



1 - GENERALITES

- ☞ Va-t-on faire de l'électronique ou de l'informatique ???
- ☞ Les 2, mon général !!!

1.1 : Les Microprocesseurs :

Apparus dès la création des premiers circuits intégrés numériques, au début des années 1970, les **microprocesseurs** constituent le cœur de presque toutes les réalisations électroniques; on en trouve dans tous les domaines, notamment : l'informatique (de la calculatrice à l'ordinateur), l'automobile (ABS, injection, ...), l'automatique (automates programmables, contrôle de processus, ...), l'électronique domestique (thermomètre, télécommande, carte à puce, ...).

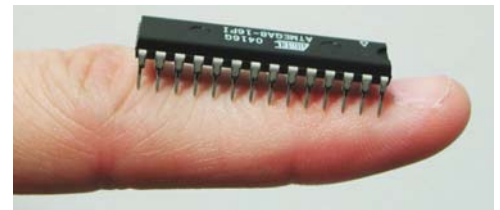


Les performances des microprocesseurs sont liées aux possibilités offertes par la technologie, en terme de capacité (nombre de portes logiques intégrées) et de vitesse, et au choix d'architectures adaptées (ou imposées pour cause de compatibilité ascendante); à l'heure actuelle, on trouve sur le marché des microprocesseurs intégrant des millions de transistors, fonctionnant à plus de 300 MHz et disposés dans des boîtiers de plusieurs centaines de broches, ils sont issus de différentes approches architecturales : CISC, RISC, DSP et VLIW.

1.2 : Les Microcontrôleurs :

Un système minimal, pour fonctionner, a besoin :

- . D'une Unité Centrale (Processeur).
- . De Mémoire *morte*, pour le programme (PROM, EPROM, ...).
- . De Mémoire *vive*, pour les calculs, pour stocker les données.
- . De circuits Interfaces, pour connecter les périphériques qui vont permettre la communication avec l'extérieur.



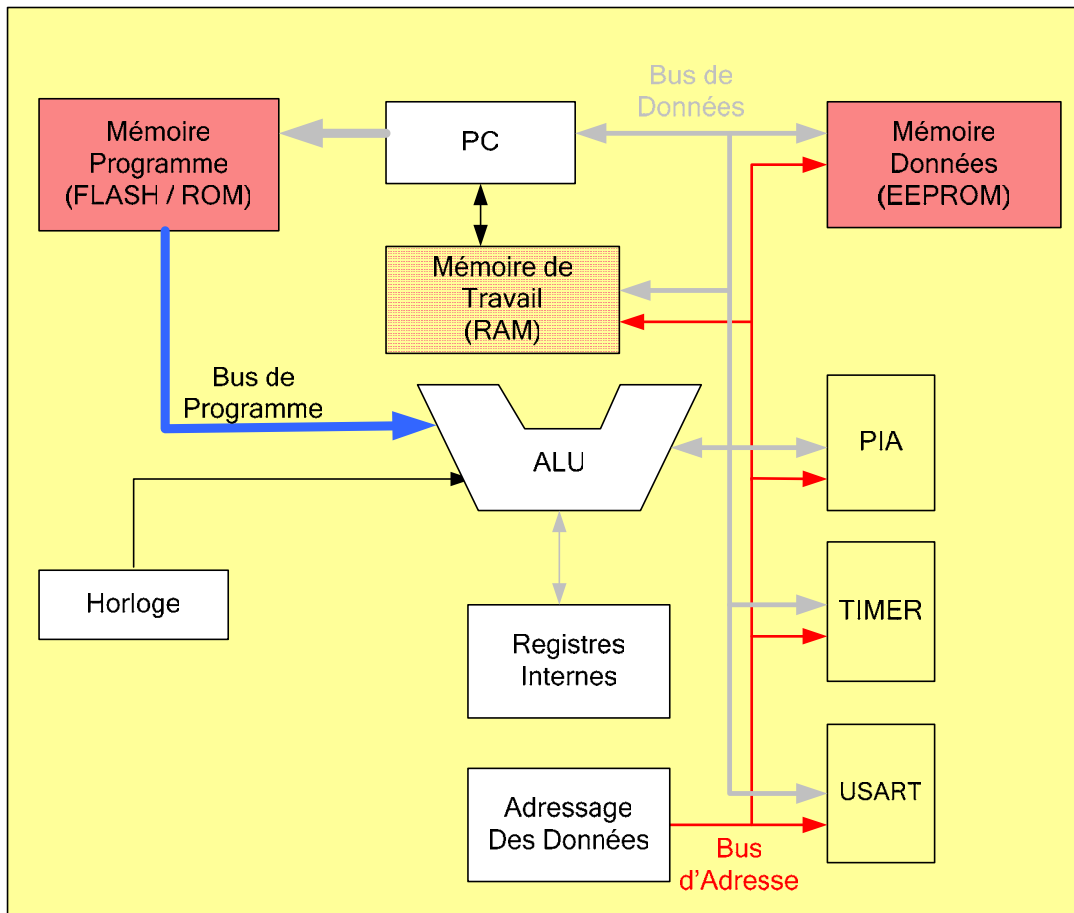
Les microcontrôleurs sont des circuits logiques qui permettent l'exécution d'un programme dont les actions dépendent de l'état des variables d'environnement du système.

Ces **circuits intégrés** à bas prix (quelques euros) sont de véritables ordinateurs miniatures autonomes utilisés dans les voitures (ordinateurs de bords, airbags, freins ABS...), les lecteurs DVD, les fours à micro-ondes, etc...

On peut définir un microcontrôleur comme une unité de traitement de l'information de type microprocesseur à laquelle on a ajouté des périphériques internes permettant de réaliser des montages sans nécessiter l'ajout de composants externes.



Architecture Interne d'un Microcontrôleur :



PC : Compteur Programme

ALU : Unité Arithmétique et Logique.

PIA : Interface Parallèle Programmable
(Parallel Interface Adaptator)

TIMER : Générateur de signaux d'horloge

USART : Transmetteur Série Synchrones - Asynchrone
(Universal Synchronous - Asynchronous Receiver/Transmitter)

BUS d'Adresse : Permet d'activer le circuit pour lequel les données sont envoyées ou lues.

BUS de Données : Contient l'information binaire devant être traitée par l'ALU ou le circuit adressé.



2 – LE MICROCONTROLEUR PIC

2.1 : Présentation

Les microcontrôleurs PIC du constructeur MICROCHIP ont été conçus sur une architecture dite HARVARD et non sur un modèle VON NEUMANN :

	Architecture VON NEUMANN (MOTOROLA, INTEL, ZILOG, ..)	Architecture HARVARD (RISC) (MICROCHIP PICS)
Concept	<ul style="list-style-type: none">basé sur un bus de données unique. Celui-ci véhicule les instructions et les données.	<ul style="list-style-type: none">basée sur deux bus de données. Un bus est utilisé pour les données et un autre pour les instructions.
Architecture		
Avantages	<ul style="list-style-type: none">Jeu d'instructions riches.Accès à la mémoire facile.	<ul style="list-style-type: none">Jeu d'instructions pauvre, maisfacile à mémoriser.Le codage des instructions est facile, chaque instruction est codée sur un mot et dure un cycle machine.Le code est plus compact.
Inconvénients	<ul style="list-style-type: none">Le temps pour exécuter une instruction est variable.Le codage des instructions se fait sur plusieurs octets.	<ul style="list-style-type: none">Le jeu d'instruction est pauvre :Par exemple pour effectuer une comparaison il faut faire une soustraction.Les accès aux registres internes et la mémoire sont très délicats.

3 familles de PIC en fonction de la largeur du bus de données:

- Base-Line : 12 bits (ex : 12F509)
- Mid - Range : 14 bits (ex : 16F84, 16F877)
- High - End : 16 bits (ex : 18F4410)

3 modes de programmation des PIC

- ROM : CR
- EPROM - EEPROM : C (ex : 16C74)
- FLASH : F (ex : 16F84)

Les PIC sont des microcontrôleurs RISC. **RISC**, de l'anglais **Reduced Instructions Set Computer**, signifie « calculateur à jeu réduit d'instructions ».

Les instructions sont au nombre de 35, soit une centaine de moins que le microcontrôleur 68HC11 de Motorola par exemple.

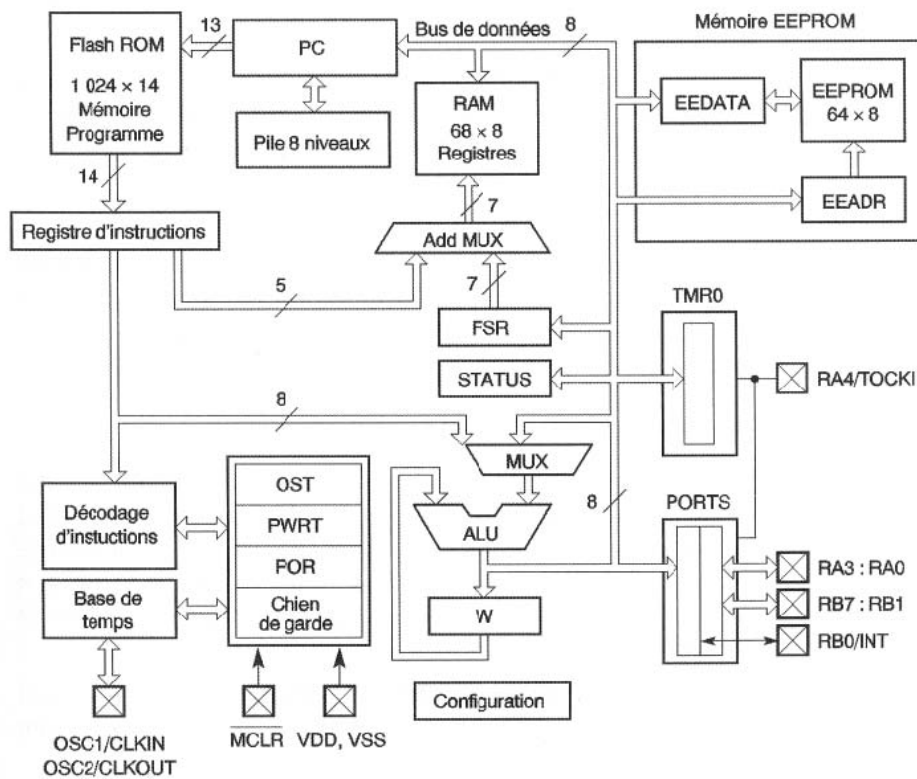


2.2 : Caractéristiques comparées des PIC 16F84 et 16F877

Caractéristiques	16F84	16F877
Mémoire Programme (mots)	1792	14336
EEPROM de données (mots)	64	256
RAM	68	368
Nb de broches d'entrées/Sorties	13	33
Boîtiers disponibles	18P (DIL18) , 18SO, 20SS	40P (DIL40) , 44ML, 44L, 44PT
Convertisseur A/N		8 de 10bits
Timers - Watchdog	1 de 8bits / 1 WDT	1 de 16bits/2 de 8bits / 1 WDT
Entrées Sorties Série		AUSART / I2C / SPI
Fréquence maximale	20Mhz	20Mhz

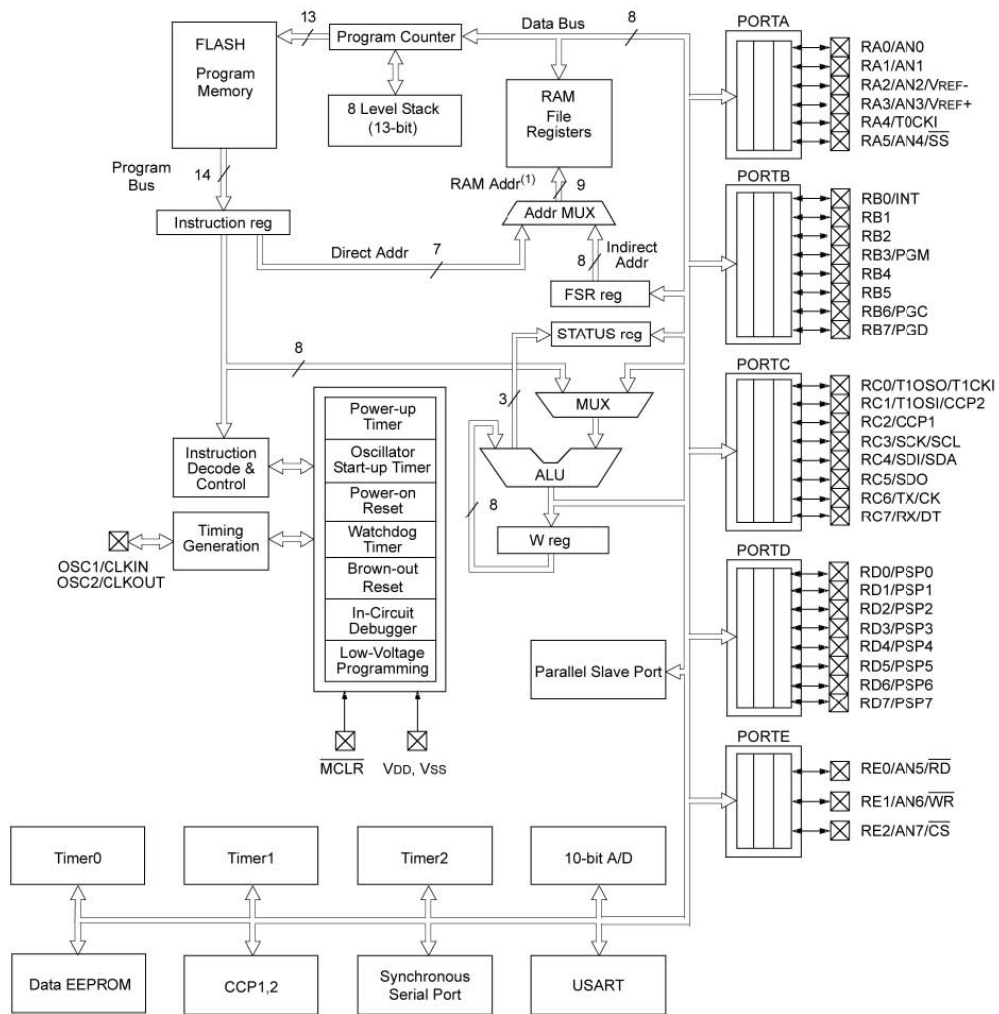
2.3 : Architectures comparée des PIC 16F84 et 16877

• 16F84





• 16F877

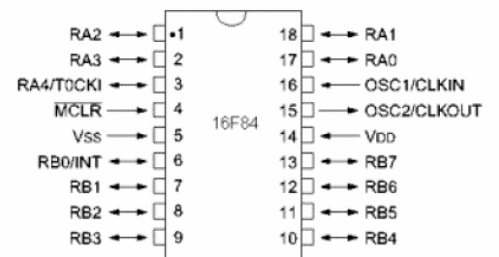


2.4 : Découverte du 16F84

Parmi les principaux constituants, on remarque :

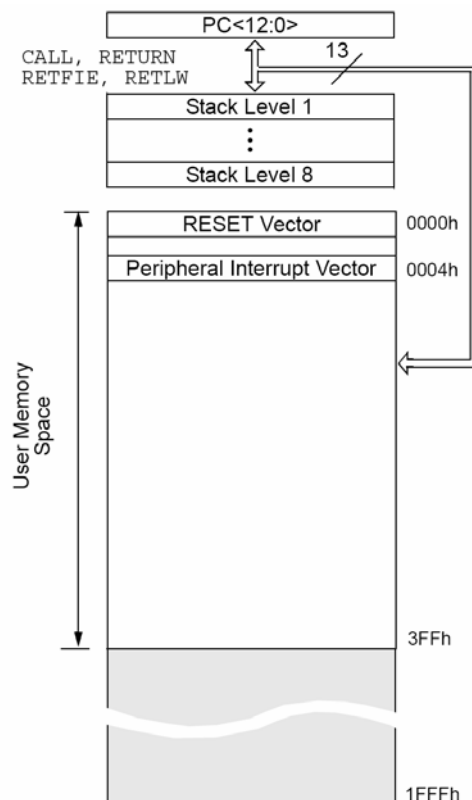
- la mémoire de type Flash ROM pour le programme ;
- la mémoire de type RAM pour les registres ;
- des registres particuliers : W, FSR et d'état ;
- l'Unité Arithmétique et Logique (ALU) ;
- les ports d'entrées/sorties ;
- la pile à 8 niveaux.
- la mémoire de type EEPROM pour les données à sauvegarder.

Brochage du PIC 16F84 (µC vu de dessus)





La mémoire de programme Comme son nom l'indique, c'est dans cette mémoire qu'est mémorisé le programme à exécuter. De type Flash ROM, elle est organisée en 1 024 mots de 14 bits, ce qui permet de coder sur un seul mot l'instruction et l'opérande, à la différence des microcontrôleurs plus classiques à 8 bits où l'instruction et l'opérande sont écrits sur plusieurs adresses consécutives. Chaque mot de cette mémoire programme est repéré par son adresse allant de 0 à 1023 (0 à 3FF en hexadécimal). Le Compteur Programme (PC) contient l'adresse de l'instruction à exécuter.



La pile Ce terme désigne les emplacements mémoire où sont empilées les adresses de retour lors des appels à des sous-programmes. La pile ne comportant que 8 niveaux, il ne faut pas imbriquer plus de 8 sous-programmes.

Les registres La mémoire RAM de 68 octets est utilisée pour stocker des données temporaires. Baptisé du nom de registre à usage général par Microchip, chacune de ces mémoires est adressable directement. À ce bloc de 68 registres à usage général, le fabricant a associé 11 registres internes et organisé les adresses selon le tableau ci-dessous :




Adresses		Adresses	
00	INDF	INDF	80
01	TMR0	OPTION	81
02	PCL	PCL	82
03	STATUS	STATUS	83
04	FSR	FSR	84
05	PORTA	TRISA	85
06	PORTB	TRISB	86
07			87
08	EEDATA	EECON1	88
09	EEADR	EECON2	89
0A	PCLATH	PCLATH	8A
0B	INTCON	INTCON	8B
0C à 4F	68 mémoires disponibles	Accès aux mêmes mémoires que page 0	8C à CF
Page 0		Page 1	

• Description des principaux registres internes

INDF : Le registre INDF d'adresse 00 est utilisé pour l'adressage indirect.

TMRO : Le registre TMRO d'adresse 01 est lui aussi particulier puisqu'il constitue un compteur programmable incrémenté continuellement à une cadence constante indépendante de l'exécution du reste du programme. Après avoir atteint la valeur 255, le compteur reprend à 0. La cadence de l'incrémentation est proportionnelle à la fréquence de l'horloge appliquée au microcontrôleur ou à la fréquence du signal appliqué sur la broche 3 (RA4/TOCKI).

 **PCL et PCLATH** : Le registre PCL d'adresse 02 contient les 8 bits de poids faible du PC qui est codé sur 13 bits. Lors d'opérations d'écriture dans le registre PCL, le nouveau compteur programme est calculé à partir des 8 bits du registre PCL et des 5 bits du registre PCLATH d'adresse 0A.



📖 **PORTA** et **PORTB** : Les registres **PORTA** d'adresse 05 et **PORTB** d'adresse 06 sont les registres des ports **d'entrées/sorties**.

PORTA :

Name	Bit0	Buffer Type	Function
RA0	bit0	TTL	Input/output
RA1	bit1	TTL	Input/output
RA2	bit2	TTL	Input/output
RA3	bit3	TTL	Input/output
RA4/T0CKI	bit4	ST	Input/output or external clock input for TMR0. Output is open drain type.

Legend: TTL = TTL input, ST = Schmitt Trigger input

PORTB :

Name	Bit	Buffer Type	I/O Consistency Function
RB0/INT	bit0	TTL/ST ⁽¹⁾	Input/output pin or external interrupt input. Internal software programmable weak pull-up.
RB1	bit1	TTL	Input/output pin. Internal software programmable weak pull-up.
RB2	bit2	TTL	Input/output pin. Internal software programmable weak pull-up.
RB3	bit3	TTL	Input/output pin. Internal software programmable weak pull-up.
RB4	bit4	TTL	Input/output pin (with interrupt-on-change). Internal software programmable weak pull-up.
RB5	bit5	TTL	Input/output pin (with interrupt-on-change). Internal software programmable weak pull-up.
RB6	bit6	TTL/ST ⁽²⁾	Input/output pin (with interrupt-on-change). Internal software programmable weak pull-up. Serial programming clock.
RB7	bit7	TTL/ST ⁽²⁾	Input/output pin (with interrupt-on-change). Internal software programmable weak pull-up. Serial programming data.

Chaque broche de chaque port peut être utilisée en **entrée** ou en **sortie** : le sens de fonctionnement des ports est défini par le contenu des registres **TRISA** et **TRISB** :

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-on Reset	Value on all other RESETS
PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0/INT	xxxx xxxx	uuuu uuuu
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	1111 1111	1111 1111

- ☞ Un bit à **0** programme la broche correspondante en **sortie**.
- ☞ Un bit à **1** la programme en **entrée**.


Quand une broche est en sortie, son état est déterminé par le bit correspondant du **registre** du port (**PORTA** ou **PORTB**).

Quand une broche est en entrée, c'est son niveau qui lui est fixé par le montage extérieur qui détermine la valeur du bit du registre correspondant. Lors d'un *Reset*, les bits des registres TRISA et TRISB étant mis à 1, toutes les broches des ports A et B sont **initialement** des **entrées**.



Exemple : indiquez quelles sont les broches du PORT B qui sont des entrées (Input) et celles qui sont des sorties (Output) si le registre **TRISB** est programmé ainsi : TRISB = %00110111

PORTB	B7	B6	B5	B4	B3	B2	B1	B0
I/O								

 **INTCON** : Le registre INTCON d'adresse 0B, présenté dans le chapitre des registres (p. 6), est utilisé pour le contrôle et la gestion des interruptions.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
GIE	EEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF

Une interruption est un évènement qui va interrompre le programme en cours et faire exécuter un second programme dit « Programme d'Interruption ». Cet évènement peut être interne, par programmation du TIMER, ou externe par la mise à 1 d'une broche d'entrée.

Le 16F84 est très pauvre à ce niveau, puisqu'il ne dispose que de 4 sources d'interruptions possibles (contre 13 pour le 16F876 par exemple). Les évènements susceptibles de déclencher une interruption sont les suivants :

- **TMR0** : Débordement du timer0 (tmr0). Une fois que le contenu du tmr0 passe de 0xff à 0x00, une interruption peut être générée.
- **EEPROM** : cette interruption peut être générée lorsque l'écriture dans une case EEPROM interne est terminée.
- **RB0/INT** : Une interruption peut être générée lorsque, la pin RB0, encore appelée INTerrupt pin, étant configurée en entrée, le niveau qui est appliqué est modifié.
- **PORTB** : De la même manière, une interruption peut être générée lors du changement d'un niveau sur une des pins RB4 à RB7. Il n'est pas possible de limiter l'interruption à une seule de ces pins. L'interruption sera effective pour les 4 pins ou pour aucune.

Le registre de contrôle des interruptions INTCON permet de paramétrer individuellement ces 4 types d'interruption et de déterminer par la mise à 1 des drapeaux (FLAG) quelle interruption a été déclenchée.



Description des bits du registre INTCON :

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
GIE	EEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF
bit 7							bit 0

GIE : (Global Interrupt Enable) Valide ou non la prise en comptes des interruptions.
1 = Toutes les interruptions non-masquables sont validées.
0 = Toutes les interruptions non-masquables sont inhibées.

EEIE : (EE Write Complete Interrupt Enable) Ecriture dans une case EEPROM
1 = Interruption Validée
0 = Interruption inhibée

TOIE : (TMR0 Overflow Interrupt Enable) Interruption sur dépassement du TIMER
1 = Interruption Validée
0 = Interruption inhibée

INTE : (RB0/INT External Interrupt Enable) Interruption sur mise à 1 de l'entrée RB0
1 = Interruption Validée
0 = Interruption inhibée

RBIE : (RB Port Change Interrupt Enable) Interruption sur Changement d'état du PORTB
1 = Interruption Validée
0 = Interruption inhibée

TOIF : (TMR0 Overflow Interrupt Flag) Drapeau d'interruption du dépassement du TIMER
1 = Le registre TMR0 a créé un Dépassement
0 = Le registre TMR0 n'a pas créé de Dépassement

INTF: (RB0/INT External Interrupt Flag) Drapeau d'interruption du Bit RBO
1 = La broche RB0 est passée à 1
0 = La broche RB0 n'est pas passée à 1

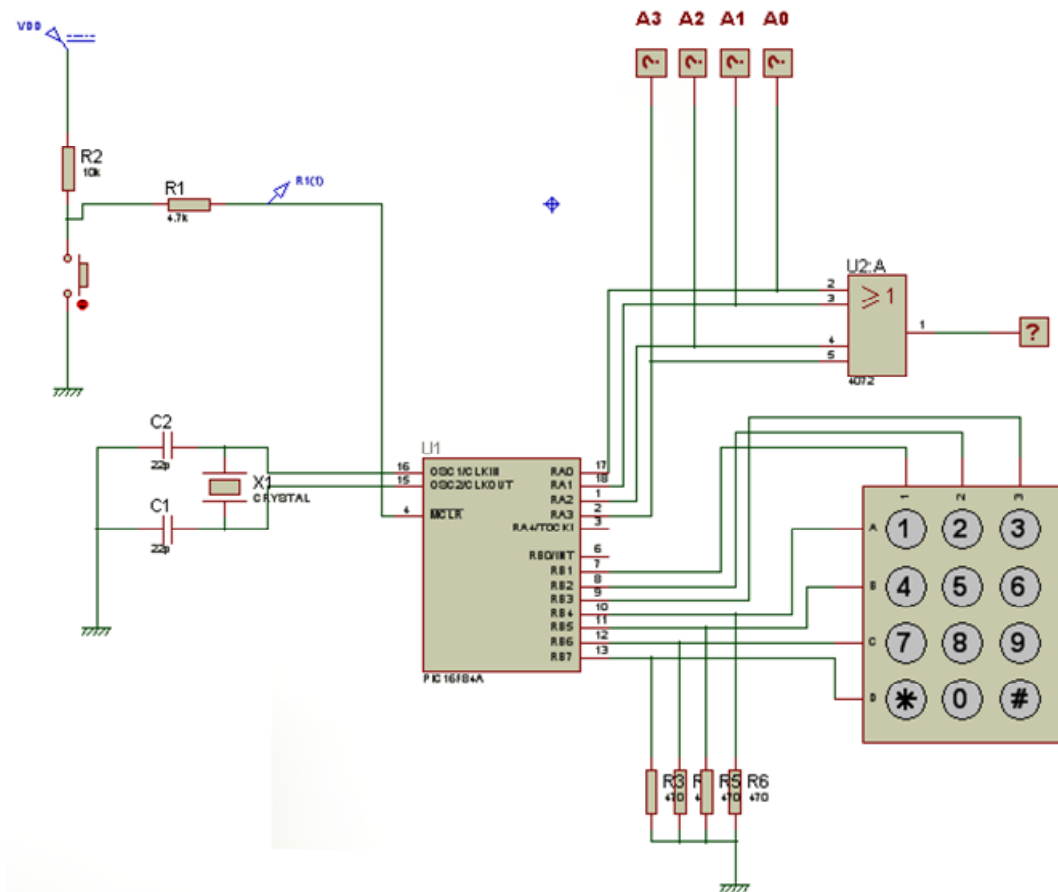
RBIF: (RB Port Change Interrupt Flag)
1 = au moins une des broches RB7 à RB4 a changé d'état
0 = aucune des broches RB7 à RB4 n'a changé d'état



Exemple : Complétez le tableau de programmation du registre INTCON si l'on désire qu'une interruption soit créée lorsque le Bit 5 du Port B change d'état, ou lorsque le PORT B.0 passe à 1.

INCTCON	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
%								

Exemple d'application : Décodage Clavier Alarme Intrusion



(cf. Exercice d'application PROGRAMMATION du CLAVIER)



2.5 : Découverte du 16F877

Les broches

- Broche **MCLR**

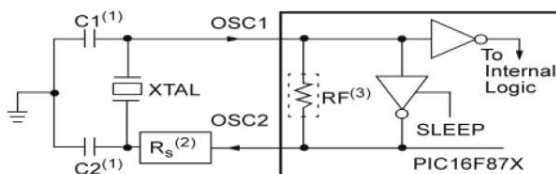
Cette broche sert à initialiser le μC , elle peut être simplement reliée à VDD si on n'a pas besoin de RESET externe. Par contre si on souhaite implanter un bouton de remise à zéro, on pourra câbler un simple réseau RC sur la broche MCLR.

Le μC génère automatiquement un RESET à la mise sous tension (lorsque celle-ci passe entre 1,2V et 1,7V)

- Broches **OSCI/OSC2** ou **CLKIN/CLOUT**. Ces broches permettent de faire fonctionner l'oscillateur interne du **PIC**. On peut utiliser **3** types d'oscillateurs :

- ☞ **Quartz ou résonateur céramique.**

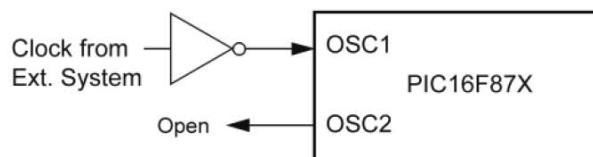
Schéma de câblage



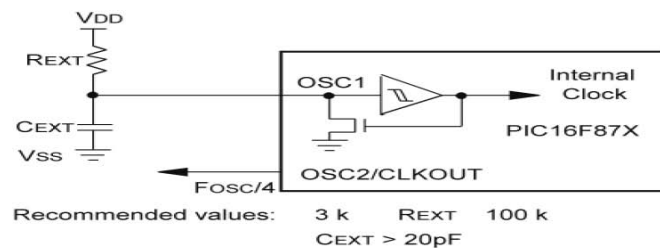
Choix des condensateurs

Osc Type	Crystal Freq.	Cap. Range C1	Cap. Range C2
LP	32 kHz	33 pF	33 pF
	200 kHz	15 pF	15 pF
XT	200 kHz	47-68 pF	47-68 pF
	1 MHz	15 pF	15 pF
	4 MHz	15 pF	15 pF
HS	4 MHz	15 pF	15 pF
	8 MHz	15-33 pF	15-33 pF
	20 MHz	15-33 pF	15-33 pF

- ☞ **Oscillateur externe**



- ☞ **Reseau RC**



Remarques



• Les instructions standards durent 1 cycle machine (sauf les instructions de sauts 2 cycles). Le μC utilise 4 coups d'horloge pour réaliser un cycle machine. Si la fréquence du QUARTZ est de 20MHz ($T=50\text{nS}$), une instruction sera exécutée toutes les 200nS, Dans ce cas là, le μC a une puissance de calcul de 5MIPS (5 Millions d'instructions par secondes ! ! !).

• La fréquence MAX est de 20MHz pour les μC dont les références se terminent par -20. Par exemples : 16F877-20 (20MHz max) et 16F877-04 (4MHz max). La fréquence MIN est le continu.

• La consommation du circuit sera d'autant plus faible que la fréquence sera petite, cela peut être intéressant pour des applications de faible consommation (alimentation autonome). Pour des applications faible consommation, on peut utiliser les séries LF (Low Frequency and Low Power).

• Broches d'alimentation : **VDD et VSS.**

Ce sont les broches d'alimentation du circuit. Les tensions qui peuvent être appliquées vont :

- De 4,5V à 6V pour la gamme standard F.
- De 2 à 6V pour la gamme étendue LF.

L'intensité du courant consommé peut aller de **1pA à 10mA**. La consommation du μC sera fonction de :

- La tension d'alimentation.
- La fréquence interne.
- Le mode de fonctionnement.

De plus ces bornes doivent être découplées par deux condensateurs :

- **1 pF** électrolytique.
- **10nF** céramique.

• Broche **RBO/INT**. Cette broche a une double fonction elle peut être utilisée comme une broche standard RBO ou comme une entrée d'interruption INT.

Si cette broche est utilisée comme une entrée d'interruption externe, elle doit être maintenue à un niveau haut par l'intermédiaire de résistances de 10 k pour ne pas déclencher d'interruptions imprévues, cela permet aussi de relier plusieurs sources d'interruptions sur une même ligne (OU CABLE).

• **Les ports d'ENTREES/SORTIES**

Le μC dispose de 5 PORTS (A,B,C,D et E) . Tous les ports d'entrées sorties Input/ Output sont bidirectionnels. La plupart des lignes de PORTs ont une double fonction :

PORT A : (5 bits) I/O pure et/ou convertisseur analogique et/ou TIMER 0. La broche RA4 du PORT A (Entrée du TIMER 0 TOCKI) est du type DRAIN OUVERT.



PORT B (8 bits) I/O pure et/ou programmation in situ ICSP/ICD (Broche RB3/PGM, RB6/PGC et RB7/PGD) et l'entrée d'interruption externe RBO/INT.

- Remarque : Si le PIC est utilisé en mode ICSP/ICD il faut laisser libre les broches RB3/PGM, RB6/PGC ainsi que RB7/PGD) et les configurer en entrée.

PORT C (8 bits) I/O pure et/ou TIMER 1 et/ou SPI / I2C et/ou USART.

PORT D (8 bits) I/O pure et/ou port parallèle 8 bits associé au PORT E.

PORT E (3 bits) I/O pure et/ou pilotage du PORT E RE0/RD, RE1/WR et RE2/CS.

Toutes les lignes des PORTS peuvent fournir un courant de 25mA par ligne de PORT. Une limite de 40mA par PORT doit être respectée pour des questions de dissipation

Configuration des lignes de ports en entrée ou en sortie :

les ports sont pilotés par deux registres :

- Le registre de **PORTx** : si le **PORT x** ou certaines lignes de **PORT X** sont configurées en sortie, ce registre détermine l'état logique des sorties.
- Le registre **TRISx**, c'est le registre de direction. Il détermine si le **PORTx** ou certaines lignes de port sont en entrée ou en sortie. L'écriture d'un 1 logique correspond à une **entrée** (1 comme Input) et l'écriture d'un 0 logique correspond à une sortie (**0** comme Output). Au **RESET** toutes les lignes de ports sont configurées en entrées.

Les fonctions intégrées

• Le convertisseur Analogique/Numérique

Le convertisseur analogique numérique est à approximations successives et il possède une résolution de 10 bits. Il est composé de :

- Un multiplexeur analogique 8 voies
- Un échantillonneur bloqueur.
- Un Convertisseur Analogique Numérique de **10** bits.

Fonctionnement du convertisseur :

La conversion se passe en **2** temps :

- 1 temps le signal à convertir est appliqué sur l'entrée à convertir, ce signal doit être présent au moins pendant le temps **Tacq** (temps d'acquisition environ 20pS pour 5V).
- 2eme temps : la conversion par approximations successives, le temps de conversion minimum est de 12 TAD (TAD c'est le temps de conversion dépendant de l'horloge interne, typiquement 1,6µS).

$$\text{valeur numérisée} = ((V_{IN} - V_{REF-}) / (V_{REF+} - V_{REF-})) * 1023 \quad \text{Si } V_{REF+} = V_{DD} = 5V \text{ et } V_{REF-} = V_{SS} = 0V \text{ alors (valeur numérisée)} = 1023 * (V_{IN}/5)$$

Avant de réaliser une conversion il faut définir la configuration du convertisseur :

- Le nombre d'entrées analogiques.
- Le nombre d'entrées logiques.
- Le type de tension de référence.

• Les TIMERS.

Il existe 3 timers disponibles :

- ☞ Le **timer 0** (8 bits): Il peut être incrémenté par les impulsions extérieures via la broche (TOCKI/ RA4) ou par l'horloge interne ($F_{osc}/4$).
- ☞ Le **timer 1** (16 bits): Il peut être incrémenté soit par l'horloge interne, par des impulsions sur la broche TICKI/RCO ou par un oscillateur (RC ou quartz) connecté sur les broches TOSO/RCO et TIOSI/RCI.
- ☞ Le **timer 2** (8 bits) : Il est incrémenté par l'horloge interne, celle peut être pré divisée.

• La liaison série USART ou SCI

La liaison série **SCI** est une interface série asynchrone de type **START / STOP**.

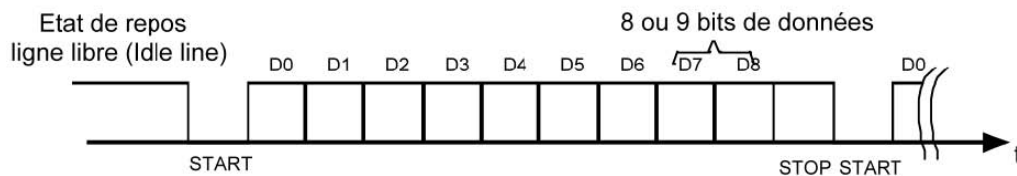
Elle permet d'effectuer des communications avec d'autres systèmes ou objets techniques sur de longues distances (quelques mètres à quelques kilomètres).

Elle dispose des fonctionnalités suivantes:

- ☞ Fonctionnement en Full Duplex, c'est à dire émission et réception de données en même temps. Transmission et réception de données (compatibles avec la norme **RS232** en utilisant une fonction d'adaptation de niveaux).
- ☞ Contrôle des erreurs de transmission et de réception.
- ☞ Transmission sur 8 ou 9 bits.
- ☞ Mode réveil automatique lors de la réception de signaux valides.
- ☞ 4 Sources d'interruptions possibles

Cette interface est composée de 3 fonctions :

- ☞ La transmission.
- ☞ La réception.
- ☞ Le générateur d'horloge (choix de la vitesse de transmission et de réception).



• Les Interruptions.

Le μ C dispose de plusieurs sources d'interruptions.

- ☞ Une interruption externe, action sur la broche INT/RBO.
- ☞ Débordement du TIMERO.
- ☞ Changement d'état logique sur une des broches du PORTB (RB4 à RB7).
- ☞ Une interruption d'un des périphériques (PEIE) :

Au début d'une interruption le sous programme d'interruption doit sauvegarder le contexte et le restituer à la fin, c'est à dire les valeurs des registres **W**, PCLATCH et STATUS. Cela permet au processus interrompu de retrouver ses registres intacts. Pour respecter ce principe il faut ajouter au début du sous programme d'interruption quelques instructions pour sauvegarder les registres **W**, PCLATCH et STATUS.

A la fin du sous programme on ajoute des instructions pour restaurer ces valeurs.

Jeux d'instructions :

Mnemonic, Operands	Description	Cycles	14-Bit Opcode				Status Affected	Notes	
			MSb	LSb					
BYTE-ORIENTED FILE REGISTER OPERATIONS									
ADDWF	f, d	Add W and f	1	00	0111	df ff	ffff	C,DC,Z	1,2
ANDWF	f, d	AND W with f	1	00	0101	df ff	ffff	Z	1,2
CLRF	f	Clear f	1	00	0001	lf ff	ffff	Z	2
CLRW	-	Clear W	1	00	0001	0xxx	xxxx	Z	
COMF	f, d	Complement f	1	00	1001	df ff	ffff	Z	1,2
DECf	f, d	Decrement f	1	00	0011	df ff	ffff	Z	1,2
DECFSZ	f, d	Decrement f, Skip if 0	1(2)	00	1011	df ff	ffff		1,2,3
INCF	f, d	Increment f	1	00	1010	df ff	ffff	Z	1,2
INCFSZ	f, d	Increment f, Skip if 0	1(2)	00	1111	df ff	ffff		1,2,3
IORWF	f, d	Inclusive OR W with f	1	00	0100	df ff	ffff	Z	1,2
MOVF	f, d	Move f	1	00	1000	df ff	ffff	Z	1,2
MOVWF	f	Move W to f	1	00	0000	lf ff	ffff		
NOP	-	No Operation	1	00	0000	0xx0	0000		
RLF	f, d	Rotate Left f through Carry	1	00	1101	df ff	ffff	C	1,2
RRF	f, d	Rotate Right f through Carry	1	00	1100	df ff	ffff	C	1,2
SUBWF	f, d	Subtract W from f	1	00	0010	df ff	ffff	C,DC,Z	1,2
SWAPF	f, d	Swap nibbles in f	1	00	1110	df ff	ffff		1,2
XORWF	f, d	Exclusive OR W with f	1	00	0110	df ff	ffff	Z	1,2
BIT-ORIENTED FILE REGISTER OPERATIONS									
BCF	f, b	Bit Clear f	1	01	00bb	bf ff	ffff		1,2
BSF	f, b	Bit Set f	1	01	01bb	bf ff	ffff		1,2
BTFSC	f, b	Bit Test f, Skip if Clear	1 (2)	01	10bb	bf ff	ffff		3
BTFSS	f, b	Bit Test f, Skip if Set	1 (2)	01	11bb	bf ff	ffff		3
LITERAL AND CONTROL OPERATIONS									
ADDLW	k	Add literal and W	1	11	111x	kk kk	kkkk	C,DC,Z	
ANDLW	k	AND literal with W	1	11	1001	kk kk	kkkk	Z	
CALL	k	Call subroutine	2	10	0kkk	kk kk	kkkk		
CLRWDt	-	Clear Watchdog Timer	1	00	0000	0110	0100	$\overline{TO}, \overline{PD}$	
GOTO	k	Go to address	2	10	1kkk	kk kk	kkkk		
IORLW	k	Inclusive OR literal with W	1	11	1000	kk kk	kkkk	Z	
MOVLW	k	Move literal to W	1	11	00xx	kk kk	kkkk		
RETFIE	-	Return from interrupt	2	00	0000	0000	1001		
RETLW	k	Return with literal in W	2	11	01xx	kk kk	kkkk		
RETURN	-	Return from Subroutine	2	00	0000	0000	1000		
SLEEP	-	Go into standby mode	1	00	0000	0110	0011	$\overline{TO}, \overline{PD}$	
SUBLW	k	Subtract W from literal	1	11	110x	kk kk	kkkk	C,DC,Z	
XORLW	k	Exclusive OR literal with W	1	11	1010	kk kk	kkkk	Z	

- Note 1: When an I/O register is modified as a function of itself (e.g., MOVF PORTB, 1), the value used will be that value present on the pins themselves. For example, if the data latch is '1' for a pin configured as input and is driven low by an external device, the data will be written back with a '0'.
- 2: If this instruction is executed on the TMR0 register (and, where applicable, d = 1), the prescaler will be cleared if assigned to the Timer0 module.
- 3: If Program Counter (PC) is modified, or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a NOP.