L'ESSENTIEL DES PICS: PERFORMANCES & SIMPLICITÉ

DIABLOTRONIC

Auteur: Galodé Alexandre

Date: 04/09/2006

NOTE

Ce livre a été écrit afin de disposer de l'essentiel à connaître sur les principaux PICs. On peut le considérer comme un databook simplifié et en français.

Le but de ce livre est donc de constituer en quelque sorte un livre de références pour comprendre, programmer et tester les PICs les plus performants & les plus répandus (12F683, 16F88, 18F2550 et par extension le 18F4550), lesquels peuvent répondre à de nombreux besoins. Un livre à la fois le plus simple & compréhensible possible, mais aussi, je l'espère, efficace. Le but n'est pas ici de faire un livre avec des tonnes de documentations ne servant jamais à rien, mais au contraire, un livre avec uniquement les informations utiles et nécessaires à la compréhension et à la mise en œuvre des principaux PICs. Nous ne nous attarderons donc pas sur des schémas détaillés, mais uniquement sur des représentations schématiques, suffisantes à son utilisation. Une fois cet ouvrage lu, à l'aide de ces PICs, vous pourrez alors concevoir nombre de montages.

Il vous sera cependant nécessaire d'avoir un minimum de connaissances en électronique et informatique afin de comprendre l'intégralité de ce livre.

Ce livre vous présentera un programmateur de PIC Mircochip, et un compilateur fort simple d'emploi qui rendra transparent à vos yeux l'utilisation des diverses fonctions du PIC.

Ce livre s'adresse donc aussi bien aux amateurs avertis, qu'aux étudiant(e)s et/ou professionnels.

Bonne lecture !!!

SOMMAIRE

1- Introduction P5

2- Généralités P9
- 2.1 Reset P10
- 2.2 Watchdog P10
- 2.3 RTCC P10
- 2.4 Mémoire flash P11
- 2.5 E²PROM P11
- 2.6 Divers P12

3- Le PIC 12F683 P13
- 3.1 Caractéristiques du PIC 12F683 P14
- 3.2 Fonctionnement du PIC 12F683 P15
  - 3.2.1 Les entrées/sorties
  - 3.2.2 Les différents mode d'horloge
  - 3.2.3 Les interruptions
  - 3.2.4 Les timers
  - 3.2.5 Le comparateur
  - 3.2.6 La PWM
  - 3.2.7 Les CAN
  - 3.2.8 La tension de référence interne

4- Le PIC 16F88 P26
- 4.1 Caractéristiques du PIC 16F88 P27
- 4.2 Fonctionnement du PIC 16F88 P29
  - 4.2.1 Les entrées/sorties
  - 4.2.2 Les différents mode d'horloge
  - 4.2.3 Les interruptions
  - 4.2.4 Les timers
  - 4.2.5 L'USART, I²C, SPI
  - 4.2.6 La tension de référence
  - 4.2.7 Les comparateurs
  - 4.2.8 La PWM
  - 4.2.9 Les CAN

5- Le PIC 18F2550 P42
- 5.1 Caractéristiques du PIC 18F2550 P43
- 5.2 Fonctionnement du PIC 18F2550 P45
  - 5.2.1 Les entrées/sorties
  - 5.2.2 Les différents mode d'horloge
  - 5.2.3 Les interruptions
  - 5.2.4 Les timers
  - 5.2.5 L'EUSART, I²C, SPI, USB
  - 5.2.6 La tension de référence
  - 5.2.7 Les comparateurs
  - 5.2.8 La PWM
  - 5.2.9 Les CAN

6- Les protocoles de communication P61
6.1 Série asynchrone, RS 232 P62
6.2 I²C, de Philips P63
6.2.1 Principe
6.2.2 Les mémoires
6.3 SPI P69
6.4 USB, HID P70
6.5 RC5 P70
6.6 Les bases du CPL P72

7- La programmation des PICs P75
7.1 Nécessaire à la programmation P76
7.2 Choix d'un langage P77
7.3 Le BASIC pour PIC P78
7.4 Un programmateur de PIC ICSP et son logiciel P79

8- Conclusion P80

9- Lexique P81

10- Liens Internet utiles P82

11- Annexes P83

1- INTRODUCTION

Deux exemples de microcontrôleurs

Aujourd'hui, les microcontrôleurs sont partout : ordinateurs, portables…

Assez faciles d'utilisation, pour la plupart, et programmables de fort nombreuses fois (plus de 1000), ou définitivement (OTP : One Time Programmable), leur souplesse d'utilisation a séduit rapidement les divers constructeurs de divers domaines. Aujourd'hui, un des géants mondiaux s'appelle Microchip.

Mais il ne faut surtout pas confondre les microcontrôleurs et les microprocesseurs. Pour résumer, on peut dire qu'un microcontrôleur est un ordinateur extrêmement miniaturisé et possédant donc assez peu de mémoire, et dont le processeur est relativement simple, alors qu'un microprocesseur ne fait qu'exécuter des instructions qui lui sont communiquées, puis renvoie les résultats.

Les principaux problèmes des microcontrôleurs sont la taille de leur mémoire et le nombre limité de périphériques qu'ils peuvent recevoir en même temps. Cependant, le nombre de ces derniers peut parfois être augmenté en associant, sur les mêmes pattes un périphérique d'entrée et un de sortie, permettant alors de doubler le nombre de périphériques connectables…

Les microcontrôleurs sont tellement miniatures, que la plupart du temps, ils sont implantés sur l'application même qu'ils sont censés piloter, comme par exemple un clavier d'ordinateur, ou la souris.

Ces systèmes sont alors appelés « systèmes embarqués ». Ils exécutent alors tout le temps, en boucle, le même programme.

Leur champ d'application ne connaît comme limite, que l'ingéniosité des divers concepteurs. Grâce à eux, la réception radio est de plus en plus fine; et leur avenir s'annonce radieux en ce début de XXI siècle.

Un microcontrôleur se décompose en diverses parties : la mémoire de programme, la mémoire de données, le processeur, les ressources auxiliaires.

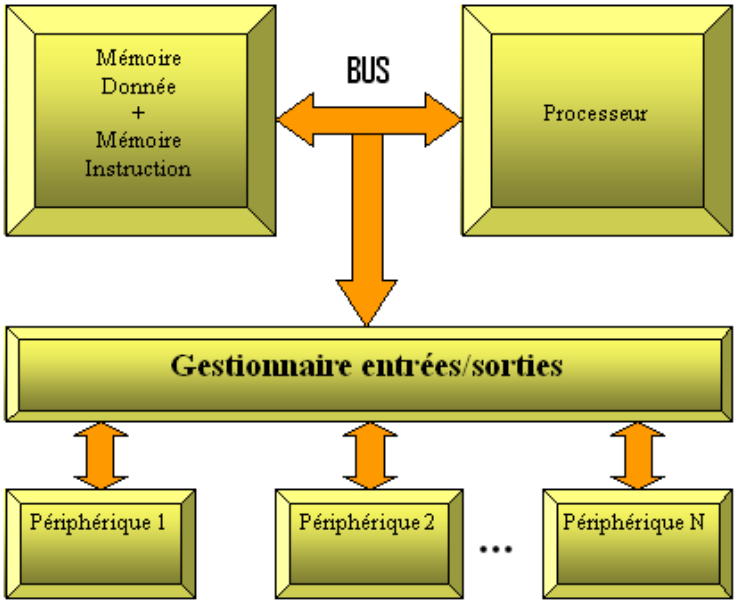
Les modèles les plus vendus sont les PICs de chez Microchip, en raison d'un excellent rapport qualité/prix et qui sont utilisés par toute la communauté électronique, que ce soit amateur, ou même par certains professionnels. Facile à programmer et à utiliser, leur prix est relativement bas, avec des fréquences de fonctionnement élevées.

Chez les PICs, il y a principalement 5 gammes (plus si on rajoute les DSPICs). Il existe la gamme 10F, 12F,14F,16F,18F.

Pour des raisons de simplicité, il est évident qu'il est préférable d'avoir quelques références permettant de répondre aux diverses attentes. Dans cette optique, 3 références de PICs ont été retenues. Le 1er critère de sélection était le nombre de pattes (taille du CI), le 2nd était les caractéristiques techniques du PIC.

Les PICs ainsi retenus sont le 12F683, le 16F88, le 18F2550. Par extension, les explications pour le 18F2550 seront valables pour le 18F4550, la différence étant au niveau des E/S.

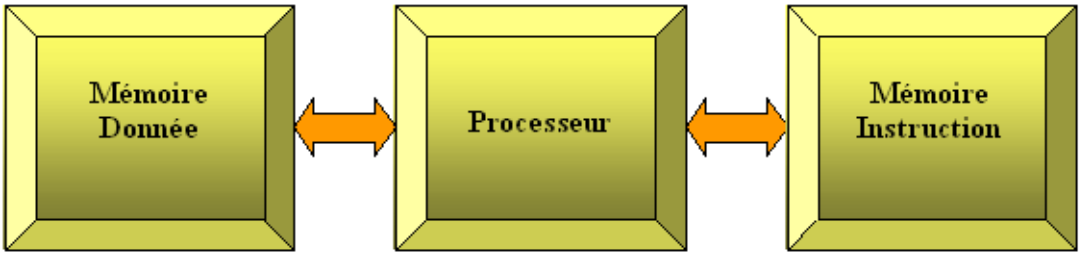
Jusqu'à une certaine époque, les microcontrôleurs respectaient l'architecture Von Neumann, inventeur de l'ENIAC, premier ordinateur au monde. Cependant, celle-ci présente des inconvénients. En effet, la vitesse d'exécution est limitée, les instructions et les données transitaient par le même bus. Pour résumer, son principal défaut était le fait qu'elle ne possédait qu'un bus pour, simultanément, la mémoire programme et la mémoire de données.



L'architecture de Von Neumann

D'où l'architecture Harvard, utilisée maintenant par les PICs. Sa particularité tient dans le fait qu'il y a deux mémoires accessibles en même temps par le processeur, par l'intermédiaire de deux bus spécifiques.

L'un sert pour les données, et l'autre pour les instructions. De ce fait, les deux peuvent être accessibles en même temps, d'où un gain de vitesse, au niveau exécution.



L'architecture Harvard

Ceci aidant, il existe depuis quelques années, un nouveau type de mémoires dites « flash », avec écriture et effacement électrique des données dans la mémoire. Elle est de type RAM, mais est associée à une mémoire E^2PROM pour des données auxiliaires.

Cependant, d'autres architectures existent, mais néanmoins moins répandues, comme par exemple, l'architecture Flynn développée en 1996.

Chaque microcontrôleur possède des registres. Il s'agit de mémoires intervenant dans les opérations de l'UAL (Unité Arithmétique et Logique). Celle-ci requiert deux opérandes (instructions) pour réaliser une opération. Quand un processeur exécute une instruction, il le fait en deux étapes : d'abord une phase de recherche, puis une phase d'exécution.

En plus du programme, les microcontrôleurs ont besoin de ce qu'on appelle un mot de configuration, tenant compte du type d'oscillateur (RC, XT,…), de l'activation ou non du Watchdog (qui empêche un dysfonctionnement), d'une temporisation au démarrage ou non, du code P (non utilisé et explicité dans ce livre, le code P, pour Protection, sert à verrouiller la programmation du PIC ; c'est-à-dire que pour le programmer, il faut un mot de passe), ...

2- Généralités

2.1 Reset

2.2 Watchdog

2.3 RTCC

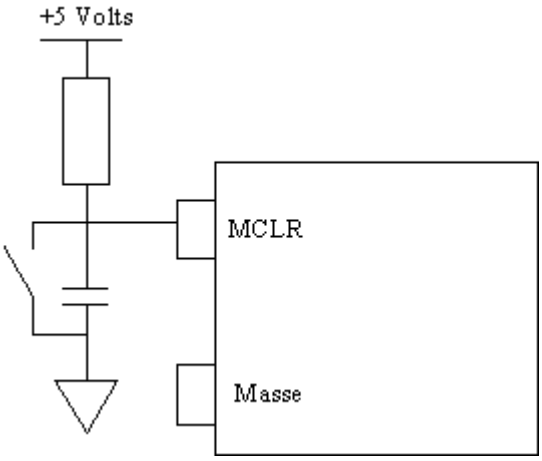
2.4 Mémoire flash

2.5 E²PROM

2.6 Divers

2- GÉNÉRALITÉS

2.1 Reset



Le schéma type du reset: une résistance de 1KOhms et un condensateur de 1 µF

Ce système électronique, permet d'effectuer un reset automatique à la mise sous tension, et laisse l'opportunité d'effectuer un reset manuel par l'intermédiaire de l'interrupteur.

IL existe cependant, une autre façon de faire un reset. En effet, dans les fusibles de configuration du PIC, existe le BODEN. Permettant s'il est activé d'effectuer un reset hardware automatique si le programme présente un problème d'exécution. Une autre solution est d'utiliser le WATCHDOG, qui assure également un reset en cas de problème du programme, mais ce reset est alors logiciel.

2.2 Watchdog

Il s'agit d'une horloge interne. Cette horloge compte et est régulièrement remise à zéro. Dans le cas où le compteur de l'horloge ferait un "overflow", c'est-à-dire qu'il dépasserait sa valeur maximale, alors cela voudrait dire qu'il y a eu un problème dans l'exécution du programme. Le WATCHDOG effectue alors un reset logiciel.

2.3 RTCC

Le RTCC est assez rarement utilisé pour les programmes classiques. Mais pour résumer il s'agit en fait d'une horloge temps réel

servant pour certains programmes et étant rattachée généralement au timer 0.

2.4 Mémoire flash

La mémoire flash est le nouveau type de mémoire E²PROM. Bien plus souple que les premières générations de ces dernières, la mémoire flash permet une écriture/effacement de toute la mémoire ou que d'une partie au choix. Ce type de mémoire possède beaucoup de caractéristiques intéressantes.

Caractéristiques de la mémoire Flash
1-Pas de différence significative entre les différents types de mémoires existantes sur le marché
2-Utilisation simplifié par rapport aux autres mémoires E²PROM
3-Faible coût des outils de développement
4-Simplification du débogage
5-Possibilité de misa a jour du Firmware
6– Grande plage de tension de programmation.

Caractéristiques	Mémoire flash
Alimentation	2-5,5 volts
Tension de programmation	Vdd
Autoprogrammable	OUI
Débogage en circuit	OUI
Technologie	0,5µm
Cycles d'effacement/écriture	100K(données), 1K(programme)
EEPROM Données	OUI
Temps de programmation/cycle	1-2 ms

2.5 E²PROM

L'E²PROM (Electrically Erasable Programmable Read Only Memory) est une mémoire interne au Pics. Il s'agit d'une mémoire non volatile dans laquelle le PIC peut stocker des données, comme par exemple les résultats d'une acquisition. Son utilisation, avec le compilateur retenu, est extrêmement simple. Sa taille varie d'un PIC à l'autre. Il est également possible de rajouter de la mémoire en externe.

2.6 Divers

Il faut savoir que les PICs ne font pas la différence entre les niveaux logiques TTL et CMOS. Il est donc alors préférable d'utiliser les deux extrêmes de signaux logiques TTL (0 et 5 volts), pour éviter tous problèmes possibles.

On ne peut affecter que deux valeurs différentes de configuration à chaque patte : un '1' pour la mettre en entrée, ou un '0' pour une sortie.

Enfin, autre point fort du PIC : la consommation, car en tant que système embarqué, il faut que ce microcontrôleur consomme peu.
Ainsi, des quelques mA de consommation en fonctionnement normal, il peut passer sous les 10 µA en fonction veille ou sommeil, comme par exemple sur un téléphone portable, quand on ne s'en sert pas, la lumière reste éteinte : il est en veille. En revanche, au moindre appui sur un bouton (ex: interruption externe), il se remet en marche. Le Pics fonctionne de la même manière.

Remarque : il faut savoir que le PIC peut fournir jusqu'à 20 mA par sortie.

Chaque patte possède généralement plusieurs fonctions, laissant ainsi de nombreuses possibilités de mise en oeuvre.

Lors de la présentation des registres utiles, les cases en orange correspondront aux cases n'ayant pas d'utilité. Il est conseillé de les considérer à 0.

Les explications détaillées ici le sont principalement à titre d'information. En effet, la plupart des compilateurs rendent la majorité des actions transparentes à l'utilisateur, permettant ainsi un développement rapide. Cependant, ces informations peuvent servir pour détecter un éventuel bug.

CHAPITRE 3 : LE PIC 12F683

3.1 Caractéristiques du PIC 12F683

3.2 Fonctionnement du PIC 12F683

3.2.1 Les entrées/sorties

3.2.2 Les différents modes d'horloge

3.2.3 Les interruptions

3.2.4 Les Timers

3.2.5 Le comparateur

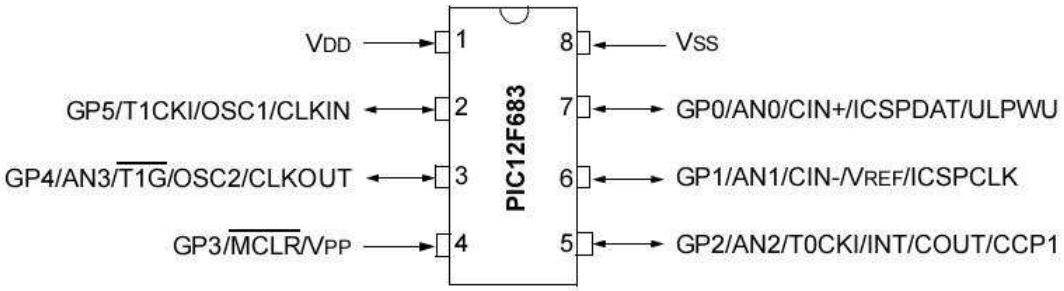
3.2.6 La PWM

3.2.7 Les CAN

3.2.8 La tension de référence interne

3- LE PIC 12F683

3.1- Caractéristiques du PIC 12F683



Ce microcontrôleur, possède jusqu'à 6 broches configurables, réparties sur un port unique (le port GP), le tout en boitier DIL 8 broches.

Alimentable de 2 à 6 V continu, il est compatible avec le mode de programmation ICSP (In Chip Serial Programming), et dispose d'un oscillateur interne pouvant monter jusqu'à 8 Mhz. Sur oscillateur externe, le PIC peut monter jusqu'à 20 MHz.

Au niveau mémoire, 2KO de mémoire flash sont disponibles pour accueillir le firmware, et 256 octets d'E²PROM peuvent accueillir d'éventuelles données en stockage. Enfin, 128 octets de RAM sont disponibles pour les variables et autres constantes du firmware.

Côté fonctionnalité, outre les E/S numériques, le 12F683 dispose de 4 CAN, d'1 comparateur aux configurations diverses et variées, de 3 timers, d'une PWM 10 bits, de 10 sources d'interruptions programmes, ...

Caractéristiques	12F683
Broches	8
E/S max	6
µy flash	2 KO
µy E²PROM	256 O
CAN	4
PWM	1 de 10 bits
TIMER	3
Comparateur	1
Interruptions	10
Oscillateur	20 MHz MAX

3.2- Fonctionnement du PIC 12F683

3.2.1- Les entrées/sorties

Dans cette partie, nous allons voir les différentes possibilités de chaque patte, en les désignant par leur numéro. A noter, qu'une seule fonction est disponible à la fois, par patte.

PATTE 1 : Alimentation positive

PATTE 2 : E/S GP5, entrée logique du timer 1, entrée quartz 1, entrée oscillateur RC

PATTE 3 : E/S GP4, CAN 3, entrée logique timer 1, entrée quartz 2, sortie horloge

PATTE 4 : E GP3, reset externe

PATTE 5 : E/S GP2, CAN 2, entrée timer 0, interruption externe, sortie comparateur, PWM

PATTE 6 : E/S GP1, CAN 1, entrée negative comparateur, tension de référence

PATTE 7 : E/S GP0, CAN 0, entrée positive comparateur, patte de réveil en mode ultra basse consommation

PATTE 8 : Masse

Les E/S se configurent via le registre TRISIO.

3.2.2- Les modes d'horloge

Dans cette partie, nous allons voir les différents modes d'horloge. Dans le cas d'un quartz (mode1), vous pourrez choisir horloge XT (jusqu'à 4 Mhz), ou HS (jusqu'à 20 MHz).

MODE 1 : EC, horloge externe avec GP4 en E/S

MODE 2 : LP, quartz low power à 32 KHz

MODE 3 : XT, quartz ou resonnateur céramique jusqu'à 4 MHz

MODE 4 : HS, quartz ou résonnateur céramique jusqu'à 20 MHz

MODE 5 : RC, oscillateur externe sur OSCIN, sortie de F/4 sur CLKOUT

MODE 6 : RCIO, idem mode 5, mais avec OSC2/CLKOUT en E/S

MODE 7 : INTOSC, oscillateur interne avec GP5 en E/S et F/4 sur OSC2

MODE 8 : INTOSCIO, idem mode 7, mais avec GP4 en E/S

Le mode est configurable sur les 3 derniers bits du registre CONFIG, de 16 bits, dédiés à la configuration du PIC. L'utilisation de ce registre est transparent à travers les options du compilateur.

FOSC2	FOSC1	FOSC0

Les deux registres qui peuvent éventuellement réellement servir sont le OSCON et le OSCTUNE.

OSCON

	IRCF2	IRCF1	IRCF0	OSTS	HTS	LTS	SCS

IRCF2-0: fréquence de l'oscillateur interne (voir tableau)
OSTS: en lecture seulement
HTS: en lecture seulement
LTS: en lecture seulement
SCS: horloge interne (1) ou définie dans CONFIG

Bit Value	Fréquence
000	31 KHz
001	125 KHz
010	250 KHz
011	500 KHz
100	1 MHZ
101	2 MHz
110	4 MHz
111	8 MHz

OSCTUNE

			TUN4	TUN3	TUN2	TUN1	TUN0

TUN4-0: de 01111 (max) à 00000 (défaut, neutre) puis 11111 à 10000 (min)

Dans le cas de l'utilisation de l'oscillateur interne, OSCTUNE permet d'ajuster précisément la fréquence.

3.2.3- Les Interruptions

Sur ce modèle, ils sont au nombre de 10.

A chaque source d'interruption activée, correspond un drapeau, un "flag", permettant de savoir si l'interruption a eu lieu. Dans les registres, les bits terminant par un E, sauf précision contraire, correspondent à une interruption. Les bits terminant par F correspondent aux drapeaux. Chaque drapeau, une fois passé à "1" est à remettre à zéro de manière logicielle.

Ainsi pour savoir si une interruption a eu lieu, il suffit de surveiller le flag de l'interruption concernée.

La mise en marche des interruptions se fait via le bit GIE du registre INTCON.

Voici les registres concernés, avec les explications de chaque bit. Nous ne détaillerons pas les bits flags (finissant par F).

INTCON:

GIE	PEIE	T0IE	INTE	GPIE	T0IF	INTF	GPIF

GIE: active ou non les interruptions INTCON du PIC
PEIE: active ou non les interruptions du registre PIE1 du PIC
T0IE: interruption du timer 0
INTE: interruption externe (GP2)
GPIE: interruption de changement des entrées du port GP (avec IOC)

PIE1:

EEIE	ADIE	CCP1IE		CMIE	OSFIE	TMR2IE	TMR1IE

EEIE: interruption d'écriture E²PROM
ADIE: interruption de fin de conversion AN
CCP1IE: interruption du module PWM
CMIE: interruption de fin de comparaison
OSFIE: interruption de problème d'oscillateur
TMR2IE: interruption de comparaison timer 2, PR2
TMR1IE: interruption de timer 1

PIR1:

EEIF	ADIF	CCP1IF		CMIF	OSFIF	TMR2IF	TMR1IF

Le registre PIR1 contient tous les flags du registre PIE1.

IOC:

		IOC5	IOC4	IOC3	IOC2	IOC1	IOC0

Ce registre permet de sélectionner les entrées à associer à l'interruption GPIE du registre INTCON.

3.2.4- Les TIMERS

Ils sont au nombre de 3. Un timer permet de compter le temps, ou des impulsions. Chaque timer possède des spécificités précises. La compréhension à 100% des timers est assez difficile. Heureusement, la plupart des compilateurs nous permettent de nous en servir de manière relativement tranparente.

TIMER 0

Ce timer est le timer de base, et se configure via le registre OPTION_REG.

Timer 8 bits, le timer 0 est un compteur pouvant être initialisé à une valeur donnée, via le registre TMR0 (charger la valeur désirée dans ce registre). Lorsque l'interruption du timer 0 est activée, le passage de 0xFF à 0x00 du timer provoque l'activation du flag du timer0.

OPTION_REG

GPPU\	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0

GPPU\: GPIO pull up activé (1) ou désactivé (0)
INTEDG: interruption sur front montant (1) ou descendant (0) de GP2
T0CS: utilisation de l'horloge interne (0, f/4) ou de l'entrée T0CKI (1)
T0SE: incrémentation sur front montant (0) ou descendant (1) de T0CKI
PSA: prédiviseur pour le watchdog (1) ou pour le timer 0 (0)
PS2-PS0: prédiviseur (voir tableau ci dessous)

Bit Value	Timer 0	Watchdog
000	1:2	1:1
001	1:4	1:2
010	1:8	1:4
011	1:16	1:8
100	1:32	1:16
101	1:64	1:32
110	1:128	1:64
111	1:256	1:128

***Remarque:* le INTEDG ne sert que si l'interruption sur GP2 est activée.**

TIMER 1

Second timer, il est sur 16 bits. Ce registre se configure via T1CON. Tout comme le timer 0, le timer 1 est initialisable en chargeant la valeur désirée dans les 2 registres dédiés: TMR1H pour l'octet supérieur, et TMR1L pour l'octet inférieur. A noter que l'entrée timer 1 (autre que T1CKI), est en fait une porte logique inverseuse.

T1CON

T1GINV	TMR1GE	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC\	TMR1CS	TMR1ON

T1GINV:entrée logique du timer 1 active à l'état haut (1) ou bas (attention, porte inverseuse)
TMR1GE: n'est utile que si TMR1ON est activé. Timer 1 actif (0), timer 1 actif uniquement quand l'entrée timer 1 est désactivée (1)
T1CKPS1-0: prédiviseur du timer 1 (voir tableau ci-après)
T1OSCEN: actif uniquement en mode 7 d'horloge

T1SYNC\: actif si TMR1CS=1. Le timer 1 se synchronise sur l'entrée horloge externe (0) ou non (1)
TMR1CS: fonctionnement sur horloge interne (0,f/4), ou T1CKI (1)
TMR1ON: active (1) ou non le timer 1

Bit Value	Timer 1
00	1:1
01	1:2
10	1:4
11	1:8

TIMER 2

Troisième timer du PIC, et second timer 8 bits, le timer 2 est notamment utilisé pour générer la PWM. Il possède également, en plus d'un prédiviseur, un postdiviseur. Il est lui initialisable via le registre TMR2. La période du timer 2 peut être configurée via PR2 (1 octet).

T2CON

	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0

TOUTPS3-0: postdiviseur (binaire)
TMR2ON: active (1) ou non le timer 2
T2CKPS1-0: prédiviseur

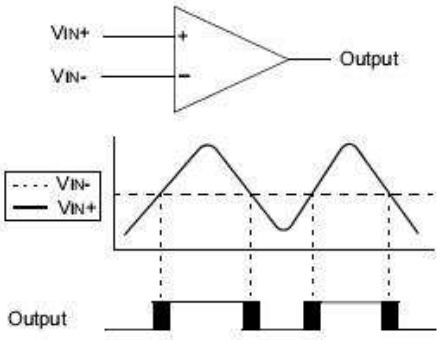
Bit Value	Timer 2
00	1:1
01	1:4
10	1:16
11	1:16

Bit Value	postdiviseur
0000	1:1
0001	1:2
...	...
1111	1:16

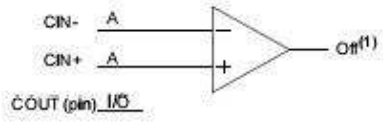
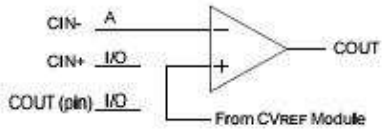
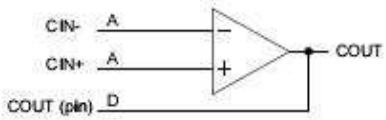
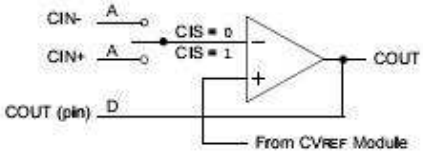
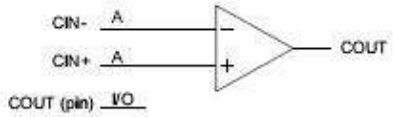
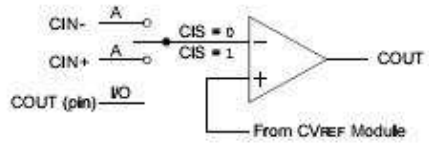
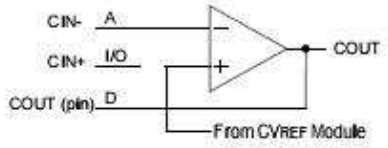
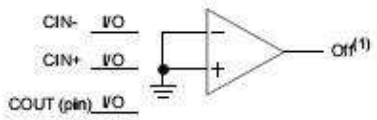
3.2.5- Le comparateur

La présence d'un comparateur analogique est fort pratique. En effet, cette fonctionnalité permet de réaliser la comparaison et de connaître au niveau logiciel & hardware le résultat de façon simple.

Le principe d'un comparateur, fort simple, est le suivant:



Si V+>V- alors la sortie =1, 0 sinon. Ce comparateur permet de comparer diverses choses en fonction du mode de configuration.

<table>
<tr>
<td>Comparator Reset (POR Default Value – low power)
CM<2:0> = 000
 
</td>
<td>Comparator w/o Output and with Internal Reference
CM<2:0> = 100
 
</td>
</tr>
<tr>
<td>Comparator with Output
CM<2:0> = 001
 
</td>
<td>Multiplexed Input with Internal Reference and Output
CM<2:0> = 101
 
</td>
</tr>
<tr>
<td>Comparator without Output
CM<2:0> = 010
 
</td>
<td>Multiplexed Input with Internal Reference
CM<2:0> = 110
 
</td>
</tr>
<tr>
<td>Comparator with Output and Internal Reference
CM<2:0> = 011
 
</td>
<td>Comparator Off (Lowest power)
CM<2:0> = 111
 
</td>
</tr>
<tr>
<td colspan="2">Legend: A = Analog Input, ports always reads '0'
I/O = Normal port I/O
CIS = Comparator Input Switch (CMCON0<3>)
D = Comparator Digital Output
Note 1: Reads as '0', unless CINV = 1.</td>
</tr>
</table>

Afin de configurer le comparateur dans le mode désiré, il faut configurer 2 registres: CMCON0 & CMCON1. Le CVREF est une référence de tension (voir 3.2.8).

CMCON0

	COUT		CINV	CIS	CM2	CM1	CM0

COUT: en lecture seulement, permet de connaître logiciellement l'état de la comparaison (sortie du comparateur)
CINV: inverse l'état de la sortie (1) ou non
CIS: sélection d'entrée dans certains modes, voir schémas précédents
CM2-0: configuration du comparateur, voir schémas précédents

CMCON1

						T1GSS	CMSYNC

T1GSS: à 0 l'entrée logique du timer 1 est connectée sur la sortie du comparateur, à 1 l'entrée logique est connectée à T1G\
CMSYNC: à 0, la sortie est asynchrone (ne dépend d'aucune horloge particulière), à 1 se synchronise sur le front descendant de l'entrée logique du timer 1

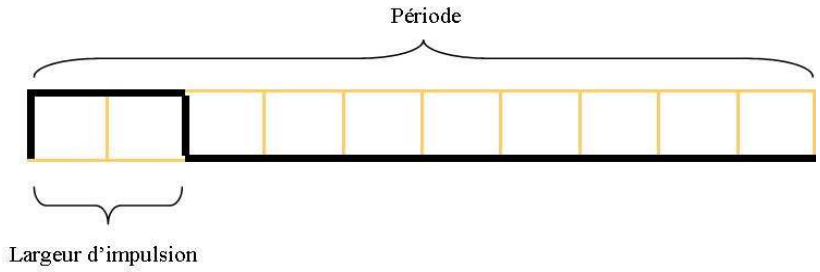
3.2.6- La PWM

La PWM fait partie d'un bloc interne du PIC: le CCP; ou Capture/Compare/PWM. Les deux premiers modes assez flous et d'une approche assez difficile ne seront pas vus ici, car très peu utilisés. Nous nous concentrerons sur la PWM, plus répendue.

La PWM pour Pulse Width Modulation, ou en français MLI pour Modulation en Largeur d'Impulsion, permet de gérer l'énergie transmise à l'extérieur. En effet, si un signal continu correspond à 100% d'énergie, un signal carré dont la durée d'état haut égale celle d'état bas correspond à 50 % d'énergie. Le pourcentage d'énergie transmis se calcule en faisant le rapport de la durée d'état haut sur la durée de la période.

La PWM peut permettre de gérer la luminosité d'une lampe, le pilotage d'un servomoteur de modélisme, contrôler la vitesse d'un moteur continu, ...

Voici une période PWM:



Deux formules permettent de calculer la durée de la période, et la largeur de l'impulsion.

PERIODE=(PR2+1)*4*Tosc*(prédiviseur timer 2)
LARGEUR=(10 bits)*Tosc*(prédiviseur timer 2)

Par Tosc, nous comprenons la période de l'oscillateur servant d'horloge au PIC. Les 10 bits correspondent à l'octet de CCPR1L et des 2 bits de DC1Bx.

Cependant cette résolution de 10 bits est la valeur maximale. En effet, selon la fréquence désirée, cette résolution peut changer. Pour connaître la résolution maximale, on utilise la formule suivante:

RESOLUTION=(log(4*(PR2+1)))/log(2) bits.

Par exemple, avec le prédiviseur timer 2 à 16, et un PR2 à 255 (0XFF), nous avons une résolution maximale de 10 bits, et une fréquence de PWM de 1,22 Khz.

Autre exemple, avec un prédiviseur à 1, et un PR2 à 31, la résolution maximale est alors de 7 bits, et une fréquence de PWM de 156,3 KHz.

Ici, les registres utiles sont le CCP1CON, PR2, CCPR1L. De plus, comme précisé précédemment, c'est le timer 2 qui est utilisé ici. De fait, il faudra également utiliser T2CON, afin d'activer le timer 2. Cependant, la plupart des compilateurs rendent toutes ces actions transparentes.

CCP1CON

		DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0

DC1B1-0: contient les 2 bits de poids faibles de la période (PWM 10 bits)
CCP1M3-0: 2 solutions pour la PWM, 110x pour une PWM active à l'état haut, out 111x pour une PWM active à l'état bas.

Les 8 autres bits doivent être chargés dans le registre CCPR1L.

3.2.7- Les CAN

Le PIC dispose de 4 CAN, répartis sur ses entrées.

2 registres permettent la configuration de ces convertisseurs, et 2 autres regsitres permmettent de récupérer les résultats (sur 10 bits).

ADCON0

ADFM	VCFG			CHS1	CHS0	GO-DONE\	ADON

ADFM: aligne les résultats à gauche (0) ou à droite
VCFG: sélection de la référence tension : VDD (0) ou entrée vref (1)
CHS1-0: sélection du CAN (00 pour le 0 à 11 pour le 3)
GO-DONE\: à 1, la conversion est en cours, à 0, elle est finie
ADON: activation (1) ou non des CAN

ANSEL

	ADCS2	ADCS1	ADCS0	ANS3	ANS2	ANS1	ANS0

ADCS2-0: détermine l'échantillonange des CAN (voir tableau ci-après)
ANS3-0: pour chaque CAN, un 1 configure l'entrée en CAN, un 0 en E/S numérique

Bit Value	Fréquence
000	f/2
001	f/8
010	f/32
x11	500 KHz max
100	f/4
101	f/16
110	f/64

<u>Remarque:</u> sauf exception particulière, vous pouvez laisser l'échantillonnage à sa valeur par défaut.

2 autres registres permettent de récupérer la valeur de la conversion: ADRESH et ADRESL. Pour des raisons de commodités de traitement de la conversion, 2 cas sont possibles selon la valeur de ADFM (ADRESH en haut)

ADFM=1

						ADR9	ADR8
ADR7	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	ADR0

ADFM=0

ADR9	ADR8	ADR7	ADR6	ADR5	ADR4	ADR3	ADR2
ADR1	ADR0						

3.2.8- La tension de référence interne

Le 12F683 possède une tension de référence interne, utilisant un réseau de résistances. Cette tension est configurable via VRCON, et est disponible sur GP1. Toutefois cette source est dédiée au comparateur (d'ou le C de CVREF)

VRCON

VREN		VRR		VR3	VR2	VR1	VR0

<u>VREN:</u> active (1) ou non la référence de tension
<u>VRR:</u> sélection de la formule de calcul de tension
<u>VR3-0:</u> si VRR=1, alors CVREF=(VR3-0 /24)*VDD, si VRR=0, alors CVREF=VDD/4+(VR3-0/32)*VDD

CHAPITRE 4 : LE PIC 16F88

4.1 Caractéristiques du PIC 16F88

4.2 Fonctionnement du PIC 16F88

4.2.1 Les entrées/sorties

4.2.2 Les différents modes d'horloge

4.2.3 Les interruptions

4.2.4 Les Timers

4.2.5 L'USART, I²C, SPI

4.2.6 La tension de référence

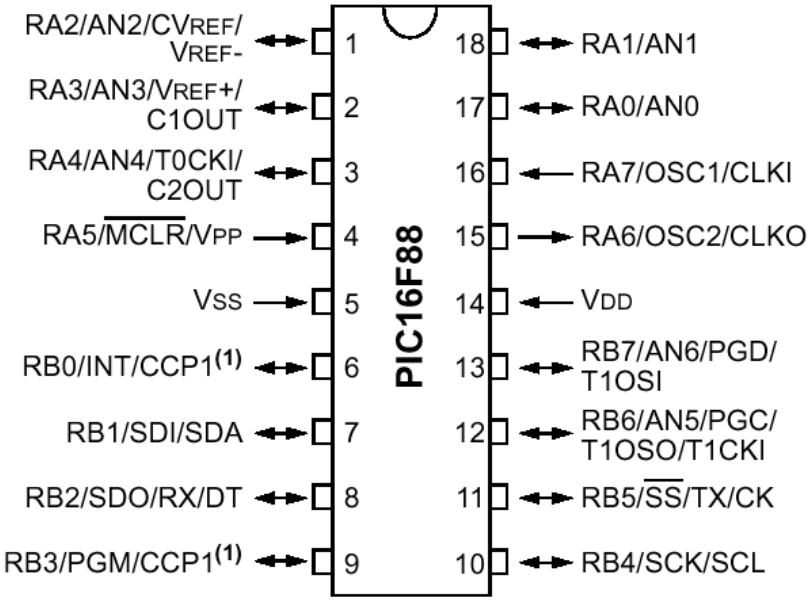
4.2.7 Les comparateurs

4.2.8 La PWM

4.2.9 Les CAN

4- LE PIC 16F88

4.1- Caractéristiques du PIC 16F88



Ce microcontrôleur, possède jusqu'à 16 broches configurables, réparties sur 2 ports, le tout en boîtier DIL 18 broches.

Alimentable de 2 à 6 V continu, il est compatible avec le mode de programmation ICSP (In Chip Serial Programming), et dispose d'un oscillateur interne pouvant monter jusqu'à 8 Mhz. Sur oscillateur externe, le PIC peut monter jusqu'à 20 MHz.

Au niveau mémoire, ce PIC dispose de 4 KO de mémoire flash pour le firmware, de 368 octets de RAM et de 256 octets d'E²PROM.

Côté fonctionnalité, outre les E/S numériques classiques, le 16f88 dispose de 7 CAN, de 2 comparateurs, d'1 PWM 10 bits, d'1 module de communication série synchrone et asynchrone, de 3 timers, ...

Et outre cela, le 16F88 est également compatible broche à broche avec le 16F84 et le 16F628, et peut donc permettre un remplacement de ces derniers. En effet, pour rappel, le 16F84 n'est plus fabriqué, et si vous en trouvez encore chez votre revendeur, il s'agit de stock.

<u>Remarque:</u> dans le cas où RA5,6 &7 sont utilisées en E/S, sachez que RA5 & 7 ne peuvent être que des entrées, et RA6 qu'une sortie.

Voici un tableau récapitulatif des principales caractéristiques du 16F88:

Caractéristiques	16F88
Broches	18
E/S max	16
µy flash	4 KO
µy E²PROM	256 O
CAN	7
PWM	1 de 10 bits
TIMER	3
Comparateur	2
Interruptions	13
Oscillateur	20 MHz MAX

4.2- Fonctionnement du PIC 16F88

4.2.1- Les entrées/sorties

Dans cette partie, nous allons voir les différentes possibilités de chaque patte, en les désignant par leur numéro. A noter, qu'une seule fonction est disponible à la fois, par patte.

PATTE 1: E/S numérique RA2, CAN2, CVREF (référence comparateur), Vref- (référence basse CAN), entrée + du comparateur 2

PATTE 2: E/S numérique RA3, CAN3, VREF+ (référence haute CAN), sortie du comparateur 1, entrée + du comparateur 1

PATTE 3: E/S numérique RA4, CAN4, entrée de comptage timer 0, sortie du comparateur 2

PATTE 4: E numérique RA5, reset

PATTE 5: Masse

PATTE 6: E/S numérique RB0, interruption externe, sortie PWM

PATTE 7: E/S numérique RB1, entrée data SPI, data I²C

PATTE 8: E/S numérique RB2, sortie data SPI, entrée asynchrone, détection synchrone

PATTE 9: E/S numérique RB3, sortie PWM

PATTE 10: E/S numérique RB4, horloge SPI, horloge I²C

PATTE 11: E/S numérique RB5, sélection esclave SPI, sortie asynchrone, horloge synchrone

PATTE 12: E/S numérique RB6, CAN5, sortie oscillateur timer1, entrée horloge timer 1

PATTE 13: E/S numérique RB7, CAN6, entrée oscillateur externe timer 1

PATTE 14: Alimentation positive, 5V

PATTE 15: S numérique RA6, entrée quartz, sortie horloge

PATTE 16: E numérique RA7, entrée quartz, entrée horloge externe

PATTE 17: E/S numérique RA0, CAN0, entrée – du comparateur 1

PATTE 18: E/S numérique RA1, CAN1, entrée – du comparateur 2

Les E/S se configurent via le registre TRISA, TRISB.

4.2.2- Les modes d'horloge

Dans cette partie, nous allons voir les différents modes d'horloge. Dans le cas d'un quartz (mode1), vous pourrez choisir horloge XT (jusqu'à 4 Mhz), ou HS (jusqu'à 20 MHz).

MODE 1 : LP, quartz low power à 32 Khz

MODE 2 : XT, quartz ou résonnateur céramique jusqu'à 4 MHz

MODE 3 : HS, quartz ou résonnateur céramique jusqu'à 20 MHz

MODE 4 : RC, oscillateur externe sur OSCIN, sortie de F/4 sur CLKOUT

MODE 5 : RCIO, idem mode 4, mais avec OSC2/CLKOUT en E/S

MODE 6 : INTIO1, oscillateur interne avec F/4 sur RA6 et E sur RA7

MODE 7 : INTIO2, oscillateur interne ace S sur RA6 et E sur RA7

MODE 8 : ECIO, horloge externe avec sur RA6

Le mode est configurable sur 3 bits du registre CONFIG, de 16 bits, dédiés à la configuration du PIC. L'utilisation de ce registre est transparent à travers les options du compilateur.

Les deux registres qui peuvent éventuellement réellement servir sont le OSCON et le OSCTUNE.

OSCON

	IRCF2	IRCF1	IRCF0	OSTS	IOFS	SCS1	SCS0

IRCF2-0: fréquence de l'oscillateur interne (voir tableau)
OSTS: en lecture seulement
IOFS: en lecture seulement
SCS1-0: oscillateur défini dans CONFIG (00), T1OSC (01), oscillateur interne RC (10)

Bit Value	Fréquence
000	31 KHz
001	125 KHz
010	250 KHz
011	500 KHz
100	1 MHZ
101	2 MHz
110	4 MHz
111	8 MHz

OSCTUNE

		TUN5	TUN4	TUN3	TUN2	TUN1	TUN0

TUN5-0: de 011111 (max) à 000000 (défaut, neutre) puis 111111 à 100000 (min)

Dans le cas de l'utilisation de l'oscillateur interne, OSCTUNE permet d'ajuster précisément la fréquence.

4.2.3- Les Interruptions

A chaque source d'interruption activée, correspond un drapeau, un "flag", permettant de savoir si l'interruption a eu lieu. Dans les registres, les bits terminant par un E, sauf précision contraire, correspondent à une interruption. Les bits terminant par F correspondent aux drapeaux. Chaque drapeau, une fois passé à "1" est à remettre à zéro de manière logicielle.

Ainsi pour savoir si une interruption a eu lieu, il suffit de surveiller le flag de l'interruption concernée.

La mise en marche des interruptions se fait via le bit GIE du registre INTCON.

Voici les registres concernés, avec les explications de chaque bit. Nous ne détaillerons pas les bits flags (finissant par F).

INTCON:

GIE	PEIE	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF

GIE: active ou non les interruptions INTCON du PIC
PEIE: active ou non les interruptions du registre PIE1 du PIC
TMR0IE: interruption du timer 0
INT0IE: interruption externe (GP2)
RBIE: changement d'état du port B (patte 4 à 7)

PIE1:

	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE

ADIE: interruption de fin de conversion des CAN
RCIE: interruption de réception de l'AUSART
TXIE: interruption de transmission de l'AUSART
SSPIE: interruption du port série synchrone (I^2C, SPI)
CCP1IE: interruption du CCP1 (inutilisé en PWM)
TMR2IE: interruption de comparaison timer 2, PR2
TMR1IE: interruption de timer 1

PIR1:

	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF

Le registre PIR1 contient tous les flags du registre PIE1.

PIE2:

OSFIE	CMIE		EEIE				

OSFIE: interruption de problème d'oscillateur
CMIE: interruption de comparateur
EEIE: interruption de fin d'écriture E^2PROM

PIR2:

OSFIF	CMIF		EEIF				

Le registre PIR2 contient tous les flags du registre PIE1.

4.2.4- Les TIMERS

Ils sont au nombre de 3. Un timer permet de compter le temps, ou des impulsions. Chaque timer possède des spécificités précises. La compréhension à 100% des timers est assez difficile. Heureusement, la plupart des compilateurs nous permettent de nous en servir de manière relativement tranparente.

TIMER 0

Ce timer est le timer de base, et se configure via le registre OPTION_REG.

Timer 8 bit, le timer 0 est un compteur pouvant être initialisé à une valeur donnée, via le registre TMR0 (charger la valeur désirée dans ce registre). Lorsque l'interruption du timer 0 est activée, le passage de 0xFF à 0x00 du timer provoque l'activation du flag du timer0.

OPTION_REG

RBPU\	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0

RBPU\: port B pull up activé (1) ou désactivé (0)
INTEDG: interruption sur front montant (1) ou descendant (0) de GP2
T0CS: utilisation de l'horloge interne (0, f/4) ou de l'entrée T0CKI (1)
T0SE: incrémentation sur front montant (0) ou descendant (1) de T0CKI
PSA: prédiviseur pour le watchdog (1) ou pour le timer 0 (0)
PS2-PS0: prédiviseur (voir tableau ci dessous)

Bit Value	Timer 0	Watchdog
000	1:2	1:1
001	1:4	1:2
010	1:8	1:4
011	1:16	1:8
100	1:32	1:16
101	1:64	1:32
110	1:128	1:64
111	1:256	1:128

<u>Remarque:</u> le INTEDG ne sert que si l'interruption sur GP2 est activée.

TIMER 1

Second timer, il est sur 16 bits. Ce registre se configure via T1CON. Tout comme le timer 0, le timer 1 est initialisable en chargeant la valeur désirée dans les 2 registres dédiés: TMR1H pour l'octet supérieur, et TMR1L pour l'octet inférieur. A noter que l'entrée timer 1 (autre que T1CKI), est en fait une porte logique inverseuse.

T1CON

	T1RUN	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC\	TMR1CS	TMR1ON

<u>T1RUN:</u> horloge système dérivée de l'oscillateur du timer 1(1) ou non
<u>T1CKPS1-0:</u> prédiviseur du timer 1 (voir tableau ci-après)
<u>T1OSCEN:</u> oscillateur timer 1 actif (1) ou non
<u>T1SYNC\:</u> actif si TMR1CS=1. Le timer 1 se synchronise sur l'entrée horloge externe (0) ou non (1)
<u>TMR1CS:</u> fonctionnement sur horloge interne (0,f/4), ou T1CKI (1)
<u>TMR1ON:</u> active (1) ou non le timer 1

Bit Value	Timer 1
00	1:1
01	1:2
10	1:4
11	1:8

TIMER 2

Troisième timer du PIC, et second timer 8 bits, le timer 2 est notamment utilisé pour générer la PWM. Il possède également, en plus d'un prediviseur, un postdiviseur. Il est, lui, initialisable via le registre TMR2. La période du timer 2 peut être configurée via PR2 (1 octet).

T2CON

	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0

<u>TOUTPS3-0:</u> postdiviseur (binaire)
<u>TMR2ON:</u> active (1) ou non le timer 2
<u>T2CKPS1-0:</u> prédiviseur

Bit Value	Timer 2
00	1:1
01	1:4
10	1:16
11	1:16

Bit Value	postdiviseur
0000	1:1
0001	1:2
...	...
1111	1:16

4.2.5- L'USART, I²C, SPI

Sous ces noms se cachent en réalité 2 grands principes: la communication série asynchrone (USART/AUSART), et série synchrone (SSP pour l'I²C et le SPI). Comme pour les autres fonctions, elles sont transparentes ou émulées avec le compilateur sélectionné dans ce livre. De fait, nous ne détaillerons pas les registres utilisés, mais nommerons tout de même ces derniers.

USART/AUSART:

l'AUSART, sur les PICs est généralement utilisé pour les communications série asynchrone, c'est-à-dire que les communications circulent sans horloge de synchronisation, un peu en "électrons libres".

Les registres utiles sont TXSTA, TXREG, RCSTA, RCREG et SPBRG (pour définir le débit) (configurer Rx et TX en entrée).

SSP:

Dans ce cas, 4 pattes peuvent être mises à contribution: SDO (data out), SDI (data in), SCK (horloge), SS\ (sélection périphérique).

Communication synchrone signifie que les données transites selon une cadence définie par une horloge commune. Les données entrantes et sortantes peuvent circuler alternativement sur le même fil (I²C), ou sur des fils dédiés (SPI).

Les registres utiles sont le SSPSTAT, SSPCON, SSPADD, SSPBUF, et bien sûr, indirectement, le TRISB, et éventuellement les registres d'interruptions.

4.2.6- La tension de référence

Le 16F88 possède une tension de référence interne, utilisant un réseau de résistances. Cette tension est configurable via VRCON. Toutefois cette source est dédiée au comparateur (d'où le C de CVREF).

VRCON

CVREN	CVROE	CVRR		CVR3	CVR2	CVR1	CVR0

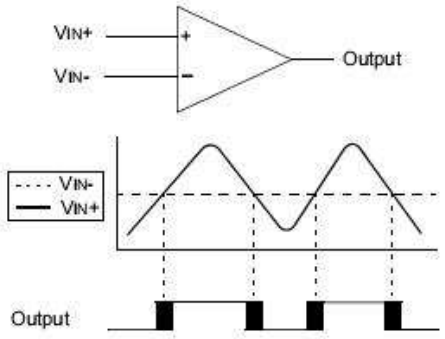
CVREN: active (1) ou non la référence de tension
CVROE: la tension de référence est connecté sur RA2 (1) ou non
CVRR: sélection de la formule de calcul de tension
CVR3-0: si CVRR=1, alors CVREF=(CVR3-0/24)*VDD, si CVRR=0, alors CVREF=(VR3-0/32)*VDD+VDD/4

Attention toutefois, car la tension maximale que peut atteindre cette référence de tension est d'environ 3,6V.

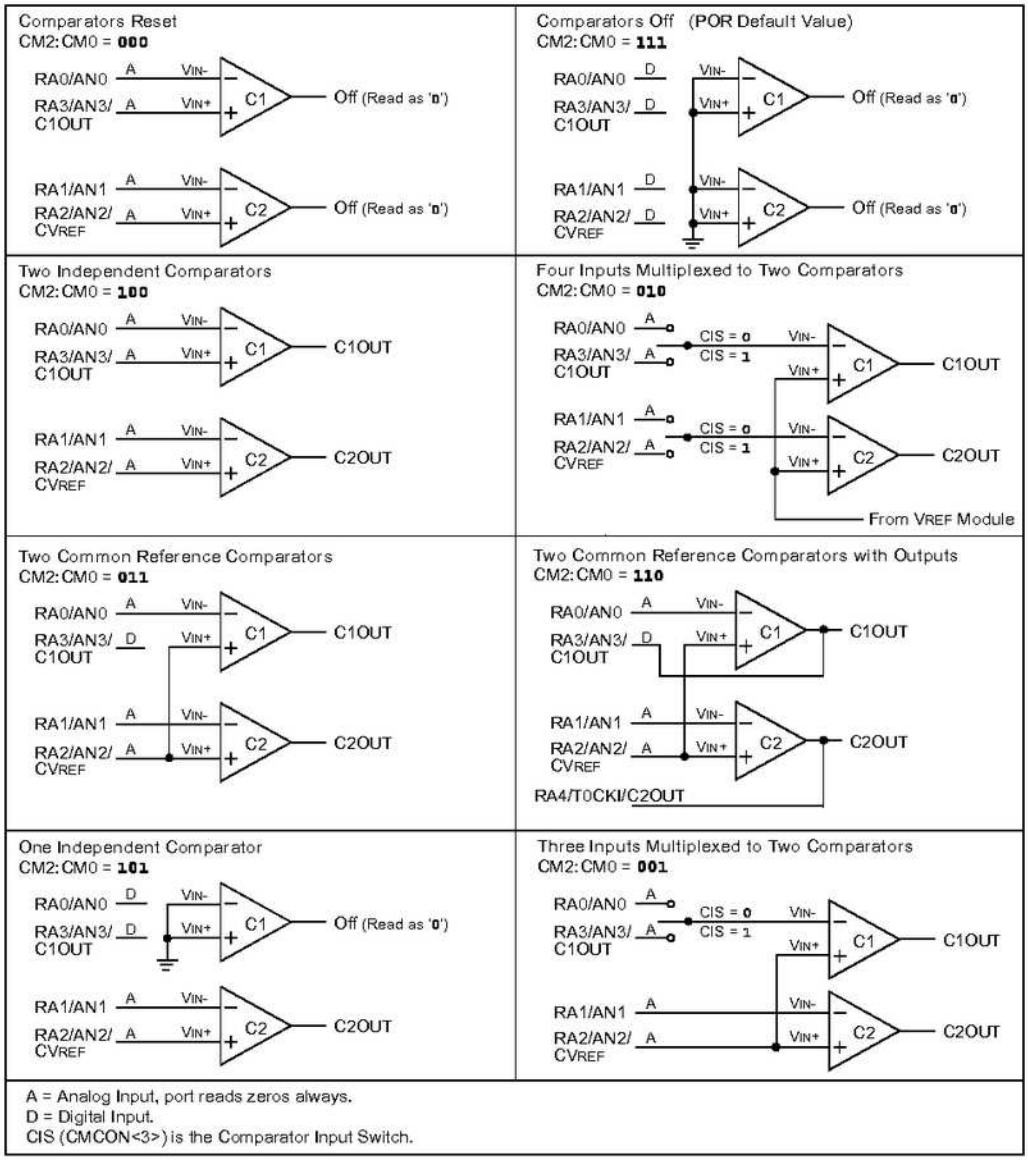
4.2.7- Les comparateurs

La présence de comparateurs analogiques est fort pratique. En effet, cette fonctionnalité permet de réaliser la comparaison et de connaître au niveau logiciel & hardware le résultat de façon simple.

Le principe d'un comparateur, fort simple, est le suivant:



Si V+>V- alors la sortie =1, 0 sinon. Ce comparateur permet de comparer diverses choses en fonction du mode de configuration.



Afin de configurer le comparateur dans le mode désiré, il faut configurer le registre CMCON. Le CVREF est une référence de tension.

CMCON

C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0

C2OUT: en lecture seulement, permet de connaître logiciellement l'état de la comparaison (sortie du comparateur 2)
C1OUT: en lecture seulement, permet de connaître logiciellement l'état de la comparaison (sortie du comparateur 1)
C2INV-C1INV: inverse l'état de la sortie (1) ou non
CIS: sélection d'entrées dans certains modes, voir schémas précédents
CM2-0: configuration du comparateur, voir schémas précédents

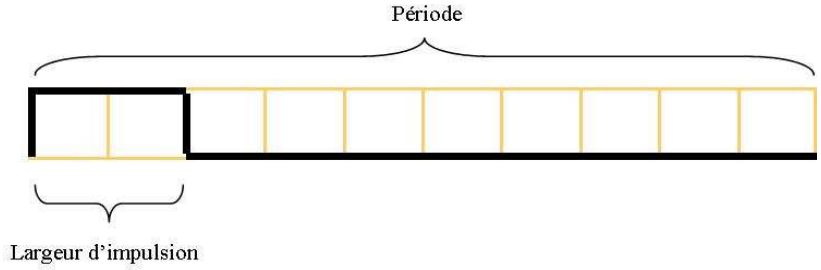
4.2.8- La PWM

La PWM fait partie d'un bloc interne du PIC: le CCP; ou Capture/Compare/PWM. Les deux premiers modes assez flous et d'une approche assez difficiles ne seront pas vus ici, car très peu utilisés. Nous nous concentrerons sur la PWM, plus répandue.

La PWM pour Pulse Width Modulation, ou en français MLI pour Modulation en Largeur d'Impulsion, permet de gérer l'énergie transmise à l'extérieur. En effet, si un signal continu correspond à 100% d'énergie, un signal carré dont la durée d'état haut égale celle d'état bas correspond à 50 % d'énergie. Le pourcentage d'énergie transmis se calcule en faisant le rapport de la durée d'état haut sur la durée de la période.

La PWM peut permettre de gérer la luminosité d'une lampe, le pilotage d'un servomoteur de modélisme, contrôler la vitesse d'un moteur continu, ...

Voici une période PWM:



Deux formules permettent de calculer la durée de la période, et la largeur de l'impulsion.

PERIODE=(PR2+1)*4*Tosc*(prédiviseur timer 2)
LARGEUR=(10 bits)*Tosc*(prédiviseur timer 2)

Par Tosc, nous comprenons la période de l'oscillateur servant d'horloge au PIC. Les 10 bits correspondent à l'octet de CCPR1L et des 2 bits de DC1Bx. Cependant cette résolution de 10 bits est la valeur maximale. En effet, selon la fréquence désirée, cette résolution peut changer. Pour connaître la résolution maximale, on utilise la formule suivante:

RESOLUTION=(log(Fosc/Fpwm))/log(2) bits.

Par exemple, avec le prédiviseur timer 2 à 16, et un PR2 à 255 (0XFF), nous avons une résolution maximale de 10 bits, et une fréquence de PWM de 1,22 Khz.

Autre exemple, avec un prédiviseur à 1, et un PR2 à 31, la résolution maximale est alors de 7 bits, et une fréquence de PWM de 156,3 KHz.

Ici, les registres utiles sont le CCP1CON, PR2, CCPR1L. De plus, comme précisé précédemment, c'est le timer 2 qui est utilisé ici. De fait, il faudra également utiliser T2CON, afin d'activer le timer 2. Cependant, la plupart des compilateurs rendent toutes ces actions transparentes.

CCP1CON

		CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0

CCP1X-Y: contient les 2 bits de poids faibles de la période (PWM 10 bits)
CCP1M3-0: Pour la PWM, égal à 11xx

Les 8 autres bits doivent être chargés dans le registre CCPR1L.

4.2.9- Les CAN

Le PIC dispose de 7 CAN, répartis sur ses entrées.

3 registres permettent la configuration de ces convertisseurs, et 2 autres regsitres permettent de récupérer les résultats (sur 10 bits).

ADCON0

ADCS1	ADCS0	CHS2	CHS1	CHS0	GO-DONE\		ADON

ADS1-0: selon l'état d'ADCS2
CHS2-0: sélection du CAN (000 pour le 0 à 110 pour le 6)
GO-DONE\: à 1, la conversion est en cours, à 0, elle est finie
ADON: activation (1) ou non des CAN

Bit Value	ADCS2=0	ADCS2=1
00	Fosc/2	Fosc/4
01	Fosc/8	Fosc/16
10	Fosc/32	Fosc/64
11		

ADCON1

ADFM	ADCS2	VCFG1	VCFG0				

ADFM: aligne les résultats à gauche (0) ou à droite
ADCS2: horloge de conversion divisée par 2 (1) ou non
VCFG1-0: défini les références de tension des convertisseurs

Bit Value	Vref+	Vref-
00	AVdd	AVss
01	AVdd	Vref-
10	Vref+	AVss
11	Vref+	Vref-

Pour utiliser les Vrefs, configurer les pattes adéquates en entrées.

ANSEL

	ADCS2	ADCS1	ADCS0	ANS3	ANS2	ANS1	ANS0

ADCS2-0: détermine l'échantillonnage des CAN (voir tableau ci-après)
ANS3-0: pour chaque CAN, un 1 configure l'entrée en CAN, un 0 en E/S numérique

Bit Value	Fréquence
000	f/2
001	f/8
010	f/32
x11	500 KHz max
100	f/4
101	f/16
110	f/64

<u>Remarque:</u> sauf exception particulière, vous pouvez laisser l'échantillonnage à sa valeur par défaut.

2 autres registres permettent de récupérer la valeur de la conversion: ADRESH et ADRESL. Pour des raisons de commodités de traitement de la conversion, 2 cas sont possibles selon la valeur de ADFM (ADRESH en haut)

ADFM=1

						ADR9	ADR8
ADR7	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	ADR0

ADFM=0

ADR9	ADR8	ADR7	ADR6	ADR5	ADR4	ADR3	ADR2
ADR1	ADR0						

CHAPITRE 5 : LE PIC 18F2550

5.1 Caractéristiques du PIC 18F2550

5.2 Fonctionnement du PIC 18F2550

5.2.1 Les entrées/sorties

5.2.2 Les différents modes d'horloge

5.2.3 Les interruptions

5.2.4 Les Timers

5.2.5 L'EUSART, I²C, SPI, USB

5.2.6 La tension de référence

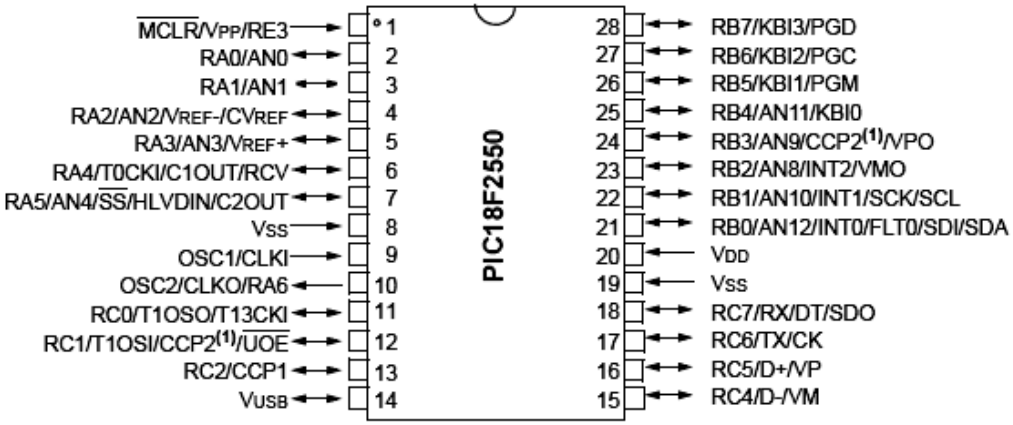
5.2.7 Les comparateurs

5.2.8 La PWM

5.2.9 Les CAN

5- LE PIC 18F2550

5.1- Caractéristiques du PIC 18F2550



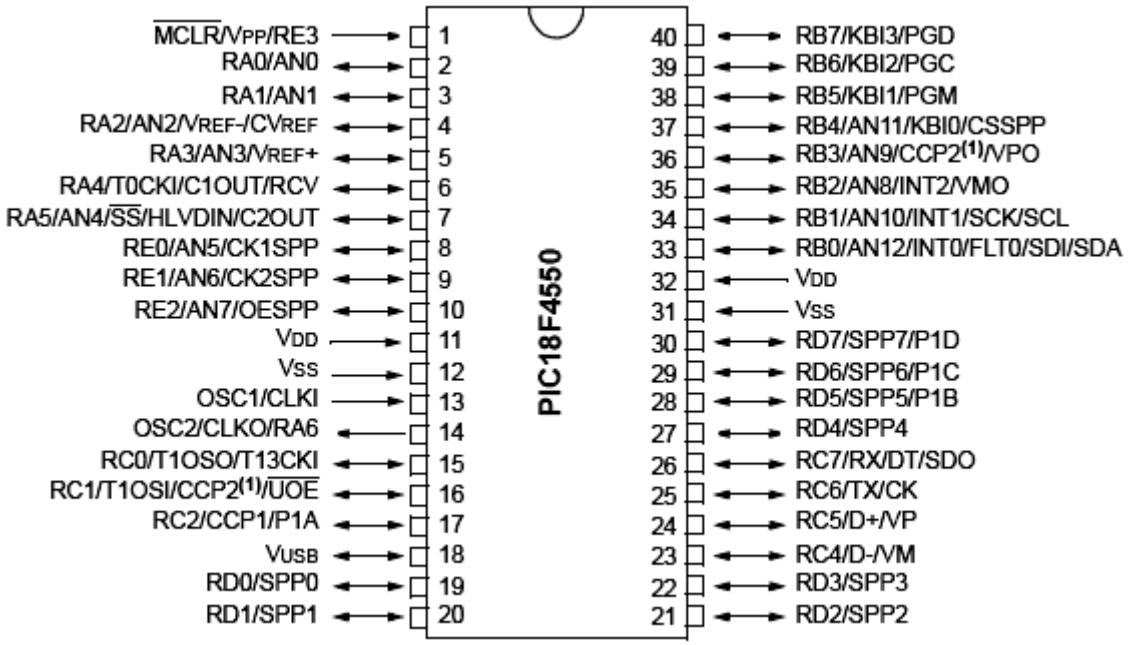
Le 18F2550 est un microcontrôleur en boîtier DIL 28 pattes possédant jusqu'à 24 E/S sur 3 ports (A, B, C, et éventuellement RE3).

Alimentable de 2 à 6 V continu, il est compatible avec le mode de programmation ICSP (In Chip Serial Programming), et dispose d'un oscillateur interne pouvant monter jusqu'à 8 Mhz. Sur oscillateur externe, le 18F2550 peut monter jusqu'à 48 MHz.

Au niveau mémoire, ce PIC dispose de 16 KO de mémoire flash pour le firmware, de 2 KO de RAM et de 256 octets d'E²PROM.

Côté fonctionnalité, outre les E/S numériques classiques, ce PIC dispose de 10 CAN, de 2 comparateurs, de 2 PWM 10 bits, d'1 module de communication série synchrone et asynchrone, d'un module de communication USB, de 4 timers, ...

Il faut savoir qu'il existe le 18F4550. Ce PIC, pouvant être considéré comme une extension du 18F2550, est en boitîer DIL 40 pattes, et possède jusqu'à 35 E/S. Possédant également un module de communication USB, leur principale différence, outre le nombre de pattes, et donc d'E/S se situe au niveau des CAN. En effet, le 18F4550 en possède 13, soit 3 de plus que le 18F2550.


PIC18F4550
MCLR/VPP/RE3 1
RA0/AN0 2
RA1/AN1 3
RA2/AN2/VREF-/CVREF 4
RA3/AN3/VREF+ 5
RA4/T0CKI/C1OUT/RCV 6
RA5/AN4/SS/HLVDIN/C2OUT 7
RE0/AN5/CK1SPP 8
RE1/AN6/CK2SPP 9
RE2/AN7/OESPP 10
VDD 11
VSS 12
OSC1/CLKI 13
OSC2/CLKO/RA6 14
RC0/T1OSO/T13CKI 15
RC1/T1OSI/CCP2(1)/UOE 16
RC2/CCP1/P1A 17
VUSB 18
RD0/SPP0 19
RD1/SPP1 20
40 RB7/KBI3/PGD
39 RB6/KBI2/PGC
38 RB5/KBI1/PGM
37 RB4/AN11/KBI0/CSSPP
36 RB3/AN9/CCP2(1)/VPO
35 RB2/AN8/INT2/VMO
34 RB1/AN10/INT1/SCK/SCL
33 RB0/AN12/INT0/FLT0/SDI/SDA
32 VDD
31 VSS
30 RD7/SPP7/P1D
29 RD6/SPP6/P1C
28 RD5/SPP5/P1B
27 RD4/SPP4
26 RC7/RX/DT/SDO
25 RC6/TX/CK
24 RC5/D+/VP
23 RC4/D-/VM
22 RD3/SPP3
21 RD2/SPP2

5.2- Fonctionnement du PIC 18F2550

5.2.1- Les entrées/sorties

Dans cette partie, nous allons voir les différentes possibilités de chaque patte, en les désignant par leur numéro. A noter, qu'une seule fonction est disponible à la fois, par patte.

PATTE 1: E numérique RE3, reset

PATTE 2: E/S numérique RA0, CAN0

PATTE 3: E/S numérique RA1, CAN1

PATTE 4: E/S numérique RA2, CAN2, tension de référence du comparateur, référence basse CAN

PATTE 5: E/S numérique RA3, CAN3, référence haute CAN

PATTE 6: E/S numérique RA4, entrée comptage timer 0, sortie comparateur 1, E module USB externe

PATTE 7: E/S numérique RA5, CAN4, sélection de périphérique synchrone, sortie de comparateur 2

PATTE 8: Masse

PATTE 9: Entrée oscillateur, entrée horloge

PATTE 10: Entrée oscillateur, sortie horloge, S numérique RA6

PATTE 11: E/S numérique RC0, sortie horloge timer 1

PATTE 12: E/S numérique RC1, entrée horloge timer 1, sortie PWM 2, S module USB externe

PATTE 13: E/S numérique RC2, sortie PWM 1

PATTE 14: Référence de tension USB 3,3V (condensateur de 470 nF)

PATTE 15: E/S numérique RC4, E/S – USB, E module USB externe VM

PATTE 16: E/S numérique RC5, E/S + USB, E module USB externe VP

PATTE 17: E/S numérique RC6, S RS232, S horloge synchrone

PATTE 18: E/S numérique RC7, E RS232, E donnée synchrone, sortie donnée SPI

PATTE 19: Masse

PATTE 20: Alimentation positive +5V

PATTE 21: E/S numérique RB0, CAN12, interruption externe 0, entrée d'erreur PWM, entrée SPI, E/S I²C

PATTE 22: E/S numérique RB1, CAN10, interruption externe 1, horloge SPI, horloge I²C

PATTE 23: E/S numérique RB2, CAN8, interruption externe 2, S module USB externe VM

PATTE 24: E/S numérique RB3, CAN9, S PWM 2, S module USB externe VP

PATTE 25: E/S numérique RB4, CAN11, interruption de changement 0

PATTE 26: E/S numérique RB5, interruption de changement 1

PATTE 27: E/S numérique RB6, interruption de changement 2

PATTE 28: E/S numérique RB7, interruption de changement 3

Les E/S se configurent via TRISA, TRISB, TRISC, TRISE.

5.2.2- Les modes d'horloge

Dans cette partie, nous allons voir les différents modes d'horloge. Dans le cas d'un quartz (mode1), vous pourrez choisir horloge XT (jusqu'à 4 Mhz), ou HS (jusqu'à 20 MHz).

MODE 1 : XT, quartz ou résonnateur céramique jusqu'à 4 MHz

MODE 2 : XTPLL, quartz avec PLL

MODE 3 : HS, quartz ou résonnateur céramique jusqu'à 20 MHz

MODE 4 : HSPLL, quartz avec PLL

MODE 5 : EC, horloge externe avec sortie F/4 sur RA6

MODE 6 : ECIO, horloge externe avec E/S sur RA6

MODE 7 : ECPLL, idem mode 5 mais avec PLL

MODE 8 : ECPIO, idem mode 6, mais avec PLL

MODE 9 : INTHS, oscillateur interne pour le PIC, quartz HS pour l'USB

MODE 10 : INTXT, oscillateur interne pour le PIC, quartz XT pour l'USB

MODE 11 : INTIO, oscillateur interne pour le PIC, EC pour l'USB, E/S sur RA6

MODE 12 : INTCKO, oscillateur interne pour le PIC, EC pour l'USB, F/4 sur RA6

Les deux registres qui peuvent éventuellement réellement servir sont le OSCON et le OSCTUNE.

OSCCON

IDLEN	IRCF2	IRCF1	IRCF0	OSTS	IOFS	SCS1	SCS0

IDLEN: toujours à 0 (pour le sommeil)
IRCF2-0: fréquence de l'oscillateur interne (voir tableau)
OSTS: en lecture seulement
IOFS: en lecture seulement
SCS1-0: oscillateur définie dans CONFIG (00), T1OSC (01), oscillateur interne RC (10)

Bit Value	Fréquence
000	31 KHz
001	125 KHz
010	250 KHz
011	500 KHz
100	1 MHZ
101	2 MHz
110	4 MHz
111	8 MHz

OSCTUNE

INTSRC			TUN4	TUN3	TUN2	TUN1	TUN0

INTSRC: source de l'oscillateur interne, toujours à 1
TUN5-0: de 011111 (max) à 000000 (défaut, neutre) puis 111111 à 100000 (min)

Dans le cas de l'utilisation de l'oscillateur interne, OSCTUNE permet d'ajuster précisément la fréquence.

5.2.3- Les Interruptions

A chaque source d'interruption activée, correspond un drapeau, un "flag", permettant de savoir si l'interruption a eu lieu. Dans les registres, les bits terminant par un E, sauf précision contraire, correspondent à une interruption. Les bits terminant par F correspondent aux drapeaux. Chaque drapeau, une fois passé à "1" est à remettre à zéro de manière logicielle.

Ainsi pour savoir si une interruption a eu lieu, il suffit de surveiller le flag de l'interruption concernée.

Voici les registres concernés, avec les explications de chaque bit. Il s'agit de INTCON, INTCON2, INTCON3, PIR1, PIR2, PIE1, PIE2, IPR1, IPR2. Nous ne détaillerons pas les bit flags (finissant par F). Dans RCON, le seul bit utile est RCON.7. Nous considèrerons ici ce bit toujours à 1. Autre chose à savoir: l'USB possède ses propres registres d'interruptions (non détaillés ici). Il s'agit de UIR, UIE, UEIR, UEIE.

INTCON

GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF

GIE/GIEH: active (1) ou non les interruptions de haut niveau
PEIE/GIEL: active (1) ou non les interruptions de bas niveau
TMR0IE: active (1) ou non l'interruption du timer 0
INT0IE: active (1) ou non l'interruption externe 0
RBIE: active (1) ou non l'interruption de changement d'état du port B
TMR0IF, INT0IF, RBIF: flags

INTCON2

RBPU\	INTEDG0	INTEDG1	INTEDG2		TMR0IP		RBIP

RBPU\: active (1) ou non tous les pull-ups du port B
INTEDG0: interruption externe 0 sur front montant (1) ou descendant
INTEDG1: interruption externe 1 sur front montant (1) ou descendant
INTEDG2: interruption externe 2 sur front montant (1) ou descendant
TMR0IP: priorité haute (1) ou basse de l'interruption du timer 0
RBIP: priorité haute (1) ou basse de l'interruption du port B

INTCON3

INT2IP	INT1IP		INT2IE	INT1IE		INT2IF	INT1IF

INT2IP: priorité haute (1) ou basse de l'interruption externe 2
INT1IP: priorité haute (1) ou basse de l'interruption externe 1
INT2IE: active (1) ou non l'interruption externe 2
INT1IE: active (1) ou non l'interruption externe 1
INT2IF, INT1IF: flags

PIE1

SPPIE	ADIE	RCIE	TXIE	SPPIE	CCP1IE	TMR2IE	TMR1IE

SPPIE: active (1) ou non l'interruption du module SPP (lecture/écriture)
ADIE: active (1) ou non l'interruption des CAN
RCIE: active (1) ou non l'interruption de réception de l'EUSART
TXIE: active (1) ou non l'interruption d'émission de l'EUSART
SSPIE: active (1) ou non l'interruption du module MSSP
CCP1IE: active (1) ou non l'interruption de la PWM1
TMR2IE: active (1) ou non l'interruption de la comparaison timer 2 - PR2
TMR1IE: active (1) ou non l'interruption du timer 1

PIR1

SPPIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF

Le registre PIR1 contient tous les flags du registre PIE1.

PIE2

OSCFIE	CMIE	USBIE	EEIE	BCLIE	HLVDIE	TMR3IE	CCP2IE

OSFIE: active (1) ou non l'interruption d'erreur d'oscillateur
CMIE: active (1) ou non l'interruption des comparateurs
USBIE: active (1) ou non l'interruption de l'USB
EEIE: active (1) ou non l'interruption de lecture/écriture E²PROM
BCLIE: à ne pas prendre en compte
HLVDIE: à ne pas prendre en compte
TMR3IE: active (1) ou non l'interruption du timer 3
CCP2IE: active (1) ou non l'interruption de la PWM 2

PIR2

OSCFIF	CMIF	USBIF	EEIF	BCLIF	HLVDIF	TMR3IF	CCP2IF

Le registre PIR2 contient les flags du registre PIE2.

IPIR1

SPPIP	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP

SPPIP: priorité haute (1) ou basse de l'interruption de SPP
ADIP: priorité haute (1) ou basse de l'interruption des CAN
RCIP: priorité haute (1) ou basse de l'interruption de réception EUSART
TXIP: priorité haute (1) ou basse de l'interruption d'émission EUSART
SSPIP: priorité haute (1) ou basse de l'interruption du MSSP
CCP1IP: priorité haute (1) ou basse de l'interruption de la PWM 1
TMR2IP: priorité haute (1) ou basse de l'interruption du timer 2
TMR1IP: priorité haute (1) ou basse de l'interruption du timer 1

IPIR2

OSCFIP	CMIP	USBIP	EEIP	BCLIP	HLVDIP	TMR3IP	CCP2IP

OSCFIP: priorité haute (1) ou basse de l'interruption de l'oscillateur
CMIP: priorité haute (1) ou basse de l'interruption comparateurs
USBIP: priorité haute (1) ou basse de l'interruption USB
EEIP: priorité haute (1) ou basse de l'interruption E²PROM

BCLIP: à ne pas prendre en compte
HLVDIP: à ne pas prendre en compte
TMR3IP: priorité haute (1) ou basse de l'interruption du timer 3
CCP2IP: priorité haute (1) ou basse de l'interruption de la PWM 2

5.2.4- Les TIMERS

Ils sont au nombre de 3. Un timer permet de compter le temps, ou des impulsions. Chaque timer possède des spécificités précises. La compréhension à 100% des timers est assez difficile. Heureusement, la plupart des compilateurs nous permettent de nous en servir de manière relativement tranparente.

TIMER 0

Ce timer est le timer de base, et se configure via le registre OPTION_REG.

Timer 8 ou 16 bits, le timer 0 est un compteur pouvant être initialisé à une valeur donnée, via le registre TMR0 (8 bits) ou TMR0L & TMR0H (charger la valeur désirée dans ce(s) registre(s)). Lorsque l'interruption du timer 0 est activée, le passage de 0xFF à 0x00 du timer provoque l'activation du flag du timer0.

T0CON

TMR0ON	T08BIT	T0CS	T0SE	PSA	T0PS2	T0PS1	T0PS0

TMR0ON: active (1) ou non le timer 0
T08BIT: configure le timer 0 en 8 bits (1) ou 16 bits
T0CS: utilisation de l'horloge interne (0, F/4) ou de l'entrée T0CKI (1)
T0SE: incrémentation sur front montant (0) ou descendant (1) de T0CKI
PSA: prédiviseur pour le watchdog (1) ou pour le timer 0 (0)
T0PS2-0: prédiviseur (voir tableau ci dessous)

Bit Value	Timer 0
000	1:2
001	1:4
010	1:8
011	1:16
100	1:32
101	1:64
110	1:128
111	1:256

TIMER 1

Second timer, il est sur 16 bits. Ce registre se configure via T1CON. Tout comme le timer 0, le timer 1 est initialisable en chargeant la valeur désirée dans les 2 registres dédiés: TMR1H pour l'octet supérieur, et TMR1L pour l'octet inférieur. A noter que l'entrée timer 1 (autre que T1CKI), est en fait une porte logique inverseuse.

T1CON

RD16	T1RUN	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC\	TMR1CS	TMR1ON

RD16: active la lecture/écriture du timer 1 en 16 bits ou 2*8bits
T1RUN: horloge système dérivée de l'oscillateur du timer 1(1) ou non
T1CKPS1-0: prédiviseur du timer 1 (voir tableau ci-après)
T1OSCEN: oscillateur timer 1 actif (1) ou non
T1SYNC\: actif si TMR1CS=1. Le timer 1 se synchronise sur l'entrée horloge externe (0) ou non (1)
TMR1CS: fonctionnement sur horloge interne (0,F/4), ou T1CKI (1)
TMR1ON: active (1) ou non le timer 1

Bit Value	Timer 1
00	1:1
01	1:2
10	1:4
11	1:8

TIMER 2

Troisième timer du PIC, 8 bits, il possède, en plus d'un prédiviseur, un postdiviseur. Il est, lui, initialisable via le registre TMR2. La période du timer 2 peut être configurée via PR2 (1 octet).

T2CON

	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2OUTPS0	TMR2ON	T2CKPS1	T2CKPS0

T2OUTPS3-0: postdiviseur (binaire)
TMR2ON: active (1) ou non le timer 2
T2CKPS1-0: prédiviseur

Bit Value	Timer 2
00	1:1
01	1:4
10	1:16
11	1:16

Bit Value	postdiviseur
0000	1:1
0001	1:2
...	...
1111	1:16

TIMER 3

Quatrième timer du PIC, 16 bits, le timer 3 est notamment utilisé pour générer la PWM. Il est initialisable via le registre TMR3H et TMR3L.

T3CON

RD16	T3CCP2	T3CKPS1	T3CKPS0	T3CCP1	T3SYNC\	TMR3CS	TMR3ON

RD16: active la lecture/écriture du timer 3 en 16 bits ou 2*8bits
T3CCP2-1: sélectionne l'horloge pour les PWM
T3CKPS1-0: prédiviseur (binaire)
T3SYNC\: si TMR3CS=1, synchronise (0) ou non le timer avec l'horloge externe
TMR3CS: sélectionne l'horloge interne (0, F/4), ou l'entrée T13CKI (1)
TMR3ON: active (1) ou non le timer 3

Bit Value	prédiviseur
00	1:1
01	1:2
10	1:4
111	1:8

Bit Value	PWM
00	Timer1
01	Timer3-PWM2 TImer1-PWM11
1x	Timer3

5.2.5- L'EUSART, I²C, SPI, USB

Sous ces noms se cachent en réalité 2 grands principes: la communication série asynchrone (USART/EUSART), série synchrone (MSSP pour l'I²C et le SPI), et l'USB. Comme pour les autres fonctions, elles sont transparentes ou émulées avec le compilateur sélectionné dans ce livre. De fait, nous ne détaillerons pas les registres utilisés, mais nommerons tout de même ces derniers.

USART/EUSART:

l'EUSART, sur les PICs est généralement utilisé pour les communications série asynchrone, c'est-à-dire que les communications circulent sans horloge de synchronisation, un peu en "électrons libres".

Les registres utiles sont TXSTA, TXREG, RCSTA, RCREG, BAUDCON, SPBRGH et SPBRG (pour définir le débit).

MSSP:

Dans ce cas, 4 pattes peuvent être mises à contribution: SDO (data out), SDI (data in), SCK (horloge), SS\ (sélection périphérique).

Communication synchrone signifie que les données transites selon une cadence définie par une horloge commune. Les données entrantes et sortantes peuvent circuler alternativement sur le même fil (I^2C), ou sur des fils dédiés (SPI).

Les registres utiles sont le SSPSTAT, SSPCON1, SSPBUF, et bien sur, indirectement, le TRISB, et éventuellement les registres d'interruptions.

USB:

A priori compliqué d'accès, impression justifié, l'utilisation de l'USB via le compilateur BASIC devient relativement facile. En effet, une fois de plus, tout est totalement transparent. De fait, ne vous souciez pas de savoir comment le port USB fonctionne précisément, mais plutot comment utiliser l'USB avec le compilateur.

Les registres permettant d'initialiser l'USB sont UCON, UCFG, USTAT, UADDR, UEPn, UFRMH ET UFMRL.

A noter cependant, que l'USB possède ses propres interruptions avec des registres dédiés.

5.2.6- La tension de référence

La tension de référence est configurable via CVRCON. Toutefois cette source est dédiée au comparateur (d'où le C de CVREF).

CVRCON

CVREN	CVROE	CVRR	CVRSS	CVR3	CVR2	CVR1	CVR0

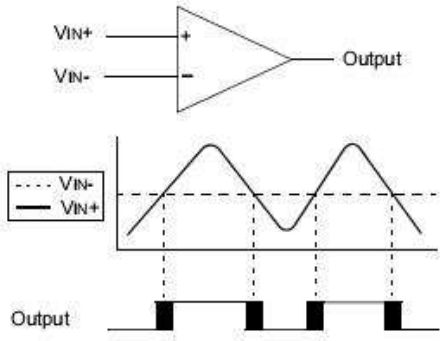
CVREN: active (1) ou non la référence de tension
CVROE: la tension de référence est connectée sur RA2 (1) ou non
CVRR: sélection de la formule de calcul de tension
CVRSS: sélectionne la tension de référence: VDD-VSS (0) ou bien (Vref+) - (Vref-)
CVR3-0: si CVRR=1, alors CVREF=(CVR3-0/24)*VDD, si CVRR=0, alors CVREF=(VR3-0/32)*VDD+VDD/4

Attention toutefois, car la tension maximale que peut atteindre cette référence de tension est d'environ 3,6V.

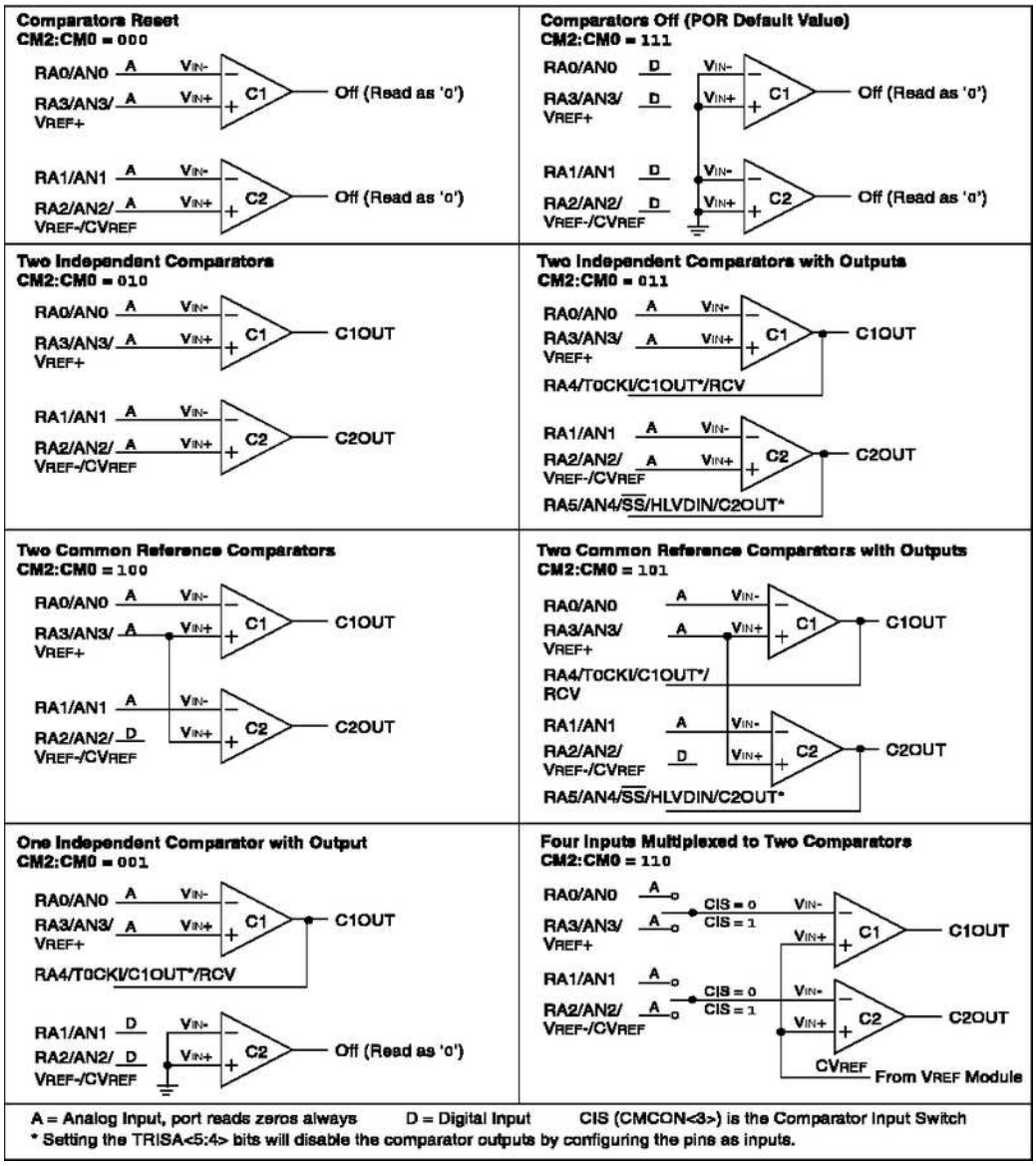
5.2.7- Les comparateurs

La présence de comparateurs analogiques est fort pratique. En effet, cette fonctionnalité permet de réaliser la comparaison et de connaître au niveau logiciel & hardware le résultat de façon simple.

Le principe d'un comparateur, fort simple, est le suivant:



Si V+>V- alors la sortie =1, 0 sinon. Ce comparateur permet de comparer diverses choses en fonction du mode de configuration.



Afin de configurer le comparateur dans le mode désiré, il faut configurer le registre CMCON. Le CVREF est une référence de tension.

CMCON

C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0

C2OUT: en lecture seulement, permet de connaître logiciellement l'état de la comparaison (sortie du comparateur 2)
C1OUT: en lecture seulement, permet de connaître logiciellement l'état de la comparaison (sortie du comparateur 1)
C2INV-C1INV: inverse l'état de la sortie (1) ou non
CIS: sélection d'entrée dans certains modes, voir schémas précédents
CM2-0: configuration du comparateur, voir schémas précédents

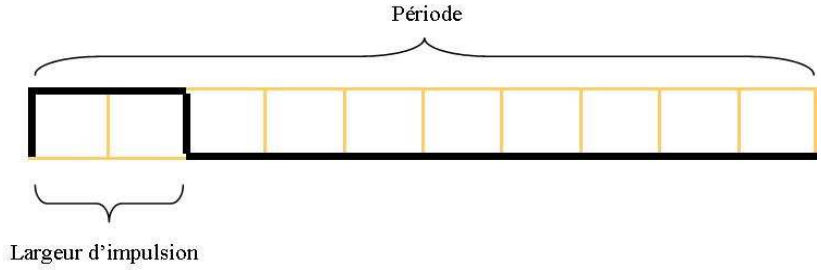
5.2.8- La PWM

La PWM fait partie d'un bloc interne du PIC: le CCP; ou Capture/Compare/PWM. Les deux premiers modes assez flous et d'une approche assez difficile ne seront pas vus ici, car très peu utilisés. Nous nous concentrerons sur la PWM, plus répandue.

La PWM pour Pulse Width Modulation, ou en français MLI pour Modulation en Largeur d'Impulsion, permet de gérer l'énergie transmise à l'extérieur. En effet, si un signal continu correspond à 100% d'énergie, un signal carré dont la durée d'état haut égale celle d'état bas correspond à 50 % d'énergie. Le pourcentage d'énergie transmis se calcule en faisant le rapport de la durée d'état haut sur la durée de la période.

La PWM peut permettre de gérer la luminosité d'une lampe, le pilotage d'un servomoteur de modélisme, contrôler la vitesse d'un moteur continu, ...

Voici une période PWM:



Deux formules permettent de calculer la durée de la période, et la largeur de l'impulsion.

PERIODE=(PR2+1)*4*Tosc*(prédiviseur timer 2)
LARGEUR=(10 bits)*Tosc*(prédiviseur timer 2)

Par Tosc, nous comprenons la période de l'oscillateur servant d'horloge au PIC. Les 10 bits correspondent à l'octet de CCPR1L et des 2 bits de DC1Bx. Cependant cette résolution de 10 bits est la valeur maximale. En effet, selon la fréquence désirée, cette résolution peut changer. Pour connaître la résolution maximale, on utilise la formule suivante:

RESOLUTION=(log(Fosc/Fpwm))/log(2) bits.

Par exemple, avec le prédiviseur timer 2 à 16, et un PR2 à 255 (0XFF), nous avons une résolution maximale de 10 bits, et une fréquence de PWM de 1,22 Khz.

Autre exemple, avec un prédiviseur à 1, et un PR2 à 31, la résolution maximale est alors de 7 bits, et une fréquence de PWM de 156,3 KHz.

Ici, les registres utiles sont le CCP1CON, CCP2CON, PR2, CCPR1L, CCPR2L. De plus, comme précisé précédemment, c'est le timer 2 qui est utilisé ici. De fait, il faudra également utiliser T2CON, afin d'activer le timer 2. Cependant, la plupart des compilateurs rendent toutes ces actions transparentes.

CCP1CON

		DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0

DC1B1-0: contient les 2 bits de poids faibles de la période (PWM 10 bits)
CCP1M3-0: Pour la PWM, égal à 11xx

Les 8 autres bits doivent être chargés dans le registre CCPR1L.

CCP2CON

		DC2B1	DC2B0	CCP2M3	CCP2M2	CCP2M1	CCP2M0

DC2B1-0: contient les 2 bits de poids faibles de la période (PWM 10 bits)
CCP2M3-0: Pour la PWM, égal à 11xx

Les 8 autres bits doivent être chargés dans le registre CCPR2L.

5.2.9- Les CAN

Le PIC dispose de 10 CAN, répartis sur ses entrées.

3 registres permettent la configuration de ces convertisseurs, et 2 autres regsitres permettent de récupérer les résultats (sur 10 bits).

ADCON0

		CHS3	CHS2	CHS1	CHS0	GO - DONE\	ADON

CHS3-0: sélection du CAN (0000 pour le 0 à 1100 pour le 12, le 5,6,7, ne sont disponibles que sur le 4550)
GO-DONE\: à 1, la conversion est en cours, à 0, elle est finie
ADON: activation (1) ou non des CAN

ADCON1

		VCFG0	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0

VCFG0: définie la source de référence négative: Vref- (1) ou Vss (0)
VCFG0: définie la source de référence positive: Vref1 (1) ou Vdd (0)
PCFG2-0: défini les CAN utilisés, et ceux inutilisés

PCFG3: PCFG0	AN12	AN11	AN10	AN9	AN8	AN7(2)	AN6(2)	AN5(2)	AN4	AN3	AN2	AN1	AN0
0000(1)	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	D	A	A	A	A	A	A	A	A	A	A	A	A
0100	D	D	A	A	A	A	A	A	A	A	A	A	A
0101	D	D	D	A	A	A	A	A	A	A	A	A	A
0110	D	D	D	D	A	A	A	A	A	A	A	A	A
0111(1)	D	D	D	D	D	A	A	A	A	A	A	A	A
1000	D	D	D	D	D	D	A	A	A	A	A	A	A
1001	D	D	D	D	D	D	D	A	A	A	A	A	A
1010	D	D	D	D	D	D	D	D	A	A	A	A	A
1011	D	D	D	D	D	D	D	D	D	A	A	A	A
1100	D	D	D	D	D	D	D	D	D	D	A	A	A
1101	D	D	D	D	D	D	D	D	D	D	D	A	A
1110	D	D	D	D	D	D	D	D	D	D	D	D	A
1111	D	D	D	D	D	D	D	D	D	D	D	D	D

A = Analog input D = Digital I/O

Pour utiliser les Vrefs, configurer les pattes adéquates en entrées.

ADCON2

ADFM		ACQT2	ACQT1	ACQT0	ADCS2	ADCS1	ADCS0

ADFM: justification à gauche (0) ou à droite
ACQT2-0: définie le temps d'acquisition
ADCS2-0: définie l'horloge pour l'échantillonnage

<u>Remarque:</u> sauf exception particulière, veuillez laisser les valeurs d'échantillonnage et d'acquisition à leurs valeurs par défaut.

2 autres registres permettent de récupérer la valeur de la conversion: ADRESH et ADRESL. Pour des raisons de commodités de traitement de la conversion, 2 cas sont possibles selon la valeur de ADFM (ADRESH en haut)

ADFM=1

ADFM=0

CHAPITRE 6 : LES PROTOCOLES DE COMMUNICATION

6.1 Série asynchrone, RS232

6.2 I²C, de Philips

6.2.1 Principe

6.2.2 Les mémoires

6.3 SPI

6.4 USB

6.5 RC5

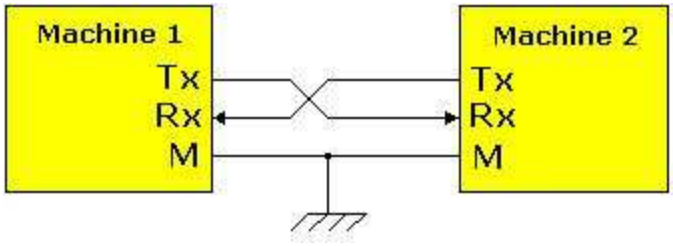
6.6 Les bases du CPL

6- Protocoles de communication

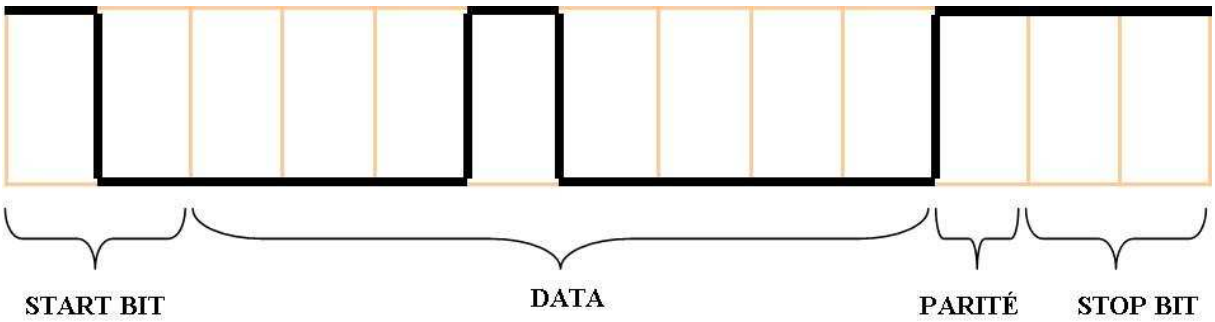
Dans cette partie du livre, nous verrons les bases de divers protocoles de communication, sans toutefois rentrer dans les détails, car nous en parlons à titre d'information. En effet, ces protocoles sont gérés par le PIC ou le compilateur, et leurs utilisations sont de ce fait transparentes pour l'utilisateur.

6.1- Série asynchrone, RS232

Il y a peu de choses à dire sur la communication RS232 tellement le nombre d'information est élevé sur le web. Le principe est simple: une sortie data (TX), une entrée data (RX), et un fil de masse. Les fils sont alors croisés entre les 2 machines communicantes, le TX1 avec le RX2 et le RX1 avec le TX2, le tout ayant un point de référence commun: la masse.



Pour le reste, une trame est toujours réalisée de la même façon: un START BIT, suivi des données (généralement un octet), un éventuel BIT de parité, et un STOP BIT.



La ligne par défaut est au niveau logique "1".

-STOP BIT: est généré en passant de l'état "1" à l'état "0"
-BITS de données: 7 ou 8 bits, généralement 1 octet complet
-BIT de parité: ce bit permet de détecter un problème de transmission
-STOP BIT: constitué de 1 ou 2 bit(s)

Pour communiquer, le RS232 se base sur le tableau ASCII, et utilise généralement des débits normalisés, en bauds (300, 600, 1200, 2400, 9600, 19200 pour les plus usités).

***Remarque:** le bit de parité est à 0 lorsque le nombre de 1 dans data est pair, et à 1 lorsque ce nombre est impair.*

6.2- I²C, de Philips

6.2.1- Principe

Les cases jaunes correspondent aux données du microcontrôleur, et les grises celles de l'E²PROM. Le R/W du mot de contrôle sera placé à 1 pour la lecture et à zéro pour l'écriture. Sont expliquées ici les différentes trames possibles, sans pour autant rentrer dans le détail du protocole. En effet, on considère que le microcontrôleur utilisé possède une interface I²C intégrée, ou bien une émulation I²C comme dans le compilateur retenu.

Ecriture simple

START BIT	MOT CONTROLE	ACK	ADRESSE	ACK	DATA	ACK	STOP BIT

Voici la trame permettant d'écrire un octet à une adresse donnée, les starts bits, stop bit et différents mots respectant les contraintes citées précédemment.

Ecriture multiple

START BIT	MOT CONTROLE	ACK	ADRESSE	ACK	DATA 1 Position n	ACK	DATA 2 Position n+1	...	ACK	STOP BIT

Cette trame est relativement semblable que pour un seul octet, à une exception près, la génération du stop bit n'est faite que quand l'on a fini de transmettre les octets.

ATTENTION : tant que le stop bit n'est pas généré, les données sont mises dans un buffer. Elles ne sont écrites qu'après le stop bit. Pour cette raison, si une coupure d'alimentation subvenait pendant l'écriture, toutes les données de la trame seraient perdues. De plus, la taille du buffer est limitée : ainsi, vous ne pouvez écrire plus de 16 octets à suivre pour un 24xx04 (pas de buffer pour le 00 donc pas valable, 8 octets pour le 01 et le 02, 16 octets pour le 04, 08 et 16, 32 octets pour le 32 et le 64).

Lecture simple d'un octet à l'adresse courante

START BIT	MOT CONTRÔLE lecture	ACK	DATA	NO ACK	STOP BIT

Ce mode permet de ne pas avoir à transmettre une adresse. L'on obtient alors la valeur que l'on vient de rentrer en dernier. A noter en lecture l'apparition du NO ACK avant le stop bit, c'est-à-dire, qu'il n'y a pas d'accusé.

Lecture simple d'un octet à une adresse donnée

START BIT	MOT CONTRÔLE écriture	ACK	ADRESSE	ACK	START BIT	MOT CONTRÔLE lecture	ACK	DATA	NO ACK	STOP BIT

Dans ce mode, avant de récupérer la donnée, nous transmettons d'abord l'adresse où se trouve la donnée que nous désirons. Pour ce faire, nous « l'écrivons » dans l'E²PROM.

Lecture multiple de plusieurs octets depuis l'adresse courante

START BIT	MOT CONTRÔLE lecture	ACK	DATA 1 Position n	ACK	DATA 2	...	NO ACK	STOP BIT

Ce mode n'a d'intérêt que si l'on vient de réécrire une donnée à une adresse antérieure, car sinon, la plupart du temps, les adresses suivantes sont vides.

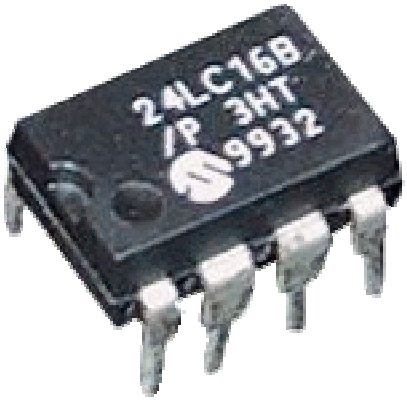
Lecture de plusieurs octets à une adresse donnée

START BIT	MOT CONTRÔLE écriture	ACK	ADRESSE	ACK	START BIT	MOT CONTRÔLE lecture	ACK	DATA 1	ACK	DATA 2	...	NO ACK	STOP BIT

Ce mode est véritablement plus intéressant que le précédent. Après avoir écrit l'adresse dans la mémoire, pour se positionner, la mémoire renvoie l'octet présent, puis passe à l'adresse suivante et recommence, et ainsi de suite, jusqu'au stop bit.

6.2.2- Les mémoires

Présentation des différents modèles



Une mémoire C16

Aujourd'hui, nombreux sont nos besoins en mémoire, avec l'explosion des appareils numériques : appareils photos, baladeurs MP3, pdas, et autres…

Un temps remplacés par les mémoires flash, alors plus rapides, avec les progrès de ces dernières années, les écarts entre ces deux technologies tendent à se réduire de plus en plus, et si les mémoires flash restent désormais ancrées dans les technologies mobiles, les mémoires EEPROMS sont largement utilisées dans le monde de l'électronique industriel.

Faciles à câbler, faciles à utiliser, à programmer, et ayant un coût relativement faible, ces mémoires s'appliquent également bien à des montages d'amateurs. Afin de répondre au mieux aux attentes des concepteurs, divers modèles ont été mis au point par Microchip.

Ces modèles se différencient principalement par la taille de stockage disponible. Les modèles vont en effet de 128 bits (16 octets, 24C00) à 512 Kbits (64 Koctets 24c512) ou plus.

Caractéristiques communes et respectives

Toutes ces mémoires possèdent le même boîtier DIP. De plus, elles sont, la plupart du temps, disponibles ou en version classique, ou

en version «LC», c'est-à-dire «low consommation», permettant ainsi de consommer moins.

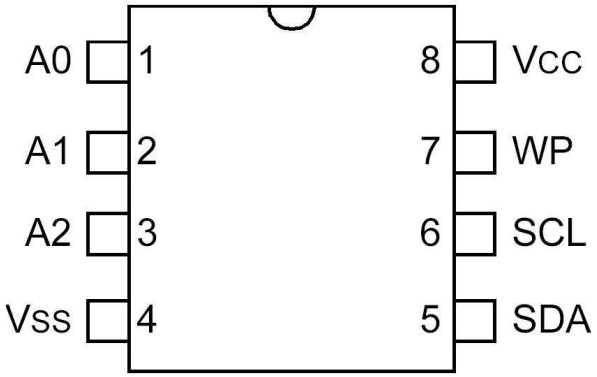
Puce	Taille Bit/octet	Alimentation	Protection écriture	Fréquence maximum	Durée écriture ms	Nombres écriture	Température fonctionnement
24AA00	128/16	1.8-6	non	400KHz	4	1 000 000	-40 +85 °c
24AA01	1024/128	1.8-5.5	oui		5		
24AA02	2048/256						
24AA04	4096/512						
24AA08	8192/1024						
24AA16	16384/2048						
24AA32	32768/4096						
24AA64	65536/8192						

Puce	Taille Bit/octet	Alimentation	Protection écriture	Fréquence maximum	Durée écriture ms	Nombres écriture	Température fonctionnement
24LC00	128/16	2.5-5.5	non	400KHz	4	1 000 000	-40 +85°c
24LC01	1024/128		oui		5		-40 +125 °c
24LC02	2048/256						
24LC04	4096/512						
24LC08	8192/1024						
24LC16	16384/2048						
24LC32	32768/4096						
24LC64	65536/8192						

Comme nous pouvons le constater à travers ces tableaux, la plupart de ces mémoires ont beaucoup de points communs. Ces tableaux résument les deux principaux types que l'on trouve les AA et les LC. Leurs principales différences tiennent dans les tensions d'alimentation, ainsi que dans leur consommation, comme dit précédemment. Qui plus est, les LC peuvent travailler dans des conditions plus extrêmes, côté température. A noter que d'autres types existe (C, LCS…).

<u>Remarque :</u> Pour une tension d'alimentation inférieure à 2.5V, la fréquence maximum de fonctionnement, passe alors à 100 Khz.

Fonctionnement

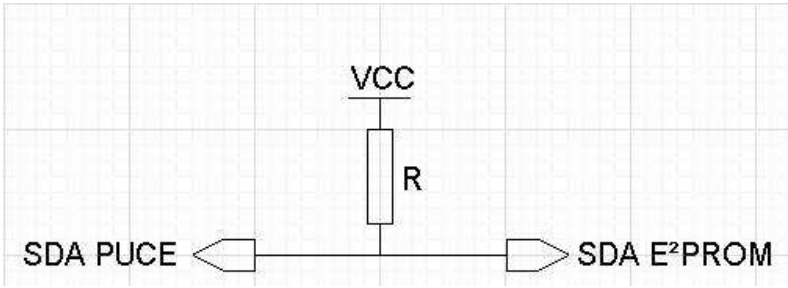


Les pattes A (A0, A1, A2), ne sont pas connectées mais peuvent servir à donner une adresse propre à la mémoire si plusieurs sont connectées au bus.

Le WP (Write Protection) sert à interdire (5V) ou autoriser (0V), de manière physique, l'écriture sur l'E²PROM.

Le SCL correspond à l'horloge de communication.

Le SDA est la patte de communication. Elle est utilisée pour les données. Cette patte doit être câblée de manière particulière, en effet, il faut une résistance de pull-up (10Kohms pour 100KHz, 2Kohms pour 400KHz).



Remarque: ***Le bit de poids fort est émis en premier. Seul le maître pilote l'horloge. Le bus est inactif si SDA=SCL=1. En bus actif, SCL=0.***

Format général d'une trame

START BIT	MOT CONTROLE	ACK	ADRESSE	ACK	DATA	ACK	STOP BIT

Les bits en jaune sont ceux émis par la puce pilotant la mémoire. Les cases en gris correspondent aux bits émis par la mémoire, les accusés de réception.

START BIT

Ce bit, correspondant au premier émis, est réalisé par un passage de SDA de l'état haut à l'état bas, pendant que SCL est à l'état haut.

STOP BIT

Ce bit, lui est réalisé par un passage de SDA de l'état bas à l'état haut, pendant que SCL est à l'état haut.

ACK

Ce ACK ou acknoledge, est important, car c'est lui qui permet de savoir si la mémoire a bien reçu les données ou non. Selon les types de transmissions, nous rencontrerons des ACK (0 logique), et des NO ACK (1 logique). Il y en a un après chaque transmission de mot adresse ou data (sauf cas particulier en lecture data).

ADRESSE/DATA

Ces deux mots font un octet chacun.

ETAT DE REPOS

A l'état de repos, les pattes SCL et SDA sont à 1 logique. En repos, la patte du microcontrôleur doit être en lecture pour savoir si le bus est occupé (SDA=0V) ou non.

ATTENTION : entre 2 communications, le bus doit être libre au minimum 5 µs. Le changement d'état des bits des octets d'adresse et de data doit se faire pendant que le SCL est à zéro.

MOT CONTRÔLE

1	0	1	0	B2	B1	B0	R/$\overline{W}$

R/W sert à indiquer à la mémoire si on désire lire des données, ou écrire des données.

L'ensemble des B (B2 B1 B0) sert à indiquer quelle page l'utilisateur désire écrire. Le tableau ci-dessous vous indiquera le nombre de pages, et leur taille selon la référence de votre mémoire.

PUCES	TAILLE DE LA PUCE octet	NOMBRE DE PAGE	TAILLE DE PAGE octet
24x00	16	1	16
24x01	128	1	128
24x02	256	1	256
24x04	512	2	256
24x08	1024	4	256
24x16	2048	8	256
24x32	4096	1	4096
24x64	8192	1	8192

GÉNÉRALITÉS

•Protection électrostatique : >4Kv
•Nombre de cycle de lecture/écriture : 1 000 000
•Durée de sauvegarde des données : >200 ans
•Fréquence maximale : 400KHz
•Alimentation : de 2.5V à 5.5V, 6.5V MAX

6.3- SPI

LE SPI (Serial Peripheral Interface) est un mode de communication série synchrone 3 fils, fonctionnant en mode maître-esclave. Plusieurs esclaves sont possibles, grâce à une patte dédiée sur le périphérique esclave SPI permettant la sélection de l'esclave avec lequel le maître désire dialoguer.

Le maître est seul capable de commencer un dialogue.

La différence du SPI par rapport à l'I²C est qu'il possède deux fils de données. Un pour les données entrantes, et l'autre pour les sortantes (comme pour le RS232). Ainsi, l'échange des données est plus rapide.

Ce protocole utilise donc 3 fils, voir 4 si l'on possède plusieurs esclaves:

- CS\ ou SS\: patte de sélection
- CMD ou SDI: entrée du périphérique
- CLK ou SCK: horloge
- DAT0 ou SDO: sortie du périphérique

***Remarque:* les cartes mémoires SD peuvent être utilisées en mode SPI. Cela est intéressant si on cherche une grande capacité mémoire et/ou une mémoire amovible.**

6.4- USB, HID

Nous allons parler rapidement ici de l'USB, en mode HID (Human Interface Device). En USB, il existe 4 grandes catégories:

- Display
- Communication (modem)
- Audio
- HID

Dans le cas du PIC 18Fx550, le compilateur retenu (voir chapitre suivant) configure le PIC en HID. L'USB est toutefois une option, mais le compilateur est alors livré avec un activex Visual Basic permettant une utilisation fort simple du PIC.

Outre le mode HID, le PIC est également configuré en mode HIGH SPEED USB 2.0.

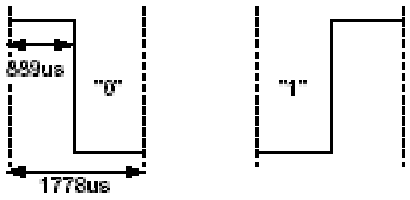
Ainsi, vous avez la possibilité de créer vos propres applications windows sur port USB, et ce simplement.

6.5- RC5

Le RC5 a été créé il y a maintenant quelques années afin de piloter les appareils audios/vidéos. Ce codage est si bon, que de nombreux industriels ont décidé de l'adopter, ce qui, de fait, a engendré une grande compatibilité des matériels. Ainsi, si vous désirez tout centraliser sur une télécommande infrarouge, il est conseillé d'utiliser du RC5.

Le RC5 est simple d'emploi, et n'impose principalement qu'une contrainte: le timing. En effet, tout se joue à la µs. Par défaut, l'état est "0".

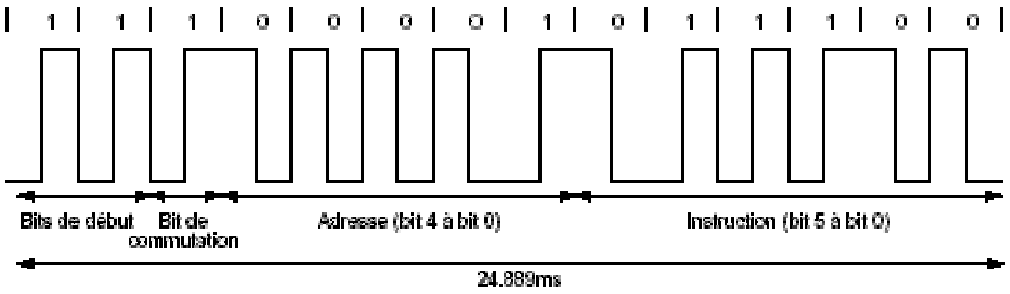
La grande particularité du RC5 est ce qu'on pourrait appeler le "codage manchester inversé". En effet, chaque bit est transmis par 2 états.



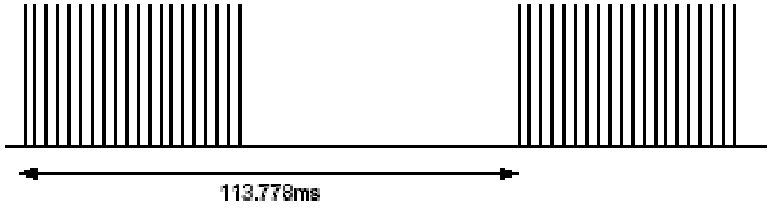
Une trame RC5 est constituée de 4 parties (14 bits en tout):

-les bits de start
-le bit de basculement (ou de commutation)
-les bits d'adressage système
-les bits d'instruction

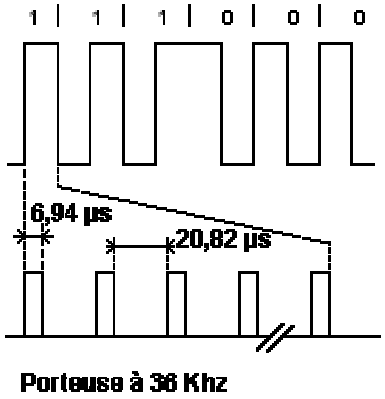
L'ensemble de ces parties fournit alors une trame complète.



Dans le cas, où l'on resterait appuyé sur la même touche, plusieurs signaux sont alors émis, à intervalle régulier: on parle de salve.



Afin d'économiser l'énergie, chaque "1" du signal est en réalité un signal carré à 36KHz, réalisé de la manière suivante:



Voyons en détail maintenant chaque bit:

START BIT: constitué de 2 zéros consécutifs, il indique à la cible le départ d'une trame RC5

Bit de commutation: Ce bit permet de savoir si l'on reste appuyé sur la touche (valeur du bit identique), ou si on appuie de manière répétitive.

Bits d'adresse: Ces bits permettent d'indiquer l'appareil cible (DVD, TV...)

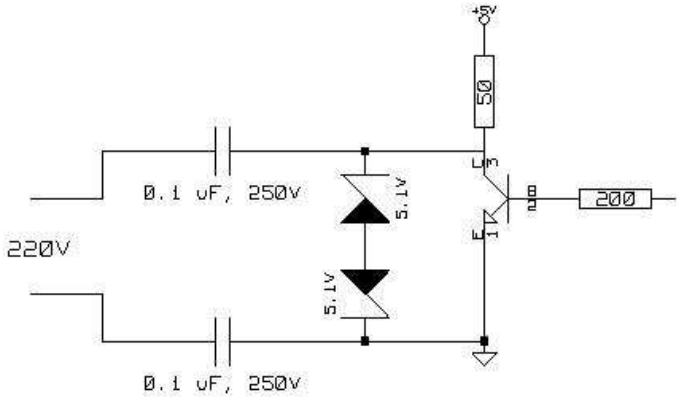
Bits d'instructions: Ces bits indiquent à la cible la fonction à réaliser.

Vous trouverez en annexe, les divers tableaux contenant les codes des appareils cibles, et des instructions.

6.6- Les bases du CPL

Le principe du CPL (Courant Porteur en Ligne) est fort simple: superposer le signal à tranmettre sur la sinusoïde secteur. Ce signal est un signal haute fréquence et à faible énergie. Cela permet d'utiliser le réseau électrique pour faire transiter des informations, et ainsi éviter de retirer du câble.

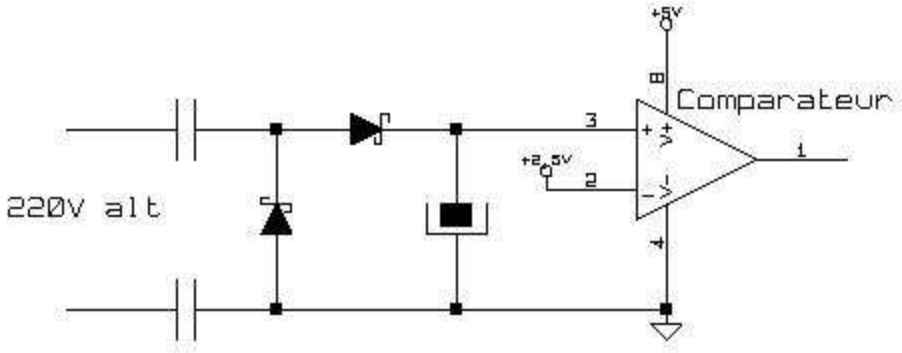
Le schéma suivant vous donne un exemple de l'électronique fort simple à mettre en oeuvre afin de pouvoir émettre des trames CPL. Comme vous pouvez le constater, le schéma est tellement simple que l'on se demande d'où sort le prix des modules des magasins. A noter toutefois, que le schéma est équivalent à un CPL bas débit.



Vous voyez que l'on se connecte directement au 220V. La résistance de 200 Ohms est connectée à un microcontrôleur qui génère le signal. Les diodes zéner permettent d'ajuster le niveau de tension. Enfin, l'astuce est au niveau des condensateurs. La résistance équivalente d'un condensateur en dynamique se calcule ainsi: Z= 1/(2*pi*f*C). Ainsi, à 50 Hz, la résistance est élevée, mais à 100 Khz, elle est très faible, permettant ainsi le passage du signal.

Le principal problème de ce type de communication est le bruit et les parasites. De fait, il y a une forte atténuation du signal, et il est souvent conseillé d'effectuer des redondances.

En ce qui concerne le récepteur, vous pouvez vous inspirer du schéma de l'émetteur, ne serait-ce que pour la partie d'isolation dynamique. Il faut ensuite, redresser le signal et supprimer la porteuse pour obtenir le signal original. Un exemple de schéma pourrait être ceci:



Ce schéma est là juste à titre indicatif; il n'a pas été testé, mais voici cependant son principe de fonctionnement: les condensateurs en entrée servent à laisser passer juste le signal de données, les diodes sont là pour le redresser. Vous noterez le choix des diodes schottky, choisies pour leurs faibles tensions de chute. Suit ensuite un

condensateur pour supprimer la porteuse et recouvrer le signal originel, et enfin un comparateur afin de reconstruire correctement le signal. On pourra choisir un rail to rail (sans tension de déchet) afin d'assurer une bonne qualité de signal.

Les principes conseillés pour les trames sont 2 starts bits, suivis des données, et d'un stop bit égal à 0. Tout comme le protocole RS 232, vous pouvez inclure un bit de parité.

Il existe deux catégories de CPL: haut et bas débit. Si un jour vous vous lancez dans la réalisation d'un tel système, il y a fort à parier que vous opterez pour le bas débit, tel le système présenté précédemment. En effet, le bas débit n'utilise qu'une simple modulation; généralement entre 100 et 120 Khz.

Ainsi, un "1" est représenté par un signal carré à 100 KHz.

Le protocole CPL le plus connu est le X10. Le bas débit est utilisé entre autre pour le KNEX ou Konnex (domotique).

CHAPITRE 7 : LA PROGRAMMATION DES PIC

7.1 Nécessaire à la programmation

7.2 Choix d'un langage

7.3 Le BASIC pour PIC

7.4 Un programmateur de PIC ICSP

7.1- Nécessaire à la programmation du PIC

Quand on veut programmer un PIC, il est nécessaire d'avoir un minimum de documents avec soi. Cependant, si ces derniers sont importants pour programmer en langage assembleur, nous verrons plus tard que selon le langage utilisé, certains deviennent quasi inutiles.

Ces documents, peu nombreux, traitent des tableaux de pages de programmation, de la répartition des ports en entrées/sorties et du type d'oscillateur utilisé (en effet, dans le cas par exemple d'un oscillateur RC, l'instruction SLEEP est alors utilisable)

Tous ces documents changent d'un PIC à l'autre. Ils sont cependant regroupés dans les chapitres détaillant chaque PIC.

Pour le reste, il vous faudra un programmateur de PIC (pour charger le programme dans le PIC) et un compilateur (pour créer le programme).

Ici, nous avons retenu le PICKIT2 de Microchip et un compilateur BASIC. Notre choix s'est appuyé sur plusieurs critères: viabilité du matériel, facilité d'utilisation, et coût d'achat.

Ainsi, avec du matériel sur port USB, un compilateur simple d'emploi, avec simulateur intégré, le tout pour moins de 100€ en version de base, cela est apparu comme un très bon rapport qualité/prix que ce soit pour un amateur ou un professionnel.

7.2- Choix d'un langage de programmation

Quand on choisit de programmer un microcontrôleur, il faut savoir que sauf exceptions, cas rare, nous avons le choix entre plusieurs langages de programmation. En ce qui concerne le PIC, les programmes se développent généralement avec le logiciel MPLAB, qui nous permet de programmer le PIC dans divers langages. Parmi ceux-ci, les principaux sont le C et le BASIC pour les langages hauts niveaux, et l'assembleur pour les langages bas niveaux.

Les langages bas niveaux sont les plus proches de la machine, généralement l'équivalent de la « langue natale » du système. L'assembleur en fait partie. Proche de la machine, ce langage permet d'optimiser le programme. Il reste cependant difficile à mettre en œuvre pour des programmes complexes, mais abordable pour des programmes de base. Adopté par nombre d'industriels comme langage de référence, il reste cependant difficilement réparable pour une tierce personne.

Les langages hauts niveaux sont des langages faciles à comprendre et à mettre en œuvre, avec un minimum de connaissance. Ce type de langage facilite la maintenance lors de problèmes, car il est aisé pour quelqu'un, pris au hasard, de comprendre un programme haut niveau. Ce type de langage est, en effet, plus proche de l'utilisateur que de la machine. Ces langages utilisent ce qu'on appelle des « compilateurs », sorte de convertisseurs, dictionnaires automatiques. Ces compilateurs font la conversion directe entre le langage haut niveau et le langage assembleur, nous épargnant ainsi une difficulté supplémentaire.

Comme cité précédemment nous avons sélectionné le langage BASIC. Cependant, pour ceux qui serait plutot intéressés par le langage C, ce type de compilateur est expliqué dans les livres sur le PIC16F84 et PIC16F628, disponibles en téléchargement sur le site de l'auteur.

Nous verrons donc dans ce livre, les bases du BASIC pour PIC. Pour rappel, un aide-mémoire de BASIC pour le compilateur retenu (PIC SIMULATOR IDE) est disponible en téléchargement sur le site de l'auteur.

7.3- Le BASIC pour PIC

Le langage BASIC est ce qu'on appelle un langage "haut niveau", c'est-à-dire que c'est un langage plus proche de l'humain que de la machine. Nous verrons ici les règles de bases du BASIC, afin que ceux qui ne connaissent pas le BASIC puissent se mettre à programmer rapidement.

Remarque: **il faut savoir que quand, sur les documentations techniques, on lit le mot « instructions », celles-ci font référence à une instruction assembleur. C'est le langage par défaut du PIC, sa « langue natale ».**

Les règles du BASIC sont simples. Un commentaire se met après un « ' ». Le programme se finit par "END", et un nom est donné à la fonction principale (ex: main:). Pour déclarer des variables, on utilise l'instruction DIM.

Voilà pour le principal. Pour le reste, les règles sont identiques aux autres langages. Veillez à utiliser des tabulations et retours à la ligne afin de rendre le code lisible. En ce qui concerne les mots clés spécifiques, ils sont détaillés par catégories dans l'aide-mémoire.

Enfin, voici pour exemple, un petit code en BASIC, faisant clignoter une diode:

```
Dim i as byte

TRISB=0x00

main:
    PORTB.0=1
    WAITMS 1000
    PORTB.0=0
    WAITMS 1000
    GOTO main
END
```

7.4- un programmateur de PIC ICSP

Nous allons voir ici, un programmateur de PIC fort pratique. Conçu par Microchip, et commandable chez Farnell en France pour environ 45€ TTC, ce programmateur de PIC se branche sur port USB (en autoalimentation), et est en fait construit sur un PIC. Le format du connecteur de sortie est en ICSP, format Microchip, que ce dernier essaie d'implanter sur tous ses PICs. D'ici quelques années, tous les PICs seront programmables via cette interface. A l'heure actuelle seuls ceux en fin de vie ne sont pas compatibles.

L'interface ICSP, pour In Chip Serial Programming, ou en français programmation embarquée (interprétation libre), permet de programmer un PIC directement sur la carte où il est implanté sans avoir à le sortir de la carte, ce qui est très appréciable.

Ce fabuleux programmateur s'appelle le PICKIT 2. Pour ceux qui voudrait plus d'informations sur l'ICSP, une doc est disponible en téléchargement sur le site de l'auteur, permettant entre autre de fabriquer une platine d'extension à ce programmateur.

Le PICKIT 2 est livré avec un logiciel dédié. Lors du lancement, la détection du PIC est automatique, et l'utilisation du logiciel même est relativement simple.

CHAPITRE 8 : CONCLUSION

Ainsi, comme nous avons pu le voir tout au long de ce livre, la maîtrise et l'exploitation d'un microcontrôleur PIC n'est pas si difficile que ça.

Difficile à appréhender en premier lieu, ils s'imposent comme éléments indispensables pour de nombreux montages, tant par leur rapport qualité/prix, que par leurs capacités ; leurs principaux handicaps étant en général le nombre limité de leurs entrées/sorties et/ou leurs capacités mémoires pour le programme.

Programmables avec un minimum de difficultés, ils sont parfaits pour les amateurs débutants. Quant aux utilisateurs confirmés, ils apprécient l'ensemble des fonctions à leurs justes valeurs, et savent repousser les limites des PICs par de petites astuces, toutes plus ingénieuses les unes que les autres.

De même, le matériel de programmation et de test, requiert le minimum de connaissances que l'on peut demander à un amateur.

De fait, l'amateur appréciera utiliser les PICs tant ils sauront lui simplifier la vie pour tous ses montages, quels qu'ils soient.

Pour conclure, nous pouvons dire que dans bien des cas, les PICs nous permettront d'arriver à nos fins avec un minimum de difficultés, et un maximum de simplicité.

CHAPITRE 9 : LEXIQUE

Adresse : correspond à l'endroit où l'on peut trouver les informations recherchées

Architecture : terme servant à définir la façon dont est organisée la structure d'un Circuit Intégré, tel un microcontrôleur

Bit : unité de base en informatique

BUS : sorte « d'autoroutes » pour les signaux, les BUS servent à transporter les informations entre deux parties du Circuit Intégré

Code : nom donné au texte constituant un programme informatique

DIL : norme de Circuit Intégré, définissant notamment les écarts entre les pattes

E²PROM : Electrically Erasable Programmable Read Only Memory ; se dit d'une mémoire qui peut s'écrire et surtout s'effacer électriquement

Flash : nouveau type de mémoire, avec de nouvelles possibilités

Fusible: se dit des bits des registres permettant de configurer le PIC

Interruption: événement peu courant devant déclencher une action spécifique

Microcontrôleur : Circuit Intégré (CI), contenant à la fois un processeur, des mémoires, et des entrées/sorties externes

Octet : ensemble de 8 bits

Oscillateur : se dit d'un composant, ou ensemble de composants capable de générer un signal régulier

PCB : nom donné aux typons, servant à tirer les plaques de circuit imprimé

Port : se dit d'un ensemble, d'un groupement d'entrées/sorties

RAM : Random Access Memory, aussi appelée mémoire vive, ce type de mémoire perd toutes ses informations stockées, lorsqu'elle cesse d'être alimentée

RISC : Reduced Instruction Set Computer ; se dit d'un processeur, ou microcontrôleur, possédant un nombre d'instructions de programmation réduit, limité

Schéma d'implantation : schéma montrant la façon, et le sens si nécessaire, dont doivent être montés les composants sur le circuit imprimé

Schéma Structurel : schéma montrant les représentations schématiques des composants, avec leurs liaisons électriques, correspondant aux pistes du PCB

Télécharger un programme : action de copier le code du PC dans le microcontrôleur

CHAPITRE 10 : LIENS INTERNET UTILES

Adresse de Microchip :

http://www.microchip.com/

Adresse de Farnell (PICKIT2):

http://www.farnell.fr/

Adresse du compilateur BASIC:

http://www.oshonsoft.com/

Adresse de l'auteur:

http://diablotronic.bzh.bz

CHAPITRE 11 : ANNEXES

11.1- Fréquence musicale

Aussi pratique soit la fonction de génération de note du compilateur BASIC, aucune note n'est enregistrée. Se pose alors la question des fréquences de chaque note. A ce titre, voici les fréquences arrondies d'une gamme complète:

NOTES	FRÉQUENCE EN Hz
DO	264
DO#	275
RÉ	297
RÉ#	317
MI	330
FA	352
FA#	371
SOL	396
SOL#	413
LA	440
LA#	475
SI	495
DO	528

11.2- Les codes RC5

Comme promis lors de l'explication du RC5, voici les codes.

Code RC5 des adresses des appareils

Adresse-Système . . . . Appareil

- 0 . . . . . .TV1
- 1 . . . . . .TV2
- 2 . . . . . .Vidéotexte
- 3 . . . . . .Extension pour TV1 et TV2
- 4 . . . . . .Laser Vision Player
- 5 . . . . . .Magnétoscope1 (VCR1)
- 6 . . . . . .Magnétoscope 2 (VCR2)
- 7 . . . . . .Réservé
- 8 . . . . . .SAT1
- 9 . . . . . .Extension pour VCR1 et VCR2
- 10 . . . . . .SAT2
- 11 . . . . . .Réservé
- 12 . . . . . .CD-Vidéo
- 13 . . . . . .Réservé
- 14 . . . . . .CD-Photo
- 15 . . . . . .Réservé
- 16 . . . . . .Préampli Audio1
- 17 . . . . . .Tuner
- 18 . . . . . .Magnétocassette analogique
- 19 . . . . . .Préampli Audio2
- 20 . . . . . .CD
- 21 . . . . . .Rack Audio ou Enregistreur
- 22 . . . . . .Récepteur satellite Audio
- 23 . . . . . .Magnéto DCC
- 24 . . . . . .Réservé
- 25 . . . . . .Réservé
- 26 . . . . . .CD Inscriptible
- 27 à 31 . . . . . .Réservé

Code RC5 des Instructions communes à toutes les adresses

Instruction . . . . . Signification

- 0 0
- 1 1
- 2 2
- 3 3
- 4 4
- 5 5
- 6 6
- 7 7
- 8 8
- 9 9
- 16 . . . . . .Volume +
- 17 . . . . . .Volume –
- 18 . . . . . .Brightness +
- 19 . . . . . .Brightness –
- 20 . . . . . .Color saturation +
- 21 . . . . . .Color saturation –
- 22 . . . . . .Bass +
- 23 . . . . . .Bass –
- 24 . . . . . .Treble +
- 25 . . . . . .Treble –
- 26 . . . . . .Balance right
- 27 . . . . . .Balance left
- 63 . . . . . .System select
- 71 . . . . . .Dim local display
- 77 . . . . . .Linear function increment
- 78 . . . . . .Linear function decrement
- 80 . . . . . .Step up
- 81 . . . . . .Step down
- 82 . . . . . .Menu on
- 83 . . . . . .Menu off
- 84 . . . . . .Display A/V system status
- 85 . . . . . .Step left
- 86 . . . . . .Step right
- 87 . . . . . .Acknowledge
- 88 . . . . . .PIP on/off (Pay TV channel + for system 3)
- 89 . . . . . .PIP shift (Pay TV channel - for system 3)
- 90 . . . . . .PIP / main swap (Radio channel + for system 3)
- 91 . . . . . .Strobe on/off (Radio system – for channel 3)
- 92 . . . . . .Multi strobe (Date + for system 9)
- 93 . . . . . .Main frozen (Date – for system 9)
- 94 . . . . . .3/9 multi-scan (Start time + for system 9)
- 95 . . . . . .PIP select (Start time – for system 9)
- 96 . . . . . .Mosaic/multi-PIP (Record program + for system 9)
- 97 . . . . . .Picture DNR (Record program – for system 9)
- 98 . . . . . .Main stored (Alternate channel for system 9)

- 99 . . . . . .PIP strobe (Stop time + for system 9)
- 100 . . . . . .Recall main picture (Stop time – for system 9)
- 101 . . . . . .PIP freeze
- 102 . . . . . .PIP step up +
- 103 . . . . . .PIP step down –
- 118 . . . . . .Sub mode
- 119 . . . . . .Options sub mode
- 123 . . . . . .Connect
- 124 . . . . . .Disconnect

Autres code RC5 : Commandes des adresses 0 et 1 (TV1 / TV2)

Commande . . . . . Signification

- 10 . . . . . .1/2/3 digits / 10
- 11 . . . . . .Freq./prog./ch./11
- 12 . . . . . .Standby
- 13 . . . . . .Mute/de-mute
- 14 . . . . . .Personal pref.
- 15 . . . . . .Display
- 28 . . . . . .Contrast +
- 29 . . . . . .Contrast –
- 30 . . . . . .Search +
- 31 . . . . . .Tint/hue –
- 32 . . . . . .Ch./prog. +
- 33 . . . . . .Ch./prog. –
- 36 . . . . . .Spatial stereo
- 37 . . . . . .Stereo/mono
- 38 . . . . . .Sleep timer
- 39 . . . . . .Tint/hue. +
- 40 . . . . . .RF switch
- 41 . . . . . .Store/execute/vote
- 42 . . . . . .Time
- 43 . . . . . .Scan fwd./increm.
- 44 . . . . . .Decrement
- 46 . . . . . .Sec con/menu
- 47 . . . . . .Show clock
- 48 . . . . . .Pause
- 49 . . . . . .Erase/correct
- 50 . . . . . .Rewind
- 51 . . . . . .Go to
- 52 . . . . . .Wind
- 53 . . . . . .Play
- 54 . . . . . .Stop
- 55 . . . . . .Record
- 56 . . . . . .External 1
- 57 . . . . . .External 2
- 59 . . . . . .Advance

- 60 . . . . . .TXT sub-mode/12
- 61 . . . . . .Sys. Standby
- 62 . . . . . .Crispener
- 70 . . . . . .Speech/music
- 79 . . . . . .Sound scroll
- 104 . . . . . .PIP size
- 105 . . . . . .Pic. Scroll
- 106 . . . . . .Act. On/off
- 107 . . . . . .Red
- 108 . . . . . .Green
- 109 . . . . . .Yellow
- 110 . . . . . .Cyan
- 111 . . . . . .Index/white
- 112 . . . . . .Next
- 113 . . . . . .Previous
- 122 . . . . . .Store open/close
- 126 . . . . . .Movie expand
- 127 . . . . . .Parental access