

TRAITEMENT PROGRAMMÉ DE L'INFORMATION

- PARTIE 6 -

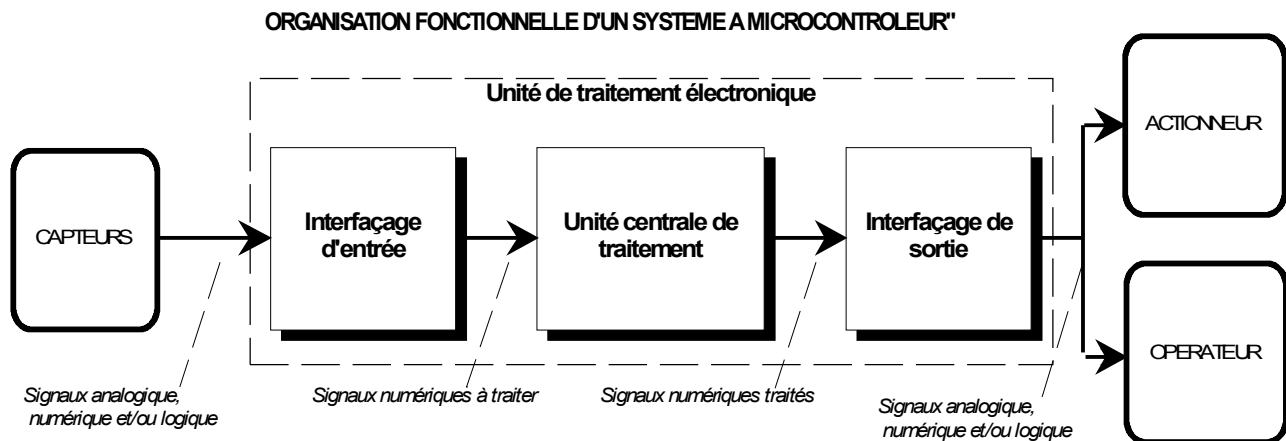
Le microcontrôleur PIC 16F877

SOMMAIRE

1	MISE EN SITUATION.....	3
2	DESCRIPTION ET STRUCTURE INTERNE.....	3
2.1	C.P.U. (Microprocesseur).....	5
2.2	Mémoires programmes.....	5
2.3	Mémoires de données.....	5
2.3.1	Organisation mémoire du PIC16F877.....	6
2.3.2	La mémoire ram des données.....	7
2.3.3	La mémoire FLASH PROGRAMME et la MEMOIRE DONNEES EEPROM.....	8
2.4	L'interface parallèle.....	10
2.4.1	Les Ports du PIC16F877.....	10
2.5	L'interface série.....	15
2.5.1	Les liaisons séries synchrones du PIC16F877.....	16
2.5.2	La liaison SPI.....	18
2.5.3	La liaison I2C en mode slave.....	19
2.5.4	La liaison série asynchrone du PIC16F877.....	20
2.6	Le CAN.....	21
2.6.1	Le Can du PIC 16F877.....	21
2.7	Le timer.....	23
2.8	Le chien de garde.....	30
2.9	Les signaux d'horloge.....	30
2.10	Les Particularités du PIC16F877.....	31
3	MODE DE FONCTIONNEMENT.....	34
3.1	Le fonctionnement en interruptions.....	34
3.2	Instructions et modes d'adressages.....	35
3.3	Les Instructions du PIC16F877.....	36
4	STRUCTURE D'UN PROGRAMME.....	37
5	EXEMPLES DE MICROCONTRÔLEURS.....	37

LE MICROCONTROLEUR PIC 16F877

1 Mise en situation.



Un objet technique, intégrant de l'électronique, fait souvent apparaître des fonctions ayant pour rôle le traitement d'informations : opérations arithmétiques (addition, multiplication...) ou logiques (ET, OU...) entre plusieurs signaux d'entrée permettant de générer des signaux de sortie.

Ces fonctions peuvent être réalisées par des circuits intégrés analogiques ou logiques. Mais, lorsque l'objet technique devient complexe, et qu'il est alors nécessaire de réaliser un ensemble important de traitements d'informations, il devient plus simple de faire appel à une structure à base de microcontrôleur.

2 Description et structure interne.

Un microcontrôleur se présente sous la forme d'un circuit intégré réunissant tous les éléments d'une structure à base de microprocesseur. Voici généralement ce que l'on trouve à l'intérieur d'un tel composant :

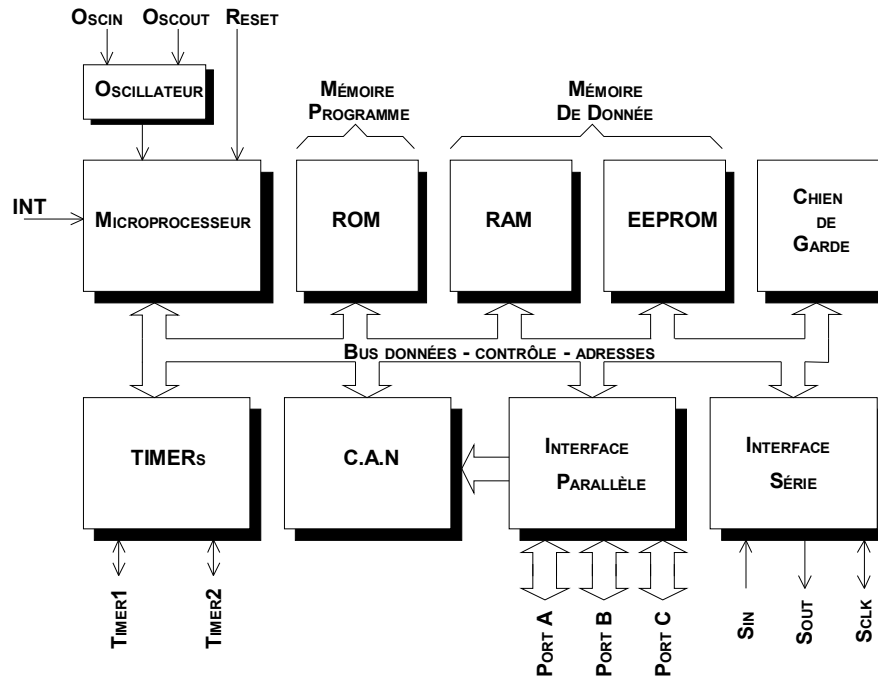
- ♦ Un microprocesseur (C.P.U.),
- ♦ De la mémoire de donnée (RAM et EEPROM),
- ♦ De la mémoire programme (ROM, OTPROM, UVPRM ou EEPROM),
- ♦ Des interfaces parallèles pour la connexion des entrées / sorties,
- ♦ Des interfaces séries (synchrone ou asynchrone) pour le dialogue avec d'autres unités,
- ♦ Des timers pour générer ou mesurer des signaux avec une grande précision temporelle,
- ♦ Des convertisseurs analogique / numérique pour le traitement de signaux analogiques.

➤ **Avantages :**

- Encombrement réduit,
- Circuit imprimé peu complexe,
- Faible consommation,
- Coût réduit.

➤ **Inconvénient :**

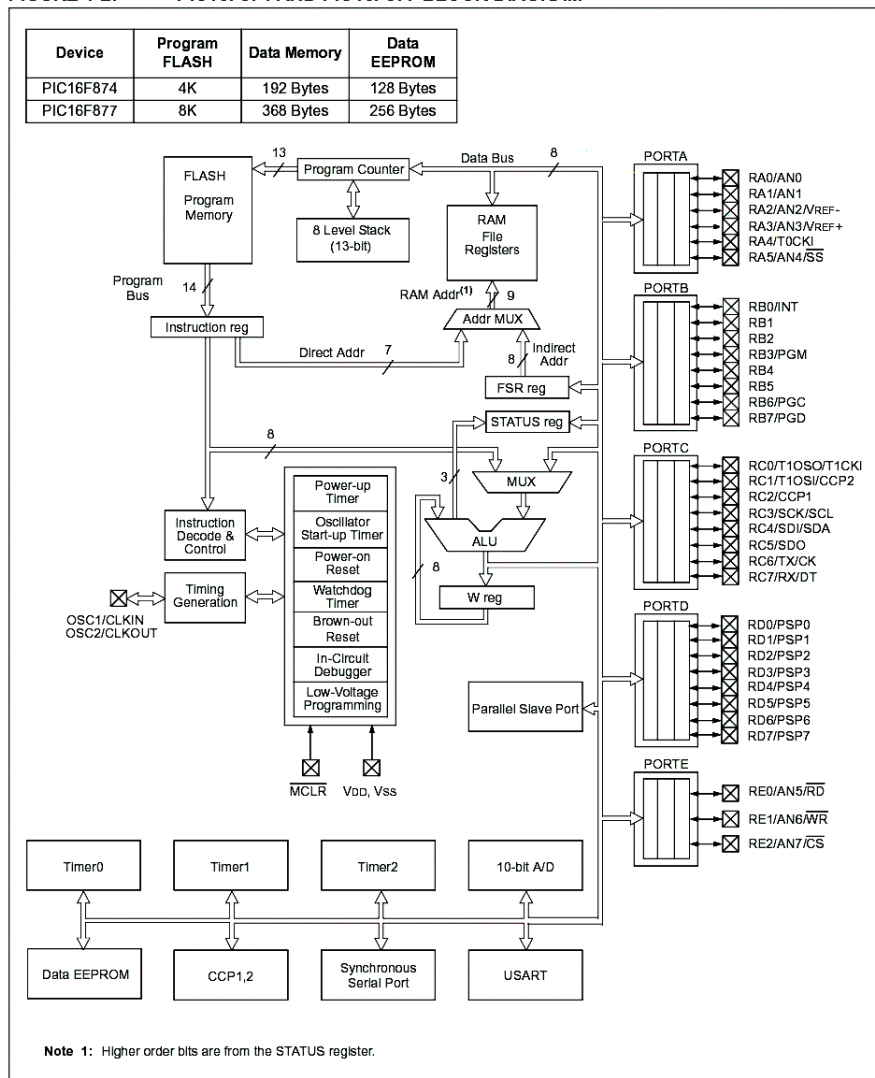
- Système de développement onéreux,
- Programmation nécessitant un matériel adapté.



Le schéma fonctionnel précédent représente une architecture de "Von Neumann" (Commune à la plupart des microprocesseurs) où la mémoire programme partage le même bus que la mémoire de donnée. L'architecture de "Harvard", qui dispose de bus distincts pour les données et pour le programme, est plus rarement utilisée.

Exemple :
l'architecture interne du PIC16F877

FIGURE 1-2: PIC16F874 AND PIC16F877 BLOCK DIAGRAM



2.1 C.P.U. (Microprocesseur).

Un microprocesseur exécute séquentiellement les instructions stockées dans la mémoire programme. Il est capable d'opérer sur des mots binaires dont la taille, en bits, est celle du bus des données (parfois le double pour certains microcontrôleurs). Il est généralement constitué des éléments suivants :

- ♦ Un ou plusieurs registres **accumulateurs** contenant temporairement les opérandes ainsi que les résultats des opérations,
- ♦ Des **registres auxiliaires** permettant de relayer les accumulateurs,
- ♦ Des **registres d'index** pour le mode d'adressage indirect,
- ♦ Un **compteur programme** pointant l'adresse de la prochaine instruction à exécuter, sa taille est celle du bus des adresses,
- ♦ Une unité arithmétique et logique (**ALU**) permettant d'effectuer des opérations entre l'accumulateur et une opérande,
- ♦ Un **registre code condition** indiquant certaines particularités en ce qui concerne le résultat de la dernière opération (retenu, zéro, interruption...).

On peut noter qu'il existe 2 catégories de microprocesseur : les CISC et les RISC.

CISC (Complex Instruction Set Computer) : Ce microprocesseur possède un nombre important d'instructions. Chacune d'elles s'exécute en plusieurs périodes d'horloges.

RISC (Reduced Instruction Set Computer) : Ce microprocesseur possède un nombre réduit d'instructions. Chacune d'elles s'exécute en une période d'horloge. C'est le cas du microcontrôleur PIC16F877 de chez Microchip

2.2 Mémoires programmes.

Ce dispositif contient les instructions du programme que doit exécuter le microprocesseur. Ce type de mémoire (appelée mémoire morte), est uniquement accessible en lecture. Sa programmation nécessite une procédure particulière et un matériel adéquate.

Il en existe différents types selon leur mode de programmation :

- ♦ De la ROM dont le contenu est programmé lors de sa fabrication.
- ♦ De la PROM programmable électriquement une seule fois par le développeur (appelée aussi OTPROM),
- ♦ De la EPROM programmable électriquement et effaçable aux U-V (appelée aussi UVPRM),
- ♦ De la EEPROM programmable et effaçable électriquement.

2.3 Mémoires de données.

Ce dispositif permet de mémoriser temporairement les données générées par le microprocesseur pendant les différentes phases du traitement numérique (résultats d'opérations, états des capteurs...). Ces mémoires sont accessibles en écriture et en lecture.

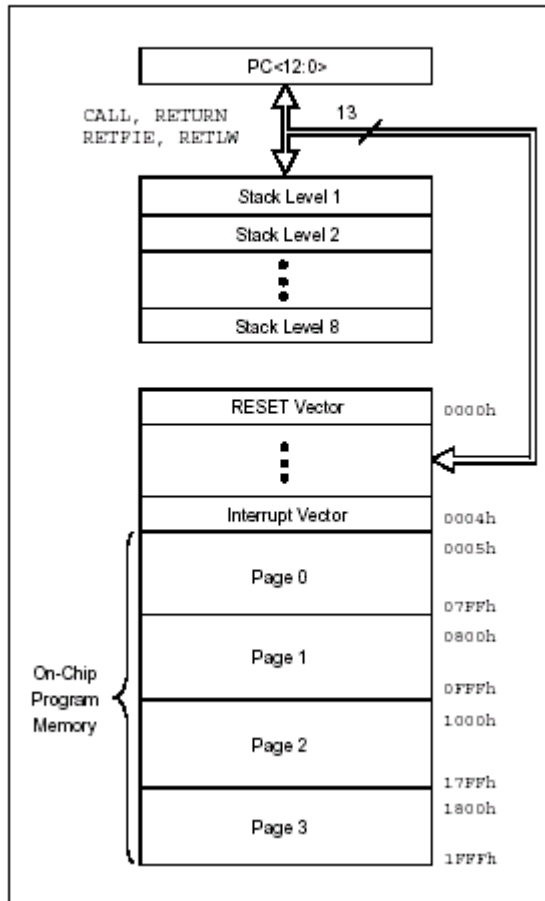
On en trouve 2 types :

- ♦ De la mémoire vive (RAM) volatile (données perdues en cas de coupure de l'alimentation) ayant un temps de lecture et écriture assez court (quelques ns),
- ♦ De la mémoire morte (EEPROM) non-volatile (données conservées en cas de coupure de l'alimentation) ayant un temps d'écriture assez élevé (quelques ms) par rapport au temps de lecture qui est assez faible (quelques ns).

2.3.1 Organisation mémoire du PIC16F877.

Le plan mémoire

FIGURE 2-1: PIC16F877/876 PROGRAM MEMORY MAP AND STACK



Stack : pile dans laquelle est sauvegardé l'état du microprocesseur (PC,W) lors d'appel à sous-programmes ou d'interruptions.

Vecteur Reset : contient l'adresse de début de programme (2 octets)

Interrupt Vecteur : contient l'adresse du début de programme d'interruption quand on utilise les interruptions.

Page : zone de mémoire contenant des registres de configuration, des données variables ou du programme

Le PIC 16F877 possède un PC de 13 bits (PC0-PC12)est permet donc une capacité d'adressage de 8K X 14 de l'adresse 0 à \$1FFF . La partie haute(PC8-PC12) du PC est accessible dans le registre PCLATH.

Il dispose de trois types de mémoire :

- La mémoire flash programme
- La mémoire de données RAM
- La mémoire de données EEPROM

Il est à noter que la mémoire programme et l'Eeprom sont sur les mêmes bus et que leur accès se différencie par le bit EEPGD (bit7) du registreEECON1.

La mémoire de donnée est, quant à elle, indépendante.

2.3.2 La mémoire ram des données

Elle contient les différents registres

FIGURE 2-3: PIC16F877/876 REGISTER FILE MAP

File Address	File Address	File Address	File Address
Indirect addr. ^(*) 00h	Indirect addr. ^(*) 80h	Indirect addr. ^(*) 100h	Indirect addr. ^(*) 180h
TMR0 01h	OPTION_REG 81h	TMR0 101h	OPTION_REG 181h
PCL 02h	PCL 82h	PCL 102h	PCL 182h
STATUS 03h	STATUS 83h	STATUS 103h	STATUS 183h
FSR 04h	FSR 84h	FSR 104h	FSR 184h
PORTA 05h	TRISA 85h		
PORTB 06h	TRISB 86h	PORTB 106h	TRISB 186h
PORTC 07h	TRISC 87h		
PORTD ⁽¹⁾ 08h	TRISD ⁽¹⁾ 88h		
PORTE ⁽¹⁾ 09h	TRISE ⁽¹⁾ 89h		
PCLATH 0Ah	PCLATH 8Ah	PCLATH 10Ah	PCLATH 18Ah
INTCON 0Bh	INTCON 8Bh	INTCON 10Bh	INTCON 18Bh
PIR1 0Ch	PIE1 8Ch	EEDATA 10Ch	EECON1 18Ch
PIR2 0Dh	PIE2 8Dh	EEADR 10Dh	EECON2 18Dh
TMR1L 0Eh	PCON 8Eh	EEDATH 10Eh	Reserved ⁽²⁾ 18Eh
TMR1H 0Fh		EEADRH 10Fh	Reserved ⁽²⁾ 18Fh
T1CON 10h			
TMR2 11h	SSPCON2 91h		
T2CON 12h	PR2 92h		
SSPBUF 13h	SSPADD 93h		
SSPCON 14h	SSPSTAT 94h		
CCPR1L 15h			
CCPR1H 16h			
CCP1CON 17h			
RCSTA 18h	TXSTA 98h	General Purpose Register 16 Bytes 117h-119h	General Purpose Register 16 Bytes 197h-199h
TXREG 19h	SPBRG 99h		
RCREG 1Ah			
CCPR2L 1Bh			
CCPR2H 1Ch			
CCP2CON 1Dh			
ADRESH 1Eh	ADRESL 9Eh		
ADCON0 1Fh	ADCON1 9Fh		
General Purpose Register 96 Bytes 20h-7Fh	General Purpose Register 80 Bytes A0h-EFh	General Purpose Register 80 Bytes 120h-16Fh	General Purpose Register 80 Bytes 1A0h-1EFh
	accesses 70h-7Fh F0h-FFh	accesses 70h-7Fh 170h-17Fh	accesses 70h-7Fh 1F0h-1FFh
Bank 0	Bank 1	Bank 2	Bank 3

■ Unimplemented data memory locations, read as '0'.
 * Not a physical register.

Note 1: These registers are not implemented on the PIC16F876.
Note 2: These registers are reserved, maintain these registers clear.

Avant d'accéder à un de ces registres il convient de s'assurer que la banque de registres est bien correctement choisie.

RP1:RP0	Bank
00	0
01	1
10	2
11	3

En positionnant les bits RP1 et RP0 du status register

STATUS REGISTER (ADDRESS 03h, 83h, 103h, 183h)

R/W-0	R/W-0	R/W-0	R-1	R-1	R/W-x	R/W-x	R/W-x	
IRP	RP1	RP0	\overline{TO}	\overline{PD}	Z	DC	C	
bit 7								bit 0

2.3.3 La mémoire FLASH PROGRAMME et la MEMOIRE DONNEES EEPROM

C'est le registre EECON1 qui permet de définir à laquelle on veut accéder.

EECON1 REGISTER (ADDRESS 18Ch)

R/W-x	U-0	U-0	U-0	R/W-x	R/W-0	R/S-0	R/S-0	
EEPGD	—	—	—	WRERR	WREN	WR	RD	
bit 7								bit 0

Exemple de procédures de lecture, écriture vers EEPROMDATA ou FLASH PROGRAM MEMORY

EXAMPLE 4-1: EEPROM DATA READ

```
BSF STATUS, RP1 ;
BCF STATUS, RP0 ;Bank 2
MOVF ADDR, W ;Write address
MOVWF EBADR ;to read from
BSF STATUS, RP0 ;Bank 3
BCF EECON1, EEPGD ;Point to Data memory
BSF EECON1, RD ;Start read operation
BCF STATUS, RP0 ;Bank 2
MOVF EBDATA, W ;W = EBDATA
```

EXAMPLE 4-2: EEPROM DATA WRITE

```
BSF STATUS, RP1 ;
BSF STATUS, RP0 ;Bank 3
BTFSC EECON1, WR ;Wait for
GOTO $-1 ;write to finish
BCF STATUS, RP0 ;Bank 2
MOVF ADDR, W ;Address to
MOVWF EBADR ;write to
MOVF VALUE, W ;Data to
MOVWF EBDATA ;write
BSF STATUS, RP0 ;Bank 3
BCF EECON1, EEPGD ;Point to Data memory
BSF EECON1, WREN ;Enable writes
;Only disable interrupts
BCF INTCON, GIE ;if already enabled,
;otherwise discard
MOVLW 0x55 ;Write 55h to
MOVWF EECON2 ;EECON2
MOVLW 0xA5 ;Write AAh to
MOVWF EECON2 ;EECON2
BSF EECON1, WR ;Start write operation
;Only enable interrupts
BSF INTCON, GIE ;if using interrupts,
;otherwise discard
BCF EECON1, WREN ;Disable writes
```


EXAMPLE 4-3: FLASH PROGRAM READ

```

BSF STATUS, RP1 ;
BCF STATUS, RP0 ;Bank 2
MOVF ADDRL, W ;Write the
MOVWF EBADR ;address bytes
MOVF ADDRH, W ;for the desired
MOVWF EBADRH ;address to read
BSF STATUS, RP0 ;Bank 3
BSF EBCON1, EEPGD ;Point to Program memory
BSF EBCON1, RD ;Start read operation
NOP ;Required two NOPS
NOP ;
BCF STATUS, RP0 ;Bank 2
MOVF EEDATA, W ;DATAH = EEDATA
MOVWF DATAH ;
MOVF EEDATH, W ;DATAH = EEDATH
MOVWF DATAH ;

```

EXAMPLE 4-4: FLASH PROGRAM WRITE

```

BSF STATUS, RP1 ;
BCF STATUS, RP0 ;Bank 2
MOVF ADDRL, W ;Write address
MOVWF EBADR ;of desired
MOVF ADDRH, W ;program memory
MOVWF EBADRH ;location
MOVF VALUEL, W ;Write value to
MOVWF EBDATA ;program at
MOVF VALUEH, W ;desired memory
MOVWF EEDATH ;location
BSF STATUS, RP0 ;Bank 3
BSF EECON1, EEPGD ;Point to Program memory
BSF EBCON1, WREN ;Enable writes
;Only disable interrupts
BCF INTCON, GIE ;if already enabled,
;otherwise discard
MOVLW 0x55 ;Write 55h to
MOVWF EBCON2 ;EECON2
MOVLW 0xAA ;Write AAh to
MOVWF EBCON2 ;EECON2
BSF EBCON1, WR ;Start write operation
NOP ;Two NOPS to allow micro
NOP ;to setup for write
;Only enable interrupts
BSF INTCON, GIE ;if using interrupts,
;otherwise discard
BCF EBCON1, WREN ;Disable writes

```

2.4 L'interface parallèle.

Ce type d'interface, répartie sur plusieurs ports (maximum 8 bits), permet de prendre en compte des états logiques appliqués en entrée (état de capteurs) ou de générer des signaux binaires en sortie (commande d'actionneurs). Les broches de ces ports peuvent donc être configurées en entrée ou en sortie, avec différentes options (résistances de rappel, sorties collecteurs ouverts, interruption...). La configuration ainsi que l'état logique de ces broches est obtenue par des opérations d'écriture ou de lecture dans différents registres associés à chaque port. On trouve généralement :

- Un registre de direction pour une configuration en entrée ou en sortie,
- Un registre de donnée recopiant les états logiques de chaque broche de port,
- Un registre d'option permettant plusieurs configurations en entrée ou en sortie.

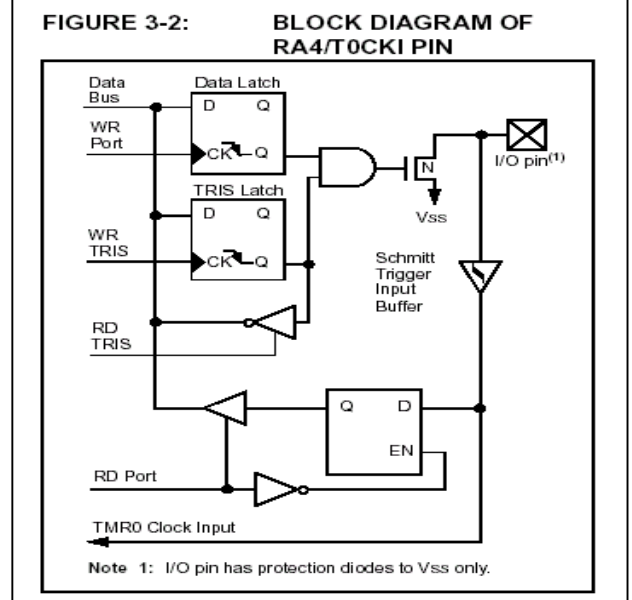
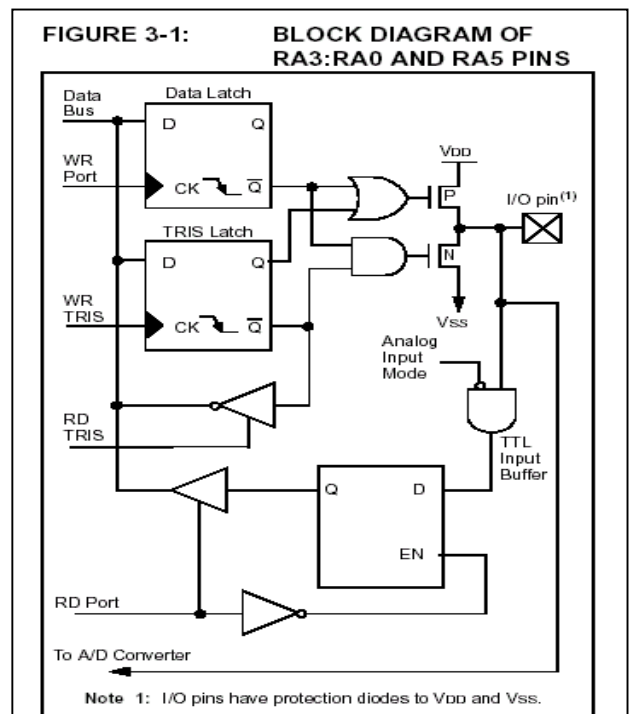
2.4.1 Les Ports du PIC16F877

Certaines lignes de port en plus de pouvoir être configurées en entrée ou en sortie, ont la particularité de pouvoir remplir d'autres fonctions comme sortie PWM, ligne usart, ligne I2C, lignes SPI...

➤ Le PORT A

```

EXAMPLE 3-1:  INITIALIZING PORTA
BCF    STATUS, RP0 ;
BCF    STATUS, RP1 ; Bank0
CLRF   PORTA      ; Initialize PORTA by
                  ; clearing output
                  ; data latches
BSF    STATUS, RP0 ; Select Bank 1
MOVLW  0x06      ; Configure all pins
MOVWF  ADCON1    ; as digital inputs
MOVLW  0x0CF     ; Value used to
                  ; initialize data
                  ; direction
MOVWF  TRISA     ; Set RA<3:0> as inputs
                  ; RA<5:4> as outputs
                  ; TRISA<7:6>are always
                  ; read as '0'.
    
```



Les registres de configuration du port A

TABLE 3-1: PORTA FUNCTIONS

Name	Bit#	Buffer	Function
RA0/AN0	bit0	TTL	Input/output or analog input.
RA1/AN1	bit1	TTL	Input/output or analog input.
RA2/AN2	bit2	TTL	Input/output or analog input.
RA3/AN3/VREF	bit3	TTL	Input/output or analog input or VREF.
RA4/T0CKI	bit4	ST	Input/output or external clock input for Timer0. Output is open drain type.
RA5/SS/AN4	bit5	TTL	Input/output or slave select input for synchronous serial port or analog input.

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

TABLE 3-2: SUMMARY OF REGISTERS ASSOCIATED WITH PORTA

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other RESETS
05h	PORTA	—	—	RA5	RA4	RA3	RA2	RA1	RA0	--0x 0000	--0u 0000
85h	TRISA	—	—	PORTA Data Direction Register						--11 1111	--11 1111
9Fh	ADCON1	ADFM	—	—	—	PCFG3	PCFG2	PCFG1	PCFG0	--0- 0000	--0- 0000

Legend: x = unknown, u = unchanged, - = unimplemented locations read as '0'.
Shaded cells are not used by PORTA.

➤ Le PORT B

TABLE 3-3: PORTB FUNCTIONS

Name	Bit#	Buffer	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/PGM ⁽³⁾	bit3	TTL	Input/output pin or programming pin in LVP mode. 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/PGC	bit6	TTL/ST ⁽²⁾	Input/output pin (with interrupt-on-change) or In-Circuit Debugger pin. Internal software programmable weak pull-up. Serial programming clock.
RB7/PGD	bit7	TTL/ST ⁽²⁾	Input/output pin (with interrupt-on-change) or In-Circuit Debugger pin. Internal software programmable weak pull-up. Serial programming data.

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

Note 1: This buffer is a Schmitt Trigger input when configured as the external interrupt.

2: This buffer is a Schmitt Trigger input when used in Serial Programming mode.

3: Low Voltage ICSP Programming (LVP) is enabled by default, which disables the RB3 I/O function. LVP must be disabled to enable RB3 as an I/O pin and allow maximum compatibility to the other 28-pin and 40-pin mid-range devices.

TABLE 3-4: SUMMARY OF REGISTERS ASSOCIATED WITH PORTB

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other RESETS
06h, 106h	PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	xxxx xxxx	uuuu uuuu
86h, 186h	TRISB	PORTB Data Direction Register								1111 1111	1111 1111
81h, 181h	OPTION_REG	RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111

Legend: x = unknown, u = unchanged. Shaded cells are not used by PORTB.

➤ **Le PORT C**

TABLE 3-5: PORTC FUNCTIONS

Name	Bit#	Buffer Type	Function
RC0/T1OSO/T1CKI	bit0	ST	Input/output port pin or Timer1 oscillator output/Timer1 clock input.
RC1/T1OSI/CCP2	bit1	ST	Input/output port pin or Timer1 oscillator input or Capture2 input/ Compare2 output/PWM2 output.
RC2/CCP1	bit2	ST	Input/output port pin or Capture1 input/Compare1 output/ PWM1 output.
RC3/SCK/SCL	bit3	ST	RC3 can also be the synchronous serial clock for both SPI and I ² C modes.
RC4/SDI/SDA	bit4	ST	RC4 can also be the SPI Data In (SPI mode) or data I/O (I ² C mode).
RC5/SDO	bit5	ST	Input/output port pin or Synchronous Serial Port data output.
RC6/TX/CK	bit6	ST	Input/output port pin or USART Asynchronous Transmit or Synchronous Clock.
RC7/RX/DT	bit7	ST	Input/output port pin or USART Asynchronous Receive or Synchronous Data.

Legend: ST = Schmitt Trigger input

TABLE 3-6: SUMMARY OF REGISTERS ASSOCIATED WITH PORTC

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other RESETS
07h	PORTC	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	xxxx xxxx	uuuu uuuu
87h	TRISC	PORTC Data Direction Register								1111 1111	1111 1111

Legend: x = unknown, u = unchanged

➤ **Le PORT D**

TABLE 3-7: PORTD FUNCTIONS

Name	Bit#	Buffer Type	Function
RD0/PSP0	bit0	ST/TTL ⁽¹⁾	Input/output port pin or parallel slave port bit0.
RD1/PSP1	bit1	ST/TTL ⁽¹⁾	Input/output port pin or parallel slave port bit1.
RD2/PSP2	bit2	ST/TTL ⁽¹⁾	Input/output port pin or parallel slave port bit2.
RD3/PSP3	bit3	ST/TTL ⁽¹⁾	Input/output port pin or parallel slave port bit3.
RD4/PSP4	bit4	ST/TTL ⁽¹⁾	Input/output port pin or parallel slave port bit4.
RD5/PSP5	bit5	ST/TTL ⁽¹⁾	Input/output port pin or parallel slave port bit5.
RD6/PSP6	bit6	ST/TTL ⁽¹⁾	Input/output port pin or parallel slave port bit6.
RD7/PSP7	bit7	ST/TTL ⁽¹⁾	Input/output port pin or parallel slave port bit7.

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

Note 1: Input buffers are Schmitt Triggers when in I/O mode and TTL buffers when in Parallel Slave Port mode.

TABLE 3-8: SUMMARY OF REGISTERS ASSOCIATED WITH PORTD

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other RESETS
08h	PORTD	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0	xxxx xxxx	uuuu uuuu
88h	TRISD	PORTD Data Direction Register								1111 1111	1111 1111
89h	TRISE	IBF	OBF	IBOV	PSPMODE	—	PORTE Data Direction Bits			0000 -111	0000 -111

Legend: x = unknown, u = unchanged, - = unimplemented, read as '0'. Shaded cells are not used by PORTD.

➤ **Le PORT E**

TABLE 3-9: PORTE FUNCTIONS

Name	Bit#	Buffer Type	Function
RE0/ \overline{RD} /AN5	bit0	ST/TTL ⁽¹⁾	I/O port pin or read control input in Parallel Slave Port mode or analog input: \overline{RD} 1 = Idle 0 = Read operation. Contents of PORTD register are output to PORTD I/O pins (if chip selected)
RE1/ \overline{WR} /AN6	bit1	ST/TTL ⁽¹⁾	I/O port pin or write control input in Parallel Slave Port mode or analog input: \overline{WR} 1 = Idle 0 = Write operation. Value of PORTD I/O pins is latched into PORTD register (if chip selected)
RE2/ \overline{CS} /AN7	bit2	ST/TTL ⁽¹⁾	I/O port pin or chip select control input in Parallel Slave Port mode or analog input: \overline{CS} 1 = Device is not selected 0 = Device is selected

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

Note 1: Input buffers are Schmitt Triggers when in I/O mode and TTL buffers when in Parallel Slave Port mode.

TABLE 3-10: SUMMARY OF REGISTERS ASSOCIATED WITH PORTE

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other RESETS
09h	PORTE	—	—	—	—	—	RE2	RE1	RE0	---- -xxx	---- -uuu
89h	TRISE	IBF	OBF	IBOV	PSPMODE	—	PORTE Data Direction Bits			0000 -111	0000 -111
9Fh	ADCON1	ADFM	—	—	—	PCFG3	PCFG2	PCFG1	PCFG0	--0- 0000	--0- 0000

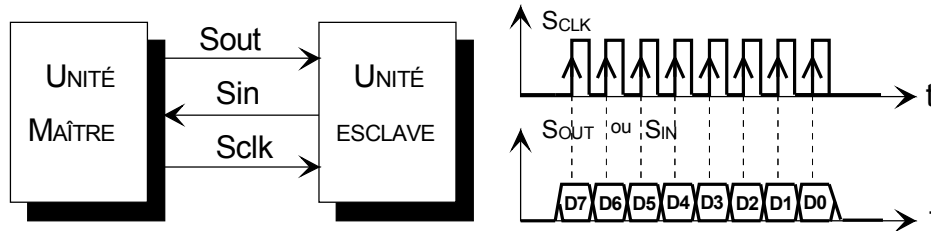
Legend: x = unknown, u = unchanged, - = unimplemented, read as '0'. Shaded cells are not used by PORTE.

2.5 L'interface série.

Ce type d'interface permet au microcontrôleur de communiquer avec d'autres systèmes à base de microprocesseur. Les données envoyées ou reçues se présentent sous la forme d'une succession temporelle (sur un seul bit) de valeurs binaires images d'un mot. Il y a 2 types de liaison série : synchrone et asynchrone.

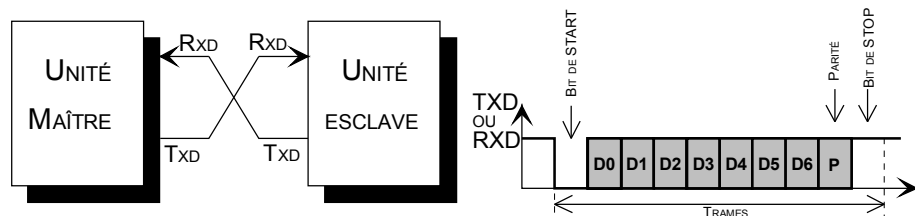
➤ Liaison série synchrone.

Dans ce dispositif la transmission est synchronisée par un signal d'horloge émis par l'unité maître.



➤ Liaison série asynchrone.

Ce dispositif ne possède pas de signal d'horloge de synchronisation. Les unités en liaison possèdent chacune une horloge interne cadencée à la même fréquence. Lorsqu'une unité veut émettre un mot binaire, elle génère un front descendant sur sa ligne émettrice. A la fin de l'émission de ce mot, la ligne repasse au niveau haut. La donnée à transmettre peut contenir un bit supplémentaire appelé "parité" et servant à la correction d'erreurs.



PARAMETRES Rentrant EN JEU POUR LA NORME RS232 :

- **Longueur des mots** : 7 bits (ex : caractère ascii) ou 8 bits
- **La vitesse de transmission** : elle est défini en bits par seconde ou bauds. Elle peut prendre des valeurs allant de 110 à 115 200 bds.
- **Parité** : le mot transmis peut être suivi ou non d'un bit de parité qui sert à détecter les erreurs éventuelles de transmission.
- **Bit de start** : la ligne au repos est à l'état logique 1 pour indiquer qu'un mot va être transmis la ligne passe à l'état bas avant de commencer le transfert. Ce bit permet de synchroniser l'horloge du récepteur.
- **Bit de stop** : après la transmission, la ligne est positionnée au repos pendant 1, 2 ou 1,5 périodes d'horloge selon le nombre de bits de stop.
- **Niveau de tension** : Un "0" logique est matérialisé par une tension comprise entre 3 et 25V, un "1" par une tension comprise entre -25 et -3 V. Des circuits spécialisés comme le MAX 232 réalise la conversion à partir de niveau TTL.

2.5.1 Les liaisons séries synchrones du PIC16F877

Le PIC16F877 permet par son module **MASTER SYNCHRONOUS SERIAL PORT** De communiquer suivant deux protocoles :

- Serial Peripheral Interface (SPI)
- Inter-Integrated Circuit (I2C)

Pour chacun de ces protocoles existent deux modes, le mode MASTER et le mode SLAVE.. En mode Master c'est le microcontrôleur qui gère la ligne SCK, alors qu'en mode SLAVE, cette horloge est piloté par un maître externe et de ce fait le microcontrôleur ne pilote pas la transmission mais la subit.

Ces liaison se configurent par les registres suivants

REGISTER 9-2: SSPCON: SYNC SERIAL PORT CONTROL REGISTER (ADDRESS 14h)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0
bit 7							bit 0

bit 7 **WCOL: Write Collision Detect bit**
Master mode:
 1 = A write to SSPBUF was attempted while the I2C conditions were not valid
 0 = No collision
Slave mode:
 1 = SSPBUF register is written while still transmitting the previous word (must be cleared in software)
 0 = No collision

bit 6 **SSPOV: Receive Overflow Indicator bit**
In SPI mode:
 1 = A new byte is received while SSPBUF holds previous data. Data in SSPSR is lost on overflow. In Slave mode, the user must read the SSPBUF, even if only transmitting data, to avoid overflows. In Master mode, the overflow bit is not set, since each operation is initiated by writing to the SSPBUF register. (Must be cleared in software.)
 0 = No overflow
In I²C mode:
 1 = A byte is received while the SSPBUF is holding the previous byte. SSPOV is a "don't care" in Transmit mode. (Must be cleared in software.)
 0 = No overflow

bit 5 **SSPEN: Synchronous Serial Port Enable bit**
In SPI mode:
 When enabled, these pins must be properly configured as input or output
 1 = Enables serial port and configures SCK, SDO, SDI, and SS as the source of the serial port pins
 0 = Disables serial port and configures these pins as I/O port pins
In I²C mode:
 When enabled, these pins must be properly configured as input or output
 1 = Enables the serial port and configures the SDA and SCL pins as the source of the serial port pins
 0 = Disables serial port and configures these pins as I/O port pins

bit 4 **CKP: Clock Polarity Select bit**
In SPI mode:
 1 = Idle state for clock is a high level
 0 = Idle state for clock is a low level
In I²C Slave mode:
 SCK release control
 1 = Enable clock
 0 = Holds clock low (clock stretch). (Used to ensure data setup time.)
In I²C Master mode:
 Unused in this mode

bit 3-0 **SSPM3:SSPM0: Synchronous Serial Port Mode Select bits**
 0000 = SPI Master mode, clock = Fosc/4
 0001 = SPI Master mode, clock = Fosc/16
 0010 = SPI Master mode, clock = Fosc/64
 0011 = SPI Master mode, clock = TMR2 output/2
 0100 = SPI Slave mode, clock = SCK pin. SS pin control enabled.
 0101 = SPI Slave mode, clock = SCK pin. SS pin control disabled. SS can be used as I/O pin.
 0110 = I²C Slave mode, 7-bit address
 0111 = I²C Slave mode, 10-bit address
 1000 = I²C Master mode, clock = Fosc / (4 * (SSPADD+1))
 1011 = I²C Firmware Controlled Master mode (slave idle)
 1110 = I²C Firmware Controlled Master mode, 7-bit address with START and STOP bit. Interrupts enabled
 1111 = I²C Firmware Controlled Master mode, 10-bit address with START and STOP bit. Interrupts enabled
 1001, 1010, 1100, 1101 = Reserved

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
- n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

REGISTER 9-3: SSPCON2: SYNC SERIAL PORT CONTROL REGISTER2 (ADDRESS 91h)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN
bit 7							bit 0

- bit 7 GCEN: General Call Enable bit (In I²C Slave mode only)
1 = Enable interrupt when a general call address (0000h) is received in the SSPSR
0 = General call address disabled
- bit 6 ACKSTAT: Acknowledge Status bit (In I²C Master mode only)
In Master Transmit mode:
1 = Acknowledge was not received from slave
0 = Acknowledge was received from slave
- bit 5 ACKDT: Acknowledge Data bit (In I²C Master mode only)
In Master Receive mode:
Value that will be transmitted when the user initiates an Acknowledge sequence at the end of a receive.
1 = Not Acknowledge
0 = Acknowledge
- bit 4 ACKEN: Acknowledge Sequence Enable bit (In I²C Master mode only)
In Master Receive mode:
1 = Initiate Acknowledge sequence on SDA and SCL pins and transmit ACKDT data bit.
Automatically cleared by hardware.
0 = Acknowledge sequence idle
- bit 3 RCEN: Receive Enable bit (In I²C Master mode only)
1 = Enables Receive mode for I²C
0 = Receive idle
- bit 2 PEN: STOP Condition Enable bit (In I²C Master mode only)
SCK Release Control:
1 = Initiate STOP condition on SDA and SCL pins. Automatically cleared by hardware.
0 = STOP condition idle
- bit 1 RSEN: Repeated START Condition Enable bit (In I²C Master mode only)
1 = Initiate Repeated START condition on SDA and SCL pins. Automatically cleared by hardware.
0 = Repeated START condition idle
- bit 0 SEN: START Condition Enable bit (In I²C Master mode only)
1 = Initiate START condition on SDA and SCL pins. Automatically cleared by hardware.
0 = START condition idle

Note: For bits ACKEN, RCEN, PEN, RSEN, SEN: If the I²C module is not in the IDLE mode, this bit may not be set (no spooling), and the SSPBUF may not be written (or writes to the SSPBUF are disabled).

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
- n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

2.5.2 La liaison SPI

FIGURE 9-1: MSSP BLOCK DIAGRAM (SPI MODE)

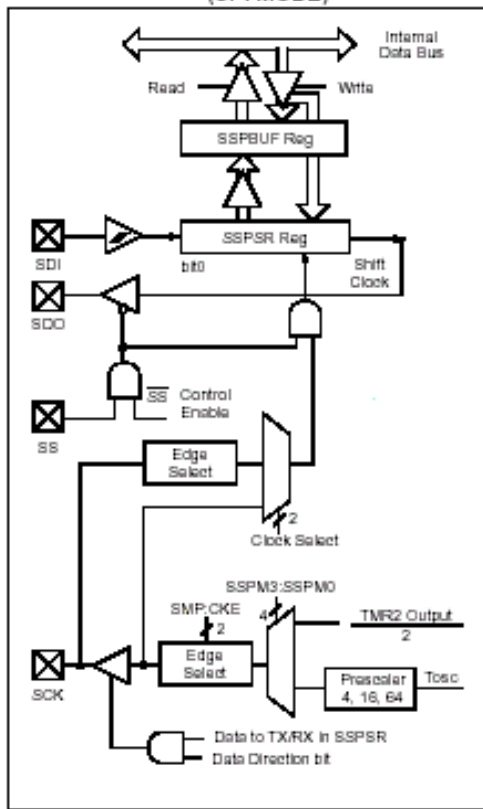


FIGURE 9-2: SPI MODE TIMING, MASTER MODE

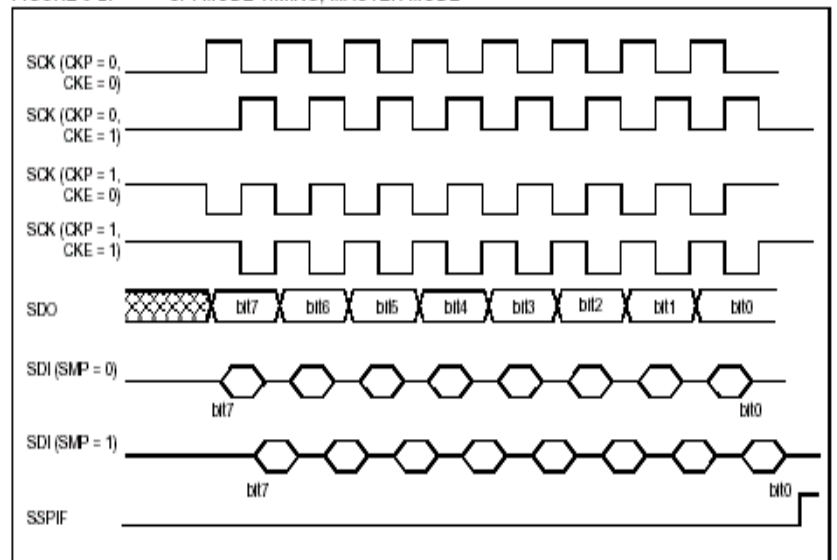


FIGURE 9-3: SPI MODE TIMING (SLAVE MODE WITH CKE = 0)

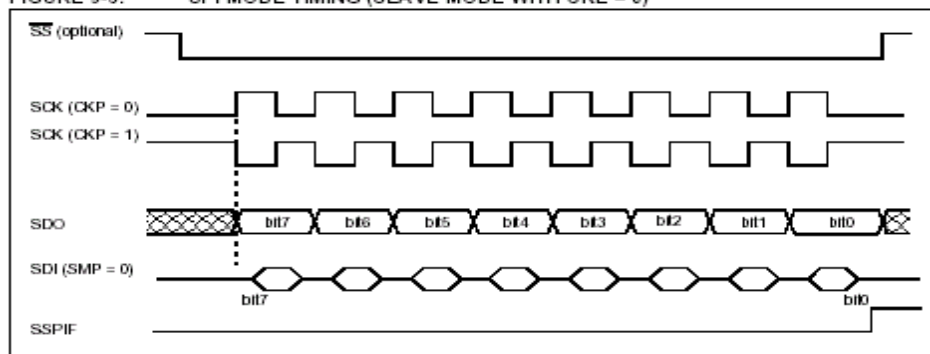
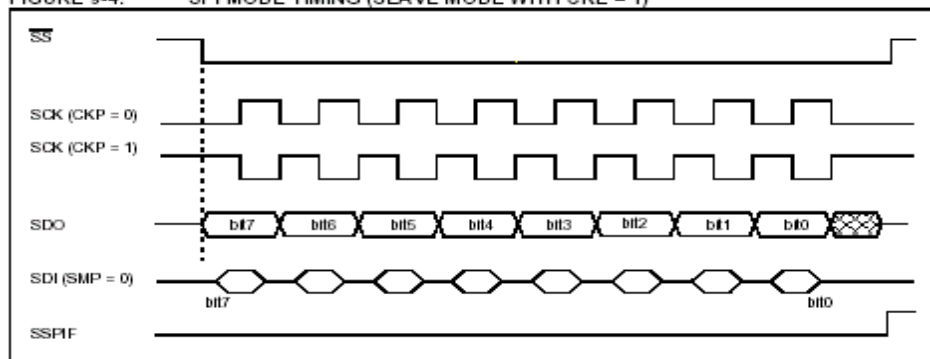


FIGURE 9-4: SPI MODE TIMING (SLAVE MODE WITH CKE = 1)



2.5.4 La liaison série asynchrone du PIC16F877

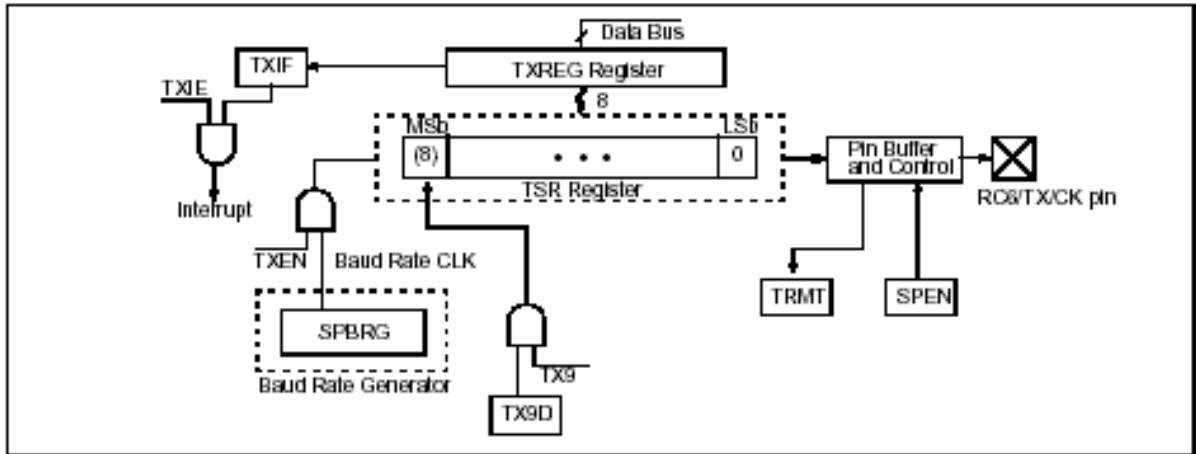
Cette liaison peut aussi être configurée en mode synchrone

**ASYNCHRONOUS RECEIVER
TRANSMITTER (USART)**

- Synchronous - Master (half duplex)

En émission

FIGURE 10-1: USART TRANSMIT BLOCK DIAGRAM



En réception

FIGURE 10-4: USART RECEIVE BLOCK DIAGRAM

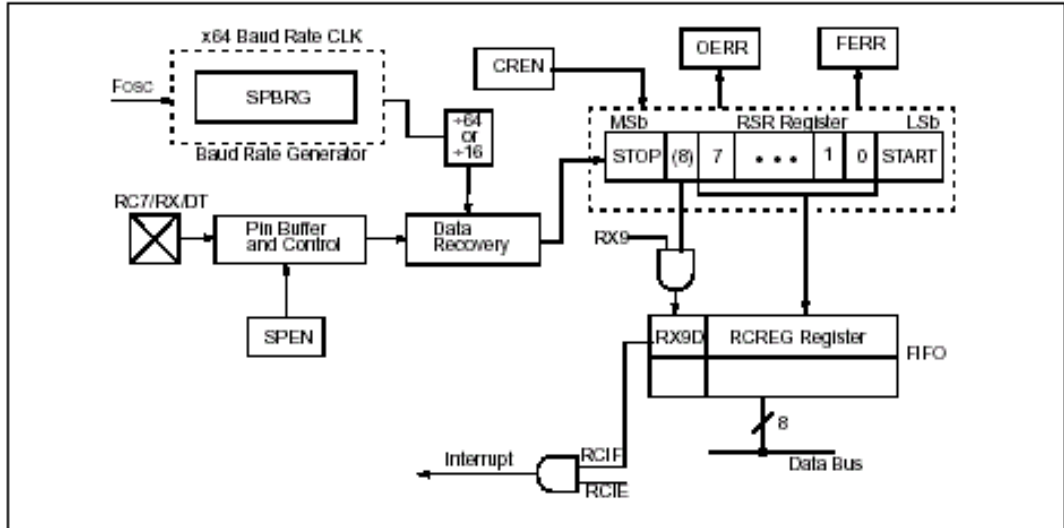
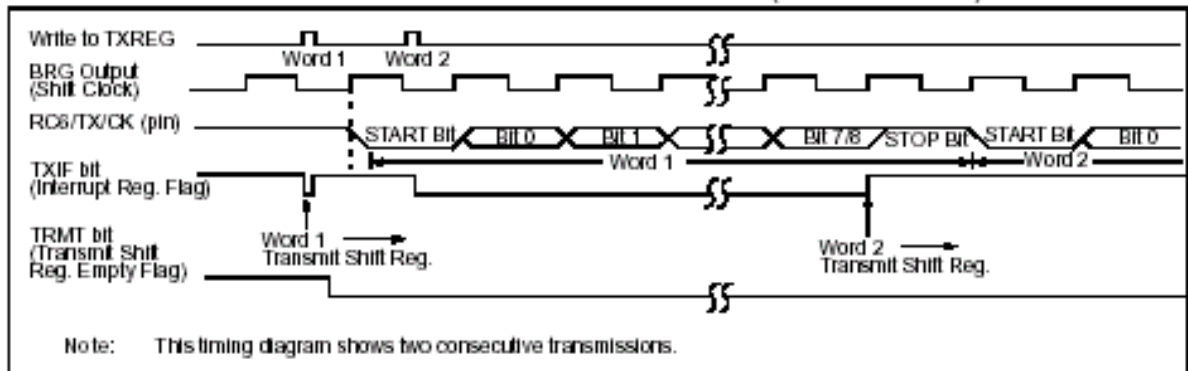


FIGURE 10-3: ASYNCHRONOUS MASTER TRANSMISSION (BACK TO BACK)



2.6 Le CAN.

Le CAN intégré dans les microcontrôleurs est généralement du type "Approximations successives". Il possède plusieurs entrées multiplexées accessibles via les broches des ports de l'interface parallèle. Le CAN possède normalement 2 registres :

- ♦ Un registre de données contenant le résultat de la conversion,
- ♦ Un registre de contrôle permettant de lancer et de surveiller la conversion.

2.6.1 Le Can du PIC 16F877

Le can interne du PIC 16F877 permet de convertir une tension d'entrée analogique en combinaison numérique sur 10 bits contenue dans les registres ADRESH et ADRESL.

Les registres suivants permettent de configurer et de lancer la conversion en fixant notamment Vref utilisée pour la conversion.

FIGURE 11-1: A/D BLOCK DIAGRAM

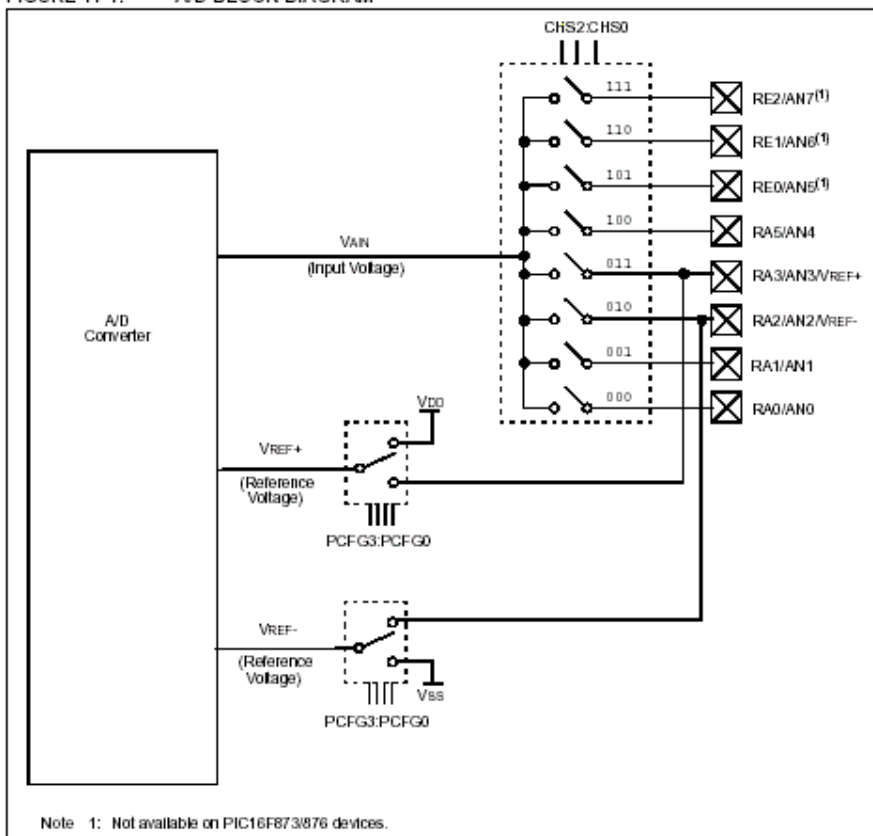
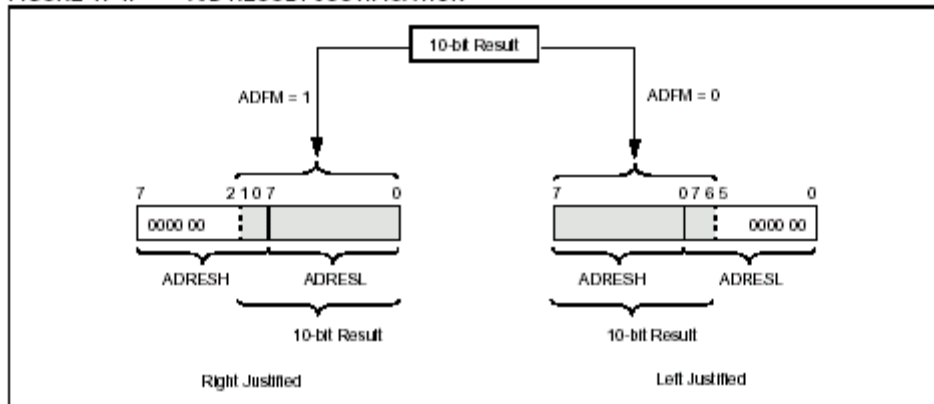


FIGURE 11-4: A/D RESULT JUSTIFICATION



REGISTER 11-1: ADCON0 REGISTER (ADDRESS: 1Fh)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0
ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/DONE	—	ADON
bit 7						bit 0	

- bit 7-6 **ADCS1:ADCS0: A/D Conversion Clock Select bits**
 00 = Fosc/2
 01 = Fosc/8
 10 = Fosc/32
 11 = Frc (clock derived from the internal A/D module RC oscillator)
- bit 5-3 **CHS2:CHS0: Analog Channel Select bits**
 000 = channel 0, (RA0/AN0)
 001 = channel 1, (RA1/AN1)
 010 = channel 2, (RA2/AN2)
 011 = channel 3, (RA3/AN3)
 100 = channel 4, (RA5/AN4)
 101 = channel 5, (RE0/AN5)⁽¹⁾
 110 = channel 6, (RE1/AN6)⁽¹⁾
 111 = channel 7, (RE2/AN7)⁽¹⁾
- bit 2 **GO/DONE: A/D Conversion Status bit**
IF ADON = 1:
 1 = A/D conversion in progress (setting this bit starts the A/D conversion)
 0 = A/D conversion not in progress (this bit is automatically cleared by hardware when the A/D conversion is complete)
- bit 1 **Unimplemented: Read as '0'**
- bit 0 **ADON: A/D On bit**
 1 = A/D converter module is operating
 0 = A/D converter module is shut-off and consumes no operating current

Note 1: These channels are not available on PIC16F873/876 devices.

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
- n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

REGISTER 11-2: ADCON1 REGISTER (ADDRESS 9Fh)

U-0	U-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	—	—	—	PCFG3	PCFG2	PCFG1	PCFG0
bit 7				bit 0			

- bit 7 **ADFM: A/D Result Format Select bit**
 1 = Right justified. 6 Most Significant bits of ADRESH are read as '0'.
 0 = Left justified. 6 Least Significant bits of ADRESL are read as '0'.
- bit 6-4 **Unimplemented: Read as '0'**
- bit 3-0 **PCFG3:PCFG0: A/D Port Configuration Control bits:**

PCFG3: PCFG0	AN7 ⁽¹⁾ RE2	AN6 ⁽¹⁾ RE1	AN5 ⁽¹⁾ RE0	AN4 RA5	AN3 RA3	AN2 RA2	AN1 RA1	AN0 RA0	VREF+	VREF-	CHAN Refs ⁽²⁾
0000	A	A	A	A	A	A	A	A	VDD	VSS	8/0
0001	A	A	A	A	VREF+	A	A	A	RA3	VSS	7/1
0010	D	D	D	A	A	A	A	A	VDD	VSS	5/0
0011	D	D	D	A	VREF+	A	A	A	RA3	VSS	4/1
0100	D	D	D	D	A	D	A	A	VDD	VSS	3/0
0101	D	D	D	D	VREF+	D	A	A	RA3	VSS	2/1
011x	D	D	D	D	D	D	D	D	VDD	VSS	0/0
1000	A	A	A	A	VREF+	VREF-	A	A	RA3	RA2	6/2
1001	D	D	A	A	A	A	A	A	VDD	VSS	6/0
1010	D	D	A	A	VREF+	A	A	A	RA3	VSS	5/1
1011	D	D	A	A	VREF+	VREF-	A	A	RA3	RA2	4/2
1100	D	D	D	A	VREF+	VREF-	A	A	RA3	RA2	3/2
1101	D	D	D	D	VREF+	VREF-	A	A	RA3	RA2	2/2
1110	D	D	D	D	D	D	D	A	VDD	VSS	1/0
1111	D	D	D	D	VREF+	VREF-	D	A	RA3	RA2	1/2

A = Analog Input D = Digital I/O

- Note 1: These channels are not available on PIC16F873/876 devices.
- Note 2: This column indicates the number of analog channels available as A/D inputs and the number of analog channels used as voltage reference inputs.

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
- n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

2.7 Le timer.

Le Timer permettent de réaliser les fonctions suivantes :

- ♦ Génération d'un signal périodique modulé ou non en largeur d'impulsion,
- ♦ Génération d'une impulsion calibrée,
- ♦ Temporisation,
- ♦ Comptage d'événements.

Plusieurs registres associés au Timer permettent de configurer les différents modes décrits précédemment.

➤ Les timers du PIC16F877.

Le Pic16F877 comporte trois timers qui sont :

- * le timer 0 qui est un compteur sur 8 bits dont on peut choisir le front de déclenchement, le rapport de division et l'horloge interne ou externe

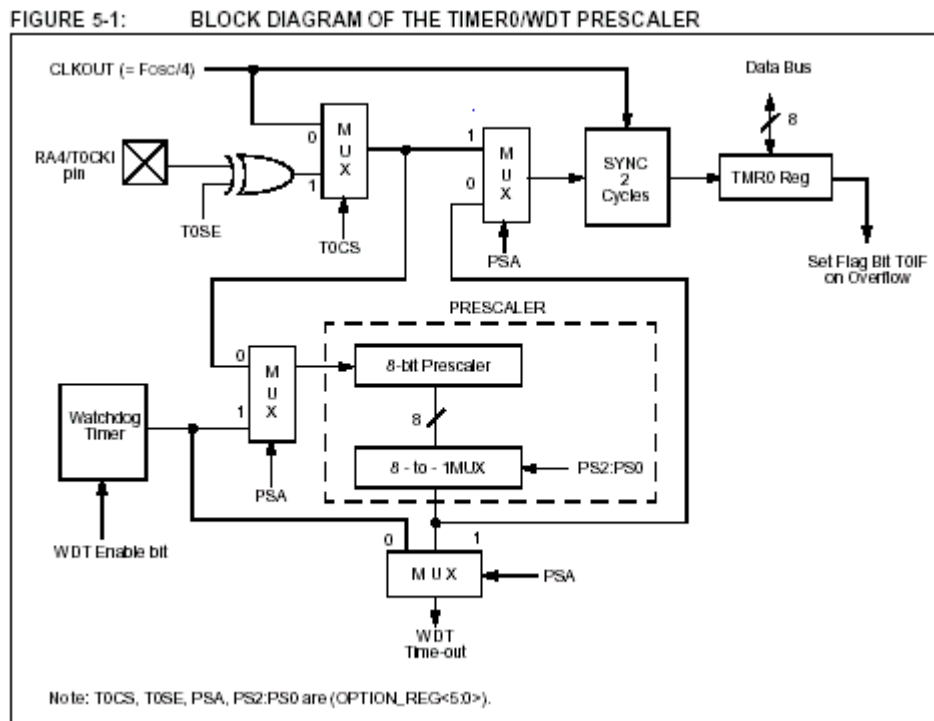


TABLE 5-1: REGISTERS ASSOCIATED WITH TIMER0

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other RESETS
01h,101h	TMR0	Timer0 Module's Register								XXXX XXXX	UUUU UUUU
0Bh,8Bh, 10Bh,18Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIF	T0IF	INTF	RBIF	0000 000x	0000 000u
81h,181h	OPTION_REG	RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111

Legend: x = unknown, u = unchanged, - = unimplemented locations read as '0'.
Shaded cells are not used by Timer0.

- le timer 1 qui est un compteur sur 16 bits dont on peut choisir le fonctionnement en mode comptage ou en mode temporisateur.

FIGURE 6-2: **TIMER1 BLOCK DIAGRAM**

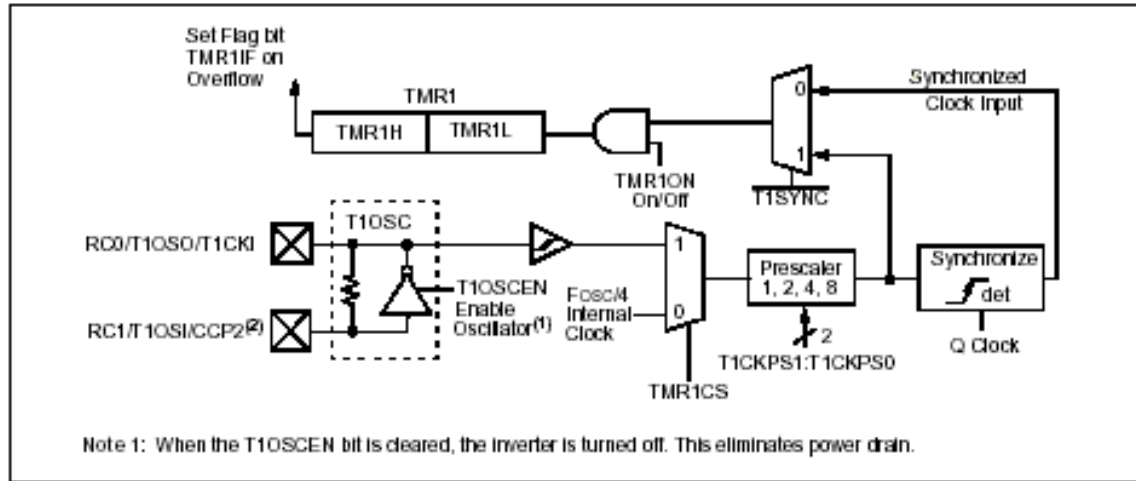


TABLE 6-2: **REGISTERS ASSOCIATED WITH TIMER1 AS A TIMER/COUNTER**

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other RESETS
0Bh, 8Bh, 10Bh, 18Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000x
0Ch	PIR1	PSPIF ⁽¹⁾	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
8Ch	PIE1	PSPIE ⁽¹⁾	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
0Eh	TMR1L	Holding Register for the Least Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	xxxx xxxx
0Fh	TMR1H	Holding Register for the Most Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	xxxx xxxx
10h	T1CON	—	—	T1CKPS1	T1CKPS0	T10SCEN	T1SYNC	TMR1CS	TMR1ON	--00 0000	--1xx xxxxx

Legend: x = unknown, u = unchanged, - = unimplemented, read as '0'. Shaded cells are not used by the Timer1 module.
Note 1: Bits PSPIE and PSPIF are reserved on the PIC16F873/876; always maintain these bits clear.

- le timer 2 qui est un compteur sur 8 bits

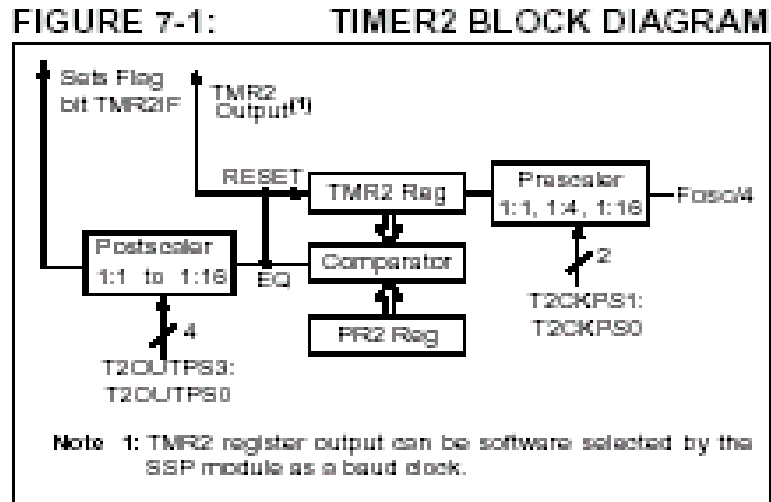


TABLE 7-1: REGISTERS ASSOCIATED WITH TIMER2 AS A TIMER/COUNTER

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other RESETS
0Bh, 8Bh, 10Bh, 18Bh	INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF ⁽¹⁾	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
8Ch	PIE1	PSPIE ⁽¹⁾	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
11h	TMR2	Timer2 Module's Register								0000 0000	0000 0000
12h	T2CON	—	TCOUTPS3	TCOUTPS2	TCOUTPS1	TCOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	-000 0000
92h	PR2	Timer2 Period Register								1111 1111	1111 1111

Legend: x = unknown, u = unchanged, - = unimplemented, read as '0'. Shaded cells are not used by the Timer2 module.
Note 1: Bits PSPIE and PSPIF are reserved on the PIC16F873/876; always maintain these bits clear.

• **Le module capture comparaison CCP1/2**

qui permet entre autre associé au timer 2 de générer des signaux PWM ou MLI (modulation en largeur d'impulsion).

TABLE 8-1: CCP MODE - TIMER RESOURCES REQUIRED

CCP Mode	Timer Resource
Capture	Timer1
Compare	Timer1
PWM	Timer2

TABLE 8-2: INTERACTION OF TWO CCP MODULES

CCPx Mode	CCPy Mode	Interaction
Capture	Capture	Same TMR1 time-base
Capture	Compare	The compare should be configured for the special event trigger, which clears TMR1
Compare	Compare	The compare(s) should be configured for the special event trigger, which clears TMR1
PWM	PWM	The PWMs will have the same frequency and update rate (TMR2 interrupt)
PWM	Capture	None
PWM	Compare	None

REGISTER 8-1: CCP1CON REGISTER/CCP2CON REGISTER (ADDRESS: 17h/1Dh)

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	
—	—	CCPxX	CCPxY	CCPxM3	CCPxM2	CCPxM1	CCPxM0	
bit 7								bit 0

bit 7-6 **Unimplemented:** Read as '0'

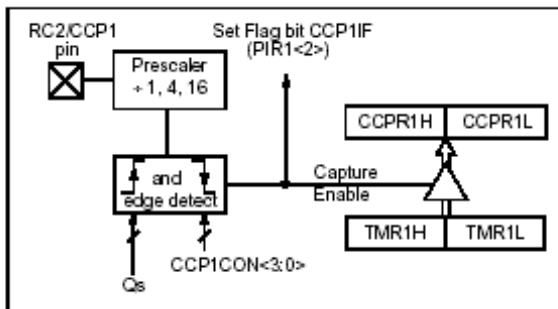
bit 5-4 **CCPxX:CCPxY:** PWM Least Significant bits
Capture mode:
 Unused
Compare mode:
 Unused
PWM mode:
 These bits are the two LSbs of the PWM duty cycle. The eight MSbs are found in CCPRxL.

bit 3-0 **CCPxM3:CCPxM0:** CCPx Mode Select bits
 0000 = Capture/Compare/PWM disabled (resets CCPx module)
 0100 = Capture mode, every falling edge
 0101 = Capture mode, every rising edge
 0110 = Capture mode, every 4th rising edge
 0111 = Capture mode, every 16th rising edge
 1000 = Compare mode, set output on match (CCPxIF bit is set)
 1001 = Compare mode, clear output on match (CCPxIF bit is set)
 1010 = Compare mode, generate software interrupt on match (CCPxIF bit is set, CCPx pin is unaffected)
 1011 = Compare mode, trigger special event (CCPxIF bit is set, CCPx pin is unaffected); CCP1 resets TMR1; CCP2 resets TMR1 and starts an A/D conversion (if A/D module is enabled)
 11xx = PWM mode

➤ mode capture

and will not generate the "false" interrupt.

FIGURE 8-1: CAPTURE MODE OPERATION BLOCK DIAGRAM



EXAMPLE 8-1: CHANGING BETWEEN CAPTURE PRESCALERS

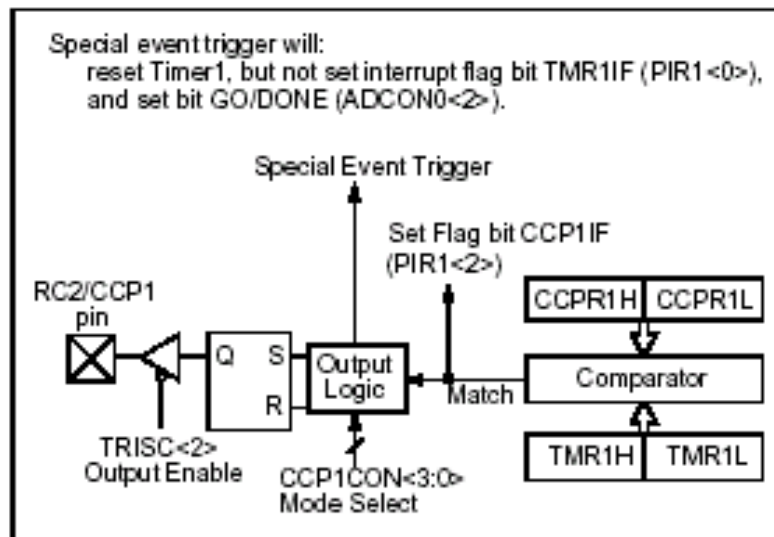
```

CLRF   CCP1CON    ; Turn CCP module off
MOVLW  NEW_CAPT_PS ; Load the W reg with
                  ; the new prescaler
MOVWF  CCP1CON    ; move value and CCP ON

MOVWF  CCP1CON    ; Load CCP1CON with this
                  ; value
    
```

➤ mode comparaison

FIGURE 8-2: COMPARE MODE OPERATION BLOCK DIAGRAM



➤ mode pwm

FIGURE 8-3: SIMPLIFIED PWM BLOCK DIAGRAM

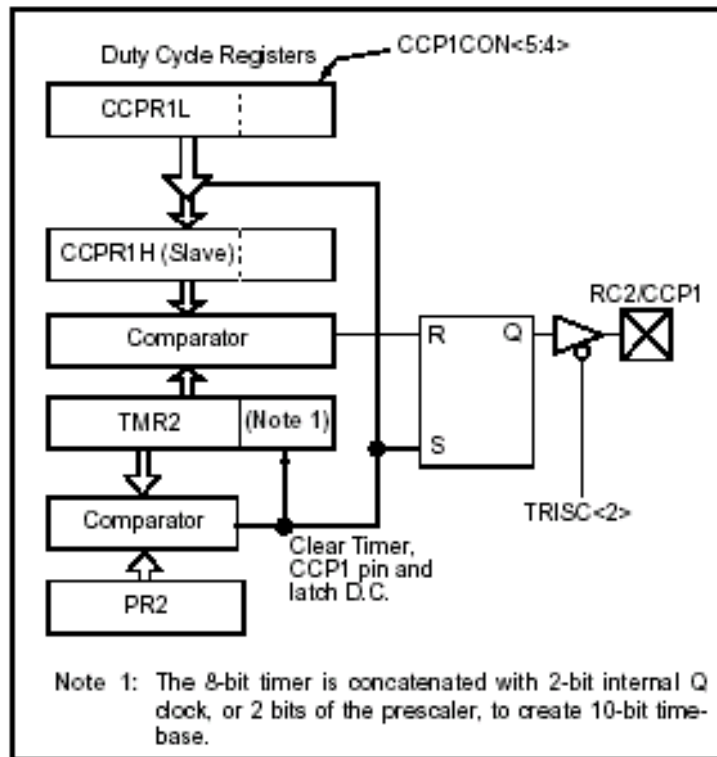
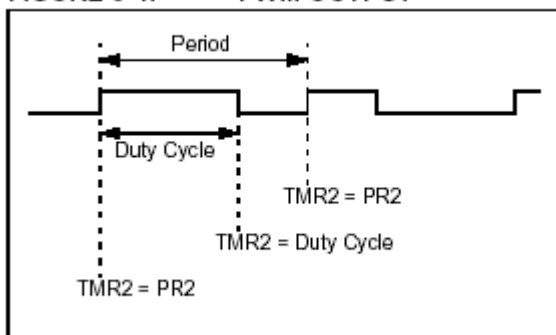


FIGURE 8-4: PWM OUTPUT



$$\text{Resolution} = \frac{\log\left(\frac{F_{OSC}}{F_{PWM}}\right)}{\log(2)} \text{ bits}$$

TABLE 8-3: EXAMPLE PWM FREQUENCIES AND RESOLUTIONS AT 20 MHz

PWM Frequency	1.22 kHz	4.88 kHz	19.53 kHz	78.12kHz	156.3 kHz	208.3 kHz
Timer Prescaler (1, 4, 16)	16	4	1	1	1	1
PR2 Value	0xFFh	0xFFh	0xFFh	0x3Fh	0x1Fh	0x17h
Maximum Resolution (bits)	10	10	10	8	7	5.5

TABLE 8-4: REGISTERS ASSOCIATED WITH CAPTURE, COMPARE, AND TIMER1

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other RESETS
0Bh,8Bh, 10Bh, 18Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF ⁽¹⁾	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
0Dh	PIR2	—	—	—	—	—	—	—	CCP2IF	---- --0	---- --0
8Ch	PIE1	PSPIE ⁽¹⁾	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
8Dh	PIE2	—	—	—	—	—	—	—	CCP2IE	---- --0	---- --0
87h	TRISC	PORTC Data Direction Register								1111 1111	1111 1111
0Eh	TMR1L	Holding Register for the Least Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	uuuu uuuu
0Fh	TMR1H	Holding Register for the Most Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	uuuu uuuu
10h	T1CON	—	—	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON	--00 0000	--uu uuuu
15h	CCPR1L	Capture/Compare/PWM Register1 (LSB)								xxxx xxxx	uuuu uuuu
16h	CCPR1H	Capture/Compare/PWM Register1 (MSB)								xxxx xxxx	uuuu uuuu
17h	CCP1CON	—	—	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0	--00 0000	--00 0000
1Bh	CCPR2L	Capture/Compare/PWM Register2 (LSB)								xxxx xxxx	uuuu uuuu
1Ch	CCPR2H	Capture/Compare/PWM Register2 (MSB)								xxxx xxxx	uuuu uuuu
1Dh	CCP2CON	—	—	CCP2X	CCP2Y	CCP2M3	CCP2M2	CCP2M1	CCP2M0	--00 0000	--00 0000

Legend: x = unknown, u = unchanged, - = unimplemented, read as '0'. Shaded cells are not used by Capture and Timer1.

Note 1: The PSP is not implemented on the PIC16F873/876; always maintain these bits clear.

2.8 Le chien de garde.

Ce dispositif est un système anti-plantage du microcontrôleur. Il s'assure qu'il n'y ait pas d'exécution prolongé d'une même suite d'instruction.

Un compteur préchargeable se décrémente régulièrement au rythme de la fréquence d'horloge. Si aucun préchargement n'est effectué avant qu'il n'atteigne la valeur "0" un Reset est généré relançant ainsi le microcontrôleur. Il faut donc penser à précharger régulièrement ce chien de garde par programme lorsqu'il est activé.

Le chien de garde. Du PIC16F877

FIGURE 12-10: WATCHDOG TIMER BLOCK DIAGRAM

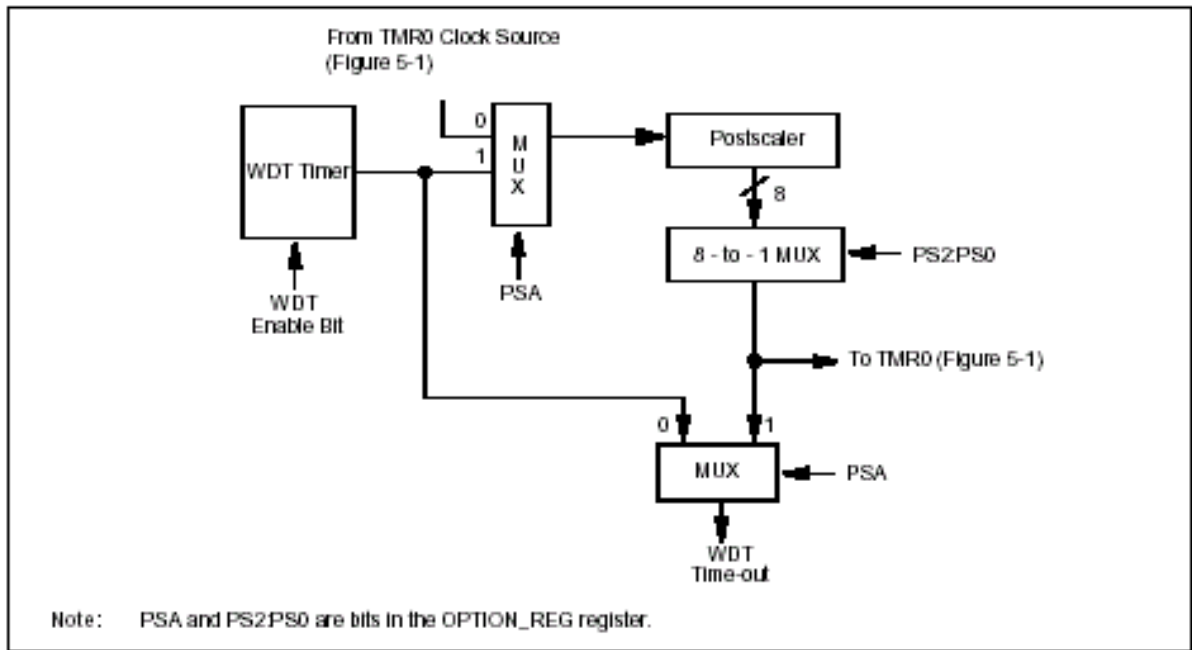


TABLE 12-7: SUMMARY OF WATCHDOG TIMER REGISTERS

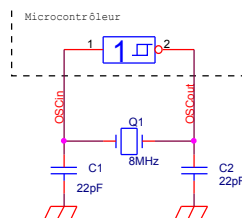
Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
2007h	Config. bits	(1)	BODEN ⁽¹⁾	CP1	CP0	PWRTE ⁽¹⁾	WDTE	FOSC1	FOSC0
81h,181h	OPTION_REG	RBPU	INTEDG	T0CS	TOSE	PSA	PS2	PS1	PS0

Legend: Shaded cells are not used by the Watchdog Timer.

Note 1: See Register 12-1 for operation of these bits.

2.9 Les signaux d'horloge.

Le signal d'horloge permet de cadencer le fonctionnement du microcontrôleur. Ce dernier intègre généralement une porte Trigger de Schmitt afin de réaliser un oscillateur. Pour l'obtenir on place un quartz entre les deux broches "OscIn" et "OscOut" comme l'indique le schéma suivant :



a) Le choix de configuration

REGISTER 12-1: CONFIGURATION WORD (ADDRESS 2007h)⁽¹⁾

	CP1	CP0	DEBUG	—	WRT	CPD	LVP	BODEN	CP1	CP0	PWRT ⁽³⁾	WDTE	FOSC1	FOSC0	
bit13															bit0
bit 13-12,	CP1:CP0: FLASH Program Memory Code Protection bits ⁽²⁾														
bit 5-4	11 = Code protection off 10 = 1F00h to 1FFFh code protected (PIC16F877, 876) 10 = 0F00h to 0FFFh code protected (PIC16F874, 873) 01 = 1000h to 1FFFh code protected (PIC16F877, 876) 01 = 0800h to 0FFFh code protected (PIC16F874, 873) 00 = 0000h to 1FFFh code protected (PIC16F877, 876) 00 = 0000h to 0FFFh code protected (PIC16F874, 873)														
bit 11	DEBUG: In-Circuit Debugger Mode 1 = In-Circuit Debugger disabled, RB6 and RB7 are general purpose I/O pins 0 = In-Circuit Debugger enabled, RB6 and RB7 are dedicated to the debugger.														
bit 10	Unimplemented: Read as '1'														
bit 9	WRT: FLASH Program Memory Write Enable 1 = Unprotected program memory may be written to by EECON control 0 = Unprotected program memory may not be written to by EECON control														
bit 8	CPD: Data EE Memory Code Protection 1 = Code protection off 0 = Data EEPROM memory code protected														
bit 7	LVP: Low Voltage In-Circuit Serial Programming Enable bit 1 = RB3/PGM pin has PGM function, low voltage programming enabled 0 = RB3 is digital I/O, HV on MCLR must be used for programming														
bit 6	BODEN: Brown-out Reset Enable bit ⁽³⁾ 1 = BOR enabled 0 = BOR disabled														
bit 3	PWRT ⁽³⁾ : Power-up Timer Enable bit ⁽³⁾ 1 = PWRT disabled 0 = PWRT enabled														
bit 2	WDTE: Watchdog Timer Enable bit 1 = WDT enabled 0 = WDT disabled														
bit 1-0	FOSC1:FOSC0: Oscillator Selection bits 11 = RC oscillator 10 = HS oscillator 01 = XT oscillator 00 = LP oscillator														

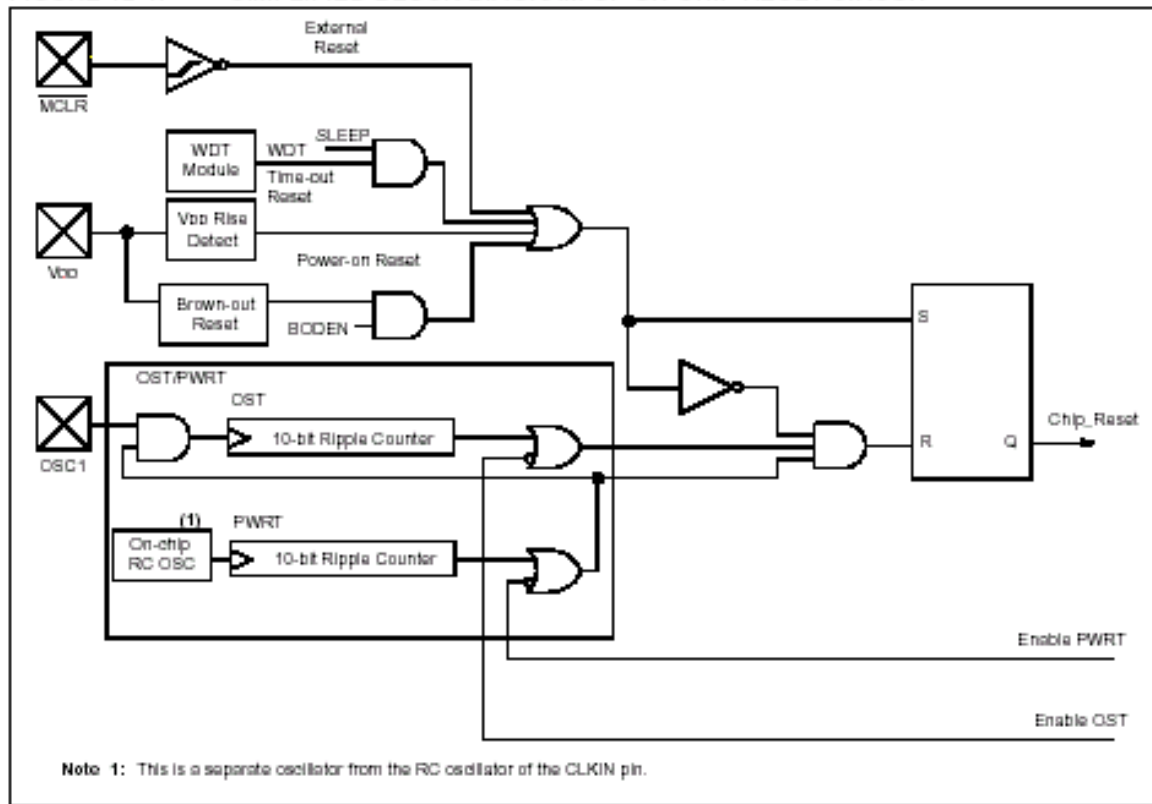
On trouve dans cette configuration des choix concernant la protection des pages de mémoire programme, de data EEPROM, de validation du chien de garde (watchdog), de sélection du type d'oscillateur pour le circuit d'horloge, de configuration du mode de RESET

b) Le reset

Il existe différentes situation pouvant entraîner un reset du circuit :

- **Power on reset** : la tension d'alimentation descend en dessous du seuil de 1,2 à 1,7v.
- **Brown on reset** : la tension d'alimentation reste vers un niveau de 4 v pendant un temps assez long
- **Watcdog reset** : action du chien de garde
- **MCLR/ activé** : action sur l'entrée MCLR

FIGURE 12-4: SIMPLIFIED BLOCK DIAGRAM OF ON-CHIP RESET CIRCUIT



Le reset initialise les différents contenus des registres concernés. (voir doc technique)

c) L'horloge

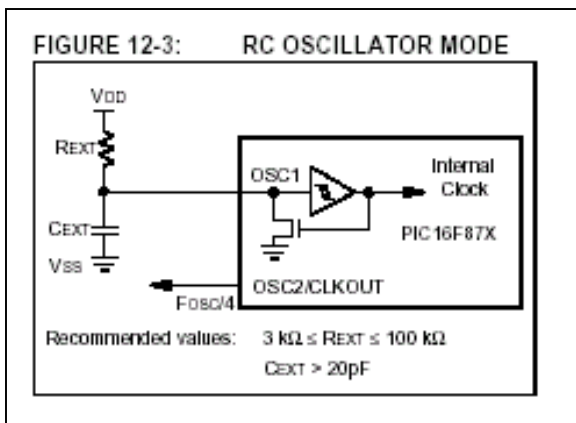
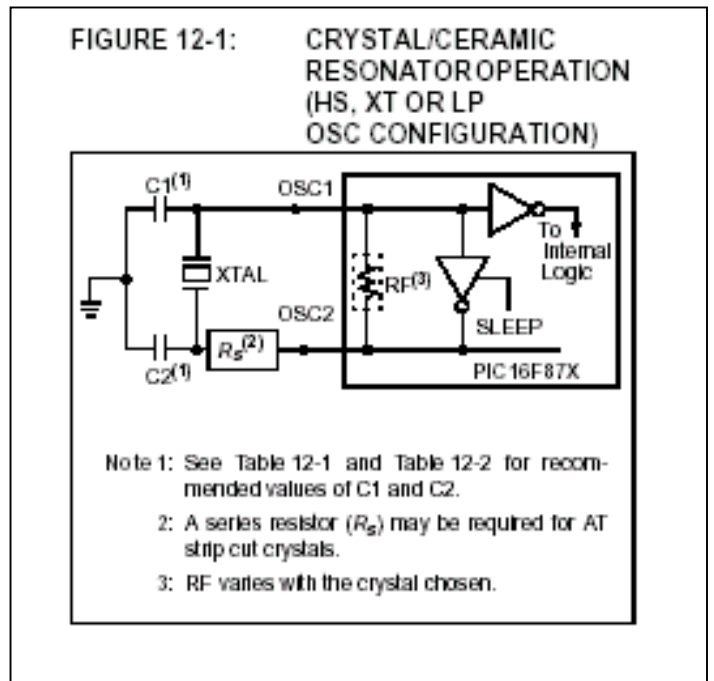
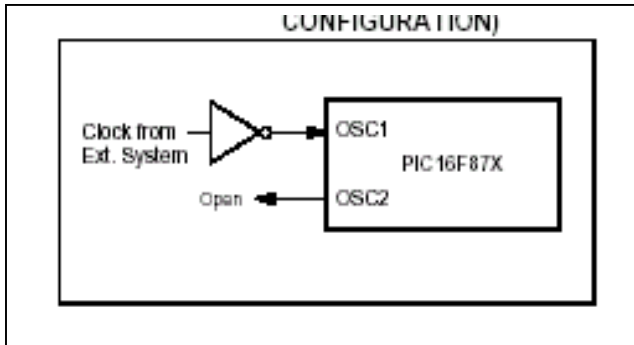


TABLE 12-1: CERAMIC RESONATORS

Ranges Tested:			
Mode	Freq.	OSC1	OSC2
XT	455 kHz	68 - 100 pF	68 - 100 pF
	2.0 MHz	15 - 68 pF	15 - 68 pF
	4.0 MHz	15 - 68 pF	15 - 68 pF
HS	8.0 MHz	10 - 68 pF	10 - 68 pF
	16.0 MHz	10 - 22 pF	10 - 22 pF
These values are for design guidance only. See notes following Table 12-2.			
Resonators Used:			
455 kHz	Panasonic EFO-A455K04B	± 0.3%	
2.0 MHz	Murata Erie CSA2.00MG	± 0.5%	
4.0 MHz	Murata Erie CSA4.00MG	± 0.5%	
8.0 MHz	Murata Erie CSA8.00MT	± 0.5%	
16.0 MHz	Murata Erie CSA16.00MX	± 0.5%	
All resonators used did not have built-in capacitors.			

TABLE 12-2: CAPACITOR SELECTION FOR CRYSTAL OSCILLATOR

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
These values are for design guidance only. See notes following this table.			
Crystals Used			
32 kHz	Epson C-001R32.768K-A	± 20 PPM	
200 kHz	STD XTL 200.000KHz	± 20 PPM	
1 MHz	ECS ECS-10-13-1	± 50 PPM	
4 MHz	ECS ECS-40-20-1	± 50 PPM	
8 MHz	EPSON CA-301 8.000M-C	± 30 PPM	
20 MHz	EPSON CA-301 20.000M-C	± 30 PPM	

3 Mode de fonctionnement

Le microprocesseur exécute séquentiellement les instructions codées en binaire et présentent dans la mémoire programme. L'initialisation de cette séquence peut se faire de différentes manières selon le mode de fonctionnement.

3.1 Le fonctionnement en interruptions.

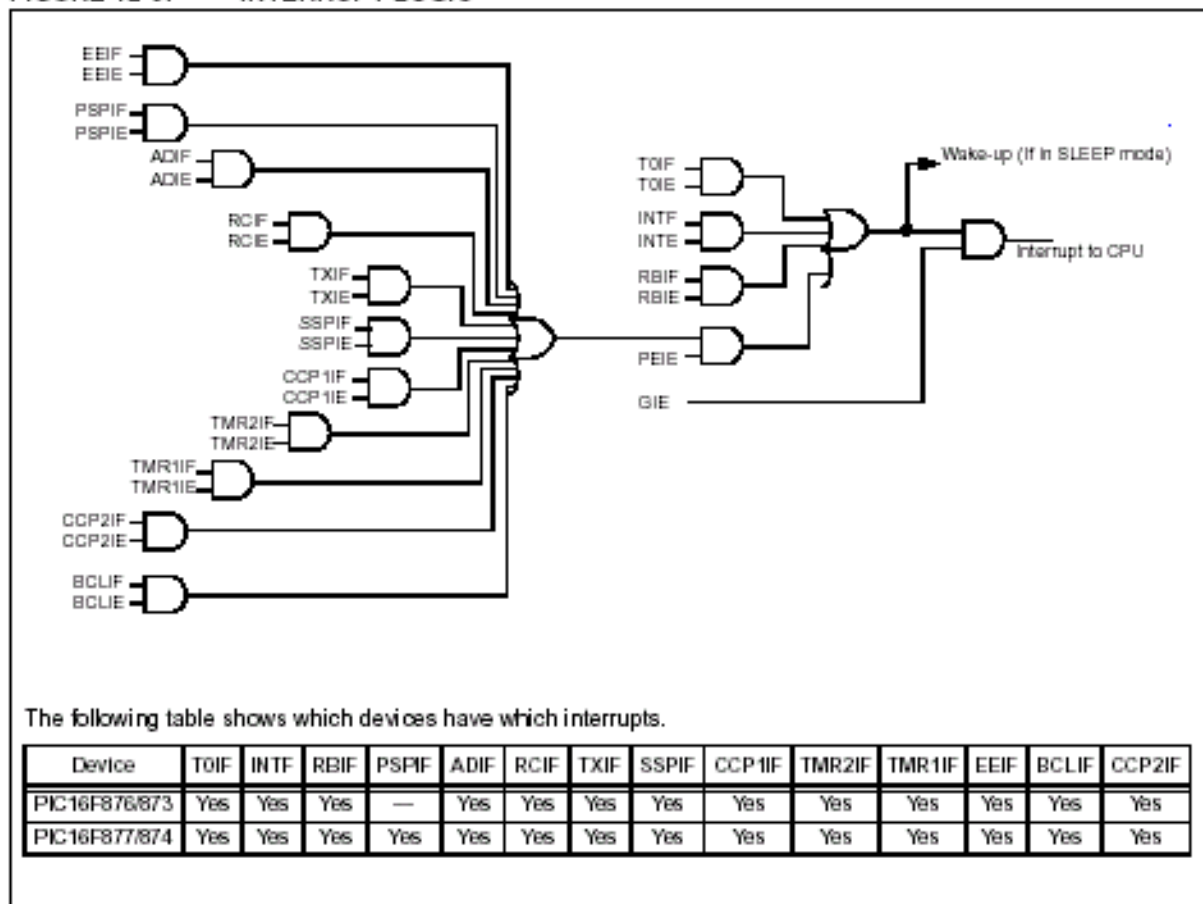
Le microcontrôleur, dans son environnement, est destiné à traiter des informations en "temps réel". L'application est couplée au monde extérieur, par l'échange fréquent de messages et de signaux à des instants prévus. Il est dans l'obligation de changer d'état en fonction des priorités relatives de l'opération en cours et de celle qui lui est demandé. Il interrompt ou non le déroulement normal du programme en fonction d'une demande externe.

Quelque soit l'entrée d'interruption activée, le microprocesseur réalise des tâches identiques :

- dans tout les cas, le programme principal est interrompu ;
- le processeur doit sauvegarder le contenu du PC dans la pile ;
- le processeur exécute une séquence privilégiée, reflet du type de traitement d'interruption
- la prise en compte d'une interruption ne se fait jamais pendant l'exécution d'une instruction.

Les interruptions du PIC16F877

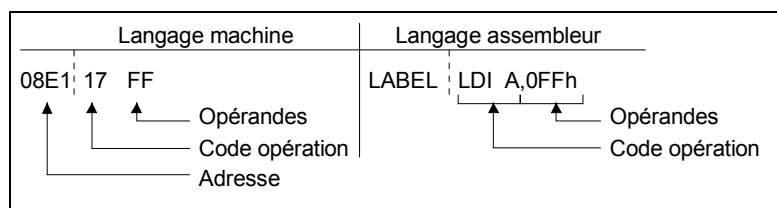
FIGURE 12-9: INTERRUPT LOGIC



En ce qui concerne le PIC 16F877, les interruptions peuvent provenir soit d'une ligne de port configurée comme telle, soit par les timers en fin de comptage, soit par le CAN pour signaler la fin de conversion, soit lors d'événements de transmissions sur les différents interfaces et liaison séries.

3.2 Instructions et modes d'adressages.

Les instructions contenues dans la mémoire programme sont une suite de mots binaires décodés puis exécutés par le microprocesseur, appelée langage machine. Ces codes sont difficilement compréhensibles par le programmeur. C'est la raison pour laquelle ils sont traduits en différents mots faisant partis du langage assembleur. Le codage d'une instruction s'effectue de la façon suivante :



Les modes d'adressages sont les différents moyens qui permettent au microprocesseur d'accéder à une opérande en vue de tester ou de modifier le contenu d'un registre ou d'une mémoire.

➤ **Mode d'adressage inhérent ou implicite.**

L'adressage inhérent concerne les instructions qui ne comportent pas d'opérande, cette dernière étant implicite. Il s'agit généralement des opérations de mise à 0 et d'incrémentation ou de décalage de bits

➤ **Mode d'adressage immédiat.**

Ce mode d'adressage permet de charger les registres internes du microprocesseur directement avec la valeur de l'opérande.

➤ **Mode d'adressage direct.**

Dans ce mode d'adressage l'opérande correspond à une adresse où est située la donnée de l'opération .

➤ **Mode d'adressage indexé ou indirect.**

Ce mode d'adressage s'applique aux registres d'index. Ces derniers contiennent une adresse mémoire dans laquelle est placée la donnée de l'opération.

➤ **Mode d'adressage relatif.**

Ce mode d'adressage est réservé pour les instructions de rupture de séquence conditionnel. La condition provient généralement du résultat de l'opération précédente (résultat nul, ayant entraîné une retenue ...) où de l'état d'un bit.

➤ **Mode d'adressage étendu.**

Ce mode d'adressage permet d'effectués des ruptures de séquence sans condition afin d'atteindre une adresse non successive dans la mémoire programme.

3.3 Les Instructions du PIC16F877.

TABLE 13-2: PIC16F87X INSTRUCTION SET

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	ffff	ffff	C,DC,Z	1,2
ANDWF	f, d	AND W with f	1	00	0101	ffff	ffff	Z	1,2
CLRF	f	Clear f	1	00	0001	1fff	ffff	Z	2
CLRWF	-	Clear W	1	00	0001	0xxx	xxxx	Z	
COMF	f, d	Complement f	1	00	1001	ffff	ffff	Z	1,2
DECF	f, d	Decrement f	1	00	0011	ffff	ffff	Z	1,2
DECFSZ	f, d	Decrement f, Skip if 0	1(2)	00	1011	ffff	ffff		1,2,3
INCF	f, d	Increment f	1	00	1010	ffff	ffff	Z	1,2
INCFSZ	f, d	Increment f, Skip if 0	1(2)	00	1111	ffff	ffff		1,2,3
IORWF	f, d	Inclusive OR W with f	1	00	0100	ffff	ffff	Z	1,2
MOVF	f, d	Move f	1	00	1000	ffff	ffff	Z	1,2
MOWWF	f	Move W to f	1	00	0000	1fff	ffff		
NOP	-	No Operation	1	00	0000	0xxx	0000		
RLF	f, d	Rotate Left f through Carry	1	00	1101	ffff	ffff	C	1,2
RRF	f, d	Rotate Right f through Carry	1	00	1100	ffff	ffff	C	1,2
SUBWF	f, d	Subtract W from f	1	00	0010	ffff	ffff	C,DC,Z	1,2
SWAPF	f, d	Swap nibbles in f	1	00	1110	ffff	ffff		1,2
XORWF	f, d	Exclusive OR W with f	1	00	0110	ffff	ffff	Z	1,2
BIT-ORIENTED FILE REGISTER OPERATIONS									
BCF	f, b	Bit Clear f	1	01	00bb	bfff	ffff		1,2
BSF	f, b	Bit Set f	1	01	01bb	bfff	ffff		1,2
BTFSC	f, b	Bit Test f, Skip if Clear	1(2)	01	10bb	bfff	ffff		3
BTFSS	f, b	Bit Test f, Skip if Set	1(2)	01	11bb	bfff	ffff		3
LITERAL AND CONTROL OPERATIONS									
ADDLW	k	Add literal and W	1	11	111x	kkkk	kkkk	C,DC,Z	
ANDLW	k	AND literal with W	1	11	1001	kkkk	kkkk	Z	
CALL	k	Call subroutine	2	10	0kck	kkck	kkck		
CLRWDT	-	Clear Watchdog Timer	1	00	0000	0110	0100	$\overline{TO}, \overline{PD}$	
GOTO	k	Go to address	2	10	1kck	kkck	kkck		
IORLW	k	Inclusive OR literal with W	1	11	1000	kkkk	kkkk	Z	
MOVLW	k	Move literal to W	1	11	00xx	kkkk	kkkk		
RETFIE	-	Return from interrupt	2	00	0000	0000	1001		
RETLW	k	Return with literal in W	2	11	01xx	kkck	kkck		
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	kkkk	kkkk	C,DC,Z	
XORLW	k	Exclusive OR literal with W	1	11	1010	kkkk	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.

Note: Additional information on the mid-range instruction set is available in the PICmicro™ Mid-Range MCU Family Reference Manual (DS33023).

4 Structure d'un programme.

La saisie d'un programme, que ça soit en langage assembleur ou évolué, doit suivre la structure suivante.

- 1- Directives d'assemblage ou de compilation
- 2- Déclaration des constantes
- 3- Déclaration des variables
- 4- Sous-programmes
- 5- Programme principal
- 6- Programme d'interruption

5 Exemples de microcontrôleurs.

REFERENCE	FABRICANT	VITESSE	RAM	ROM / EPROM / FLASH	EEPROM	E / S LOGIQUES	TIMER	ENTREES ANALOGIQUES	PARTICULARITE
8051	Intel	12 Mhz	128 o	4 Ko	X	32	2	0	
16C71	Microchip	20 Mhz	36 o	1Kx14	X	13	1	4	RISC
6805 S2	Motorola	4 MHz	64	1 Ko	X	16	2	8	
68HC11 A1	Motorola	8 MHz	256 o	X	512	22	1	8	Etendu
AT90S 8515	Atmel	20 MHz	512 o	4 Ko	512	32	3	8	RISC
ST 6265	Thomson	8 MHz	128 o	4 Ko	64 o	21	2	13	
PIC 16F8XX	Microchip	20 MHz	128 o	8Kx14	128o		3	5	RISC
SX28AC	SCENIX	20MHZ							