRF	f, a	1	0110	101a	ffff	ffff	Z
COMF	f, d, a	1	0001	11da	ffff	ffff	Z, N
CPFSEQ	f, a	1 (2 ou 3)	0110	001a	ffff	ffff	Aucun
CPFSGT	f, a	1 (2 ou 3)	0110	010a	ffff	ffff	Aucun
CPFSLT	f, a	1 (2 ou 3)	0110	000a	ffff	ffff	Aucun
DECF	f, d, a	1	0000	01da	ffff	ffff	C, DC, Z, OV, N
DECFSZ	f, d, a	1 (2 ou 3)	0010	11da	ffff	ffff	Aucun
DCFSNZ	f, d, a	1 (2 ou 3)	0100	11da	ffff	ffff	Aucun
INCF	f, d, a	1	0010	10da	ffff	ffff	C, DC, Z, OV, N
INCFSZ	f, d, a	1 (2 ou 3)	0011	11da	ffff	ffff	Aucun
INFSNZ	f, d, a	1 (2 ou 3)	0100	10da	ffff	ffff	Aucun
IORWF	f, d, a	1	0001	00da	ffff	ffff	Z, N
MOVF	f, d, a	1	0101	00da	ffff	ffff	Z, N
MOVFF	f _s , f _d	2	1100	ffff	ffff	ffff	Aucun
			1111	ffff	ffff	ffff	
MOVWF	f, a	1	0110	111a	ffff	ffff	Aucun
MULWF	f, a	1	0000	001a	ffff	ffff	Aucun
NEGF	f, a	1	0110	110a	ffff	ffff	C, DC, Z, OV, N
RLCF	f, d, a	1	0011	01da	ffff	ffff	C, Z, N
RLNCF	f, d, a	1	0100	01da	ffff	ffff	Z, N
RRCF	f, d, a	1	0011	00da	ffff	ffff	C, Z, N
RRNCF	f, d, a	1	0100	00da	ffff	ffff	Z, N
SETF	f, a	1	0110	100a	ffff	ffff	Aucun
SUBFWB	f, d, a	1	0101	01da	ffff	ffff	C, DC, Z, OV, N
SUBWF	f, d, a	1	0101	11da	ffff	ffff	C, DC, Z, OV, N
SUBWFB	f, d, a	1	0101	10da	ffff	ffff	C, DC, Z, OV, N
SWAPF	f, d, a	1	0011	10da	ffff	ffff	Aucun
TSTFSZ	f, a	1 (2 ou 3)	0110	011a	ffff	ffff	Aucun
XORWF	f, d, a	1	0001	10da	ffff	ffff	Z, N

Les opérations « Bit-oriented »

Mnémoniques, Opérandes	Description	Cycles	Mot d'instruction				États affectés
BCF	f, b, a	1	1001	bbba	ffff	ffff	Aucun
BSF	f, b, a	1	1000	bbba	ffff	ffff	Aucun
BTFSZ	f, b, a	1 (2 or 3)	1011	bbba	ffff	ffff	Aucun
BTSS	f, b, a	1 (2 or 3)	1010	bbba	ffff	ffff	Aucun
BTG	f, d, a	1	0111	bbba	ffff	ffff	Aucun

Les opérations « Literal »

Mnémoniques, Opérandes		Description	Cycles	Mot d'instruction				États affectés
ADDLW	K	Additionne k et WREG	1	0000	1111	kkkk	kkkk	C, DC, Z, OV, N
ANDLW	K	ET logique entre k et WREG	1	0000	1011	kkkk	kkkk	Z, N
IORLW	k	OU logique inclusif entre k et WREG	1	0000	1001	kkkk	kkkk	Z, N
LFSR	f, k	Déplace k (12-bit) dans le registre FSR(f)	2	1110	1110	00ff	kkkk	Aucun
				1111	0000	kkkk	kkkk	
MOVLB	k	Déplace k dans BSR<3:0>	1	0000	0001	0000	kkkk	Aucun
MOVLW	k	Déplace k dans le registre WREG	1	0000	1110	kkkk	kkkk	Aucun
MULLW	k	Multiplie k avec WREG	1	0000	1101	kkkk	kkkk	Aucun
RETLW	k	RETURN en mettant k dans WREG	2	0000	1100	kkkk	kkkk	Aucun
SUBLW	k	Soustrait WREG de k	1	0000	1000	kkkk	kkkk	C, DC, Z, OV, N
XORLW	k	OU logique exclusif entre k et WREG	1	0000	1010	kkkk	kkkk	Z, N

Les opérations « Control »

Mnémoniques, Opérandes		Description	Cycles	Mot d'instruction				États affectés
BC	n	Branchement si Carry == 1	1 (2)	1110	0010	nnnn	nnnn	Aucun
BN	n	Branchement si Negative == 1	1 (2)	1110	0110	nnnn	nnnn	Aucun
BNC	n	Branchement si Carry == 0	1 (2)	1110	0011	nnnn	nnnn	Aucun
BNN	n	Branchement si Negative == 0	1 (2)	1110	0111	nnnn	nnnn	Aucun
BNOV	n	Branchement si Overflow == 0	1 (2)	1110	0101	nnnn	nnnn	Aucun
BNZ	n	Branchement si Zero == 0	1 (2)	1110	0001	nnnn	nnnn	Aucun
BOV	n	Branchement si Overflow == 1	1 (2)	1110	0100	nnnn	nnnn	Aucun
BRA	n	Branchement inconditionnel	2	1101	0nnn	nnnn	nnnn	Aucun
BZ	n	Branchement si Zero == 1	1 (2)	1110	0000	nnnn	nnnn	Aucun
CALL	n, s	Appeler une sous-routine	2	1110	110s	kkkk	kkkk	Aucun
				1111	kkkk	kkkk	kkkk	Aucun
CLRWDT	—	Effacer le « Watchdog Timer »	1	0000	0000	0000	0100	
DAW	—	Ajuste les décimales dans WREG	1	0000	0000	0000	0111	TO, PD
GOTO	n	Aller à l'adresse n	2	1110	1111	kkkk	kkkk	C
				1111	kkkk	kkkk	kkkk	Aucun
NOP	—	No OPeration	1	0000	0000	0000	0000	
NOP	—	No OPeration	1	1111	xxxx	xxxx	xxxx	Aucun
POP	—	Retire un élément de la pile	1	0000	0000	0000	0110	Aucun
PUSH	—	Ajoute un élément à la pile	1	0000	0000	0000	0101	Aucun
RCALL	n	CALL Relatif	2	1101	1nnn	nnnn	nnnn	Aucun
RESET		Réinitialisation	1	0000	0000	1111	1111	Tous
RETFIE	s	RETURN d'une interruption	2	0000	0000	0001	000s	GIE/GIEH, PEIE/GIEL
RETLW	k	RETURN en mettant k dans WREG	2	0000	1100	kkkk	kkkk	Aucun
RETURN	s	RETURN d'une sous-routine	2	0000	0000	0001	001s	Aucun
SLEEP	—	Mettre le microcontrôleur en veille.	1	0000	0000	0000	0011	TO, PD

Configuration

Vous trouverez ici toutes les configurations disponibles pour le PIC18F4550. Les options ci-dessous sont les paramètres que nous donnons en début de code du fichier assembleur, avec la ligne :

« CONFIG option1=valeur, option2=valeur, etc. ».

Fréquence d'horloge

La fréquence de l'horloge est définie, d'abord et avant tout, par le choix de l'oscillateur. Dans un premier temps, il est possible de choisir entre deux oscillateurs internes, soit un de 8MHz et un de 31 kHz, mais ces options ne sont pas possibles si on désire activer la communication USB. Dans un deuxième temps, il est possible d'utiliser un oscillateur externe, dont la fréquence est laissée à votre discrétion, pour autant qu'elle ne dépasse pas 48 MHz.

Avec l'utilisation d'un oscillateur externe, il est possible de diminuer sa vitesse afin de le rendre compatible avec des applications USB qui fonctionnent avec des oscillateurs à faible vitesse. Cette diminution, voir division de la fréquence, s'effectue de deux manières. Il y a avec, et sans l'utilisation du PLL (Phase Locked Loop).

Le rôle du PLL est de répéter 24 fois, voir multiplier, chaque phase d'un signal de 4 MHz, pour émettre un signal de 96 MHz à sa sortie. Ce signal sera ensuite divisé par deux pour rester dans les limites du 48 MHz. Ceci est également très pratique pour augmenter la fréquence de l'horloge. Le PLL est activé avec les modes d'oscillateur HSPLL, XTPLL, ECPLL et ECPIO.

Comme spécifié ci-dessus, il nous faut obligatoirement avoir une fréquence de 4 MHz à l'entrée du PLL. C'est pourquoi nous avons recours au « PLL prescaler », qui a pour traduction libre « prédiviseur du PLL », c'est-à-dire qu'on ajuste la fréquence de l'oscillateur avant l'entrée dans le PLL. Donc si nous avons un oscillateur externe de 20 MHz, il nous faudra donner une valeur de 5 au « PLL prescaler » afin d'obtenir une fréquence de 4 MHz à l'entrée du PLL.

Options pour le « PLL Prescaler »		
Options		
PLLDIV = 1	Pas de mise à l'échelle	[pour un oscillateur de 4 MHz en entrée]
PLLDIV = 2	Division par 2	[pour un oscillateur de 8 MHz en entrée]
PLLDIV = 3	Division par 3	[pour un oscillateur de 12 MHz en entrée]
PLLDIV = 4	Division par 4	[pour un oscillateur de 16 MHz en entrée]
PLLDIV = 5	Division par 5	[pour un oscillateur de 20 MHz en entrée]
PLLDIV = 6	Division par 6	[pour un oscillateur de 24 MHz en entrée]
PLLDIV = 10	Division par 10	[pour un oscillateur de 40 MHz en entrée]
PLLDIV = 12	Division par 12	[pour un oscillateur de 48 MHz en entrée]

Pour diminuer la fréquence de l'oscillateur après avoir passé par le PLL, on utilise le « PLL postscaler », qui a pour traduction libre « postdiviseur du PLL », c'est-à-dire qu'on ajuste la fréquence après le passage dans le PLL. Ceci pourrait être utile, par exemple, si nous avons besoin d'atteindre une fréquence de 24 MHz, avec un

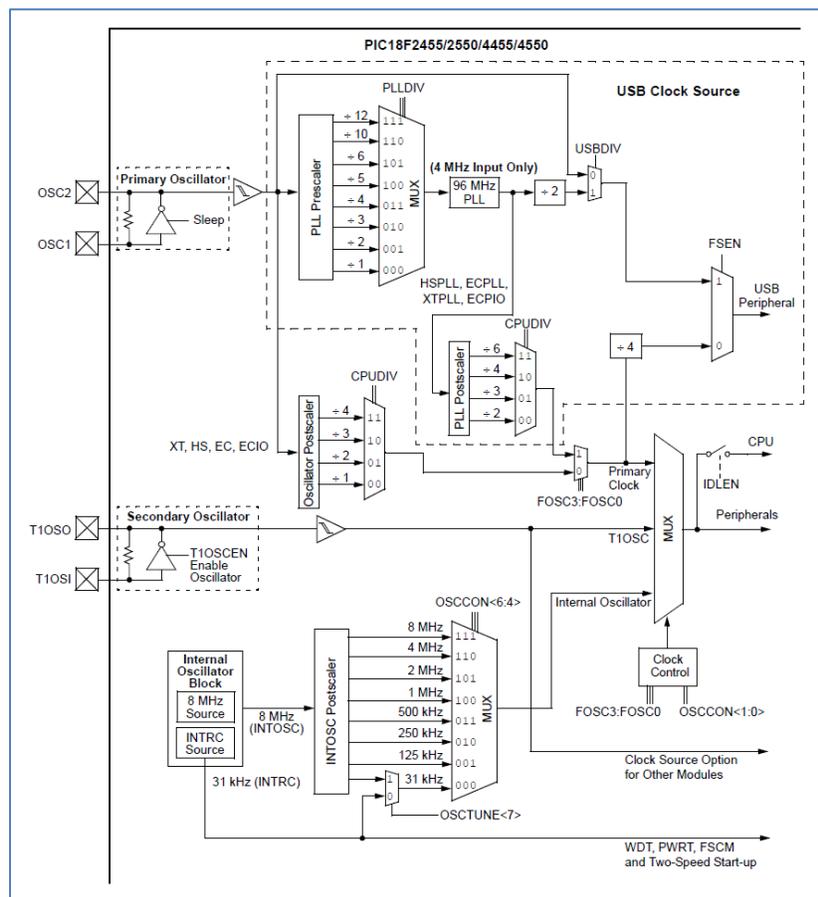
oscillateur de fréquence inférieur, en utilisant d’abord le « PLL prescaler », puis en mettant une valeur de 4 dans le « PLL postscaler ».

Pour diviser directement la fréquence de l’oscillateur externe, sans utiliser le PLL, on utilise l’ « oscillator postscaler », dont je vous laisse le soin de faire la traduction libre. L’ « oscillator postscaler » ajuste la fréquence à la sortie de l’oscillateur.

Une des grandes utilités des deux « postscalers » est de permettre d’avoir deux fréquences différentes, soit une pour le microcontrôleur et une pour le périphérique USB. Pour ce faire, il faut utiliser à la fois l’ « oscillator postscaler » et le « PLL postscaler ». Notez que les deux « postscaler » utilisent le même registre d’option, mais ont des diviseurs différents pour la même valeur dans le registre.

Options pour les deux « Postscaler »		
Options	Oscillateur	PLL
CPUDIV = OSC1_PLL2	Pas de mise à l’échelle	96 MHz : Division par 2
CPUDIV = OSC2_PLL3	Division par 2	96 MHz : Division par 3
CPUDIV = OSC3_PLL4	Division par 3	96 MHz : Division par 4
CPUDIV = OSC4_PLL6	Division par 4	96 MHz : Division par 6

Voici le diagramme complet des différentes sources d’horloge.



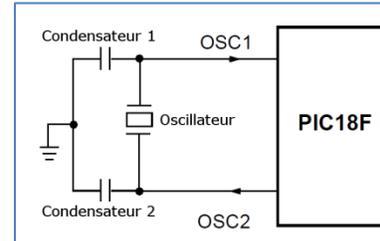
Sélection de l'horloge USB (seulement avec le mode USB « Full-Speed »)

- USBDIV = 1 La source de l'horloge USB vient directement de l'oscillateur, sans « postscaler ».
- USBDIV = 2 La source de l'horloge USB vient du PLL, soit 96 MHz divisé par 2.

Sélection du type d'oscillateur

Les types d'oscillateur compatibles sont les suivants.

- **XT** : Cristal/Résonateur
- **HS** : Cristal/Résonateur haute vitesse
- **EC** (External Clock) : Horloge externe



Le montage de l'oscillateur sur le circuit électronique, dont on peut voir le diagramme ci-dessus, nécessite l'utilisation de deux condensateurs. Voici une liste des condensateurs recommandés.

Oscillateur à résonateur de céramique		
Mode	Fréquence	Valeur des condensateurs
XT	4 MHz	33 pF
HS	8 MHz	27 pF
	16 MHz	22 pF

Oscillateur à cristal de quartz		
Mode	Fréquence	Valeur des condensateurs
XT	4 MHz	27 pF
HS	4 MHz	27 pF
	8 MHz	22 pF
	20 MHz	15 pF

Finalement, voici les options à utiliser pour la configuration désirée.

- FOSC = XT_XT oscillateur XT
- FOSC = XTPLL_XT oscillateur XT, PLL activé (XTPLL)
- FOSC = ECIO_EC oscillateur EC, port RA6 en entrée/sortie (ECIO)
- FOSC = EC_EC oscillateur EC, port RA6 en tant que CLKO (EC)
- FOSC = ECPLLIO_EC oscillateur EC, PLL activé, port RA6 en entrée/sortie (ECPIO)
- FOSC = ECPLL_EC oscillateur EC, PLL activé, port RA6 en tant que CLKO (ECPLL)
- FOSC = INTOSCIO_EC oscillateur interne, port RA6 en entrée/sortie, osci. EC utilisé par l'USB (INTIO)
- FOSC = INTOSC_EC oscillateur interne, port RA6 en tant que CLKO, osci. EC utilisé par l'USB (INTCKO)
- FOSC = INTOSC_XT oscillateur interne, oscillateur XT utilisé par l'USB (INTXT)
- FOSC = INTOSC_HS oscillateur interne, oscillateur HS utilisé par l'USB (INTHS)
- FOSC = HS oscillateur HS
- FOSC = HSPLL_HS oscillateur HS, PLL activé (HSPLL)

Fail-Safe Clock Monitor

Le FSCM (Fail-Safe Clock Monitor), qui a pour rôle de protéger contre les défaillances de l'horloge, permet au microcontrôleur de continuer à fonctionner en cas d'une défaillance de l'oscillateur externe, en commutant automatiquement le dispositif d'horloge sur l'oscillateur interne.

FCMEN = OFF	FSCM désactivé.
FCMEN = ON	FSCM activé.

Le « Démarrage à deux vitesses »

Le « démarrage à deux vitesses » est une fonctionnalité qui permet de minimiser la période de latence, du démarrage de l'oscillateur jusqu'à l'exécution de code, en permettant au microcontrôleur d'utiliser l'oscillateur interne comme source d'horloge jusqu'à ce que l'oscillateur externe soit disponible.

Le « démarrage à deux vitesses » doit être activé uniquement si le mode d'oscillateur primaire est XT, HS, XTPLL ou HSPLL (À base de cristaux de quartz). Pour les autres sources, le « démarrage à deux vitesses » doit être désactivé.

IESO = OFF	Commutateur d'oscillateur désactivé.
IESO = ON	Commutateur d'oscillateur activé.

Power-up Timer (PWRT)

Le PWRT procure un délai fixe d'approximativement 65,5 millisecondes (2048 x 32 µs) au démarrage. Durant cette période, le microcontrôleur est maintenu en mode reset.

PWRT = ON	PWRT activé.
PWRT = OFF	PWRT désactivé.

Brown-out Reset (BOR)

Le BOR (Brown-Out Reset) est une réinitialisation qui se lance automatiquement lorsque la tension descend en dessous d'un seuil critique, défini par l'utilisateur (voir page suivante). Lorsque le BOR est activé, il est entièrement contrôlé par le microcontrôleur, à moins que l'option « SOFT » soit choisie. Dans ce cas, l'utilisateur peut décider, via son logiciel, quand désactiver ou activer le BOR. Pour ce faire, il suffit de mettre respectivement à '0' ou '1' le bit SBOREN (bit no.6) du registre « RCON ».

BOR = OFF	BOR désactivé au niveau matériel et au niveau logiciel.
BOR = SOFT	BOR activé et contrôlé au niveau logiciel.
BOR = ON_ACTIVE	BOR activé au niveau matériel seulement et désactivé dans le mode « Sleep ».
BOR = ON	BOR activé au niveau matériel seulement.

Les quatre valeurs possibles pour le seuil critique du BOR sont :

BORV = 46	4.6V
BORV = 43	4.3V
BORV = 28	2.8V
BORV = 21	2.1V

Régulateur de voltage USB

Le PIC18 possède un régulateur de voltage de 3,3v pour alimenter l'émetteur/récepteur USB interne. Il est possible de désactiver ce régulateur de voltage dans le cas où l'on désire utiliser un régulateur externe.

VREGEN = OFF	Régulateur de voltage USB désactivé.
VREGEN = ON	Régulateur de voltage USB activé.

Watchdog Timer (WDT)

Le WDT (WatchDog Timer) est une minuterie qui, lorsqu'elle atteint zéro, réveille ou réinitialise le microcontrôleur, selon qu'il soit en sommeil ou en situation de blocage. Le délai de cette minuterie est défini par l'usage. Voir la section « Postscale » du Watchdog Timer ci-dessous pour les détails. À noter que le WDT n'utilise pas l'horloge générée par l'oscillateur externe, mais l'horloge générée un des deux oscillateurs internes du microcontrôleur, soit l'INTRC de 31 kHz.

WDT = OFF	WDT désactivé.
WDT = ON	WDT activé.

« Postscale » du Watchdog Timer

Le WDT a une période nominale de 4 millisecondes. Pour atteindre un délai près de celui désiré, nous utilisons les ratios ci-dessous. Par exemple, avec « WDTPS=64 », nous obtiendrions un délai de 256 millisecondes (4 x 64).

WDTPS = 1	1:1
WDTPS = 2	1:2
WDTPS = 4	1:4
WDTPS = 8	1:8
WDTPS = 16	1:16
WDTPS = 32	1:32
WDTPS = 64	1:64
WDTPS = 128	1:128
WDTPS = 256	1:256
WDTPS = 512	1:512
WDTPS = 1024	1:1024
WDTPS = 2048	1:2048
WDTPS = 4096	1:4096
WDTPS = 8192	1:8192
WDTPS = 16384	1:16384
WDTPS = 32768	1:32768

Configuration A/D de « PORTB »

Le « PORTB » bénéficie d'une option particulière pour la configuration de ses broches suite à une réinitialisation. Cela va comme suit :

PBADEN = OFF	Les broches 0 à 4 seront configurées en E/S <u>numérique</u> .
PBADEN = ON	Les broches 0 à 4 seront configurées en E/S <u>analogique</u> .

Low-Power Timer 1 Oscillator Enable bit:

LPT1OSC = OFF	Timer1 configured for higher power operation
LPT1OSC = ON	Timer1 configured for low-power operation

Activation de la broche « MCLR »

La broche numéro 1 du PIC18 peut être configurée soit en tant que MCLR (Master Clear), soit en tant qu'entrée numérique. À noter que cette broche ne peut être configurée en tant qu'entrée analogique ou sortie numérique.

MCLRE = OFF	RE3 sera activé et MCLR sera désactivé.
MCLRE = ON	MCLR sera activé et RE3 sera désactivé.

Pile des retours d'appel

La pile des retours d'appel (return address stack) accumule les adresses de retour pour les appels de fonction (call) ou les interruptions. Elle permet jusqu'à 31 combinaisons d'appels et d'interruptions. Lorsqu'on atteint ou passe outre cette limite, un drapeau est levé. Il en est de même si on tente de descendre sous la limite inférieure (position 0) de la pile. Lorsqu'un tel drapeau est levé, on peut choisir d'effectuer ou non une réinitialisation.

STVREN = OFF	Pas de provocation de réinitialisation.
STVREN = ON	Provocation d'une réinitialisation.

Protection du code

Il est possible pour l'utilisateur de lire et d'écrire dans l'espace de mémoire du programme. Dans une telle situation, il pourrait être utile de voir à protéger le code, ou du moins, une partie de celui-ci.

Pour lire et écrire dans l'espace de mémoire du programme, on utilise une table d'accès. Cette dernière se divise en un pointeur sur trois registre (TBLPTRU, TBLPTRH et TBLPTRL) et d'un registre pour l'échange de la valeur (TBLPTRU), tant en lecture qu'en écriture.

L'espace de mémoire du programme est divisée en cinq zones. Ce qui implique également cinq code de protection principal. Voici, à la page suivante, le diagramme de la structure de la protection du code. Les options de contrôle suivent le diagramme.

Accès en lecture pour la mémoire programme

EBTR0 = ON	Accès en lecture <u>autorisé</u> pour le block (000800-001FFFh)
EBTR0 = OFF	Accès en lecture <u>interdit</u> pour le block (000800-001FFFh)
EBTR1 = ON	Accès en lecture <u>autorisé</u> pour le block (002000-003FFFh)
EBTR1 = OFF	Accès en lecture <u>interdit</u> pour le block (002000-003FFFh)
EBTR2 = ON	Accès en lecture <u>autorisé</u> pour le block (004000-005FFFh)
EBTR2 = OFF	Accès en lecture <u>interdit</u> pour le block (004000-005FFFh)
EBTR3 = ON	Accès en lecture <u>autorisé</u> pour le block (006000-007FFFh)
EBTR3 = OFF	Accès en lecture <u>interdit</u> pour le block (006000-007FFFh)

Accès en écriture pour le « Boot Block »

WRTB = ON	Accès en écriture <u>autorisé</u> pour le Boot block (000000-0007FFh)
WRTB = OFF	Accès en écriture <u>interdit</u> pour le Boot block (000000-0007FFh)

Accès en écriture pour la mémoire programme

WRT0 = ON	Accès en écriture <u>autorisé</u> pour le block (000800-001FFFh)
WRT0 = OFF	Accès en écriture <u>interdit</u> pour le block (000800-001FFFh)
WRT1 = ON	Accès en écriture <u>autorisé</u> pour le block (002000-003FFFh)
WRT1 = OFF	Accès en écriture <u>interdit</u> pour le block (002000-003FFFh)
WRT2 = ON	Accès en écriture <u>autorisé</u> pour le block (004000-005FFFh)
WRT2 = OFF	Accès en écriture <u>interdit</u> pour le block (004000-005FFFh)
WRT3 = ON	Accès en écriture <u>autorisé</u> pour le block (006000-007FFFh)
WRT3 = OFF	Accès en écriture <u>interdit</u> pour le block (006000-007FFFh)

Protection du registre de configuration

WRTC = ON	Registre de configuration protégé.
WRTC = OFF	Registre de configuration non protégé.

La réinitialisation (reset)

Lorsqu'on réinitialise le microcontrôleur, certaines valeurs des registres de configuration sont remises à leur état initial, certaines autres conservent les valeurs déterminées par l'utilisateur. Voici les tableaux des différents registres de configuration, ainsi que les valeurs prises après une réinitialisation, selon le type de réinitialisation. Voyez la légende ci-contre pour les détails.

Légende	
0	Prend la valeur zéro
1	Prend la valeur un
u	Valeur inchangé
x	Valeur inconnu
-	Bit non implémenté
q	Dépend des conditions
Co	Compteur ordinal

Les registres RCON et STKPTR

Les registres RCON et STKPTR font partie du SFR (Special Function Registers), mais étant donné qu'il s'agit respectivement du registre d'état concernant les différents types de réinitialisation, et du registre du pointeur de la pile d'adresse, nous les verrons plus en détail ici.

RCON (Reset Control register)

Le registre RCON contient les bits d'état qui sont utilisés pour déterminer la cause de la dernière réinitialisation ou de la sortie du mode sommeil (sleep). RCON contient également le bit IPEN qui détermine les priorités d'interruption.

Signification des bits du registre RCON	
Bit 7	IPEN : Interrupt Priority Enable
Bit 6	SBOREN : BOR Software Enable
Bit 5	Unimplemented: Read as '0'
Bit 4	RI : RESET Instruction Flag
Bit 3	TO : Watchdog Time-out Flag
Bit 2	PD : Power-Down Detection Flag
Bit 1	POR : Power-on Reset Status
Bit 0	BOR : Brown-out Reset Status

STKPTR

Le registre STKPTR contient le pointeur de la pile d'adresses, le bit d'état STKFUL (pile pleine) et le bit d'état STKUNF (débordement vers le bas). La valeur du pointeur de la pile, bits <4:0>, peut être de 0 à 31.

Signification des bits du registre STKPTR	
Bit 7	STKFUL : Stack Full Flag
Bit 6	STKUNF : Stack Underflow Flag
Bit 5	Unimplemented: Read as '0'
Bit 4:0	SP4:SP0 : Stack Pointer Location

Tableau des valeurs après réinitialisation

Condition	Compteur ordinal	Registre RCON					Registre STKPTR	
		RI	TO	PD	POR	BOR	STKFUL	STLUNF
Réinitialisation POR (Power-On Reset)	0000h	1	1	1	0	0	0	0
Instruction « RESET »	0000h	0	u	u	u	u	u	u
Réinitialisation BOR (Brown-out Reset)	0000h	1	1	1	u	0	u	u
Réinitialisation MCLR (en mode gestion de puissance)	0000h	u	1	u	u	u	u	u
Réinitialisation MCLR (en mode pleine puissance)	0000h	u	u	u	u	u	u	u
Réinitialisation MCLR (en mode sommeil)	0000h	u	1	0	u	u	u	u
Réinitialisation WDT (en mode fonctionnement)	0000h	u	0	u	u	u	u	u
Réinitialisation WDT (en mode sommeil)	+2	u	0	0	u	u	u	u
Réinitialisation « Pile pleine »	0000h	u	u	u	u	u	1	u
Réinitialisation « Débordement de pile vers le bas »	0000h	u	u	u	u	u	u	1
Erreur de débordement de pile vers le bas	0000h	u	u	u	u	u	u	1
Réveil par interruption (en mode gestion de puissance)	+2 ⁽³⁾	u	u	0	u	u	u	u

³ Si le réveil est due à une interruption et que le bit de haute priorité d'interruption (GIEH) ou de basse priorité d'interruption (GIEL) du registre INTCON est activé, alors le compteur ordinal prendra l'adresse 008h ou 0018h, selon le cas.

Les registres du SFR (Special Function Registers)

Adresse	Registre	Réinitialisation POR	Réinitialisation MCLR Réinitialisation WDT Instruction « RESET » Débordement de la pile	Réveil par le WDT ou par une interruption
FFFh	TOSU	---0 0000	---0 0000	---0 uuuu ⁽⁴⁾
	TOSH	0000 0000	0000 0000	uuuu uuuu ⁽⁴⁾
	TOSL	0000 0000	0000 0000	uuuu uuuu ⁽⁴⁾
	STKPTR	00-0 0000	uu-0 0000	uu-u uuuu ⁽⁴⁾
	PCLATU	---0 0000	---0 0000	---u uuuu
	PCLATH	0000 0000	0000 0000	uuuu uuuu
	PCL	0000 0000	0000 0000	Co + 2 ⁽⁵⁾
	TBLPTRU	--00 0000	--00 0000	--uu uuuu
	TBLPTRH	0000 0000	0000 0000	uuuu uuuu
	TBLPTRL	0000 0000	0000 0000	uuuu uuuu
	TABLAT	0000 0000	0000 0000	uuuu uuuu
	PRODH	xxxx xxxx	uuuu uuuu	uuuu uuuu
	PRODL	xxxx xxxx	uuuu uuuu	uuuu uuuu
	INTCON	0000 000x	0000 000u	uuuu uuuu ⁽⁶⁾
	INTCON2	1111 -1-1	1111 -1-1	uuuu -u-u ⁽⁶⁾
	INTCON3	11-0 0-00	11-0 0-00	uu-u u-uu ⁽⁶⁾
	INDF0	S/O	S/O	S/O
	POSTINC0	S/O	S/O	S/O
	POSTDEC0	S/O	S/O	S/O
	PREINC0	S/O	S/O	S/O
	PLUSW0	S/O	S/O	S/O
	FSR0H	---- 0000	---- 0000	---- uuuu
	FSR0L	xxxx xxxx	uuuu uuuu	uuuu uuuu
	WREG	xxxx xxxx	uuuu uuuu	uuuu uuuu
	INDF1	S/O	S/O	S/O
	POSTINC1	S/O	S/O	S/O
	POSTDEC1	S/O	S/O	S/O
	PREINC1	S/O	S/O	S/O
	PLUSW1	S/O	S/O	S/O
	FSR1H	---- 0000	---- 0000	---- uuuu
	FSR1L	xxxx xxxx	uuuu uuuu	uuuu uuuu
	FE0h	BSR	---- 0000	---- 0000

⁴ Si le réveil est due à une interruption et que le bit de haute priorité d'interruption (GIEH) ou de basse priorité d'interruption (GIEL) du registre INTCON est activé, alors les registres TOSU, TOSH et TOSL seront mis à jour avec la valeur courante du compteur ordinal. Le pointeur de pile sera modifié pour pointer sur la prochaine location dans la pile.

⁵ Si le réveil est due à une interruption et que le bit de haute priorité d'interruption (GIEH) ou de basse priorité d'interruption (GIEL) du registre INTCON est activé, alors le compteur ordinal prendra l'adresse 008h ou 0018h, selon le cas.

⁶ Un ou plusieurs bits des registres INTCONx et PIRx seront affectés (pour provoquer le réveil).

Adresse	Registre	Réinitialisation POR	Réinitialisation MCLR Réinitialisation WDT Instruction « RESET » Débordement de la pile	Réveil par le WDT ou par une interruption
FDh	INDF2	S/0	S/0	S/0
	POSTINC2	S/0	S/0	S/0
	POSTDEC2	S/0	S/0	S/0
	PREINC2	S/0	S/0	S/0
	PLUSW2	S/0	S/0	S/0
	FSR2H	---- 0000	---- 0000	---- uuuu
	FSR2L	xxxx xxxx	uuuu uuuu	uuuu uuuu
	STSTATUS	---x xxxx	---u uuuu	---u uuuu
	TMR0H	0000 0000	0000 0000	uuuu uuuu
	TMR0L	xxxx xxxx	uuuu uuuu	uuuu uuuu
	T0CON	1111 1111	1111 1111	uuuu uuuu
	OSCCON	0100 q000	0100 00q0	uuuu uuqu
	HLVDCON	0-00 0101	0-00 0101	u-uu uuuu
	WDTCON	---- --0	---- --0	---- --u
	RCON	0q-1 11q0	0q-q qquu	uq-u qquu
	TMR1H	xxxx xxxx	uuuu uuuu	uuuu uuuu
	TMR1L	xxxx xxxx	uuuu uuuu	uuuu uuuu
	T1CON	0000 0000	u0uu uuuu	uuuu uuuu
	TMR2	0000 0000	0000 0000	uuuu uuuu
	PR2	1111 1111	1111 1111	1111 1111
	T2CON	-000 0000	-000 0000	-uuu uuuu
	SSPBUF	xxxx xxxx	uuuu uuuu	uuuu uuuu
	SSPAD	0000 0000	0000 0000	uuuu uuuu
	SSPSTAT	0000 0000	0000 0000	uuuu uuuu
	SSPCON1	0000 0000	0000 0000	uuuu uuuu
	SSPCON2	0000 0000	0000 0000	uuuu uuuu
	ADRESH	xxxx xxxx	uuuu uuuu	uuuu uuuu
	ADRESL	xxxx xxxx	uuuu uuuu	uuuu uuuu
	ADCON0	--00 0000	--00 0000	--uu uuuu
	ADCON1	--00 0qqq	--00 0qqq	--uu uuuu
FC0h	ADCON2	0-00 0000	0-00 0000	u-uu uuuu

Adresse	Registre	Réinitialisation POR	Réinitialisation MCLR Réinitialisation WDT Instruction « RESET » Débordement de la pile	Réveil par le WDT ou par une interruption
FBFh	CCPR1H	xxxx xxxx	uuuu uuuu	uuuu uuuu
	CCPR1L	xxxx xxxx	uuuu uuuu	uuuu uuuu
	CCP1CON	0000 0000	0000 0000	uuuu uuuu
	CCPR2H	xxxx xxxx	uuuu uuuu	uuuu uuuu
	CCPR2L	xxxx xxxx	uuuu uuuu	uuuu uuuu
	CCP2CON	--00 0000	--00 0000	--uu uuuu
	BAUDCON	0100 0-00	0100 0-00	uuuu u-uu
	ECCP1DEL	0000 0000	0000 0000	uuuu uuuu
	ECCP1AS	0000 0000	0000 0000	uuuu uuuu
	CVRCON	0000 0000	0000 0000	uuuu uuuu
	CMCON	0000 0111	0000 0111	uuuu uuuu
	TMR3H	xxxx xxxx	uuuu uuuu	uuuu uuuu
	TMR3L	xxxx xxxx	uuuu uuuu	uuuu uuuu
	T3CON	0000 0000	uuuu uuuu	uuuu uuuu
	SPBRGH	0000 0000	0000 0000	uuuu uuuu
	SPBTG	0000 0000	0000 0000	uuuu uuuu
	RCREG	0000 0000	0000 0000	uuuu uuuu
	TXREG	0000 0000	0000 0000	uuuu uuuu
	TXSTA	0000 0010	0000 0010	uuuu uuuu
	RCSTA	0000 000x	0000 000x	uuuu uuuu
EEADR	0000 0000	0000 0000	uuuu uuuu	
EEDATA	0000 0000	0000 0000	uuuu uuuu	
EECON2	0000 0000	0000 0000	0000 0000	
FA0h	EECON1	xx-0 x000	uu-0 u000	uu-0 u000

Adresse	Registre	Réinitialisation POR	Réinitialisation MCLR Réinitialisation WDT Instruction « RESET » Débordement de la pile	Réveil par le WDT ou par une interruption
F9Fh	IPR2	1111 1111	1111 1111	uuuu uuuu
	PIR2	0000 0000	0000 0000	uuuu uuuu
	PIE2	0000 0000	0000 0000	uuuu uuuu
	IPR1	1111 1111	1111 1111	uuuu uuuu
	PIR1	0000 0000	0000 0000	uuuu uuuu
	PIE1	0000 0000	0000 0000	uuuu uuuu
	OCSTUNE	0--0 0000	0--0 0000	u--u uuuu
	TRISE	---- -111	---- -111	---- -uuu
	TRISD	1111 1111	1111 1111	uuuu uuuu
	TRISC	11-- -111	11-- -111	uu-- -uuu
	TRISB	1111 1111	1111 1111	uuuu uuuu
	TRISA	-111 1111	-111 1111	-uuu uuuu
	LATE	---- -xxx	---- -uuu	---- -uuu
	LATD	xxxx xxxx	uuuu uuuu	uuuu uuuu
	LATC	xx-- -xxx	uu-- -uuu	uu-- -uuu
	LATB	xxxx xxxx	uuuu uuuu	uuuu uuuu
	LATA	-xxx xxxx	-uuu uuuu	-uuu uuuu
	PORTE	0--- x000	0--- x000	u--- uuuu
	PORTD	xxxx xxxx	uuuu uuuu	uuuu uuuu
	PORTC	xxxx -xxx	uuuu -uuu	uuuu -uuu
PORTB	xxxx xxxx	uuuu uuuu	uuuu uuuu	
F80h	PORTA	-x0x 0000	-u0u 0000	-uuu uuuu

Adresse	Registre	Réinitialisation POR	Réinitialisation MCLR Réinitialisation WDT Instruction « RESET » Débordement de la pile	Réveil par le WDT ou par une interruption
F7Fh	UEP15	---0 0000	---0 0000	---u uuuu
	UEP14	---0 0000	---0 0000	---u uuuu
	UEP13	---0 0000	---0 0000	---u uuuu
	UEP12	---0 0000	---0 0000	---u uuuu
	UEP11	---0 0000	---0 0000	---u uuuu
	UEP10	---0 0000	---0 0000	---u uuuu
	UEP9	---0 0000	---0 0000	---u uuuu
	UEP8	---0 0000	---0 0000	---u uuuu
	UEP7	---0 0000	---0 0000	---u uuuu
	UEP6	---0 0000	---0 0000	---u uuuu
	UEP5	---0 0000	---0 0000	---u uuuu
	UEP4	---0 0000	---0 0000	---u uuuu
	UEP3	---0 0000	---0 0000	---u uuuu
	UEP2	---0 0000	---0 0000	---u uuuu
	UEP1	---0 0000	---0 0000	---u uuuu
	UEP0	---0 0000	---0 0000	---u uuuu
	UCFG	00-0 0000	00-0 0000	uu-u uuuu
	UADDR	-000 0000	-000 0000	-uuu uuuu
	UCON	-0x0 000-	-0x0 000-	-uuu uuu-
	USTAT	-xxx xxx-	-xxx xxx-	-uuu uuu-
	UEIE	0--0 0000	0--0 0000	u--u uuuu
UEIR	0--0 0000	0--0 0000	u--u uuuu	
UIE	-000 0000	-000 0000	-uuu uuuu	
UIR	-000 0000	-000 0000	-uuu uuuu	
UFRMH	---- -xxx	---- -xxx	---- -uuu	
UFRML	xxxx xxxx	xxxx xxxx	uuuu uuuu	
SPPCON	---- --00	---- --00	---- --uu	
SPPEPS	00-0 0000	00-0 0000	uu-u uuuu	
SPPCFG	0000 0000	0000 0000	uuuu uuuu	
F60h	SPPDATA	0000 0000	0000 0000	uuuu uuuu

La gestion des événements et des interruptions

Les événements peuvent provoquer des interruptions de plusieurs manières. Il y a les événements externes au microcontrôleur, c'est-à-dire l'envoi d'un signal sur une broche d'interruption, et les événements internes au microcontrôleur, c'est-à-dire ceux provoqués par les périphériques internes, comme les minuteurs par exemple.

Pour ce qui est des événements externes, il existe trois broches qui peuvent être dédiées aux interruptions, soit INT0, INT1 et INT2 qui sont respectivement les broches numéro 33, 34 et 35. Chacune de ces broches peut être activée ou désactivée indépendamment des autres, et possède chacune un bit d'état qui lui est propre.

Il existe également une quatrième façon de récupérer un événement externe. Il s'agit du port 'B', sur ses entrées RB3 à RB7, soit les broches 36 à 40. Par contre, ces broches ne peuvent être configurées individuellement. Elles partagent le même bit d'activation et le même bit d'état.

Les événements internes quant à eux offrent une plus grande source d'interruption. Il y a quatre minuteurs, appelés timerX, mais également le port de flux parallèle (Streaming Parallel Port), le convertisseur analogique/numérique (A/D converter), le récepteur EUSART, le transmetteur EUSART, le port série synchrone (Master Synchronous Serial Port), les modules « Capture/Compare/ Pulse-Width Modulation » (CCP1 et CCP2), la défaillance de l'oscillateur, le comparateur, l'USB, l'écriture dans la mémoire de données EEPROM, les collisions dans le bus, puis finalement la détection du bas/haut voltage.

En plus d'avoir plusieurs sources d'interruption, il est possible de les diviser sur deux niveaux de priorité, soit les hautes et les basses priorités. Ceci est valable pour chacune des sources d'interruption individuellement, à l'exception de INT0 qui est toujours une interruption de haute priorité.

Les interruptions de haute priorité ne peuvent être interrompues par aucun autre événement, étant donné qu'elles désactivent automatiquement tous les masques d'interruption. Ceux-ci sont automatiquement réactivés lorsqu'une interruption prend fin avec l'instruction « retfie ». Cette dernière n'étant rien d'autre qu'une instruction « return » qui active également le bit d'activation des interruptions.

Pour ce qui est des interruptions de basse priorité, elles ne peuvent s'interrompre entre elles. Seule une interruption de haute priorité peut les interrompre.

Ces sources d'interruption utilisent les registres décrits dans les pages suivantes. Vous remarquerez dans les noms des bits que 'IE' signifie « Interrupt Enable », que 'IF' signifie « Interrupt Flag », que 'IP' signifie « Interrupt Priority » et que 'EDG' signifie « Edge ».

Note : Le bit IPEN du registre RCON influence les bits 6 et 7 du registre INTCON ci-dessous. Le bit IPEN est celui qui détermine si le double niveau de priorité est activé (1) ou désactivé (0). Lorsque le double niveau de priorité est désactivé, toutes les interruptions sont traitées avec une haute priorité.

INTCON Interrupt Control Register			
Concerne : Activation des interruptions globales Activation des interruptions périphériques (internes) Timer0 INT0 RBx			
No. bit	Nom bit	Description (anglais)	Valeurs
7	GIE/GIEH	Global Interrupt Enable bit	Lorsque RCON <IPEN> = 0 1 = Active les interruptions sans masque 0 = Désactive toutes les interruptions
			Lorsque RCON <IPEN> = 1 1 = Active les interruptions HAUTE priorité 0 = Désactive toutes les interruptions
6	PEIE/GIEL	Peripheral Interrupt Enable bit	Lorsque RCON <IPEN> = 0 1 = Active les interruptions de périphériques sans masque 0 = Désactive les interrup. de périphériques
			Lorsque RCON <IPEN> = 1 1 = Active les interruptions de périphériques BASSE priorité 0 = Désactive les interrup .de périphériques BASSE priorité
5	TMR0IE	Timer0 Overflow Interrupt Enable bit	1 = Active les interruptions de débordement sur Timer0 0 = Désactive les interruptions de débordement sur Timer0
4	INT0IE	INT0 External Interrupt Enable bit	1 = Active les interruptions externes sur INT0 0 = Désactive les interruptions externes sur INT0
3	RBIE	RB Port Change Interrupt Enable bit	1 = Active les interruptions de modifications sur RBx 0 = Désactive les interrup. de modifications sur RBx
2	TMR0IF	Timer0 Overflow Interrupt Flag bit	1 = Timer0 a débordé 0 = Timer0 n'a pas débordé
1	INT0IF	INT0 External Interrupt Flag bit	1 = Interruption externe reçue sur INT0 0 = Pas d'interruption externe reçue sur INT0
0	RBIF	RB Port Change Interrupt Flag bit	1 = Au moins une broche entre RB7 et RB3 a changé d'état 0 = Aucune broche entre RB7 et RB3 n'a changé d'état

INTCON2 Interrupt Control Register			
Concerne : « Pull-up » de PORTB Choix du front du signal pour les interruptions externes Choix du niveau de priorité pour Timer0 et RBx			
No. bit	Nom bit	Description (anglais)	Valeurs
7	RBPU	PORTB Pull-up Enable bit	1 = Tous les « pull-up » sont désactivés sur le PORTB 0 = Les « pull-up » sont activés les valeurs individuelles
6	INTEDG0	INT0 Edge Select bit	1 = Interruption sur le front ASCENDANT du signal 0 = Interruption sur le front DESCENDANT du signal
5	INTEDG1	INT1 Edge Select bit	1 = Interruption sur le front ASCENDANT du signal 0 = Interruption sur le front DESCENDANT du signal
4	INTEDG2	INT2 Edge Select bit	1 = Interruption sur le front ASCENDANT du signal 0 = Interruption sur le front DESCENDANT du signal
3		<i>Non implanté</i>	
2	TMR0IP	Timer0 Overflow Interrupt Priority bit	1 = Haute priorité 0 = Basse priorité
1		<i>Non implanté</i>	
0	RBIP	RB Port Change Interrupt Priority bit	1 = Haute priorité 0 = Basse priorité

INTCON3 Interrupt Control Register			
Concerne : INT1 et INT2			
No. bit	Nom bit	Description (anglais)	Valeurs
7	INT2IP	INT2 External Interrupt Priority bit	1 = Haute priorité 0 = Basse priorité
6	INT1IP	INT1 External Interrupt Priority bit	1 = Haute priorité 0 = Basse priorité
5		<i>Non implanté</i>	
4	INT2IE	INT2 External Interrupt Enable bit	1 = Active les interruptions externes sur INT2 0 = Désactive les interruptions externes sur INT2
3	INT1IE	INT1 External Interrupt Enable bit	1 = Active les interruptions externes sur INT1 0 = Désactive les interruptions externes sur INT1
2		<i>Non implanté</i>	
1	INT2IF	INT2 External Interrupt Flag bit	1 = Interruption externe reçue sur INT2 0 = Pas d'interruption externe reçue sur INT2
0	INT1IF	INT1 External Interrupt Flag bit	1 = Interruption externe reçue sur INT1 0 = Pas d'interruption externe reçue sur INT1

PEI Peripheral Interrupt <u>Enable</u> Register			
Concerne : Interruptions des périphériques internes (activation)			
No. bit	Nom bit	Description (anglais)	Valeurs
7	SPPIE	Streaming Parallel Port Read/Write Interrupt Enable bit	Ce bit est réservé aux microcontrôleurs de 28 broches. Il doit toujours être à '0'.
6	ADIE	A/D Converter Interrupt Enable bit	1 = Active les interruptions de lecture/écriture du SPP 0 = Désactive les interruptions de lecture/écriture du SPP
5	RCIE	EUSART Receive Interrupt Enable bit	1 = Active les interruptions de réception du EUSART 0 = Désactive les interruptions de réception du EUSART
4	TXIE	EUSART Transmit Interrupt Enable bit	1 = Active les interruptions de transmission du EUSART 0 = Désactive les interrup. de transmission du EUSART
3	SSPIE	Master Synchronous Serial Port (MSSP) Interrupt Enable bit	1 = Active les interruptions du MSSP 0 = Désactive les interruptions du MSSP
2	CCP1IE	CCP1 Interrupt Enable bit	1 = Active les interruptions du CCP1 0 = Désactive les interruptions du CCP1
1	TMR2IE	Timer2 to PR2 Match Interrupt Enable bit	1 = Active les interrup. de correspondance Timer2 → PR2 0 = Désactive les interrup. de corresp. Timer2 → PR2
0	TMR1IE	Timer1 Overflow Interrupt Enable bit	1 = Active les interruptions de débordement sur Timer1 0 = Désactive les interruptions de débordement sur Timer1

PEI2 Peripheral Interrupt <u>Enable</u> Register			
No. bit	Nom bit	Description (anglais)	Valeurs
7	OSCFIE	Oscillator Fail Interrupt Enable bit	1 = Active les interruptions de défaillance de l'oscillateur 0 = Désactive les interrup. de défaillance de l'oscillateur
6	CMIE	Comparator Interrupt Enable bit	1 = Active les interruptions du comparateur 0 = Désactive les interruptions du comparateur
5	USBIE	USB Interrupt Enable bit	1 = Active les interruptions USB 0 = Désactive les interruptions USB
4	EEIE	Data EEPROM/Flash Write Operation Interrupt Enable bit	1 = Active les interruptions d'écriture dans l'EEPROM 0 = Désactive les interruptions d'écriture dans l'EEPROM
3	BCLIE	Bus Collision Interrupt Enable bit	1 = Active les interruptions de collision dans le BUS 0 = Désactive les interruptions de collision dans le BUS
2	HLVDIE	High/Low-Voltage Detect Interrupt Enable bit	1 = Active les interruptions de voltage 0 = Désactive les interruptions de voltage
1	TMR3IE	Timer3 Overflow Interrupt Enable bit	1 = Active les interruptions de débordement sur Timer3 0 = Désactive les interruptions de débordement sur Timer3
0	CCP2IE	CCP2 Interrupt Enable bit	1 = Active les interruptions du CCP2 0 = Désactive les interruptions du CCP2

IPR Peripheral Interrupt <u>P</u> riority Register			
Concerne : Interruptions des périphériques internes (activation)			
No. bit	Nom bit	Description (anglais)	Valeurs
7	SPIIP	Streaming Parallel Port Read/Write Interrupt Priority bit	1 = Haute priorité 0 = Basse priorité
6	ADIP	A/D Converter Interrupt Priority bit	1 = Haute priorité 0 = Basse priorité
5	RCIP	EUSART Receive Interrupt Priority bit	1 = Haute priorité 0 = Basse priorité
4	TXIP	EUSART Transmit Interrupt Priority bit	1 = Haute priorité 0 = Basse priorité
3	SSPIIP	Master Synchronous Serial Port (MSSP) Interrupt Priority bit	1 = Haute priorité 0 = Basse priorité
2	CCP1IP	CCP1 Interrupt Priority bit	1 = Haute priorité 0 = Basse priorité
1	TMR2IP	Timer2 to PR2 Match Interrupt Priority bit	1 = Haute priorité 0 = Basse priorité
0	TMR1IP	Timer1 Overflow Interrupt Priority bit	1 = Haute priorité 0 = Basse priorité

IPR2 Peripheral Interrupt <u>P</u> riority Register			
No. bit	Nom bit	Description (anglais)	Valeurs
7	OSCFIP	Oscillator Fail Interrupt Priority bit	1 = Haute priorité 0 = Basse priorité
6	CMIP	Comparator Interrupt Priority bit	1 = Haute priorité 0 = Basse priorité
5	USBIP	USB Interrupt Priority bit	1 = Haute priorité 0 = Basse priorité
4	EEIP	Data EEPROM/Flash Write Operation Interrupt Priority bit	1 = Haute priorité 0 = Basse priorité
3	BCLIP	Bus Collision Interrupt Priority bit	1 = Haute priorité 0 = Basse priorité
2	HLVDIP	High/Low-Voltage Detect Interrupt Priority bit	1 = Haute priorité 0 = Basse priorité
1	TMR3IP	Timer3 Overflow Interrupt Priority bit	1 = Haute priorité 0 = Basse priorité
0	CCP2IP	CCP2 Interrupt Priority bit	1 = Haute priorité 0 = Basse priorité

PIR Peripheral Interrupt <u>Request (flag)</u> Register			
Concerne : Interruptions des périphériques internes (activation)			
No. bit	Nom bit	Description (anglais)	Valeurs
7	SPPIF	Streaming Parallel Port Read/Write Interrupt flag bit	1 = Une interruption est survenue 0 = Aucune interruption n'est survenue
6	ADIF	A/D Converter Interrupt flag bit	1 = Une interruption est survenue 0 = Aucune interruption n'est survenue
5	RCIF	EUSART Receive Interrupt flag bit	1 = Une interruption est survenue 0 = Aucune interruption n'est survenue
4	TXIF	EUSART Transmit Interrupt flag bit	1 = Une interruption est survenue 0 = Aucune interruption n'est survenue
3	SSPIF	Master Synchronous Serial Port (MSSP) Interrupt flag bit	1 = Une interruption est survenue 0 = Aucune interruption n'est survenue
2	CCP1IF	CCP1 Interrupt flag bit	1 = Une interruption est survenue 0 = Aucune interruption n'est survenue
1	TMR2IF	Timer2 to PR2 Match Interrupt flag bit	1 = Une interruption est survenue 0 = Aucune interruption n'est survenue
0	TMR1IF	Timer1 Overflow Interrupt flag bit	1 = Une interruption est survenue 0 = Aucune interruption n'est survenue

PIR2 Peripheral Interrupt <u>Request (flag)</u> Register			
No. bit	Nom bit	Description (anglais)	Valeurs
7	OSCFIF	Oscillator Fail Interrupt flag bit	1 = Une interruption est survenue 0 = Aucune interruption n'est survenue
6	CMIF	Comparator Interrupt flag bit	1 = Une interruption est survenue 0 = Aucune interruption n'est survenue
5	USBIF	USB Interrupt flag bit	1 = Une interruption est survenue 0 = Aucune interruption n'est survenue
4	EEIF	Data EEPROM/Flash Write Operation Interrupt flag bit	1 = Une interruption est survenue 0 = Aucune interruption n'est survenue
3	BCLIF	Bus Collision Interrupt flag bit	1 = Une interruption est survenue 0 = Aucune interruption n'est survenue
2	HLVDIF	High/Low-Voltage Detect Interrupt flag bit	1 = Une interruption est survenue 0 = Aucune interruption n'est survenue
1	TMR3IF	Timer3 Overflow Interrupt flag bit	1 = Une interruption est survenue 0 = Aucune interruption n'est survenue
0	CCP2IF	CCP2 Interrupt flag bit	1 = Une interruption est survenue 0 = Aucune interruption n'est survenue

Les entrées analogiques

Les entrées analogiques touchent cinq registres. Il y a d'abord ADRESH et ADRESL qui contiennent le résultat sur des conversions Analogique/Numérique (dit A/D, de l'anglais Analog/Digital). Dans le cas d'une conversion sur dix bits, ADRESH contient les deux bits les plus significatifs et ADRESL contient les huit bits les moins significatifs. Pour une conversion sur huit bits, seul ADRESH est utilisé. Les trois autres registres sont ADCON0, ADCON1 et ADCON2 qui servent principalement à la configuration et au lancement des conversions A/D.

ADCON0

Bit <7:6>	Non implanté
------------------------	---------------------

Bit <5:2>	Sélection du canal analogique
0000	= Canal 0 (AN0)
0001	= Canal 1 (AN1)
0010	= Canal 2 (AN2)
0011	= Canal 3 (AN3)
0100	= Canal 4 (AN4)
0101	= Canal 5 (AN5)
0110	= Canal 6 (AN6)
0111	= Canal 7 (AN7)
1000	= Canal 8 (AN8)
1001	= Canal 9 (AN9)
1010	= Canal 10 (AN10)
1011	= Canal 11 (AN11)
1100	= Canal 12 (AN12)
1101	= <i>Non implanté</i>
1110	= <i>Non implanté</i>
1111	= <i>Non implanté</i>

Bit <1>	Bit d'état de conversion, aussi appelé GO/DONE - Mettre ce bit à 1 pour lancer la conversion. - Revient automatiquement à 0 lorsque la conversion est terminée.
1 = Conversion en cours 0 = Prêt pour conversion / Conversion terminée	

Bit <0>	Bit d'activation du convertisseur A/D
1 = Convertisseur activé 0 = Convertisseur désactivé	

ADCON1

Bit <7:6>	Non implanté
------------------------	---------------------

Bit <5>	Bit de configuration pour la référence du voltage négatif - Définit où prendre la référence pour le voltage négatif
1 = V_{REF-} (AN2)	
0 = V_{SS} (voltage négatif du microcontrôleur, soit 0 volt)	

Bit <4>	Bit de configuration pour la référence du voltage positif - Définit où prendre la référence pour le voltage positif
1 = V_{REF+} (AN3)	
0 = V_{DD} (voltage positif du microcontrôleur, soit 5 volts)	

Bit <3:0>	Contrôle de la configuration des ports A/D - Définit quels ports seront Analogiques et quels ports seront Numériques												
ADCON1 Bits <3:0>	AN12	AN11	AN10	AN9	AN8	AN7	AN6	AN5	AN4	AN3	AN2	AN1	AN0
0000	A	A	A	A	A	A	A	A	A	A	A	A	A
0001	A	A	A	A	A	A	A	A	A	A	A	A	A
0010	A	A	A	A	A	A	A	A	A	A	A	A	A
0011	N	A	A	A	A	A	A	A	A	A	A	A	A
0100	N	N	A	A	A	A	A	A	A	A	A	A	A
0101	N	N	N	A	A	A	A	A	A	A	A	A	A
0110	N	N	N	N	A	A	A	A	A	A	A	A	A
0111	N	N	N	N	N	A	A	A	A	A	A	A	A
1000	N	N	N	N	N	N	A	A	A	A	A	A	A
1001	N	N	N	N	N	N	N	A	A	A	A	A	A
1010	N	N	N	N	N	N	N	N	A	A	A	A	A
1011	N	N	N	N	N	N	N	N	N	A	A	A	A
1100	N	N	N	N	N	N	N	N	N	N	A	A	A
1101	N	N	N	N	N	N	N	N	N	N	N	A	A
1110	N	N	N	N	N	N	N	N	N	N	N	N	A
1111	N	N	N	N	N	N	N	N	N	N	N	N	N

ADCON2

Bit <7>	Configuration du format du résultat
1	Alignement à droite (Conversion sur dix bits)
0	Alignement à gauche (Conversion sur huit bits)

Bit <6>	Non implanté
---------	--------------

Bit <5>	Bit de configuration pour la référence du voltage négatif - Définit où prendre la référence pour le voltage négatif
1	V_{REF-} (AN2)
0	V_{SS} (voltage négatif du microcontrôleur, soit 0 volt)

Bit <5:3>	Configuration du temps d'acquisition sur le canal ANx choisi
111	20 TAD
110	16 TAD
101	12 TAD
100	8 TAD
011	6 TAD
010	4 TAD
001	2 TAD
000	0 TAD

Bit <2:0>	Configuration de l'horloge pour le convertisseur A/D
111	FRC (Horloge dérivée de l'oscillateur RC du convertisseur)
110	Horloge primaire / 64
101	Horloge primaire / 16
100	Horloge primaire / 4
011	FRC (Horloge dérivée de l'oscillateur RC du convertisseur)
010	Horloge primaire / 32
001	Horloge primaire / 8
000	Horloge primaire / 2