LE PIC 16F628: L'ESSENTIEL

Auteur: Galodé Alexandre

Date: 08/02/2005

DIABLOTRONIC

INTRODUCTION

Ce livre, à pour but de vous présenter le PIC16F628, et d'apprendre à le programmer. Nous pouvons le considérer comme complémentaire au livre « Le PIC16F84 : L'essentiel ».

Tout comme pour ce dernier, nous ne verrons que le strict nécessaire, sans superflu inutile, qui parfois nous embrouille l'esprit. Seules les principales fonctions et celles les plus utilisées seront vues ici.

Ainsi, certaines possibilités du 628, comme par exemple le PWM (modulation en largeur d'impulsion), ne seront pas expliquées ici.

Bonne lecture et bon apprentissage!!!

SOMMAIRE

1- Le PIC 16F628 P2
1.1 Présentation du microcontrôleur P3
1.2 Caractéristiques du PIC 16F628 P8
1.3 Fonctionnement du PIC 16F628 P10
1.3.1 Les entrées/sorties
1.3.2 Les différents mode d'horloge
1.3.3 Le reset
1.3.4 Les interruptions
1.3.5 Les timers
1.3.6 L'USART
1.3.7 La tension de référence
1.3.8 Les comparateurs

2- Programmation du PIC 16F628 P16
2.1 Nécessaire à la programmation P17

3- Programmer le PIC P37
3.1 Un programmateur de PIC P38
3.2 Les logiciels de programmation P41

4- Un testeur de PIC P

5- En bref, tout ce qui est utile P

6- Conclusion P

7- Lexique P

8- Liens Internet utiles P

CHAPITRE 1 : LE PIC 16F628

1.1 Présentation du microcontrôleur

1.2 Caractéristiques du PIC 16F628

1.3 Fonctionnement du PIC 16F628

1.3.1 Les entrées/sorties

1.3.2 Les différents modes d'horloge

1.3.3 Le Reset

1.3.4 Les interruptions

1.3.5 Les Timers

1.3.6 L'USART

1.3.7 La tension de référence

1.3.8 Les comparateurs

1- LE PIC 16F628

1.1- Présentation du microcontrôleur :

Le PIC 16F628

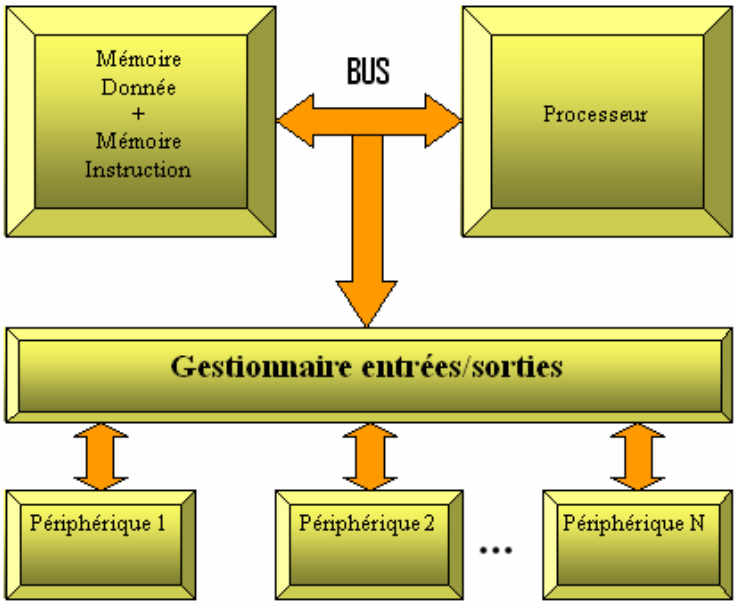
Le PIC 16F628, que nous appellerons 628 par commodité, est souvent présenté et à juste titre, comme le successeur du PIC 16F84. En effet, le 628 est compatible patte à patte avec le 16F84. De plus, au niveau programmation, seul un léger rajout, au niveau de l'initialisation des différents registres est nécessaire. De ce fait, avec un prix encore plus bas que le PIC 16F84, il est indéniable, que le 628 finisse par s'imposer.

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…

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

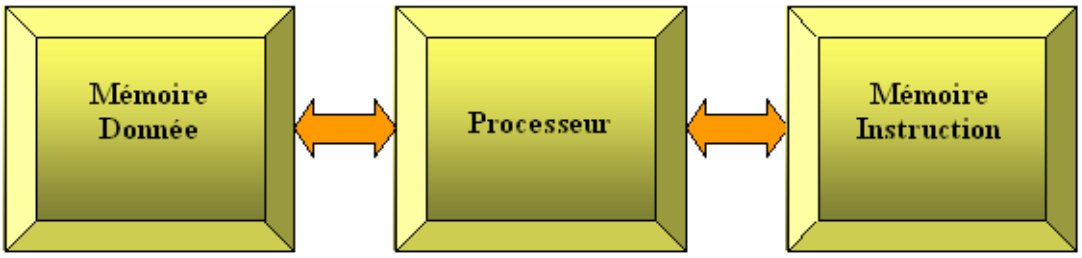
Jusqu'à une certaine époque, les microcontrôleurs respectaient l'architecture Von Neumann, inventeur de l'ENIAC, premier ordinateur au monde. Cependant, celle-ci présente des inconvénients. En effet, la vitesse d'exécution est limitée, les instructions et les données transitaient par le même bus. Pour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



L'architecture de Von Neumann

D'où l'architecture Harvard, utilisée maintenant par les PIC. 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.

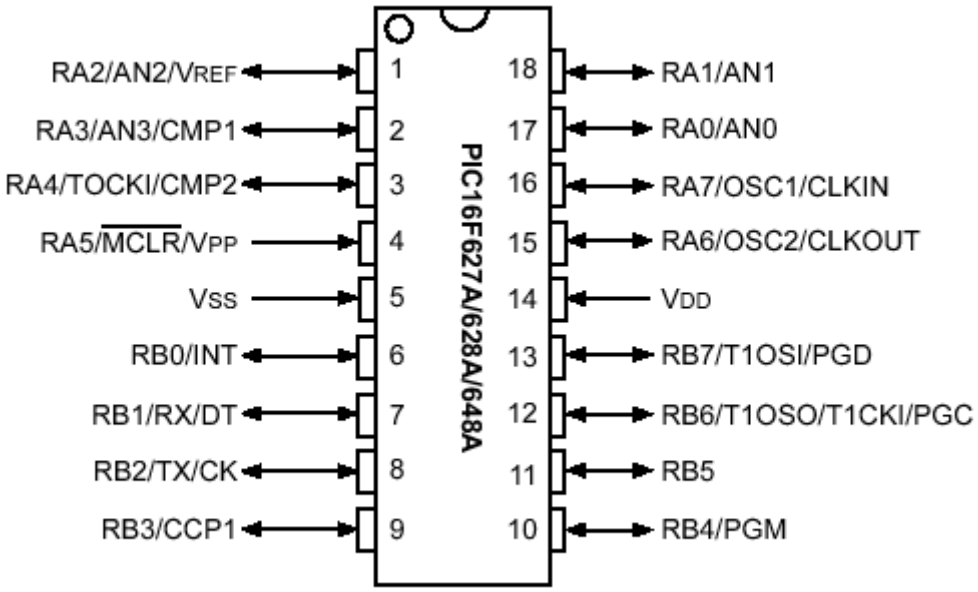
L'un sert pour les données, et l'autre pour les instructions. De ce fait, les deux peuvent être accessibles en même temps, d'où un gain de vitesse, au niveau exécution.



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 E²PROM pour des données auxiliaires.

1.2- Caractéristiques du PIC 16F628 :



Le 628 possède 13 pattes d'E/S, tout comme le 16F84, nombre qui peut monter jusqu'à 16 E/S, selon les configurations. Ces E/S sont, tout comme chez le 16F84, réparties sur deux ports (A et B).

***Rappel :* 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.**

Le port A possède, normalement 5 broches (nommées RA1 à RA4), et ce nombre peut monter jusqu'à 8 (RA1 à RA7).

Le port B, lui, possède 8 broches (de RB0 à RB7)

Outre ces caractéristiques, le 628 possède des fonctions intégrées particulièrement intéressantes. Ainsi, en plus des fonctions du 16F84, le 628, possède 4 CAN (aux entrées des comparateurs), 2 comparateurs, dont les valeurs de sortie peut être connue logiciellement, une source de tension de référence configurable de 0 à 3,6V, une connectique de transmission asynchrone (série Tx et Rx), une connectique de transmission synchrone (DT et Ck), 2 timers supplémentaires (3 en tout), enfin, un système d'horloge RC intégré à 4 MHz (fréquence légèrement variable selon la température ambiante du 628). Voilà, ainsi, résumées les principales améliorations du 628. Il va sans dire, que toutes ces

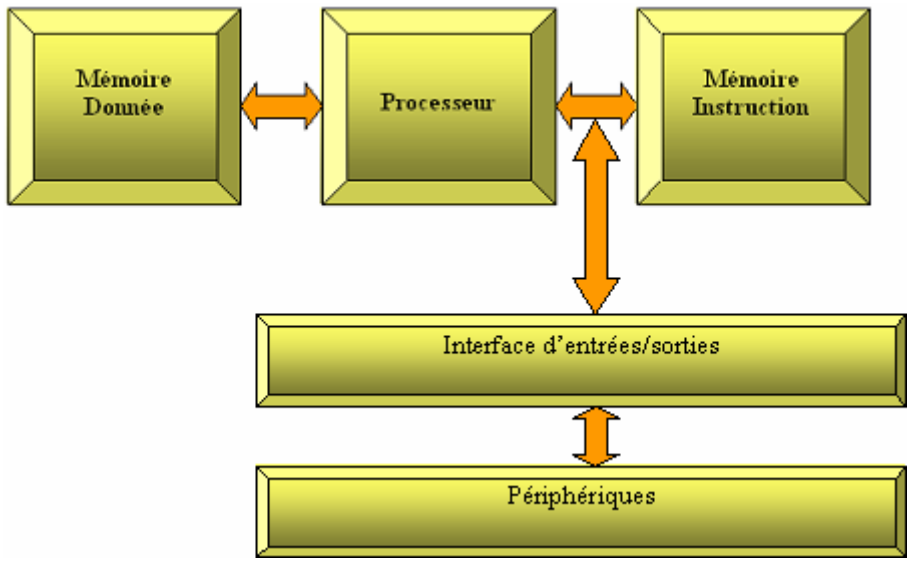
fonctions ne peuvent pas être utilisées simultanément, il qu'il faudra bien choisir ce dont on a besoin.

Voici enfin, les caractéristiques générales du PIC 16F628 fourni par Microchip :

- Mémoire de programme : 2KO, type Flash
- Mémoire de données RAM : 224 octets
- Mémoire de données E²PROM : 128 octets
- Cause d'interruption : 10
- Fréquence max de travail : 20 MHz (horloge interne de 4 MHz)
- Lignes E/S numérique : de 13 à 16 (attention, RA5 n'est qu'une entrée)
- Temporisateur : 3 pour l'utilisateur, un pour le Watchdog
- Tension d'alimentation : 3 à 5,5 V continu
- Tension de programmation : 12 à 14 V continu (compatible 16F84)
- Boîtier : DIL 18

1.3- Fonctionnement du PIC 16F628:

1.3.1- 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. Chaque valeur issue d'un CAN peut être comparée (dans les comparateurs) avec la tension de référence interne.

PATTE 1 : E/S RA2, CAN 2, tension de référence

PATTE 2 : E/S RA3, CAN 3, Sortie du comparateur 1

PATTE 3 : E/S RA4, entrée Timer 0 TOCKI, Sortie comparateur 2

PATTE 4 : E RA5, reset externe, programmation type F84 (compatible)

PATTE 5 : masse

PATTE 6 : E/S RB0, source d'interruption

PATTE 7 : E/S RB1, Rx (réception asynchrone, série), Dt broche de données (transmission synchrone)

PATTE 8 : E/S RB2, Tx (transmission asynchrone, série), Ck broche de

synchronisation de transmission synchrone

PATTE 9 : E/S RB3, Capture avec le timer 1, Comparaison avec le timer 1, PWM avec le timer 2

PATTE 10 : E/S RB4, programmation basse tension (non expliquée ici)

PATTE 11 : E/S RB5

PATTE 12 : E/S RB6, T1OSC0 (à utiliser avec T1OSC1) quartz externe pour le timer 1 (de 32 à 200 KHz), entrée d'horloge pour le timer 1

PATTE 13 : E/S RB7, T1OSC1 (voir ci-dessus)
PATTE 14 : Alimentation positive (5V typique)

PATTE 15 : E/S RA6, OSC2 entrée horloge (brancher une résistance pour un oscillateur RC) ou quartz (à utiliser avec un quartz), CLKOUT sortie d'horloge pour le mode 4 & 6

PATTE 16 : E/S RA7, OSC1 (voir ci-dessus), CLKIN entrée horloge externe (type GBF, ou autre)

PATTE 17 : E/S RA0, CAN 0

PATTE 18 : E/S RA1, CAN 1

1.3.2- Les modes d'horloge :

Dans cette partie, nous allons voir les différents modes d'horloge. On peut les dénombrer aux nombres de 6. Le mode de fonctionnement se fait sous ICPROG, dans la fenêtre oscillateur. Nous préciserons entre parenthèse les broches utilisées pour l'horloge. Dans le cas d'un quartz (mode1), vous pourrez choisir à la programmation ou horloge XT (jusqu'à 4 Mhz), ou HS (jusqu'à 20 MHz).

MODE 1 : fonctionnement classique avec quartz (OSC1 & OSC2)

MODE 2 : Horloge externe (ex : NE555, GBF,…) (OSC1)

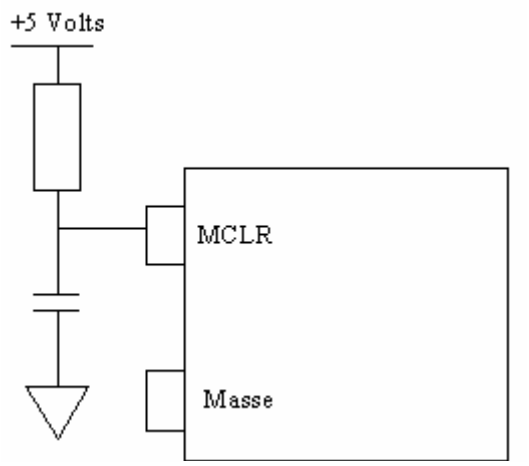
MODE 3 : Horloge interne. Fréquence unique : 4 MHz (Aucune)

MODE 4 : idem MODE 3, mais OSC2 génère l'horloge divisée par 4

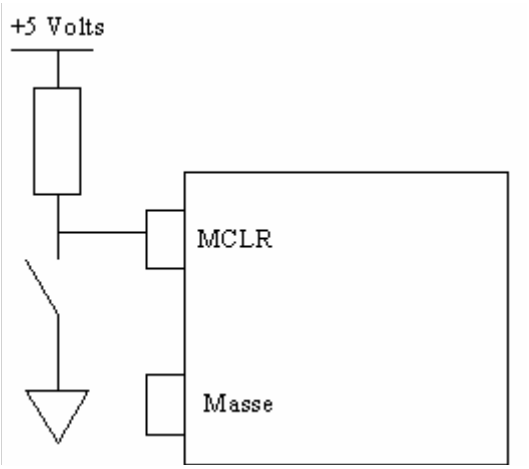
MODE 5 : Horloge externe. Branchez une R entre OSC1 et la masse

MODE 6 : Idem MODE 5, mais OSC2 génère l'horloge divisée par 4

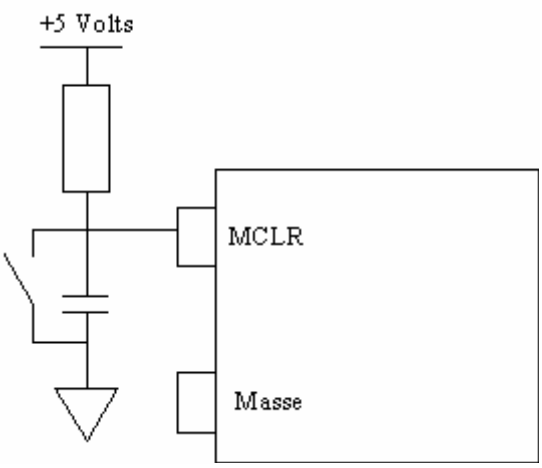
1.3.3- Le reset :



Le reset automatique (condensateur de 1µF et une résistance de 1 K)



Le reset manuel (avec une résistance de 1K)



Le modèle mixte, permettant un reset automatique lors de la mise sous tension et également un interrupteur de reset manuel

Le reset sur le 628 peut être de deux types. Classique, comme sur le 16F84, ou bien automatique (case BODEN à cocher sous ICprog, pour la surveillance watchdog et case PWRT pour un reset au démarrage).

1.3.4- La mémoire E²PROM :

L'E²PROM(Electrically Erasable Programmable Read Only Memory) est une mémoire interne au Pic. Il s'agit d'une mémoire non volatile dans laquelle le PIC peut stocker des données, comme par exemple les résultats d'une acquisition.

1.3.5- La 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

1.3.6- Les Interruptions :

Les interruptions passent ici au nombre de 10. Nous allons ici voir la liste des interruptions possibles :

- Externe : RB0
- Débordement du timer 0
- Changement d'état des broches RB4 à RB7
- Modules de comparaison
- USART
- Module CCP (pour la modulation PWM)
- Débordement du timer 1
- Timer 2

1.3.7- Les TIMERS :

Chez le 628, ils sont au nombre de 3.

Le premier (timer 0) est identique à celui du 16F84 (8 bits).
Le second (timer 1) est sur 16 bits, permettant ainsi d'étendre les possibilités.
Le troisième (timer 2), lui sur 8 bits, possède un pré et un post-diviseur, permettant ainsi de générer un signal PWM.

Timer0 :
Il s'agit d'un temporisateur interne de 8 bits, qui peut être initialisé à une valeur donnée. A chaque passage de FF à 00 (en hexa), le bit de

débordement est activé. Il faut alors le remettre à zéro, pour pouvoir détecter un autre débordement (non automatique).

Il possède deux modes de fonctionnement possible, dont le choix s'effectue par la mise à 1 ou à 0 du bit TOSC (voir chap. nécessaire à la programmation), l'entrée horloge devenant alors la patte RA4, en mode dit « TOCKI ».

Ces deux modes sont :

► temporisateur interne (peut alors servir pour des fonctions de temps)
► Compteur d'évènements (peut servir pour compter des évènements extérieurs par l'intermédiaire de RA4)

Remarque : la patte RA4 doit être définie en entrée dans le cas du compteur d'évènements.

1.3.8- L'USART :

Sous ce nom un peu barbare se cache en fait la possibilité de communiquer avec un autre système ou microcontrôleur de façon asynchrone ou synchrone.

1.3.9- La tension de référence:

Le 628 possède une source de tension de référence interne. Nous pouvons ainsi sur la patte 1 avoir, pour une tension d'alimentation de 5 V, une tension de référence comprise entre 0 et 3,6 V. Pour choisir cette tension, on utilise le registre VRCON :

VREN	VROE	VRR		VR3	VR2	VR1	VR0

Le bit VREN sert à la mise en marche (niveau 1) ou non (niveau 0) de la tension de référence (la patte 1 du 628 sert alors de tension de référence externe). Le bit VROE, lui sert à appliquer cette Vref à l'entrée du comparateur (niveau 1) ou non (niveau 0). Le VRR, lui sert à choisir deux intervalles de tensions possibles (VR3.0 désignera la valeur décimale codée sous forme binaire dans VR3, VR2, VR1 et VR0) :

VRR=0 :

Vref= Valim / 4+ (VR3.0 / 32)* Valim

Nous obtenons ici une gamme allant de 1.25 à 3.59 V

VRR=1 :

Vref= (VR3.0 / 24)* Valim

Ici, la gamme va de 0 à 3.13V

Toutes ces tensions sont obtenues, pour informations, par l'intermédiaire d'un réseau de résistance.

1.3.10- Les comparateurs:

Le 628 possède deux comparateurs intégrés. Deux états de sorties sont possibles : niveau 1 si V+>V-, niveau 0 si V+<V-

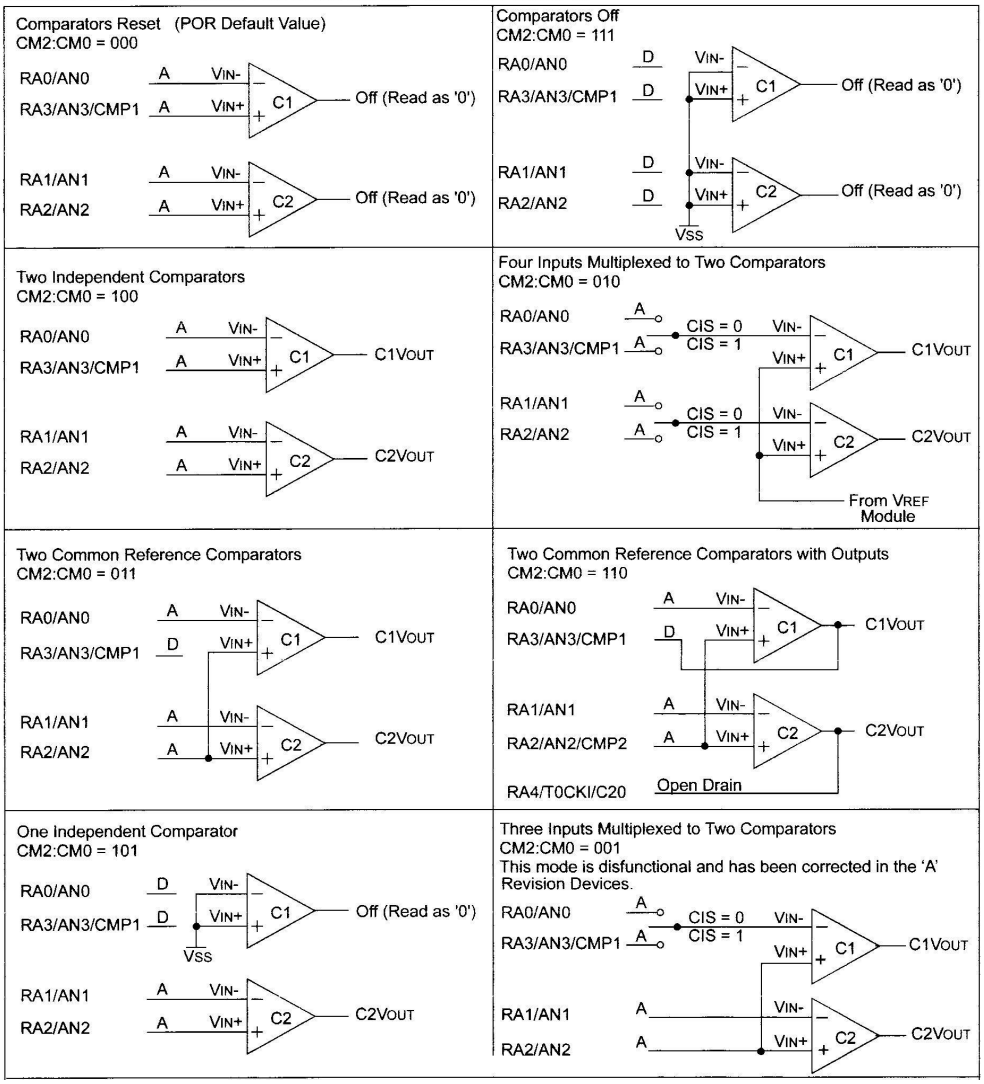
C'est le registre CMCON qui sert à configurer les comparateurs :

C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0

C2OUT : image de sortie du comparateur 2 (non écrivable)
C1OUT : image de sortie du comparateur 1 (non écrivable)
C2INV : inversion de la sortie du comparateur 2
C1INV : inversion de la sortie du comparateur 1
CIS : multiplexage d'entrées
CM2, CM1, CM0 : permettent de définir un mode de fonctionnement des Comparateurs

Les comparateurs fonctionnent selon le tableau suivant :

FIGURE 9-1: COMPARATOR I/O OPERATING MODES



Ce tableau montre, selon la configuration des CMx, les différents types de fonctionnement possible, ainsi que le câblage interne effectué.

1.3.11- Le Watchdog :

Littéralement le « chien de garde », le Watchdog est un système de surveillance du bon déroulement du programme. Il s'agit d'un compteur, qui est réinitialisé régulièrement dans le cas d'un fonctionnement normal. Mais dans le cas d'un dysfonctionnement, le compteur va jusqu'au bout et déclenche alors un reset interne, par débordement, réinitialisant le Pic.

Le compteur peut fonctionner à la fréquence de l'oscillateur, ou bien à une fréquence spécifique, désignée par la fréquence de l'oscillateur modifiée par un diviseur.

***Remarque :* il n'est pas obligatoire de l'utiliser. Son utilisation est à activer ou non, lors de la programmation du PIC**

1.3.12- Le RTCC:

Il s'agit d'une horloge interne destinée au fonctionnement du timer 0 (TMR0) dans le cas d'un fonctionnement de ce dernier sur horloge interne.

CHAPITRE 2 : PROGRAMMATION DU PIC 16F628

2.1 Nécessaire à la programmation

2.1- Nécessaire à la programmation du PIC :

Les bases de programmation, en C, sont les mêmes que pour le 16F84. Les noms de registres devront être écrits en majuscule. Tous les paramètres expliqués ici, seront vus d'un point de vu, programmation en C.

Nous allons voir dans cette partie, les registres, encore non vus, permettant la programmation du 628

STATUS :

IRP	RP1	RP0	/TO	/PD	Z	DC	C

Ce registre contient tous les bits qui servent pour nous renseigner sur l'état du PIC, et pour une partie de la programmation. Seuls trois bits nous intéressent : le C, le RP1 et le RP0.

Le bit C (pour carry, retenue en français), est à un ou zéro selon que l'opération effectuée nécessitait une retenue ou non.

Le RP0 et le RP1 sont les bits permettant d'avoir accès au tableau de programmation, vu précédemment (et donc au bit système des registres) et aux différents espaces mémoires.

Ce registre ne sert principalement qu'en assembleur et pour les programmeurs confirmés ; raison pour laquelle nous ne nous attarderontspas dessus.

TRISA :

RA7	RA6	RA5	RA4	RA3	RA2	RA1	RA0

Le TRISA est le registre permettant de définir les entrées/sorties du Port A. En remplaçant les RAx par leur valeur correspondante (1 pour une entrée, 0 pour une sortie), on obtient alors le mot binaire à rentrer dans TRISA. Quand aux bits 7, 6 et 5, leurs valeurs dépendent qu'ils soient ou non utilisés en E/S.

TRISB :

RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0

Le TRISB fonctionne de la même manière que le TRISA.

OPTION :

/RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0

Ce registre permet de définir le choix d'un diviseur et son affectation au Watchdog ou au RTCC

Le choix de ce diviseur, se fait par l'intermédiaire de PS2, PS1, et PS0, selon le tableau suivant :

PS2	PS1	PS0	Rapport de division	
			WDT	RTCC
0	0	0	:1	2
0	0	1	2	4
0	1	0	4	8
0	1	1	8	16
1	0	0	16	32
1	0	1	32	64
1	1	0	64	128
1	1	1	128	256

L'affectation au Watchdog (WDT) ou à l'horloge temps réel (RTCC) se fait par l'intermédiaire du bit PSA.

Dans le cas de l'affectation du diviseur au RTCC, le bit TOSE permet de choisir entre un déclenchement sur front montant ou sur front descendant.

Le bit INTEDG permet lui, de choisir le déclenchement de l'interruption (sur front montant ou sur front descendant).

Enfin, le bit /RBPU, permet de choisir de la mise en place ou non d'une résistance de rappel entre RB4 et RB7.

Remarque : Dans le cas d'un éventuel reset, ou à la mise sous tension, tous les bits sont par défaut à '1'.

Tout ceci est résumé dans le tableau suivant :

Bit PSA	0 = Choix du RTCC
	1 = Choix du WDT
Bit TOSE	0 = Front montant
	1 = Front descendant
Bit TOCS	0 = Horloge interne
	1 = Front sur la broche RTCC
Bit INTEDG	0 = interruption (RB0) validé sur front descendant
	1 = interruption (RB0) validé sur front montant
$\overline{\text{Bit RBPU}}$	0 = Connection d'une résistance de rappel entre RB4 et RB7
	1 = Aucune résistance de rappel

INTCON :

GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF

Bit du registre INTCON	NOM	Fonction
Bit 7	GIE (Global Interrupt Enabled Bit)	Bit de validation général des interruptions (à 1 pour activer les interruptions)
Bit 6	PEIE	Activation ou non des périphériques
Bit 5	TOIE (Timer 0 Interrupt Enabled Bit)	Interruption pour le Timer 0
Bit 4	INTE (INTerrupt pin Enabled Bit)	Interruption sur la patte RB0
Bit 3	RBIE (RB port change Interrupt Enabled bit)	Interruption sur les pattes RB4 à RB7
Bit 2	TOIF (Timer 0 Interrupt Flag bit)	Signal que l'interruption vient du débordement du Timer 0
Bit 1	INTF (INTerrupt pin Flag bit)	Signal que l'interruption provient d'un changement d'état de RB0
Bit 0	RBIF (poRt B Interrupt Flag bit)	Signal que l'interruption provient des pattes RB4 à RB7

Ce tableau résume la façon dont on peut activer et vérifier les interruptions actives.

Module Timer1 (T1CON) :

-	-	T1CKPS1	T1CKPS0	T1OSCEN	/T1SYNC	TMR1CS	TMR1ON

T1CKPS1 & T1CKPS0 : valeur de la prédivision d'horloge du timer 1

11 : division par 8
10 : division par 4
01 : division par 2
00 : division par 1

T1OSCEN : activation (1) ou non (0) de l'oscillateur du timer 1

/T1SYNC : bit de contrôle de l'entrée d'horloge externe de Synchronisation. Dépend de TMR1CS. Si ce dernier égal 0, utilisation de l'horloge interne. Si TMR1CS égal 1, si /T1SYNC égal 0, alors il y a synchronisation avec l'horloge externe ; sinon, il n'y aura pas de synchronisation.

TMR1CS : bit de sélection d'horloge externe (1 : depuis RB6, front montant) ou horloge interne (0 : Fosc/4)

TMR1ON : activation (0) ou non (1) du timer 1

PIE1 :

EEIE	CMIE	RCIE	TXIE	-	CCP1IE	TMR2IE	TMR1IE

EEIE : active (1) ou non (0) l'interruption d'écriture EEprom

CMIE : active (1) ou non (0) l'interruption des comparateurs

RCIE : active (1) ou non (0) l'interruption de réception USART

TXIE : active (1) ou non (0) l'interruption de transmission USART

CCP1IE : active (1) ou non (0) l'interruption du CCP1

TMR2IE : active (1) ou non (0) l'interruption de comparaison entre Timer2 et PR2

TMR1IE : active (1) ou non (0) l'interruption de débordement du timer 1

PIR1 :

EEIF	CMIF	RCIF	TXIF	-	CCP1IF	TMR2IF	TMR1IF

EEIF : indique si l'écriture E²PROM est finie (1, à remettre à 0 logiciellement) ou non terminé (0), ou non commencée (0)

CMIF : indique si la sortie des comparateurs à changée (1) ou non (0)

RCIF : indique si le buffer réception de l'USART est plein (1) ou non (0)

TXIF : indique si le buffer transmission de l'USART est plein (1) ou non

CCP1IF :

- mode capture : une capture TMR1 a été effectuée (1, reset logiciel) ou non
- mode comparaison : une comparaison TMR1 a été effectuée (1, reset logiciel) ou non

TMR2IF : une comparaison entre Timer2 et PR2 a été effectuée (1, reset logiciel) ou non (PR2 est le registre de période du timer2, utilisation en PWM)

TMR1IF : débordement du timer 1 (1, reset logiciel) ou non (0)

Module Timer2 (T2CON) :

-	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0

TOUTPS3/TOUTPS2/TOUTPS1/TOUTPS0 : valeur en binaire du prédéviseur de sortie du timer 2 (0000 (1:1) à 1111(1:16) par incrémentation de 1)

TMR2ON : activation (1) ou non (0) du timer 2

T2CKPS1/T2CKPS0 : valeur en binaire du prédéviseur de l'horloge du timer 2 (00 (1:1), 01 (1:4), 1x (1:16))

RCSTA :

SPEN	RX9	SREN	CREN	ADEN	FERR	OERR	RX9D

SPEN : Bit d'activation (1) ou non (0) du port série

RX9 : Bit de sélection du nombre de bits en réception (8 :0 ou 9 :1)

SREN : en mode synchrone maître, active (1) ou non la simple réception. Ce bit est effacé après chaque réception

CREN : en mode asynchrone, active (1) ou non la réception continue. En mode synchrone, active la réception continue jusqu'à mise à zéro de CREN

ADEN : en asynchrone 9 bits, active (1) ou non (0) la détection d'@, les interruptions et chargement du buffer de réception.

FERR : erreur (1) ou non (0) de réception

OERR : erreur (1) ou non de sur-écriture

RX9D : Bit de parité (bit en lecture uniquement)

USART (TXSTA) :

CSRC	TX9	TXEN	SYNC	-	BRGH	TRMT	TX9D

CSRC : en mode synchrone, mode maître (1) ou esclave (0)

TX9 : Bit de sélection du nombre de bits en transmission (8 :0 ou 9 :1)

TXEN : transmission activée (1) ou non (0)

SYNC : sélection synchrone (1) ou asynchrone (0)

BRGH : en asynchrone, haute vitesse (1) ou basse (0)

TRMT : registre de transmission plein (0) ou vide (1)

TX9D : 9éme bit de transmission (parité)

Calcul de la vitesse de transmission, et tableaux associés :

Desired Baud rate = Fosc / (64(X + 1))

9600 = 16000000 / (64(+1))X
X = i25.042°

Calculated Baud Rate = 16000000 / (64(25 + 1))
= 9615

$$\text{Error} = \frac{\text{Calculated Baud Rate} = \text{Desired Baud Rate}}{\text{Desired Baud Rate}}$$

= (9615 - 9600)/ 9600
= 0.16%

TABLE 12-1: BAUD RATE FORMULA

SYNC	BRGH = 0 (Low Speed)	BRGH = 1 (High Speed)
0	(Asynchronous) Baud Rate = Fosc/(64(X+1))	Baud Rate= Fosc/(16(X+1))
1	(Synchronous) Baud Rate = Fosc/(4(X+1))	NA

Legend: X = value in SPBRG (0 to 255)

TABLE 12-2: REGISTERS ASSOCIATED WITH BAUD RATE GENERATOR

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR	Value on all other RESETS
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
18h	RCSTA	SPEN	RX9	SREN	CREN	ADEN	FERR	OERR	RX9D	0000 -00x	0000 -00x
99h	SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

Legend: x = unknown, - = unimplemented read as '0'.
Shaded cells are not used by the BRG.

TABLE 12-3: BAUD RATES FOR SYNCHRONOUS MODE

BAUD RATE (K)	Fosc = 20 MHz		SPBRG value (decimal)	16 MHz		SPBRG value (decimal)	10 MHz		SPBRG value (decimal)
	KBAUD	ERROR		KBAUD	ERROR		KBAUD	ERROR	
0.3	NA	—	—	NA	—	—	NA	—	—
1.2	NA	—	—	NA	—	—	NA	—	—
2.4	NA	—	—	NA	—	—	NA	—	—
9.6	NA	—	—	NA	—	—	9.766	+1.73%	255
19.2	19.53	+1.73%	255	19.23	+0.16%	207	19.23	+0.16%	129
76.8	76.92	+0.16%	64	76.92	+0.16%	51	75.76	-1.36%	32
96	96.15	+0.16%	51	95.24	-0.79%	41	96.15	+0.16%	25
300	294.1	-1.96	16	307.69	+2.56%	12	312.5	+4.17%	7
500	500	0	9	500	0	7	500	0	4
HIGH	5000	—	0	4000	—	0	2500	—	0
LOW	19.53	—	255	15.625	—	255	9.766	—	255

BAUD RATE (K)	Fosc = 7.15909 MHz		SPBRG value (decimal)	5.0688 MHz		SPBRG value (decimal)	4 MHz		SPBRG value (decimal)
	KBAUD	ERROR		KBAUD	ERROR		KBAUD	ERROR	
0.3	NA	—	—	NA	—	—	NA	—	—
1.2	NA	—	—	NA	—	—	NA	—	—
2.4	NA	—	—	NA	—	—	NA	—	—
9.6	9.622	+0.23%	185	9.6	0	131	9.615	+0.16%	103
19.2	19.24	+0.23%	92	19.2	0	65	19.231	+0.16%	51
76.8	77.82	+1.32	22	79.2	+3.13%	15	75.923	+0.16%	12
96	94.20	-1.88	18	97.48	+1.54%	12	1000	+4.17%	9
300	298.3	-0.57	5	316.8	5.60%	3	NA	—	—
500	NA	—	—	NA	—	—	NA	—	—
HIGH	1789.8	—	0	1267	—	0	100	—	0
LOW	6.991	—	255	4.950	—	255	3.906	—	255

BAUD RATE (K)	Fosc = 3.579545 MHz		SPBRG value (decimal)	1 MHz		SPBRG value (decimal)	32.768 MHz		SPBRG value (decimal)
	KBAUD	ERROR		KBAUD	ERROR		KBAUD	ERROR	
0.3	NA	—	—	NA	—	—	0.303	+1.14%	26
1.2	NA	—	—	1.202	+0.16%	207	1.170	-2.48%	6
2.4	NA	—	—	2.404	+0.16%	103	NA	—	—
9.6	9.622	+0.23%	92	9.615	+0.16%	25	NA	—	—
19.2	19.04	-0.83%	46	19.24	+0.16%	12	NA	—	—
76.8	74.57	-2.90%	11	83.34	+8.51%	2	NA	—	—
96	99.43	+3.57%	8	NA	—	—	NA	—	—
300	298.3	0.57%	2	NA	—	—	NA	—	—
500	NA	—	—	NA	—	—		—	—
HIGH	894.9	—	0	250	—	0	8.192	—	0
LOW	3.496	—	255	0.9766	—	255	0.032	—	255

TABLE 12-4: BAUD RATES FOR ASYNCHRONOUS MODE (BRGH = 0)

BAUD RATE (K)	Fosc = 20 MHz		SPBRG value (decimal)	16 MHz		SPBRG value (decimal)	10 MHz		SPBRG value (decimal)
	KBAUD	ERROR		KBAUD	ERROR		KBAUD	ERROR	
0.3	NA	—	—	NA	—	—	NA	—	—
1.2	1.221	+1.73%	255	1.202	+0.16%	207	1.202	+0.16%	129
2.4	2.404	+0.16%	129	2.404	+0.16%	103	2.404	+0.16%	64
9.6	9.469	-1.36%	32	9.615	+0.16%	25	9.766	+1.73%	15
19.2	19.53	+1.73%	15	19.23	+0.16%	12	19.53	+1.73V	7
76.8	78.13	+1.73%	3	83.33	+8.51%	2	78.13	+1.73%	1
96	104.2	+8.51%	2	NA	—	—	NA	—	—
300	312.5	+4.17%	0	NA	—	—	NA	—	—
500	NA	—	—	NA	—	—	NA	—	—
HIGH	312.5	—	0	250	—	0	156.3	—	0
LOW	1.221	—	255	0.977	—	255	0.6104	—	255

BAUD RATE (K)	Fosc = 7.15909 MHz		SPBRG value (decimal)	5.0688 MHz		SPBRG value (decimal)	4 MHz		SPBRG value (decimal)
	KBAUD	ERROR		KBAUD	ERROR		KBAUD	ERROR	
0.3	NA	—	—	0.31	+3.13%	255	0.3005	-0.17%	207
1.2	1.203	+0.23%	92	1.2	0	65	1.202	+1.67%	51
2.4	2.380	-0.83%	46	2.4	0	32	2.404	+1.67%	25
9.6	9.322	-2.90%	11	9.9	+3.13%	7	NA	—	—
19.2	18.64	-2.90%	5	19.8	+3.13%	3	NA	—	—
76.8	NA	—	—	79.2	+3.13%	0	NA	—	—
96	NA	—	—	NA	—	—	NA	—	—
300	NA	—	—	NA	—	—	NA	—	—
500	NA	—	—	NA	—	—	NA	—	—
HIGH	111.9	—	0	79.2	—	0	62.500	—	0
LOW	0.437	—	255	0.3094	—	255	3.906	—	255

BAUD RATE (K)	Fosc = 3.579545 MHz		SPBRG value (decimal)	1 MHz		SPBRG value (decimal)	32.768 MHz		SPBRG value (decimal)
	KBAUD	ERROR		KBAUD	ERROR		KBAUD	ERROR	
0.3	0.301	+0.23%	185	0.300	+0.16%	51	0.256	-14.67%	1
1.2	1.190	-0.83%	46	1.202	+0.16%	12	NA	—	—
2.4	2.432	+1.32%	22	2.232	-6.99%	6	NA	—	—
9.6	9.322	-2.90%	5	NA	—	—	NA	—	—
19.2	18.64	-2.90%	2	NA	—	—	NA	—	—
76.8	NA	—	—	NA	—	—	NA	—	—
96	NA	—	—	NA	—	—	NA	—	—
300	NA	—	—	NA	—	—	NA	—	—
500	NA	—	—	NA	—	—	NA	—	—
HIGH	55.93	—	0	15.63	—	0	0.512	—	0
LOW	0.2185	—	255	0.0610	—	255	0.0020	—	255

TABLE 12-5: BAUD RATES FOR ASYNCHRONOUS MODE (BRGH = 1)

BAUD RATE (K)	Fosc = 20 MHz		SPBRG value (decimal)	16 MHz		SPBRG value (decimal)	10 MHz		SPBRG value (decimal)
	KBAUD	ERROR		KBAUD	ERROR		KBAUD	ERROR	
9600	9.615	+0.16%	129	9.615	+0.16%	103	9.615	+0.16%	64
19200	19.230	+0.16%	64	19.230	+0.16%	51	18.939	-1.36%	32
38400	37.878	-1.36%	32	38.461	+0.16%	25	39.062	+1.7%	15
57600	56.818	-1.36%	21	58.823	+2.12%	16	56.818	-1.36%	10
115200	113.636	-1.36%	10	111.111	-3.55%	8	125	+8.51%	4
250000	250	0	4	250	0	3	NA	—	—
625000	625	0	1	NA	—	—	625	0	0
1250000	1250	0	0	NA	—	—	NA	—	—

BAUD RATE (K)	Fosc = 7.16 MHz		SPBRG value (decimal)	5.068 MHz		SPBRG value (decimal)	4 MHz		SPBRG value (decimal)
	KBAUD	ERROR		KBAUD	ERROR		KBAUD	ERROR	
9600	9.520	-0.83%	46	9598.485	0.016%	32	9615.385	0.160%	25
19200	19.454	+1.32%	22	18632.35	-2.956%	16	19230.77	0.160%	12
38400	37.286	-2.90%	11	39593.75	3.109%	7	35714.29	-6.994%	6
57600	55.930	-2.90%	7	52791.67	-8.348%	5	62500	8.507%	3
115200	111.860	-2.90%	3	105583.3	-8.348%	2	125000	8.507%	1
250000	NA	—	—	316750	26.700%	0	250000	0.000%	0
625000	NA	—	—	NA	—	—	NA	—	—
1250000	NA	—	—	NA	—	—	NA	—	—

BAUD RATE (K)	Fosc = 3.579 MHz		SPBRG value (decimal)	1 MHz		SPBRG value (decimal)	32.768 MHz		SPBRG value (decimal)
	KBAUD	ERROR		KBAUD	ERROR		KBAUD	ERROR	
9600	9725.543	1.308%	22	8.928	-6.994%	6	NA	NA	NA
19200	18640.63	-2.913%	11	20833.3	8.507%	2	NA	NA	NA
38400	37281.25	-2.913%	5	31250	-18.620%	1	NA	NA	NA
57600	55921.88	-2.913%	3	62500	+8.507	0	NA	NA	NA
115200	111243.8	-2.913%	1	NA	—	—	NA	NA	NA
250000	223687.5	-10.525%	0	NA	—	—	NA	NA	NA
625000	NA	—	—	NA	—	—	NA	NA	NA
1250000	NA	—	—	NA	—	—	NA	NA	NA

TABLE 12-7: REGISTERS ASSOCIATED WITH ASYNCHRONOUS RECEPTION

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR	Value on all other RESETS
0Ch	PIR1	EEIF	CMIF	RCIF	TXIF	—	CCP1IF	TMR2IF	TMR1IF	0000 -000	0000 -000
18h	RCSTA	SPEN	RX9	SREN	CREN	ADEN	FERR	OERR	RX9D	0000 -00x	0000 -00x
1Ah	RCREG	USART Receive Register								0000 0000	0000 0000
8Ch	PIE1	EEIE	CMIE	RCIE	TXIE	—	CCP1IE	TMR2IE	TMR1IE	0000 -000	0000 -000
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

Legend: x = unknown, - = unimplemented locations read as '0'. Shaded cells are not used for Asynchronous Reception.

TABLE 12-8: REGISTERS ASSOCIATED WITH ASYNCHRONOUS RECEPTION

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR	Value on all other RESETS
0Ch	PIR1	EEIF	CMIF	RCIF	TXIF	—	CCP1IF	TMR2IF	TMR1IF	0000 -000	0000 -000
18h	RCSTA	SPEN	RX9	SREN	CREN	ADEN	FERR	OERR	RX9D	0000 -00x	0000 -00x
1Ah	RCREG	RX7	RX6	RX5	RX4	RX3	RX2	RX1	RX0	0000 0000	0000 0000
8Ch	PIE1	EEIE	CMIE	RCIE	TXIE	—	CCP1IE	TMR2IE	TMR1IE	0000 -000	0000 -000
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

Legend: x = unknown, - = unimplemented locations read as '0'. Shaded cells are not used for Asynchronous Reception.

TABLE 12-9: REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER TRANSMISSION

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR	Value on all other RESETS
0Ch	PIR1	EEIF	CMIF	RCIF	TXIF	—	CCP1IF	TMR2IF	TMR1IF	0000 -000	0000 -000
18h	RCSTA	SPEN	RX9	SREN	CREN	ADEN	FERR	OERR	RX9D	0000 -00x	0000 -00x
19h	TXREG	USART Transmit Register								0000 0000	0000 0000
8Ch	PIE1	EEIE	CMIE	RCIE	TXIE	—	CCP1IE	TMR2IE	TMR1IE	0000 -000	0000 -000
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

Legend: x = unknown, - = unimplemented, read as '0'. Shaded cells are not used for Synchronous Master Transmission.

TABLE 12-10: REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER RECEPTION

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR	Value on all other RESETS
0Ch	PIR1	EEIF	CMIF	RCIF	TXIF	—	CCP1IF	TMR2IF	TMR1IF	0000 -000	0000 -000
18h	RCSTA	SPEN	RX9	SREN	CREN	ADEN	FERR	OERR	RX9D	0000 -00x	0000 -00x
1Ah	RCREG	USART Receive Register								0000 0000	0000 0000
8Ch	PIE1	EEPIE	CMIE	RCIE	TXIE	—	CCP1IE	TMR2IE	TMR1IE	-000 0000	-000 -000
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

Legend: x = unknown, - = unimplemented read as '0'. Shaded cells are not used for Synchronous Master Reception.

TABLE 12-11: REGISTERS ASSOCIATED WITH SYNCHRONOUS SLAVE TRANSMISSION

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR	Value on all other RESETS
0Ch	PIR1	EEIF	CMIF	RCIF	TXIF	—	CCP1IF	TMR2IF	TMR1IF	0000 -000	0000 -000
18h	RCSTA	SPEN	RX9	SREN	CREN	ADEN	FERR	OERR	RX9D	0000 -00x	0000 -00x
19h	TXREG	USART Transmit Register								0000 0000	0000 0000
8Ch	PIE1	EEIE	CMIE	RCIE	TXIE	—	CCP1IE	TMR2IE	TMR1IE	0000 -000	0000 -000
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

Legend: x = unknown, - = unimplemented read as '0'. Shaded cells are not used for Synchronous Slave Transmission.

TABLE 12-12: REGISTERS ASSOCIATED WITH SYNCHRONOUS SLAVE RECEPTION

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR	Value on all other RESETS
0Ch	PIR1	EEIF	CMIF	RCIF	TXIF	—	CCP1IF	TMR2IF	TMR1IF	0000 -000	0000 -000
18h	RCSTA	SPEN	RX9	SREN	CREN	ADEN	FERR	OERR	RX9D	0000 -00x	0000 -00x
1Ah	RCREG	USART Receive Register								0000 0000	0000 0000
8Ch	PIE1	EEIE	CMIE	RCIE	TXIE	—	CCP1IE	TMR2IE	TMR1IE	0000 -000	0000 -000
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

Legend: x = unknown, - = unimplemented read as '0'. Shaded cells are not used for Synchronous Slave Reception.

CHAPITRE 3 : PROGRAMMER LE PIC

3.1 Un programmateur de PIC

3.2 Les logiciels de programmation

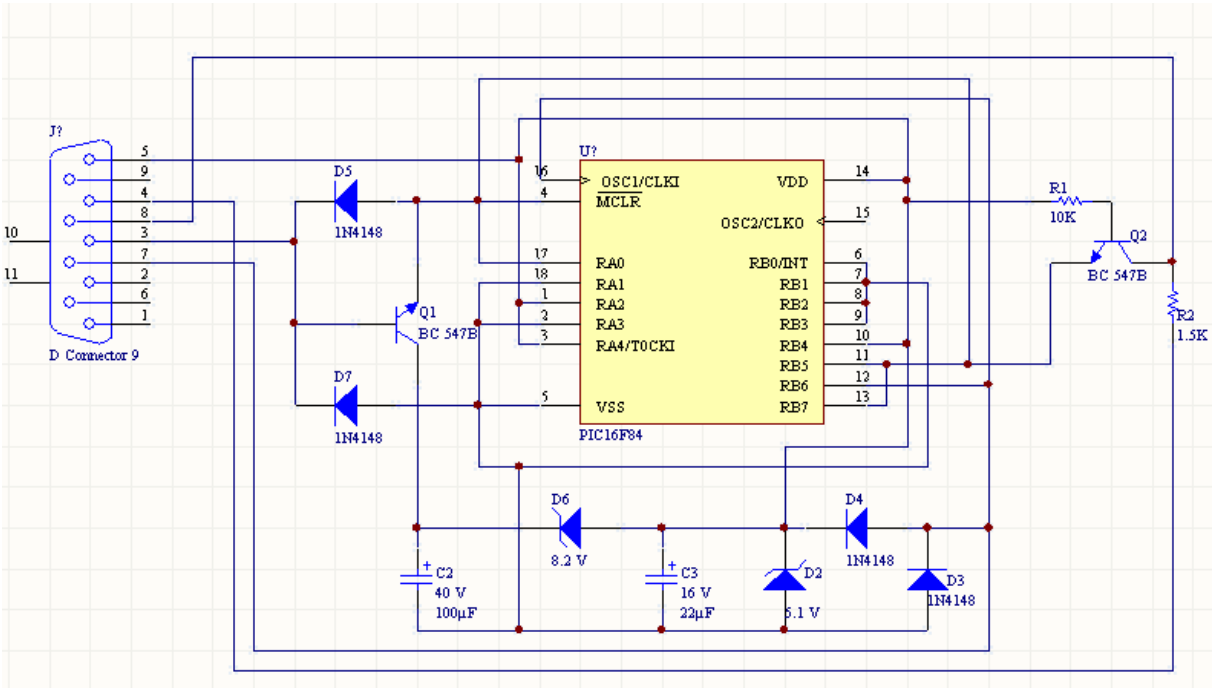
3- Programmer le PIC

3.1- un programmateur de PIC :

Nous allons voir ici, un programmateur de PIC fort simple. Bien que certains disent qu'il commence à vieillir pour la simple raison qu'il se branche sur un port série, je pense qu'il reste le meilleur programmateur. En effet, pas besoin de lui fournir une alimentation externe comme certains programmateurs. De plus, étant relativement simple, et ancien, on peut trouver de nombreuses docs sur le net, notamment, pour ceux que ça intéressent, sur son fonctionnement. Il s'agit d'un programmateur de PIC 16F84, de type JDM.

Nous ne verrons ici, que la façon de le fabriquer, sans nous attarder sur son fonctionnement. Nous verrons donc son schéma structurel, le PCB et le schéma d'implantation.

Le structurel :

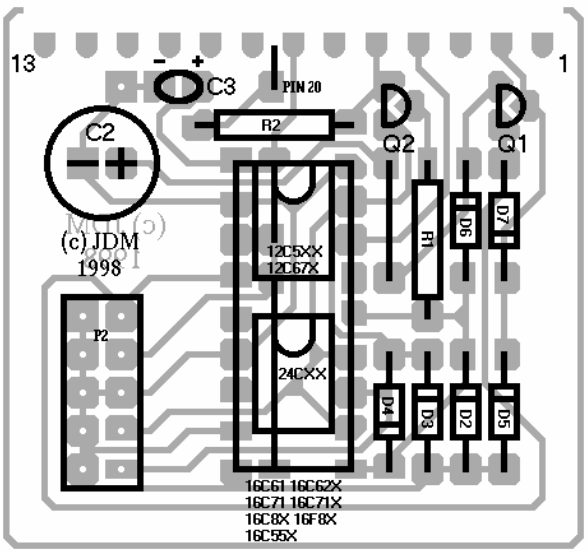


Il vous faudra quelques composants pour le fabriquer :

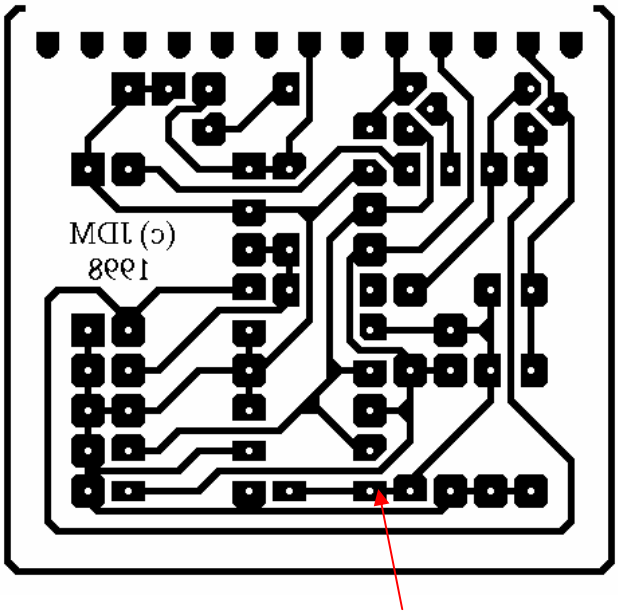
R1	10K	Resistor
R2	1.5K	Resistor
D2	5.1V/0.5W	Zener
D3	1N4148 or 1N4448	Diode
D4	1N4148 or 1N4448	Diode
D5	1N4148 or 1N4448	Diode
D6	8.2V/0.5W	Zener
D7	1N4148 or 1N4448	Diode
C2	100µF/25V	Capacitor electrolytic
C3	22µF/16V or 47µF/6.3V	Capacitor tantal
Q1	BC547B	Transistor NPN
Q2	BC547B	Transistor NPN
P1	DS25 (female)	25 pol DSUB connector
P2		Connector for In Circuit Programming

Le PCB vient d'un site internet (http://www.jdm.homepage.dk/pcb2.htm), mais pour ceux qui ne disposent pas d'accès, voici les PCB, tels qu'ils sont mis sur le net. A noter que le schématique du site est avec un connecteur parallèle, alors, que le schématique de ce livre est pour un connecteur série.

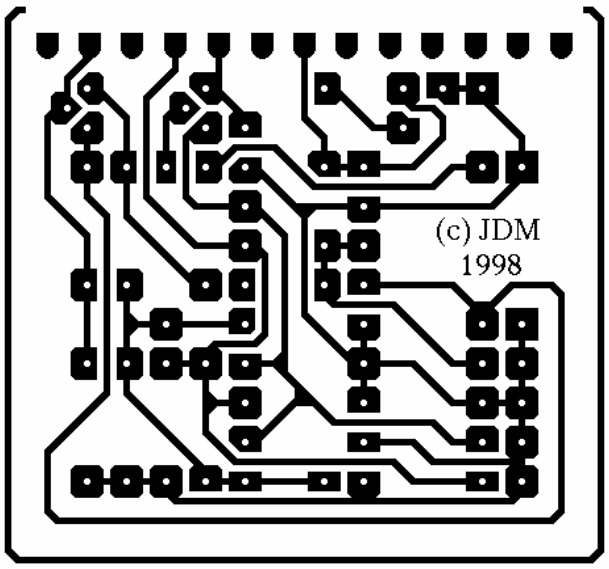
Implantations des composants :



Vue de dessus (par transparence) :



Vue côté cuivre :



Pour information, le cadre fait 3,8 cm de large et 3,6 cm de haut. Attention donc, quand vous imprimez le typon.

***Attention*: Ce schéma, est valable pour le PIC16F84. Pour pouvoir programmer le PIC 16F628, il faut déconnecter la patte RB4, et la laisser dans le vide. Concrètement, sur le typon, il faut couper la**

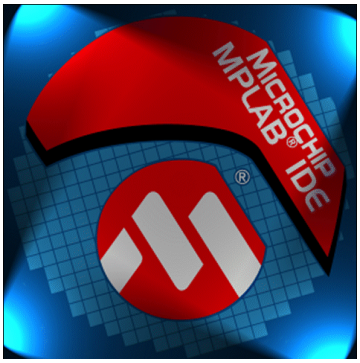
piste à droite de la pastille signalée par la flèche rouge (voir typon ci-dessus). Théoriquement, vous pouvez alors, programmer à la fois les PIC16F628, et le 16F84. Si vous êtes perfectionnistes, ou si vous avez des problèmes, vous pouvez rajouter un interrupteur, afin de faire ou non, la liaison.

3.2- Les logiciels de programmation :

Afin de programmer le PIC, nous aurons besoin de deux logiciels (MPlab et IC-Prog), ainsi que d'un compilateur (CC5Xfree).

Nous verrons donc le logiciel de programmation avec son compilateur (MPlab et CC5Xfree), et le logiciel pour télécharger le programme dans le PIC (IC-Prog)

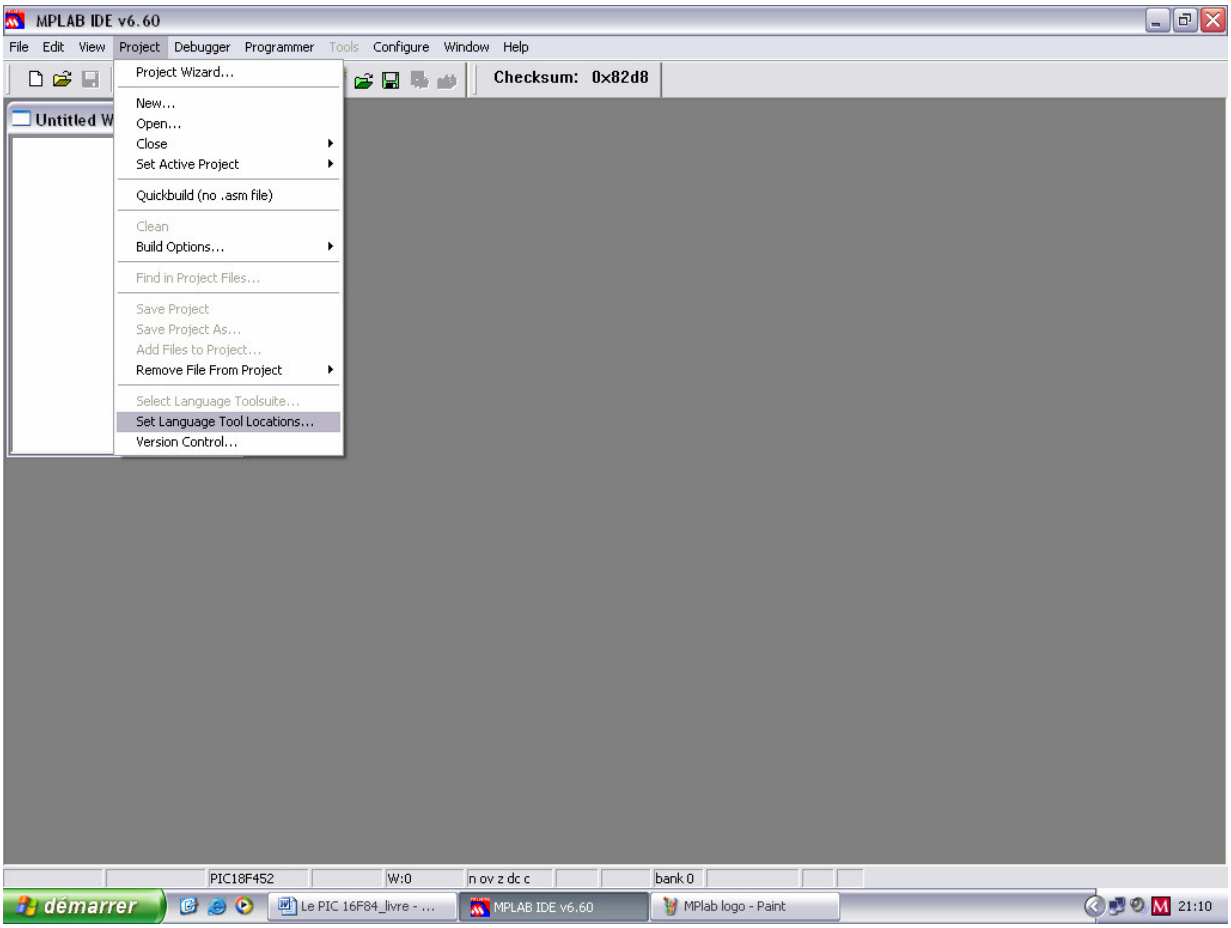
MPlab est un logiciel gratuit de programmation, développé et proposé par Microchip, le fabricant du microcontrôleur PIC.



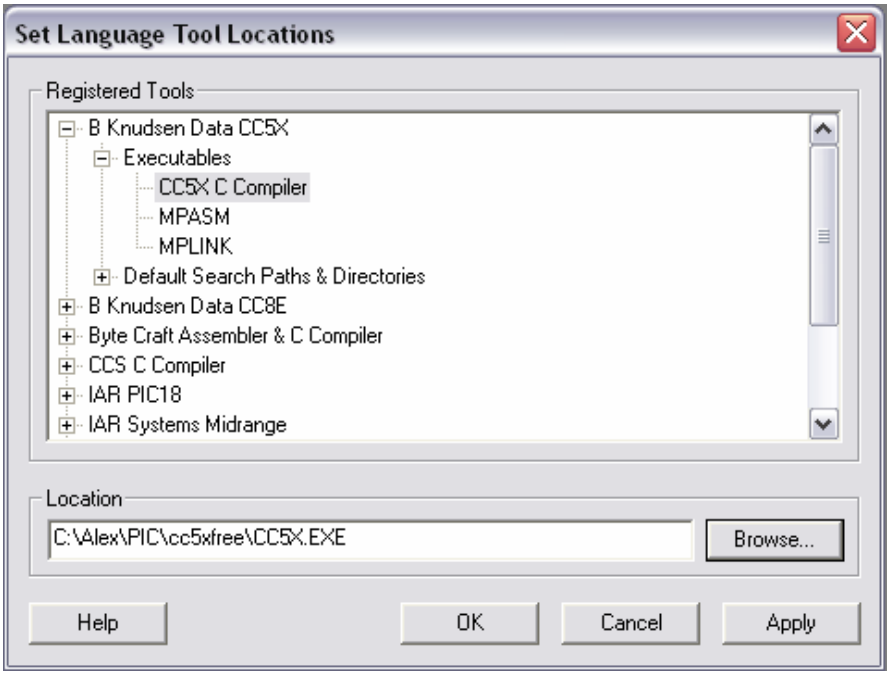
Les étapes que nous verrons dans l'immédiat consistent à déclarer le compilateur dans MPlab. De fait, cette opération n'est à effectuer qu'une seule fois. Ces étapes seront illustrées en images.

La première chose à faire est d'ouvrir le logiciel MPlab. Bien, maintenant, nous allons pouvoir commencer.

Etape 1 : ouvrez le menu ''Project'' et cliquez sur ''Set Language Tool Locations

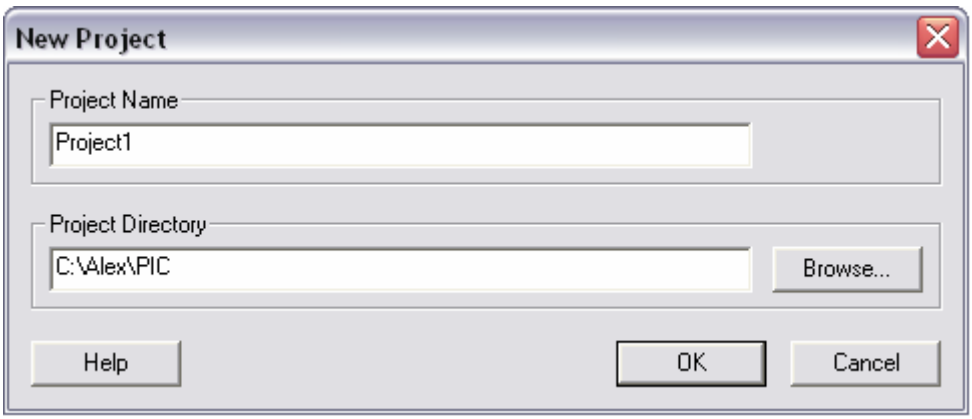


Etape 2 : il faut préciser à MPlab où se trouve le compilateur CC5X, qui nous permettra d'écrire en C.

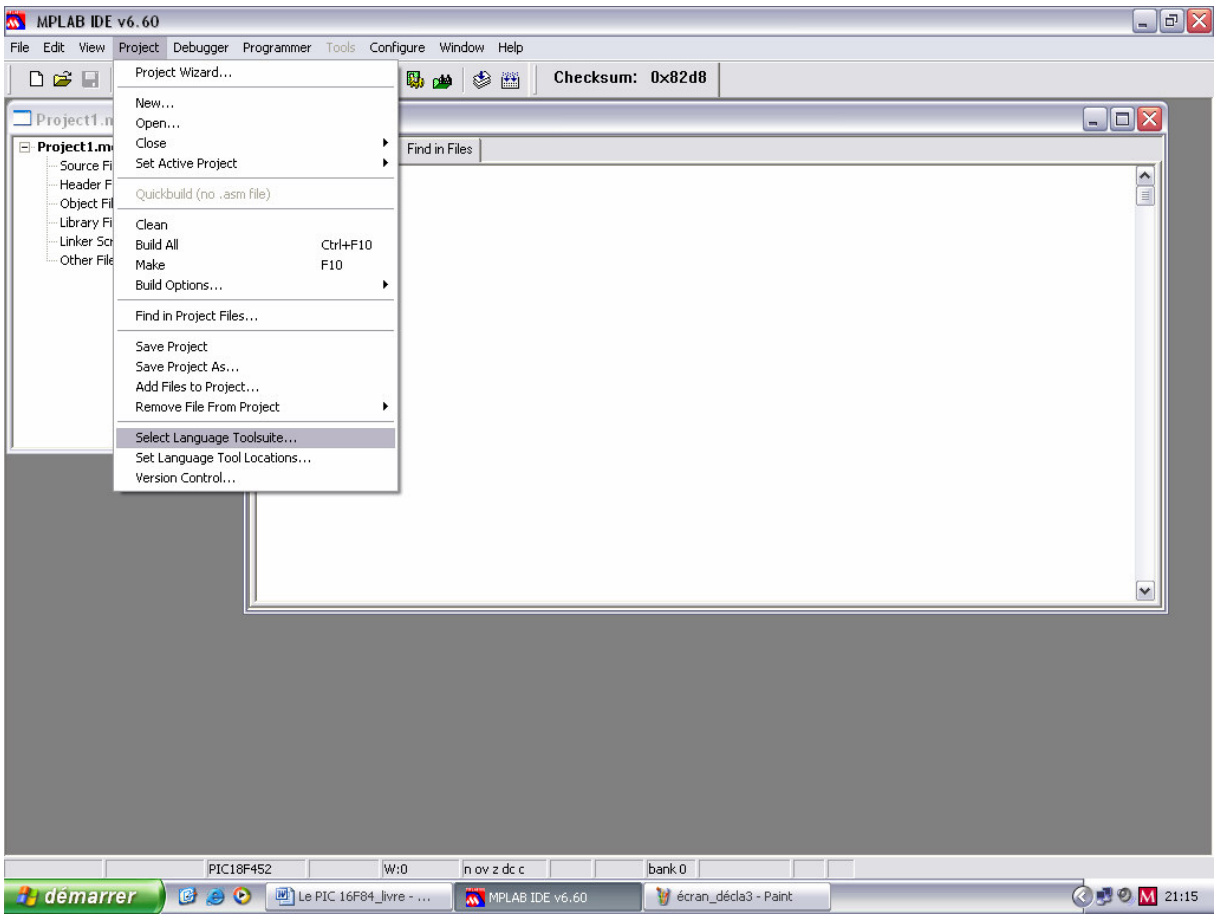


Vous devez cliquer, comme vous pouvez le voir sur ''B Knudsen Data CC5X'', puis sur ''Executables'', et enfin sélectionner ''CC5X C Compiler''. Ensuite vous devez cliquer sur ''Browse'' pour lui dire où trouver le compilateur, en lui indiquant le chemin.

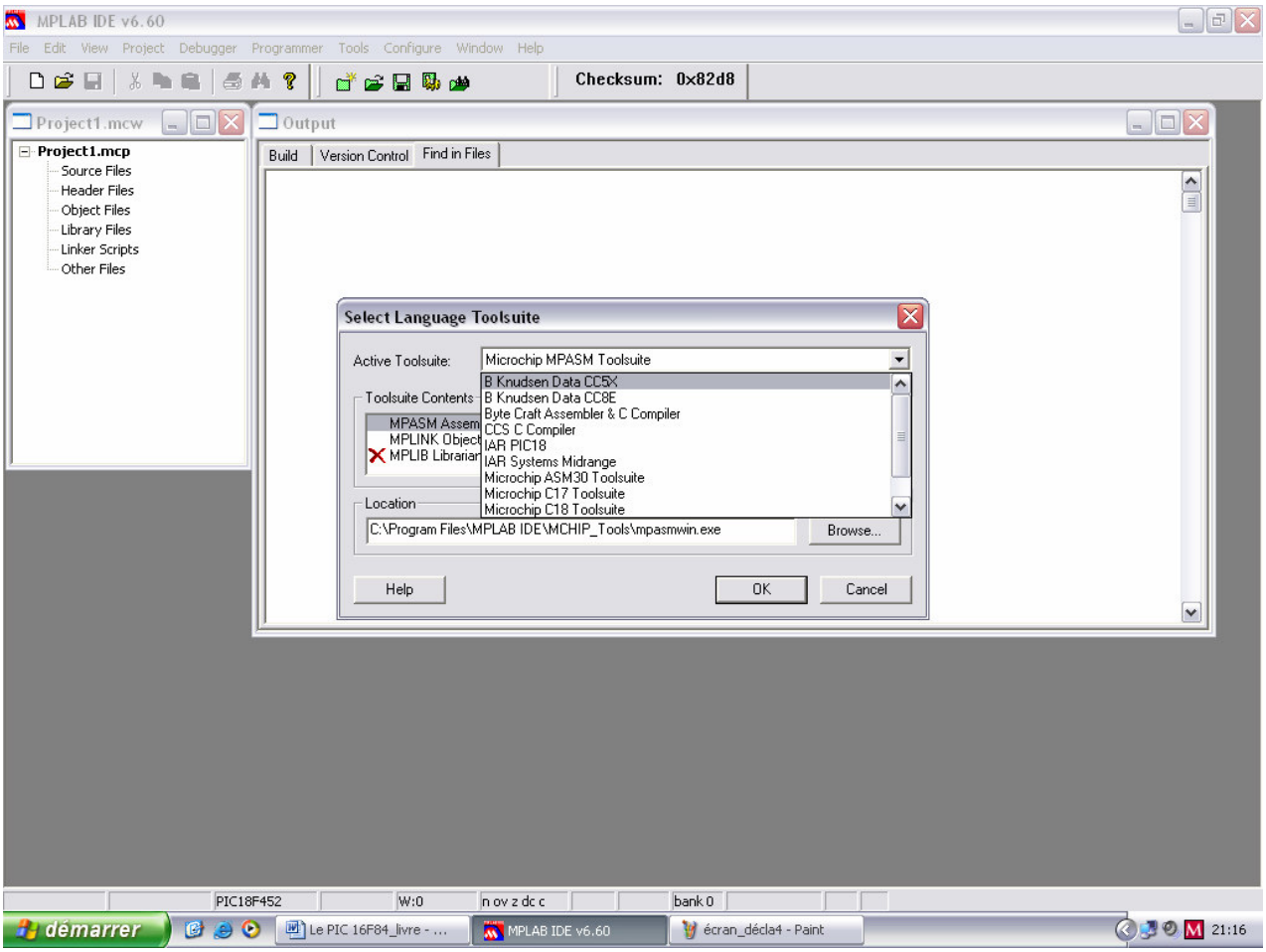
Etape 3 : la troisième étape consiste à déclarer un nouveau projet, en cliquant sur ''Project'', puis sur ''new''

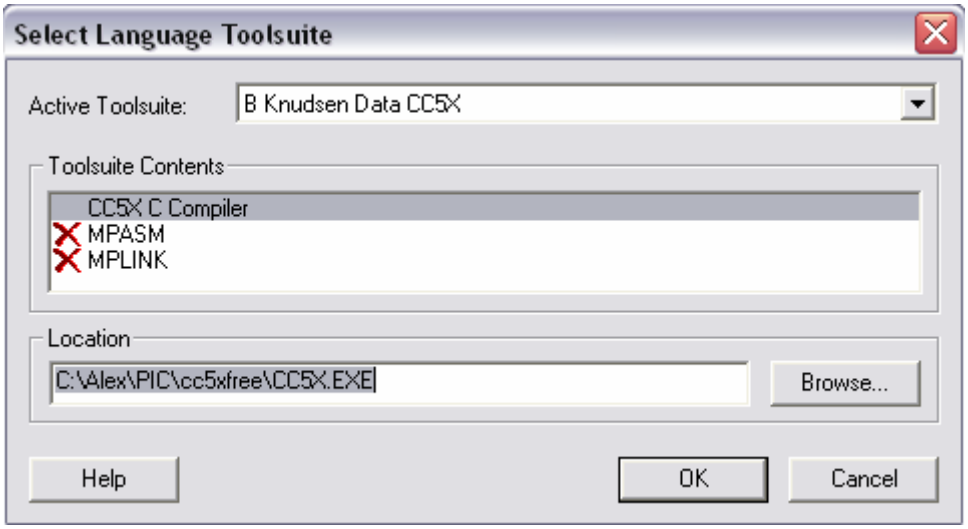


Etape 4 : après avoir dit à MPlab où était le compilateur CC5X, il faut lui préciser, que c'est le compilateur que l'on veut utiliser. Pour cela cliquez sur le menu ''Project'', puis sur ''Select Language Toolsuite''.

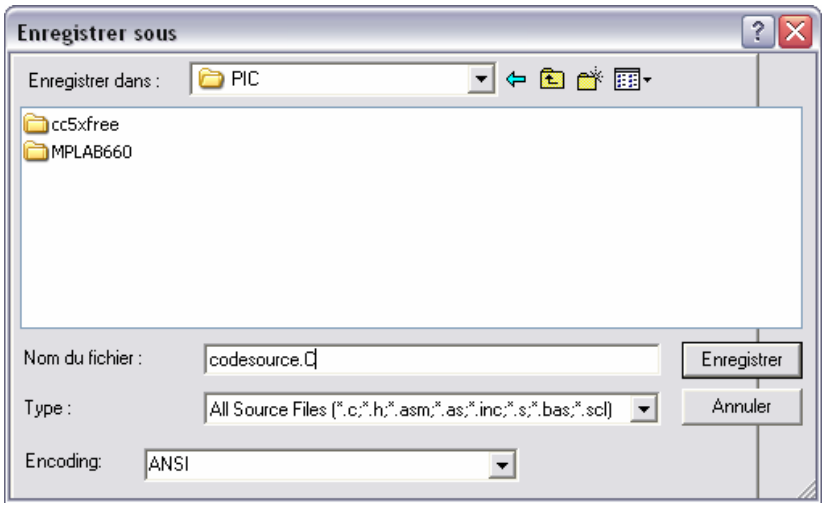


Vous obtenez alors la fenêtre suivante, dans laquelle vous devez sélectionner dans le menu déroulant ''active toolsuite'' la ligne ''B Knudsen Data CC5X''. Cliquez alors sur ''CC5X C Compiler'' et vérifiez que la ligne ''Location'' indique le bon chemin à MPlab.

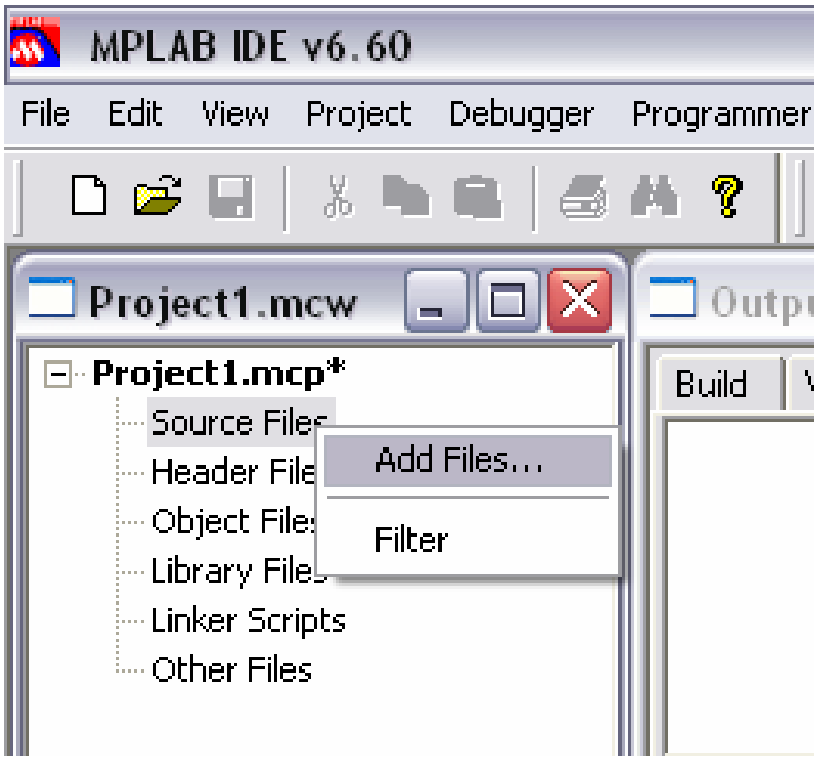

MPLAB IDE v6.60
File Edit View Project Debugger Programmer Tools Configure Window Help
Checksum: 0x82d8
Project1.mcw
Project1.mcp
Source Files
Header Files
Object Files
Library Files
Linker Scripts
Other Files
Output
Build
Version Control
Find in Files
Select Language Toolsuite
Active Toolsuite:
Microchip MPASM Toolsuite
B Knudsen Data CC5X
B Knudsen Data CC8E
Byte Craft Assembler & C Compiler
CCS C Compiler
IAR PIC18
IAR Systems Midrange
Microchip ASM30 Toolsuite
Microchip C17 Toolsuite
Microchip C18 Toolsuite
Toolsuite Contents
Location
C:\Program Files\MPLAB IDE\MCHIP_Tools\mpasmwin.exe
Browse...
Help
OK
Cancel
PIC18F452
W:0
bank 0
démarrer
Le PIC 16F84_livre - ...
MPLAB IDE v6.60
écran_décla4 - Paint
21:16


Select Language Toolsuite
Active Toolsuite:
B Knudsen Data CC5X
Toolsuite Contents
CC5X C Compiler
MPASM
MPLINK
Location
C:\Alex\PIC\cc5xfree\CC5X.EXE
Browse...
Help
OK
Cancel

Voilà, nous avons configuré le compilateur ; maintenant, nous allons voir comment créer la fenêtre pour le code source. Cela est très simple : cliquez sur ''File'', pui sur ''New''. Une fenêtre apparaît alors. Cliquez alors sur ''File'', puis sur ''Save As''. Rentrez alors le nom que vous voulez, suivit de ''.C'', afin de préciser le type de document que vous allez écrire.

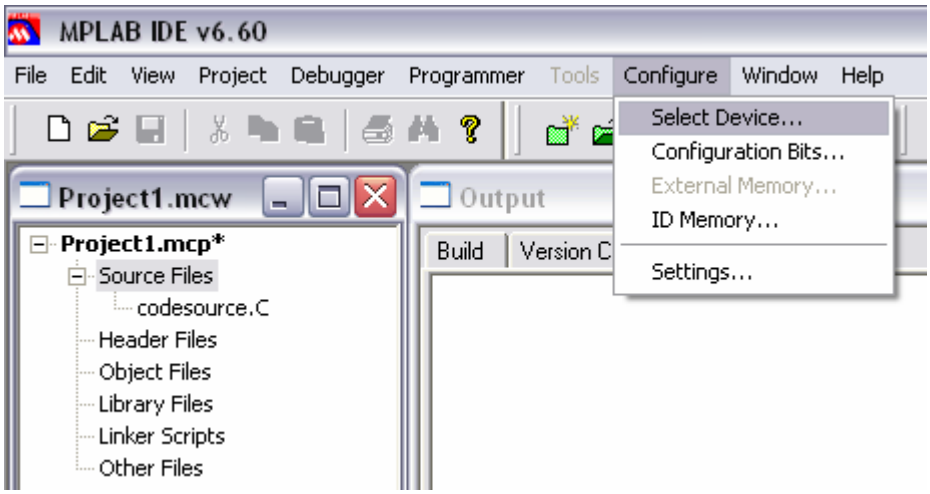


Il ne vous reste alors plus qu'à lier le code source à votre projet :

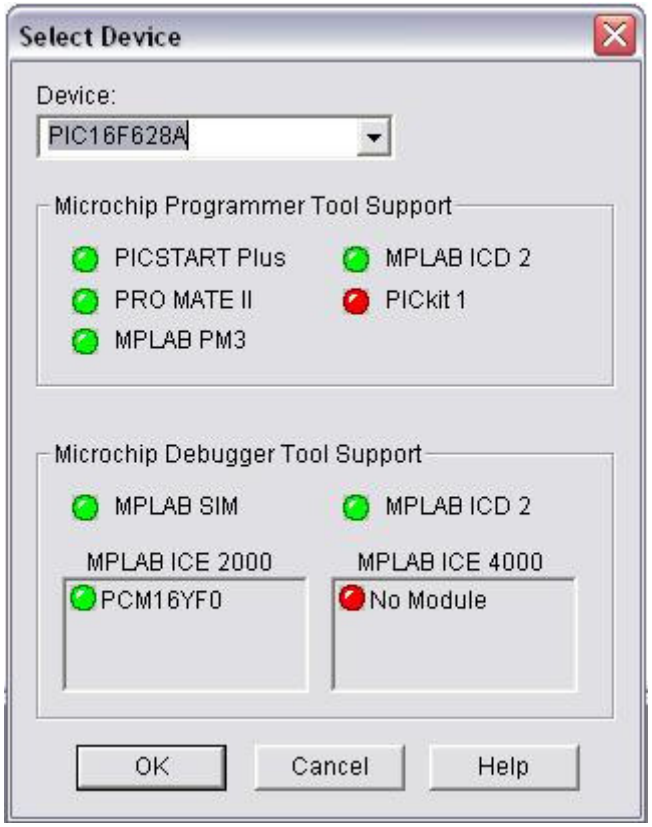


Les derniers paramètres :il s'agit de préciser à MPlab le composant auquel est destiné le programme.

Pour ce faire, cliquez sur ''Configure'', puis sur ''select device''

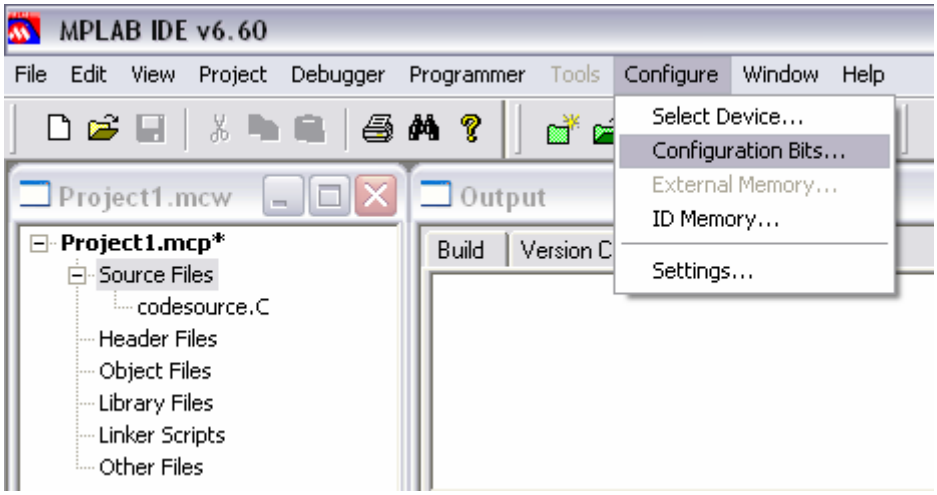


Apparaît alors l'écran suivant :

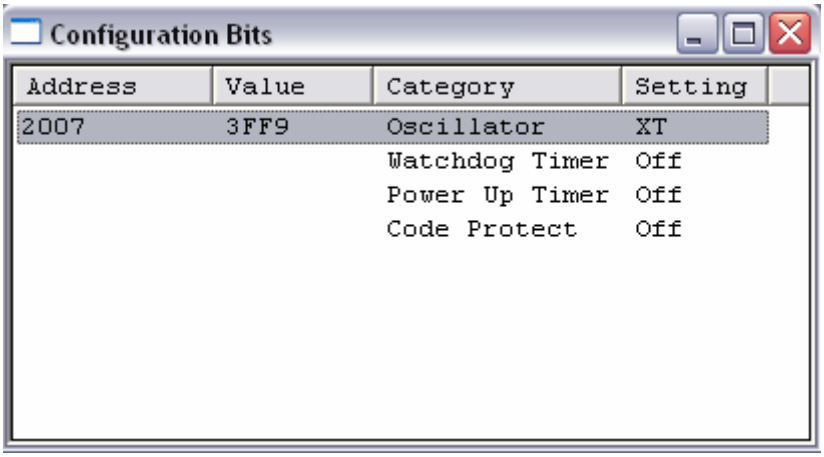


Maintenant, vous devez définir les paramètres de configuration : le Watchdog, le type d'oscillateur , le power up timer (mise sous tension retardée) et le code P (code protect).

Cliquez sur ''Configure'', ensuite sur ''Configuration Bits'' :

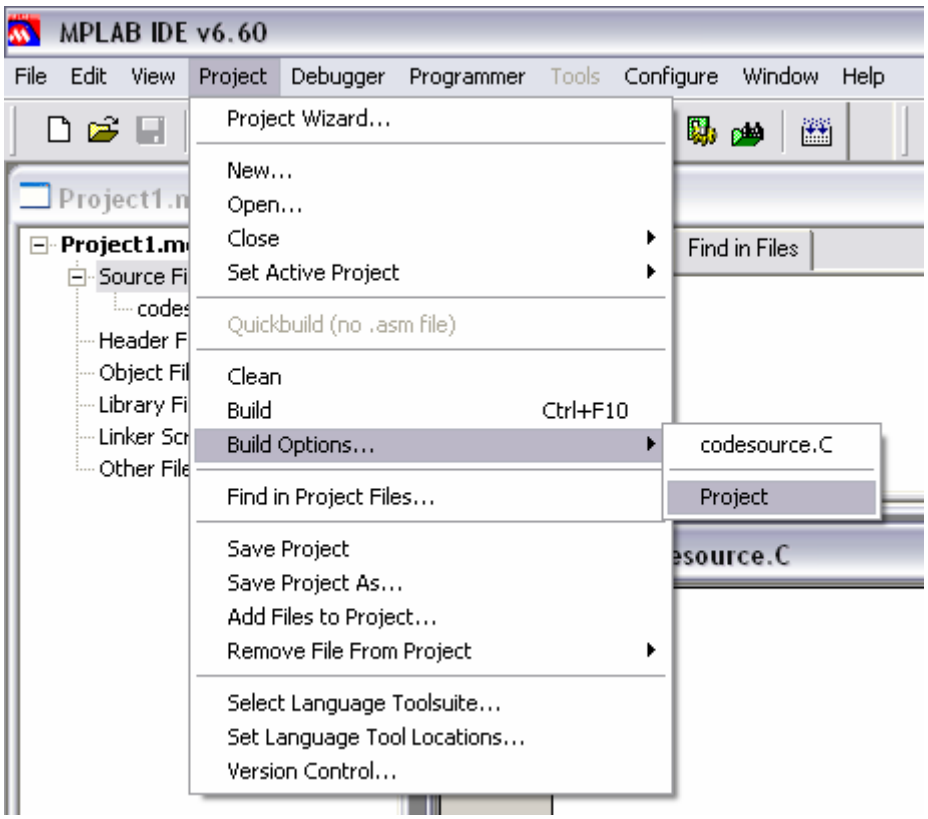


Veuillez alors à rentrer les mêmes valeurs que dans cette capture d'écran. A noter, que cette fenêtre correspond avec un quartz. Tout le reste doit être à off.

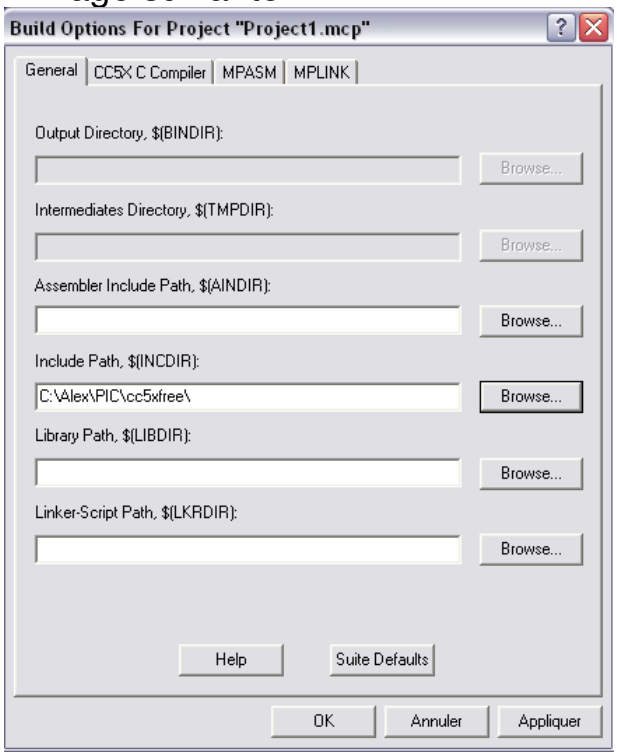


Comme nous l'avons vu précédemment dans la partie sur le langage C, nous aurons besoin de librairies. Celles-ci sont propres au compilateur. Il est donc nécessaire de définir l'endroit où elles se trouvent.

Pour cela, cliquez sur le menu ''Project'', puis sur ''Build Options'' et enfin sur ''Project''.



Et vous obtenez l'image suivante :



Il faut alors que vous indiquiez dans le champ ''Include Path'' le chemin du dossier du compilateur CC5X.

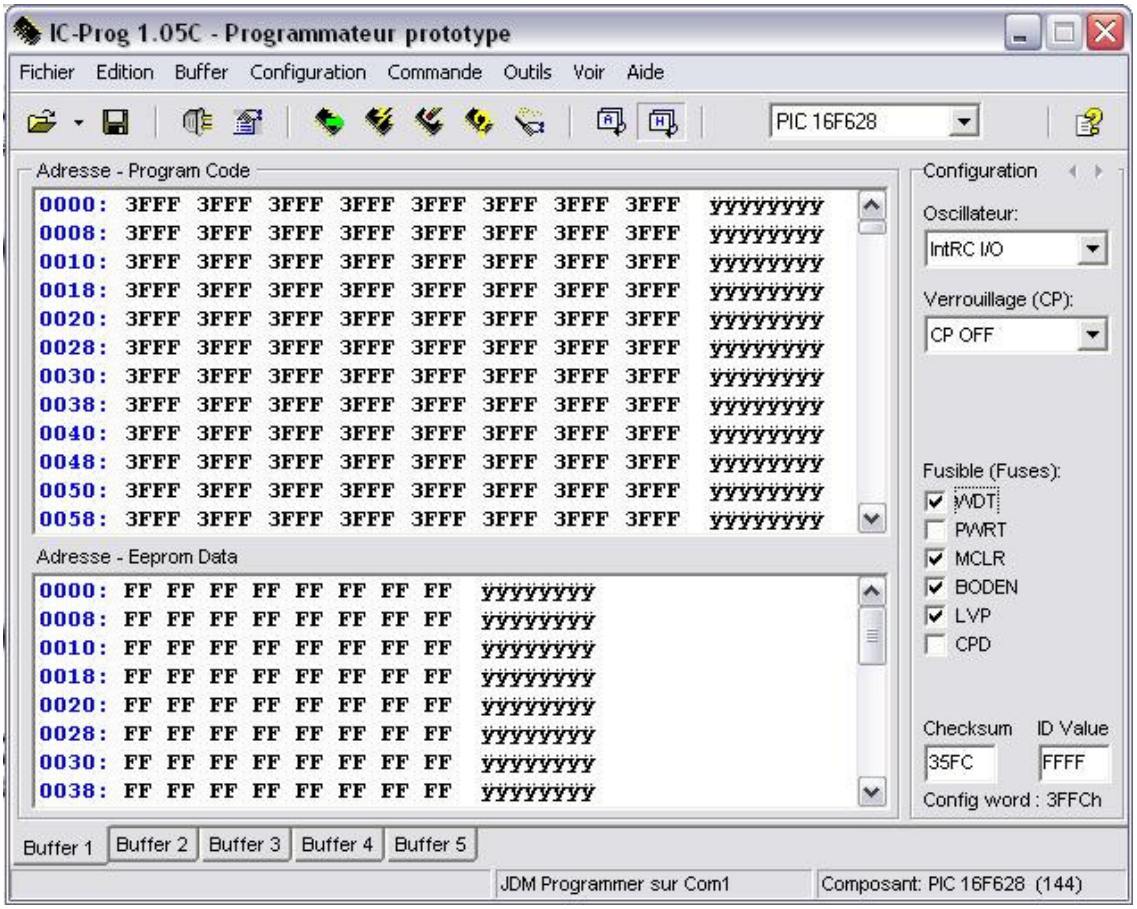
Voilà, votre logiciel est enfin configuré pour programmer.

Une indication utile toutefois. Quand votre programme sera terminé, pour le compiler et vérifier que tout va bien (si ce n'est pas le cas, il vous affichera un message d'erreur), il faut utiliser le bouton ''build'', correspondant à l'icône : 

Nous allons maintenant voir le logiciel, qui permet, par l'intermédiaire du programmateur vu précédemment, de télécharger notre programme dans le PIC.

<u>Remarque :</u> C'est le ''.hex'' que l'on doit ouvrir. Si celui-ci n'est pas présent dans votre dossier, ou si MPlab vous indique lors de la compilation qu'il n'a pas pu être généré, alors ouvrez le menu Windows, et cliquez sur ''Exécuter''. Dans le champ qui s'ouvre tapez alors la ligne :
regsvr32 ''C:\Program Files\MPLAB IDE\dlls\MPProgram.dll''
Cette ligne devrait résoudre le problème.

Nous allons commencer par ouvrir le logiciel ICPROG :



Chose importante, sélectionner le PIC 16F628 comme composant. Les fusibles sont à cocher selon l'utilisation de votre PIC. Pour une utilisation des plus classiques, vous pouvez vous contentez de l'horloge interne (intRC I/O)

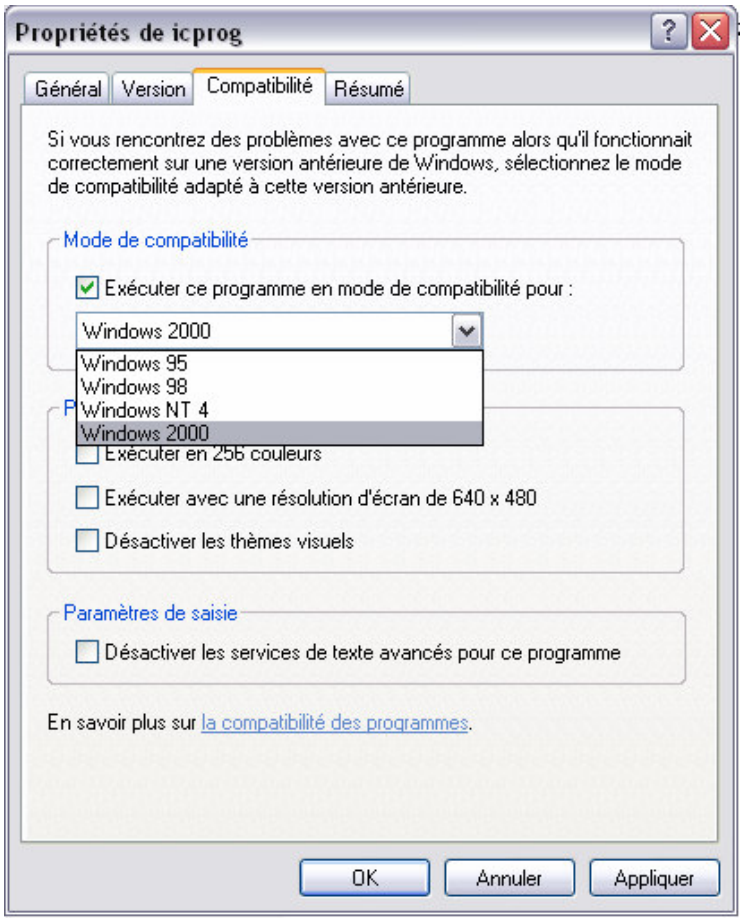
NOM	Fonction
WDT	Active ou non le watchdog
PWRT	Active ou non un reset à la mise sous tension
MCLR	Reset externe activé (RC) ou non
BODEN	Surveille une défaillance éventuelle de l'alim ou non
LVP	Active ou non le mode de programmation compatible 16F84
CPD	Active ou non la protection de la lecture de l'E²PROM

A ce stade, vous êtes prêts à programmer, ne vous reste alors plus qu'à télécharger votre programme. Mais avant de voir comment faire nous allons faire une remarque importante.

***Remarque :* Toutes les étapes pour ICPROG vues jusqu'ici sont bonnes pour toutes les versions de Windows. Cependant, pour windows 2000, NT, et XP, il est nécessaire de rajouter des drivers spéciaux à ICPROG. Voici comment faire :**

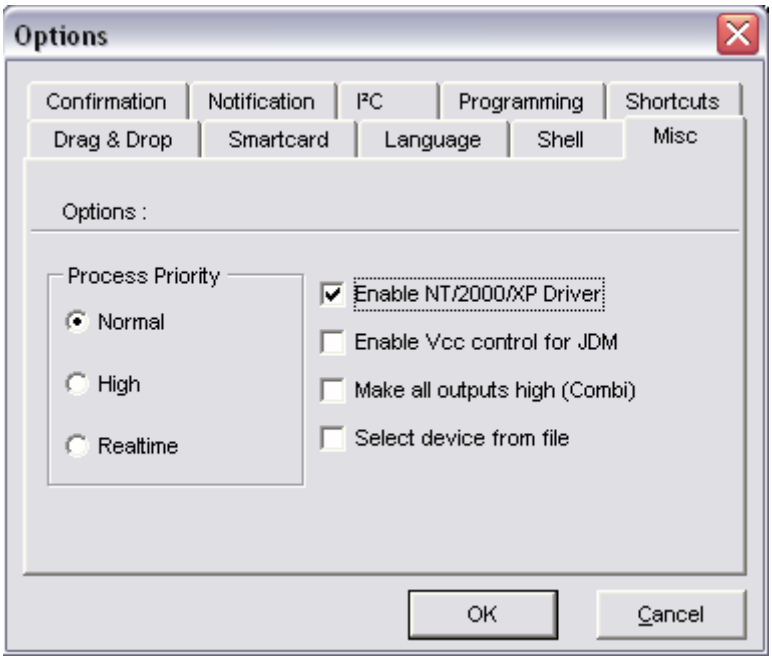
Lors de la première mise sous tension, vous obtiendrez des messages d'erreur de violations. Refermez alors le logiciel et allez dans le répertoire où se trouve l'exécutable. Faites un clique droit sur l'exécutable, et ouvrez les propriétés.

Allez alors dans compatibilité, cochez la première case, et demandez à exécuter le programme en mode de compatibilité pour windows 2000, puis validez.

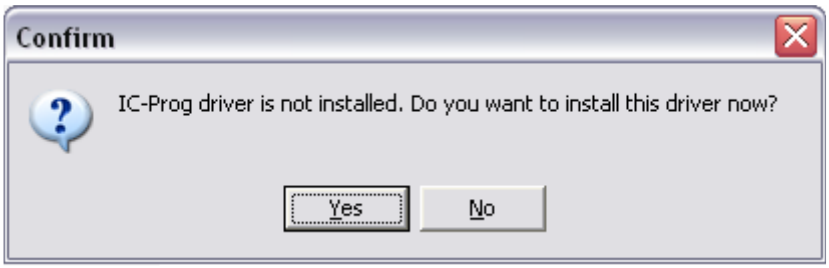


Rouvrez alors ICPROG,. Vous obtiendrez un nouveau message d'erreur. Cliquez sur OK, puis cliquez sur le menu ''Settings'', puis sur ''Options''.

Dans le menu qui apparaît, allez dans l'onglet ''MISC'', et cochez l'option ''Enable NT/2000/XP Drivers''

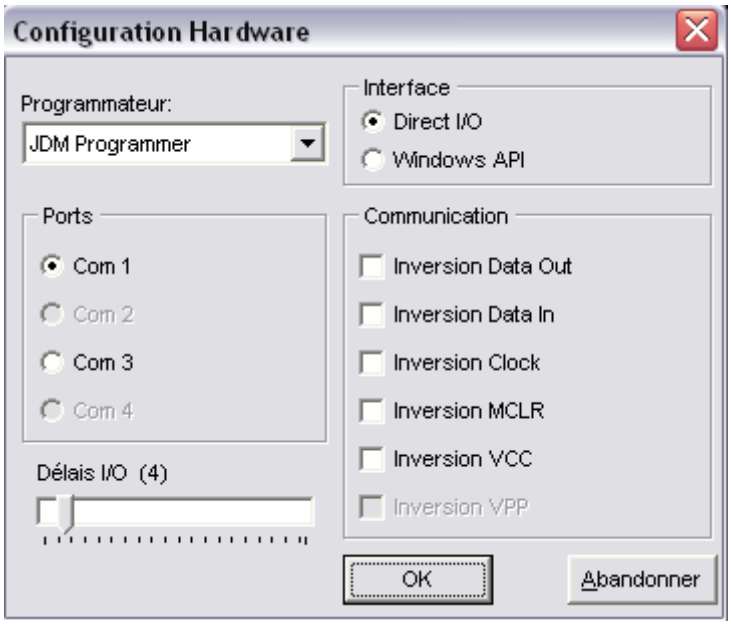


Vous devez alors redémarrer ICPROG pour que les changements prennent effet. Au redémarrage, il vous affiche cette fenêtre :



Après vous être assuré que le driver (icprog.sys) est présent dans le même dossier que l'exécutable, cliquez sur YES.

Maintenant, précisez à ICPROG, le programmateur que vous utilisez. Pour ce faire, appuyez sur F3, et réglez les paramètres comme suit :



Voilà, ICPROG est configuré pour Windows NT, 2000 et XP.

<u>Remarque :</u> avec un ordinateur portable, il se peut que la programmation échoue (erreur de type :''Echec de la vérification à l'adresse 0000h''). Ceci peut venir de votre ordinateur ne fournissant pas les bonnes tensions pour la programmation, ou alors, dans certains cas un câble série trop long (la résistance du fil entraînant une chute de tension entre le port série et le programmateur).

Bien. Maintenant, nous allons voir comment télécharger un programme dans le PIC.

Voici à quoi ressemble la barre des tâches de ICPROG :

Nous détaillerons ces icônes une à une :

1-Ouvrir un fichier (n'ouvrez que des fichiers .hex, les seuls aptes à être téléchargés)

2-Enregistrez (ne sert que rarement)

3-Hardware configurations (configuration déjà effectuée)

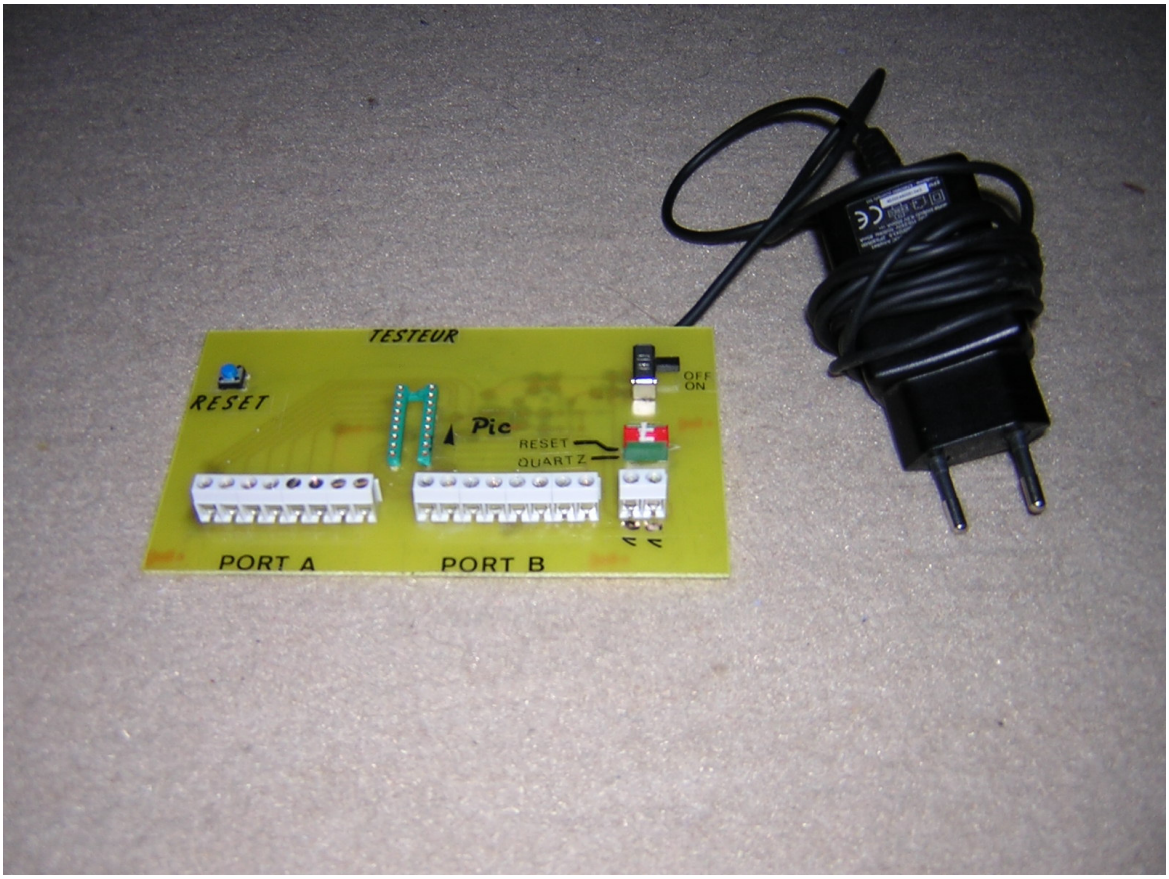
4-Options configurations (configuration déjà effectuée)
5-Sert à lire le code présent dans le composant. Le résultat s'affiche dans la fenêtre sous forme de code hexa (si pas d'erreur).
6-Sert à télécharger le programme dans le composant
7-Sert à effacer le contenu du composant
8-Sert à vérifier que le composant a bien été programmé
9-Assistant SmartCard (ne sert pas)
10-Sert à voir le code en assembleur
11-Sert à voir le code en hexadécimal

Télécharger le programme :

Pour télécharger un programme, il faut commencer par ouvrir un fichier .hex. Pour cela, cliquez sur l'icône 1 et allez chercher votre fichier . hex. Puis, réglez votre oscillateur, et vos paramètres comme indiqués plus haut. Effacez le composant en cliquant sur l'icône 7, puis programmez le avec l'icône 6.

***Remarque :* Dans certains cas, avant d'importer votre .hex, il est préférable de vérifier s'il n'y à rien sur votre composant, à l'aide de l'icône 5.**

CHAPITRE 4 : UN TESTEUR DE PIC

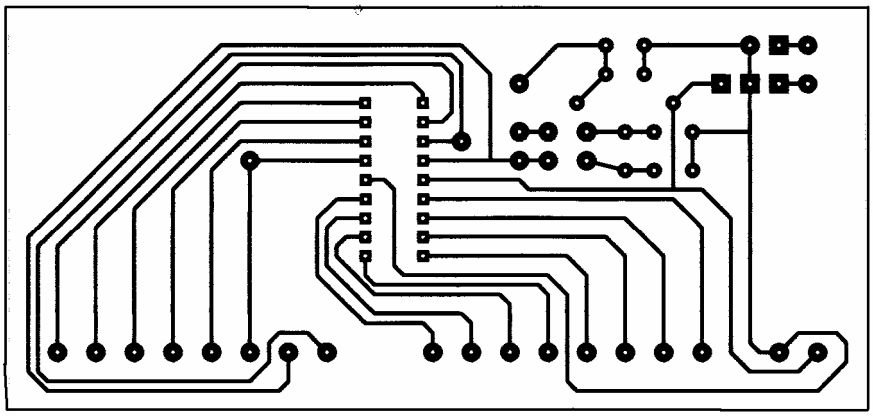


Dans cette partie, nous verrons un testeur de PIC, le principe, et le but de celui-ci.

Ce testeur, composé avec uniquement quelques composants, nous permet de tester nos programmes, et de vérifier le bon fonctionnement de ces derniers.

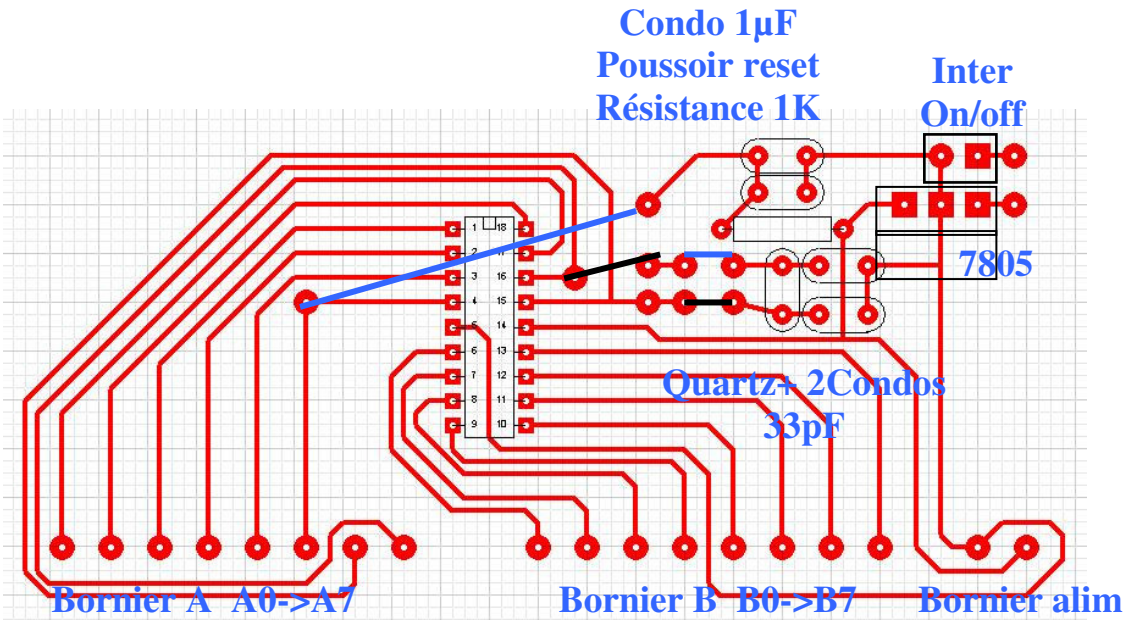
Son principe est simple. Le testeur possède sur sa plaque un quartz, un régulateur, un système de reset mixte et des séries de borniers. Un jeu de switch permet également de connecter ou non, le quartz, ainsi que le système de reset. Alimenté par un transformateur 220V-6V, le testeur délivre, grâce au régulateur (un 7805) une tension de 5 volts au PIC. Le système Reset assure le rôle qui lui est alloué, à la mise sous tension, et lorsque l'utilisateur le veut. Le quartz, cadencé à 4 Mhz, sert à délivrer le signal d'horloge au PIC. Les séries de borniers, assurent une connectivité des ports avec l'extérieur (une série de borniers par ports), ainsi, qu'une alimentation externe de 5 V.

En voici le typon (vu de dessus, attention à l'impression sur calque) :



Vous pouvez scanner et imprimer, sur calque, ce typon, afin de pouvoir tirer une plaque. Passez par un ordinateur, pour vous assurer de la bonne échelle. Le cadre fait 5,3 cm de haut et 11,4 cm de large. Attention au sens, le typon est ici vu à travers la plaque.

Voici l'implantation des composants (vu de dessus) :



La plupart des composants sont faciles à identifier. En ce qui concerne le 7805, il doit être placé, le dos vers le bas, c'est-à-dire, de gauche à droite, broche 3-2-1.

Les traits bleus correspondent aux positions des interrupteurs, l'idéal, étant de les déporter plus loin, et d'utiliser des fils pour les connecter. Les traits noirs correspondent à des vias.

Remarque : ***le via au niveau du quartz, traits bleu et noirs cote à cote, peut être remplacé par un interrupteur, mais cela est inutile, et constituait donc une dépense vaine. Cependant, l'espace a été prévu dans le cadre d'une future évolution possible.***

Deux quartz de 33 pF sont à prévoir sur chaque broche du quartz, afin d'éliminer d'éventuels parasites.

Nous avons choisi un transfo secteur de chargeur de téléphone, qui nous permet d'avoir jusqu'à 0.5A disponible. Si on tient compte du fait qu'un PIC peut fournir maximum environ 20mA par patte, dans le cas d'un 16F84, 240mA sont disponibles sur le bornier alim. Quand au bornier, notre choix s'est porté sur des bornier à vis pour CI.

Le but de ce testeur est donc de permettre de tester notre programme complet ou partiel en nous permettant de connecter facilement les périphériques, au choix ; le principal inconvénient de ce testeur étant sa fréquence de fonctionnement fixe, à 4 MHz.

Petite astuce : Vous pouvez ôter le régulateur et brancher le circuit sur une prise USB d'un ordinateur. En effet, la prise délivre une tension de 5V régulée. Cependant, cette manip reste réservée aux personnes possédant un minimum de connaissance. En effet, un mauvais branchement peut détruire votre PIC.

CHAPITRE 5 : EN BREF, TOUT CE QUI EST UTILE

Le but de cette partie est simple : y rassembler tous les éléments vus, et qui sont nécessaires à la programmation du PIC. Bonne programmation.

Tension de référence :

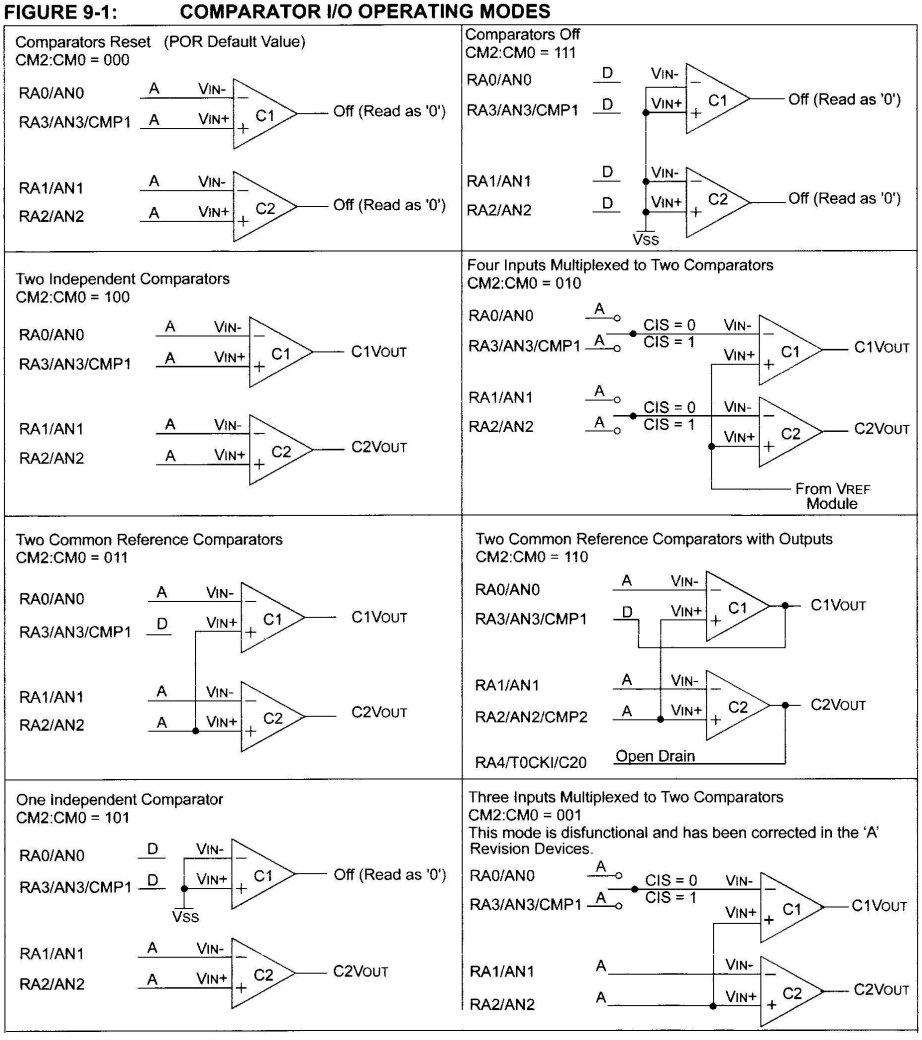
VREN	VROE	VRR		VR3	VR2	VR1	VR0

Comparateurs CMCON :

C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0

Configuration des comparateurs :

FIGURE 9-1: COMPARATOR I/O OPERATING MODES



Configuration du registre OPTION :

/RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0

PS2	PS1	PS0	Rapport de division	
			WDT	RTCC
0	0	0	:1	2
0	0	1	2	4
0	1	0	4	8
0	1	1	8	16
1	0	0	16	32
1	0	1	32	64
1	1	0	64	128
1	1	1	128	256

Bit PSA	0 = Choix du RTCC
	1 = Choix du WDT
Bit TOSE	0 = Front montant
	1 = Front descendant
Bit TOCS	0 = Horloge interne
	1 = Front sur la broche RTCC
Bit INTEDG	0 = interruption (RB0) validé sur front descendant
	1 = interruption (RB0) validé sur front montant
$\overline{\text{Bit RBPU}}$	0 = Connection d'une résistance de rappel entre RB4 et RB7
	1 = Aucune résistance de rappel

TRISA : (1 pour une entrée, 0 pour une sortie)

RA7	RA6	RA5	RA4	RA3	RA2	RA1	RA0

TRISB :

Configuration du Port B 1:entrée 0:sortie							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Registre interruptions INTCON :

GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF

Bit du registre INTCON	NOM	Fonction
Bit 7	GIE (Global Interrupt Enabled Bit)	Bit de validation général des interruptions (à 1 pour activer les interruptions)
Bit 6	PEIE	Activation ou non des périphériques
Bit 5	TOIE (Timer 0 Interrupt Enabled Bit)	Interruption pour le Timer 0
Bit 4	INTE (INTerrupt pin Enabled Bit)	Interruption sur la patte RB0
Bit 3	RBIE (RB port change Interrupt Enabled bit)	Interruption sur les pattes RB4 à RB7
Bit 2	TOIF (Timer 0 Interrupt Flag bit)	Signal que l'interruption vient du débordement du Timer 0
Bit 1	INTF (INTerrupt pin Flag bit)	Signal que l'interruption provient d'un changement d'état de RB0
Bit 0	RBIF (poRt B Interrupt Flag bit)	Signal que l'interruption provient des pattes RB4 à RB7

Module Timer1 (T1CON) :

-	-	T1CKPS1	T1CKPS0	T1OSCEN	/T1SYNC	TMR1CS	TMR1ON

PIE1 :

EEIE	CMIE	RCIE	TXIE	-	CCP1IE	TMR2IE	TMR1IE

PIR1 :

EEIF	CMIF	RCIF	TXIF	-	CCP1IF	TMR2IF	TMR1IF

Module Timer2 (T2CON) :

-	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0

RCSTA :

SPEN	RX9	SREN	CREN	ADEN	FERR	OERR	RX9D

USART (TXSTA) :

CSRC	TX9	TXEN	SYNC	-	BRGH	TRMT	TX9D

Desired Baud rate = Fosc / (64(X + 1))

9600 = 16000000 / (64(+1))X
X = i25.042°

Calculated Baud Rate = 16000000 / (64(25 + 1))
= 9615

Error = (Calculated Baud Rate = Desired Baud Rate) / Desired Baud Rate

= (9615 - 9600)/ 9600
= 0.16%

TABLE 12-1: BAUD RATE FORMULA

SYNC	BRGH = 0 (Low Speed)	BRGH = 1 (High Speed)
0	(Asynchronous) Baud Rate = Fosc/(64(X+1))	Baud Rate= Fosc/(16(X+1))
1	(Synchronous) Baud Rate = Fosc/(4(X+1))	NA

Legend: X = value in SPBRG (0 to 255)

TABLE 12-2: REGISTERS ASSOCIATED WITH BAUD RATE GENERATOR

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR	Value on all other RESETS
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
18h	RCSTA	SPEN	RX9	SREN	CREN	ADEN	FERR	OERR	RX9D	0000 -00x	0000 -00x
99h	SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

Legend: x = unknown, - = unimplemented read as '0'.
Shaded cells are not used by the BRG.

TABLE 12-3: BAUD RATES FOR SYNCHRONOUS MODE

BAUD RATE (K)	Fosc = 20 MHz		SPBRG value (decimal)	16 MHz		SPBRG value (decimal)	10 MHz		SPBRG value (decimal)
	KBAUD	ERROR		KBAUD	ERROR		KBAUD	ERROR	
0.3	NA	—	—	NA	—	—	NA	—	—
1.2	NA	—	—	NA	—	—	NA	—	—
2.4	NA	—	—	NA	—	—	NA	—	—
9.6	NA	—	—	NA	—	—	9.766	+1.73%	255
19.2	19.53	+1.73%	255	19.23	+0.16%	207	19.23	+0.16%	129
76.8	76.92	+0.16%	64	76.92	+0.16%	51	75.76	-1.36%	32
96	96.15	+0.16%	51	95.24	-0.79%	41	96.15	+0.16%	25
300	294.1	-1.96	16	307.69	+2.56%	12	312.5	+4.17%	7
500	500	0	9	500	0	7	500	0	4
HIGH	5000	—	0	4000	—	0	2500	—	0
LOW	19.53	—	255	15.625	—	255	9.766	—	255

BAUD RATE (K)	Fosc = 7.15909 MHz		SPBRG value (decimal)	5.0688 MHz		SPBRG value (decimal)	4 MHz		SPBRG value (decimal)
	KBAUD	ERROR		KBAUD	ERROR		KBAUD	ERROR	
0.3	NA	—	—	NA	—	—	NA	—	—
1.2	NA	—	—	NA	—	—	NA	—	—
2.4	NA	—	—	NA	—	—	NA	—	—
9.6	9.622	+0.23%	185	9.6	0	131	9.615	+0.16%	103
19.2	19.24	+0.23%	92	19.2	0	65	19.231	+0.16%	51
76.8	77.82	+1.32	22	79.2	+3.13%	15	75.923	+0.16%	12
96	94.20	-1.88	18	97.48	+1.54%	12	1000	+4.17%	9
300	298.3	-0.57	5	316.8	5.60%	3	NA	—	—
500	NA	—	—	NA	—	—	NA	—	—
HIGH	1789.8	—	0	1267	—	0	100	—	0
LOW	6.991	—	255	4.950	—	255	3.906	—	255

BAUD RATE (K)	Fosc = 3.579545 MHz		SPBRG value (decimal)	1 MHz		SPBRG value (decimal)	32.768 MHz		SPBRG value (decimal)
	KBAUD	ERROR		KBAUD	ERROR		KBAUD	ERROR	
0.3	NA	—	—	NA	—	—	0.303	+1.14%	26
1.2	NA	—	—	1.202	+0.16%	207	1.170	-2.48%	6
2.4	NA	—	—	2.404	+0.16%	103	NA	—	—
9.6	9.622	+0.23%	92	9.615	+0.16%	25	NA	—	—
19.2	19.04	-0.83%	46	19.24	+0.16%	12	NA	—	—
76.8	74.57	-2.90%	11	83.34	+8.51%	2	NA	—	—
96	99.43	+3.57%	8	NA	—	—	NA	—	—
300	298.3	0.57%	2	NA	—	—	NA	—	—
500	NA	—	—	NA	—	—		—	—
HIGH	894.9	—	0	250	—	0	8.192	—	0
LOW	3.496	—	255	0.9766	—	255	0.032	—	255

TABLE 12-4: BAUD RATES FOR ASYNCHRONOUS MODE (BRGH = 0)

BAUD RATE (K)	Fosc = 20 MHz		SPBRG value (decimal)	16 MHz		SPBRG value (decimal)	10 MHz		SPBRG value (decimal)
	KBAUD	ERROR		KBAUD	ERROR		KBAUD	ERROR	
0.3	NA	—	—	NA	—	—	NA	—	—
1.2	1.221	+1.73%	255	1.202	+0.16%	207	1.202	+0.16%	129
2.4	2.404	+0.16%	129	2.404	+0.16%	103	2.404	+0.16%	64
9.6	9.469	-1.36%	32	9.615	+0.16%	25	9.766	+1.73%	15
19.2	19.53	+1.73%	15	19.23	+0.16%	12	19.53	+1.73V	7
76.8	78.13	+1.73%	3	83.33	+8.51%	2	78.13	+1.73%	1
96	104.2	+8.51%	2	NA	—	—	NA	—	—
300	312.5	+4.17%	0	NA	—	—	NA	—	—
500	NA	—	—	NA	—	—	NA	—	—
HIGH	312.5	—	0	250	—	0	156.3	—	0
LOW	1.221	—	255	0.977	—	255	0.6104	—	255

BAUD RATE (K)	Fosc = 7.15909 MHz		SPBRG value (decimal)	5.0688 MHz		SPBRG value (decimal)	4 MHz		SPBRG value (decimal)
	KBAUD	ERROR		KBAUD	ERROR		KBAUD	ERROR	
0.3	NA	—	—	0.31	+3.13%	255	0.3005	-0.17%	207
1.2	1.203	+0.23%	92	1.2	0	65	1.202	+1.67%	51
2.4	2.380	-0.83%	46	2.4	0	32	2.404	+1.67%	25
9.6	9.322	-2.90%	11	9.9	+3.13%	7	NA	—	—
19.2	18.64	-2.90%	5	19.8	+3.13%	3	NA	—	—
76.8	NA	—	—	79.2	+3.13%	0	NA	—	—
96	NA	—	—	NA	—	—	NA	—	—
300	NA	—	—	NA	—	—	NA	—	—
500	NA	—	—	NA	—	—	NA	—	—
HIGH	111.9	—	0	79.2	—	0	62.500	—	0
LOW	0.437	—	255	0.3094	—	255	3.906	—	255

BAUD RATE (K)	Fosc = 3.579545 MHz		SPBRG value (decimal)	1 MHz		SPBRG value (decimal)	32.768 MHz		SPBRG value (decimal)
	KBAUD	ERROR		KBAUD	ERROR		KBAUD	ERROR	
0.3	0.301	+0.23%	185	0.300	+0.16%	51	0.256	-14.67%	1
1.2	1.190	-0.83%	46	1.202	+0.16%	12	NA	—	—
2.4	2.432	+1.32%	22	2.232	-6.99%	6	NA	—	—
9.6	9.322	-2.90%	5	NA	—	—	NA	—	—
19.2	18.64	-2.90%	2	NA	—	—	NA	—	—
76.8	NA	—	—	NA	—	—	NA	—	—
96	NA	—	—	NA	—	—	NA	—	—
300	NA	—	—	NA	—	—	NA	—	—
500	NA	—	—	NA	—	—	NA	—	—
HIGH	55.93	—	0	15.63	—	0	0.512	—	0
LOW	0.2185	—	255	0.0610	—	255	0.0020	—	255

TABLE 12-5: BAUD RATES FOR ASYNCHRONOUS MODE (BRGH = 1)

BAUD RATE (K)	Fosc = 20 MHz		SPBRG value (decimal)	16 MHz		SPBRG value (decimal)	10 MHz		SPBRG value (decimal)
	KBAUD	ERROR		KBAUD	ERROR		KBAUD	ERROR	
9600	9.615	+0.16%	129	9.615	+0.16%	103	9.615	+0.16%	64
19200	19.230	+0.16%	64	19.230	+0.16%	51	18.939	-1.36%	32
38400	37.878	-1.36%	32	38.461	+0.16%	25	39.062	+1.7%	15
57600	56.818	-1.36%	21	58.823	+2.12%	16	56.818	-1.36%	10
115200	113.636	-1.36%	10	111.111	-3.55%	8	125	+8.51%	4
250000	250	0	4	250	0	3	NA	—	—
625000	625	0	1	NA	—	—	625	0	0
1250000	1250	0	0	NA	—	—	NA	—	—

BAUD RATE (K)	Fosc = 7.16 MHz		SPBRG value (decimal)	5.068 MHz		SPBRG value (decimal)	4 MHz		SPBRG value (decimal)
	KBAUD	ERROR		KBAUD	ERROR		KBAUD	ERROR	
9600	9.520	-0.83%	46	9598.485	0.016%	32	9615.385	0.160%	25
19200	19.454	+1.32%	22	18632.35	-2.956%	16	19230.77	0.160%	12
38400	37.286	-2.90%	11	39593.75	3.109%	7	35714.29	-6.994%	6
57600	55.930	-2.90%	7	52791.67	-8.348%	5	62500	8.507%	3
115200	111.860	-2.90%	3	105583.3	-8.348%	2	125000	8.507%	1
250000	NA	—	—	316750	26.700%	0	250000	0.000%	0
625000	NA	—	—	NA	—	—	NA	—	—
1250000	NA	—	—	NA	—	—	NA	—	—

BAUD RATE (K)	Fosc = 3.579 MHz		SPBRG value (decimal)	1 MHz		SPBRG value (decimal)	32.768 MHz		SPBRG value (decimal)
	KBAUD	ERROR		KBAUD	ERROR		KBAUD	ERROR	
9600	9725.543	1.308%	22	8.928	-6.994%	6	NA	NA	NA
19200	18640.63	-2.913%	11	20833.3	8.507%	2	NA	NA	NA
38400	37281.25	-2.913%	5	31250	-18.620%	1	NA	NA	NA
57600	55921.88	-2.913%	3	62500	+8.507	0	NA	NA	NA
115200	111243.8	-2.913%	1	NA	—	—	NA	NA	NA
250000	223687.5	-10.525%	0	NA	—	—	NA	NA	NA
625000	NA	—	—	NA	—	—	NA	NA	NA
1250000	NA	—	—	NA	—	—	NA	NA	NA

TABLE 12-7: REGISTERS ASSOCIATED WITH ASYNCHRONOUS RECEPTION

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR	Value on all other RESETS
0Ch	PIR1	EEIF	CMIF	RCIF	TXIF	—	CCP1IF	TMR2IF	TMR1IF	0000 -000	0000 -000
18h	RCSTA	SPEN	RX9	SREN	CREN	ADEN	FERR	OERR	RX9D	0000 -00x	0000 -00x
1Ah	RCREG	USART Receive Register								0000 0000	0000 0000
8Ch	PIE1	EEIE	CMIE	RCIE	TXIE	—	CCP1IE	TMR2IE	TMR1IE	0000 -000	0000 -000
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

Legend: x = unknown, - = unimplemented locations read as '0'. Shaded cells are not used for Asynchronous Reception.

TABLE 12-8: REGISTERS ASSOCIATED WITH ASYNCHRONOUS RECEPTION

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR	Value on all other RESETS
0Ch	PIR1	EEIF	CMIF	RCIF	TXIF	—	CCP1IF	TMR2IF	TMR1IF	0000 -000	0000 -000
18h	RCSTA	SPEN	RX9	SREN	CREN	ADEN	FERR	OERR	RX9D	0000 -00x	0000 -00x
1Ah	RCREG	RX7	RX6	RX5	RX4	RX3	RX2	RX1	RX0	0000 0000	0000 0000
8Ch	PIE1	EEIE	CMIE	RCIE	TXIE	—	CCP1IE	TMR2IE	TMR1IE	0000 -000	0000 -000
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

Legend: x = unknown, - = unimplemented locations read as '0'. Shaded cells are not used for Asynchronous Reception.

TABLE 12-9: REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER TRANSMISSION

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR	Value on all other RESETS
0Ch	PIR1	EEIF	CMIF	RCIF	TXIF	—	CCP1IF	TMR2IF	TMR1IF	0000 -000	0000 -000
18h	RCSTA	SPEN	RX9	SREN	CREN	ADEN	FERR	OERR	RX9D	0000 -00x	0000 -00x
19h	TXREG	USART Transmit Register								0000 0000	0000 0000
8Ch	PIE1	EEIE	CMIE	RCIE	TXIE	—	CCP1IE	TMR2IE	TMR1IE	0000 -000	0000 -000
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

Legend: x = unknown, - = unimplemented, read as '0'. Shaded cells are not used for Synchronous Master Transmission.

TABLE 12-10: REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER RECEPTION

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR	Value on all other RESETS
0Ch	PIR1	EEIF	CMIF	RCIF	TXIF	—	CCP1IF	TMR2IF	TMR1IF	0000 -000	0000 -000
18h	RCSTA	SPEN	RX9	SREN	CREN	ADEN	FERR	OERR	RX9D	0000 -00x	0000 -00x
1Ah	RCREG	USART Receive Register								0000 0000	0000 0000
8Ch	PIE1	EEPIE	CMIE	RCIE	TXIE	—	CCP1IE	TMR2IE	TMR1IE	-000 0000	-000 -000
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

Legend: x = unknown, - = unimplemented read as '0'. Shaded cells are not used for Synchronous Master Reception.

TABLE 12-11: REGISTERS ASSOCIATED WITH SYNCHRONOUS SLAVE TRANSMISSION

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR	Value on all other RESETS
0Ch	PIR1	EEIF	CMIF	RCIF	TXIF	—	CCP1IF	TMR2IF	TMR1IF	0000 -000	0000 -000
18h	RCSTA	SPEN	RX9	SREN	CREN	ADEN	FERR	OERR	RX9D	0000 -00x	0000 -00x
19h	TXREG	USART Transmit Register								0000 0000	0000 0000
8Ch	PIE1	EEIE	CMIE	RCIE	TXIE	—	CCP1IE	TMR2IE	TMR1IE	0000 -000	0000 -000
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

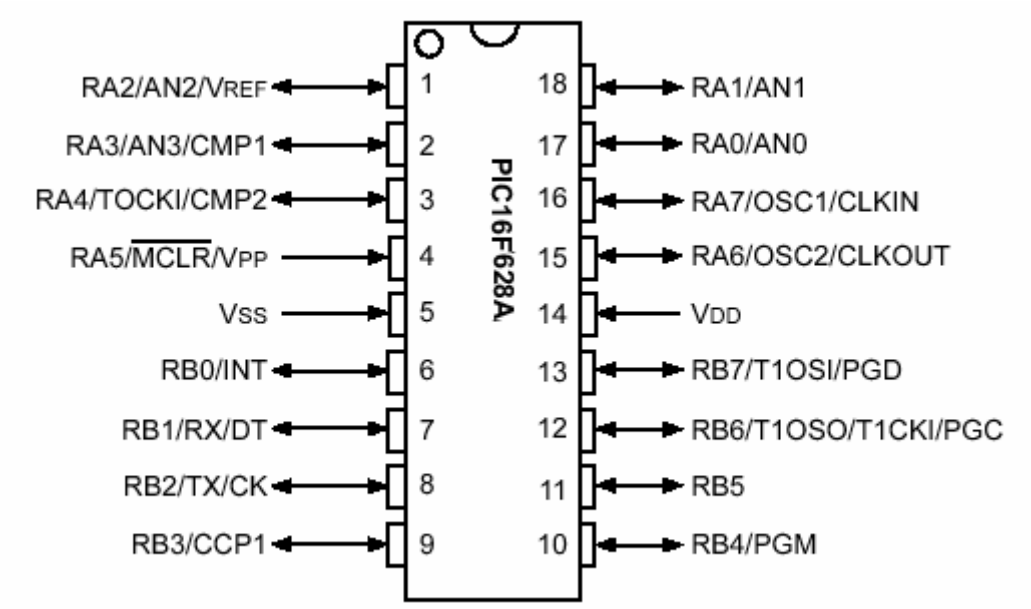
Legend: x = unknown, - = unimplemented read as '0'. Shaded cells are not used for Synchronous Slave Transmission.

TABLE 12-12: REGISTERS ASSOCIATED WITH SYNCHRONOUS SLAVE RECEPTION

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR	Value on all other RESETS
0Ch	PIR1	EEIF	CMIF	RCIF	TXIF	—	CCP1IF	TMR2IF	TMR1IF	0000 -000	0000 -000
18h	RCSTA	SPEN	RX9	SREN	CREN	ADEN	FERR	OERR	RX9D	0000 -00x	0000 -00x
1Ah	RCREG	USART Receive Register								0000 0000	0000 0000
8Ch	PIE1	EEIE	CMIE	RCIE	TXIE	—	CCP1IE	TMR2IE	TMR1IE	0000 -000	0000 -000
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

Legend: x = unknown, - = unimplemented read as '0'. Shaded cells are not used for Synchronous Slave Reception.

Répartitions des pattes du PIC



CHAPITRE 6 : CONCLUSION

Ainsi, comme nous avons pu le voir, le PIC 16F628 possède des qualités indéniables, supérieures à celles du PIC 16F84. Cependant, ce dernier est extrêmement bien ancré dans certains milieux, ce qui explique que le 628 ne soit pas encore très répandu, ni très connu.

Compatible avec le PIC 16F84, le 628 finira un jour par le remplacer. Etant donné le coût inférieur du 628, il est même, économiquement parlant, plus viable de remplacer les PIC 16F84 par des 628, pour peu qu'on puisse en trouver chez le revendeur électronique du coin.

La grande compatibilité, également au niveau logiciel, facilite encore plus la transition, le passage du 84 au 628.

De plus, le matériel de programmation est le même que pour le 84, limitant ainsi les investissements, et d'avoir de multiple matériel.

Tout comme avec le 16F84, le 628 nous permettra de réaliser de nombreux projets avec un minimum de difficultés, et de composants, grâce à ses fonctions supplémentaires intégrées.

CHAPITRE 7 : 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

Asynchrone : a l'inverse de synchrone, ce terme est utilisé pour désigner des actions (signaux) ayant un décalage temporel l'un par rapport à l'autre

Bit : unité de base en informatique

Buffer : zone de mémoire vive permettant de stocker des données temporairement

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

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 capables de générer un signal régulier

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

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

PR2 : Registre appartenant au timer 2, sert à la PWM (voir plus loin)

PWM : Pulse Width Modulation (modulation en largeur d'impulsions, utilisé notamment dans les variateurs de lumière ; mode que possède le PIC 628, mode non explicité ici.

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 réduits, limités

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

Synchrone : terme utilisé pour signifier des actions (signaux) qui se déroulent en même temps

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

TOCKI : mode de la patte RA4, permettant au timer 0 de compter des impulsions externes, sur front montant ou descendant selon la configuration

USART : Universal Synchronous Asynchronous Receiver Transmitter, partie du 628 permettant la communication avec le monde externe en mode synchrone ou asynchrone

CHAPITRE 8 : LIENS INTERNET UTILES

Hervé Hollard, cours de langage C :

- http://perso.club-internet.fr/hhollard/
-http://perso.club-internet.fr/hhollard/prog_pic_c.htm

Adresse pour le programmateur de PIC :

-http://www.jdm.homepage.dk/newpic.htm
-http://www.jdm.homepage.dk/pcb2.htm

Adresse de Microchip :

http://www.microchip.com/

Adresse du compilateur CC5X :

http://www.bknd.com/cc5x/index.shtml

Adresse de l'auteur:

http://diablotronic.bzh.bz