

1^{ère} Partie :

LES MICROCONTROLEURS (PICs de Microchip)

1- Introduction

Les PICs sont des microcontrôleurs à architecture RISC (Reduce Instructions Construction Set), ou encore composant à jeu d'instructions réduit. L'avantage est que plus on réduit le nombre d'instructions, plus leur décodage sera rapide ce qui augmente la vitesse de fonctionnement du microcontrôleur. La famille des PICs est subdivisée en 3 grandes familles : La famille **Base-Line**, qui utilise des mots d'instructions de 12 bits, la famille **Mid-Range**, qui utilise des mots de 14 bits (et dont font partie le 16F86 et 16F877), et la famille **High-End**, qui utilise des mots de 16 bits. Les PICs sont des composants STATIQUES, Ils peuvent fonctionner avec des fréquences d'horloge allant du continu jusqu'à une fréquence max spécifique à chaque circuit. Un PIC16F876-04 peut fonctionner avec une horloge allant du continu jusqu'à 4 MHz. Nous nous limiterons dans ce document à la famille Mid-Range et particulièrement au PIC 16F876/877, sachant que si on a tout assimilé, on pourra facilement passer à une autre famille, et même à un autre microcontrôleur.

PIC	FLASH	RAM	EEPROM	I/O	A/D	Port //	Port Série
16F876	8K	368	256	22	5	NON	USART/MSSP
16F877	8K	368	256	33	8	PSP	USART/MSSP

Tableau I.1 : caractéristiques de la famille 16F876 – 16F877

Les éléments essentiels du PIC 16F876 et 16F877 sont :

- Une mémoire programme de type EEPROM flash de 8K mots de 14 bits,
- Une RAM donnée de 368 octets,

- Une mémoire EEPROM de 256 octets,
- Trois ports d'entrée sortie, A (6 bits), B (8 bits), C (8 bits),
- Convertisseur Analogiques numériques 10 bits à 5 canaux pour le 16F876, (8 canaux pour le 16F877)
- USART, Port série universel, mode asynchrone (RS232) et mode synchrone
- SSP, Port série synchrone supportant I2C
- Trois TIMERS avec leurs Prescalers, TMR0, TMR1, TMR2
- Deux modules de comparaison et Capture CCP1 et CCP2
- Un chien de garde,
- 13 sources d'interruption,
- Générateur d'horloge, à quartz (jusqu' à 20 MHz) ou à Oscillateur RC
- Fonctionnement en mode sleep pour réduction de la consommation,
- Programmation par mode ICSP (In Circuit Serial Programming) 12V ou 5V,
- Possibilité aux applications utilisateur d'accéder à la mémoire programme,
- Tension de fonctionnement de 2 à 5V,
- Jeux de 35 instructions

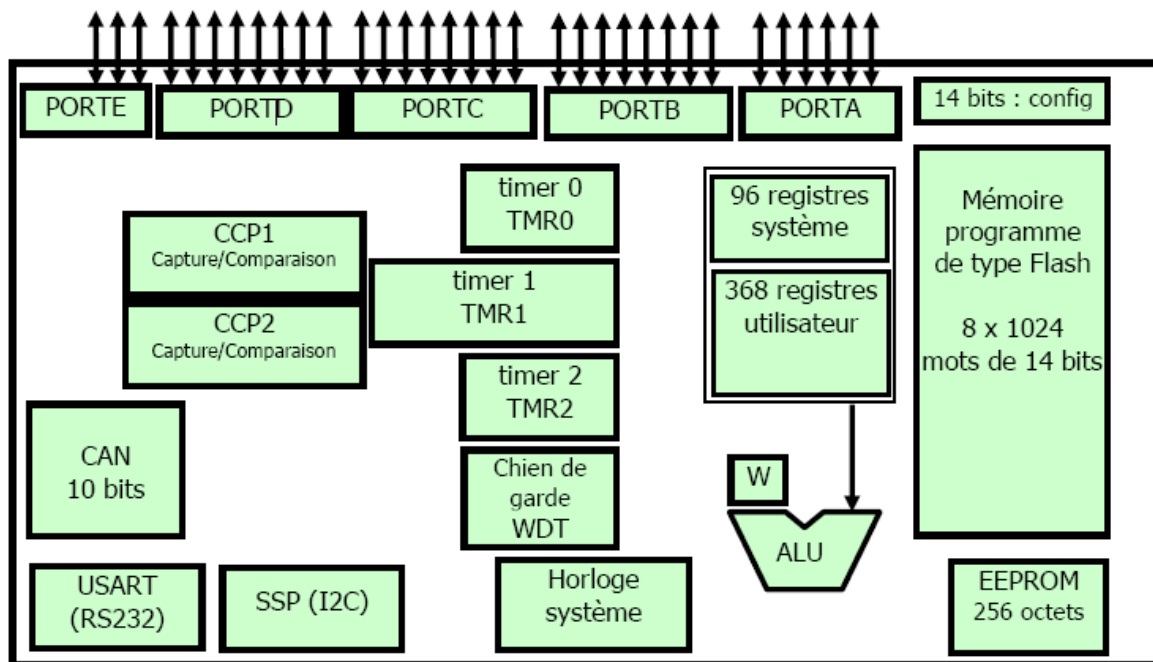
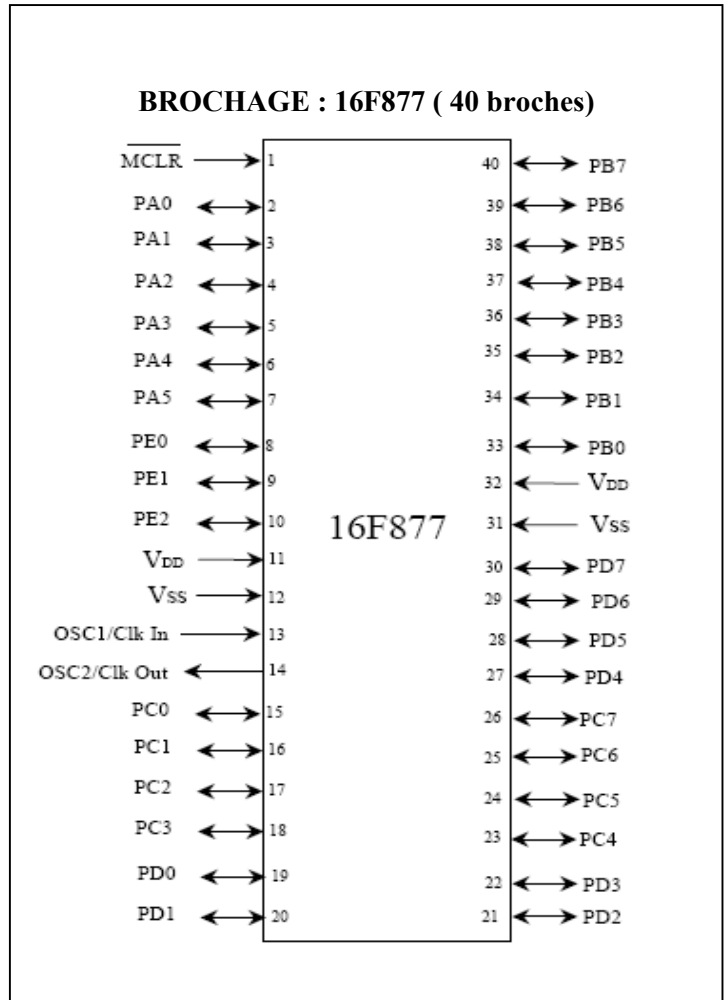
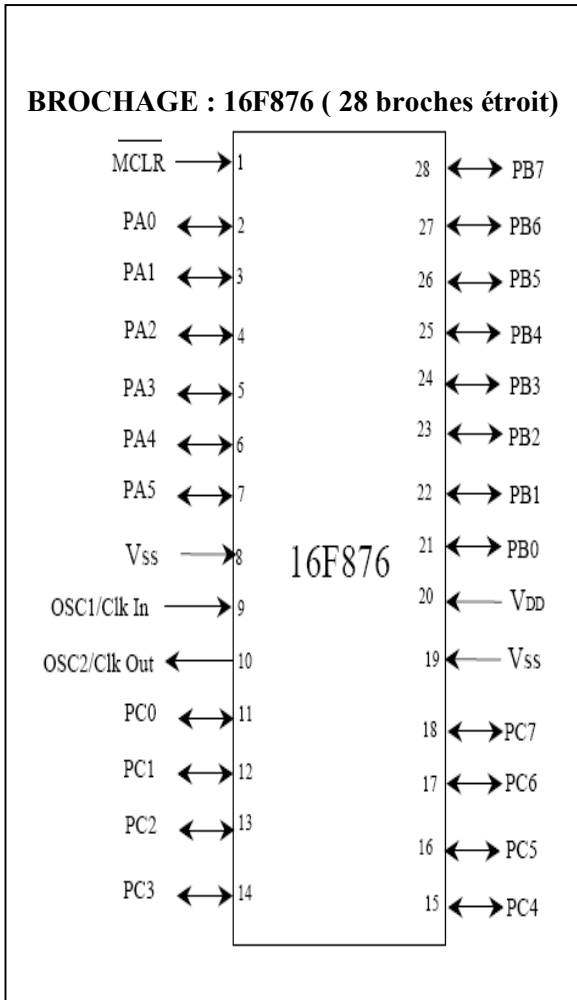


Fig. I.1 : Les éléments constitutifs du PIC 16F877

Le port D (8 bits) et le port E (3 bits) ne sont pas disponibles sur 16F876.



2- Les éléments de base du PIC 16F876/877

L'Horloge

L'horloge peut être soit interne soit externe. L'horloge interne est constituée d'un oscillateur à quartz ou d'un oscillateur RC.

Avec l'oscillateur à Quartz, on peut avoir des fréquences allant jusqu'à 20 MHz selon le type de μC . Le filtre passe bas (R_s , C_1 , C_2) limite les harmoniques dus à l'écrêtage et réduit l'amplitude de l'oscillation, il n'est pas obligatoire.

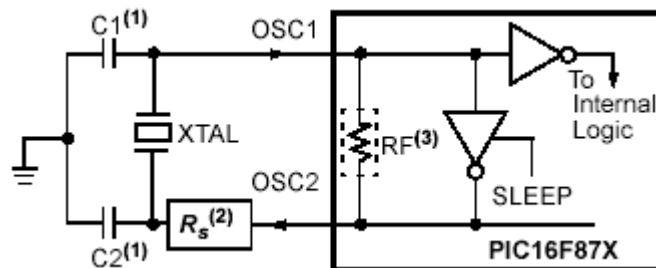


Fig. I.2 : Oscillateur à quartz du PIC 16F87x

Avec un oscillateur RC, la fréquence de l'oscillation est fixée par Vdd, Rext et Cext. Elle peut varier légèrement d'un circuit à l'autre.

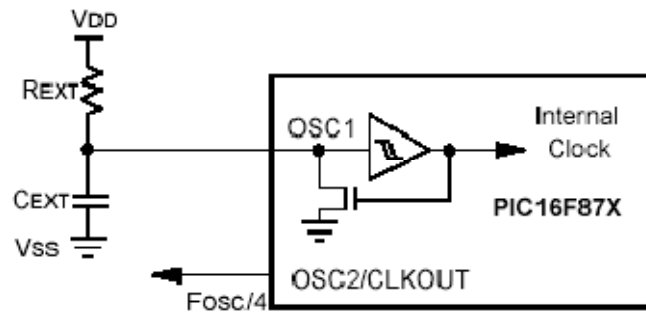


Fig. I.3 : Oscillateur RC du PIC 16F87x

Dans certains cas, une horloge externe au microcontrôleur peut être utilisée pour synchroniser le PIC sur un processus particulier.

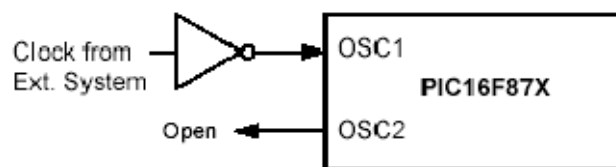


Fig. I.4 : horloge externe du PIC 16F87x

Quelque soit l'oscillateur utilisé, l'horloge système dite aussi horloge instruction est obtenue en divisant la fréquence par 4. Dans la suite de ce document on utilisera le terme $F_{osc}/4$ pour désigner l'horloge système.

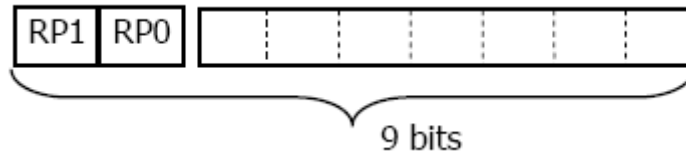
Avec un quartz de 4 MHz, on obtient une horloge instruction de 1 MHz, soit le temps pour exécuter une instruction de $1\mu s$.

Organisation de la mémoire RAM

L'espace mémoire RAM adressable est de **512** positions de 1 octet chacune :

- 96 positions sont réservées au SFR (Special Function Registers) qui sont les registres de configuration du PIC.
- Les 416 positions restantes constituent les registres GPR (General Purpose Registers) ou RAM utilisateur. Sur le 16F876 et 16F877, 3 blocs de 16 octets chacun ne sont pas implantés physiquement d'où une capacité de RAM utilisateur de 368 GPR.

Pour pouvoir adresser les 512 positions accessibles, il faut **9 bits** d'adresse. Pour avoir ces 9 bits, le PIC complète les 7 bits venant de l'instruction par deux bits situés dans le registre de configuration STATUS. Ces bits sont appelés RP0 et RP1 et doivent être positionnés correctement avant toute instruction qui accède à la RAM par l'adressage direct.



La RAM apparaît alors organisée en 4 banks de 128 octets chacun. L'adresse instruction permet d'adresser à l'intérieur d'un bank alors que les bits RP0 et RP1 du registre STATUS permettent de choisir un bank.

La Figure I-5 montre l'organisation de la RAM avec les zones allouées au SFR et aux GPR. Les zones hachurées ne sont pas implantées physiquement. Si on essaye d'y accéder, on est aiguillé automatiquement vers la zone [70h,7Fh] appelée zone commune.

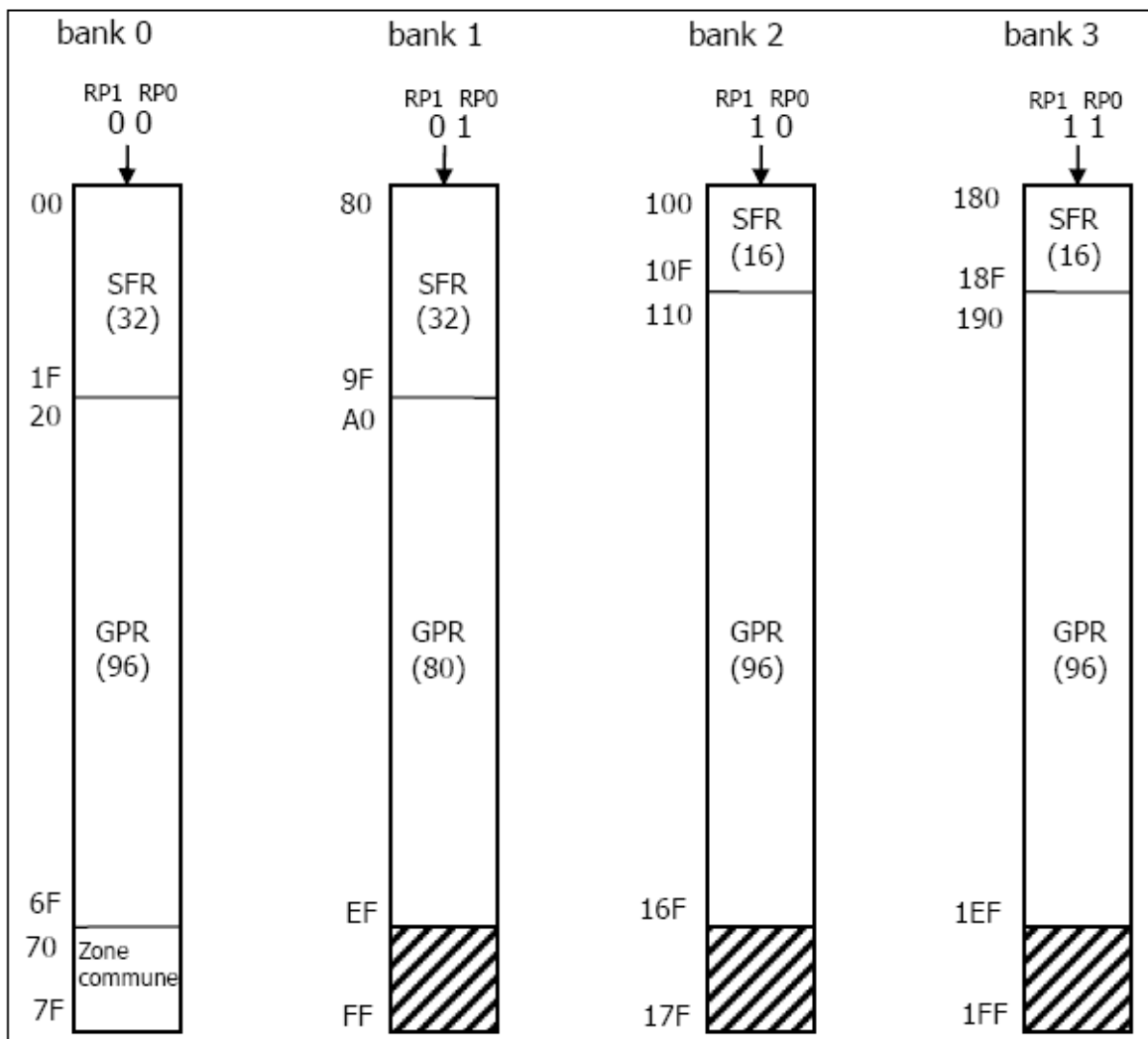


Figure I-5 : organisation de la RAM du 16F876/877

Registres de configuration et leurs positions dans la RAM

Bank 0		Bank 1		Bank 2		Bank 3	
00h	INDF	80h	INDF	100h	INDF	180h	INDF
01h	TMR0	81h	OPTION_REG	101h	TMR0	181h	OPTION_REG
02h	PCL	82h	PCL	102h	PCL	182h	PCL
03h	STATUS	83h	STATUS	103h	STATUS	183h	STATUS
04h	FSR	84h	FSR	104h	FSR	184h	FSR
05h	PORTA	85h	TRISA	105h		185h	
06h	PORTB	86h	TRISB	106h	PORTB	186h	TRISB
07h	PORTC	87h	TRISC	107h		187h	
08h	PORTD (*)	88h	TRISD (*)	108h		188h	
09h	PORTE (*)	89h	TRISE (*)	109h		189h	
0Ah	PCLATH	8Ah	PCLATH	10Ah	PCLATH	18Ah	PCLATH
0Bh	INTCON	8Bh	INTCON	10Bh	INTCON	18Bh	INTCON
0Ch	PIR1	8Ch	PIE1	10Ch	EEDATA	18Ch	EECON1
0Dh	PIR2	8Dh	PIE2	10Dh	EEADR	18Dh	EECON2
0Eh	TMR1L	8Eh	PCON	10Eh	EEDATH	18Eh	
0Fh	TMR1H	8Fh		10Fh	EEDARH	18Fh	
10h	T1CON	90h					
11h	TMR2	91h	SSPCON2				
12h	T2CON	92h	PR2				
13h	SSPBUF	93h	SSPADD				
14h	SSPCON	94h	SSPSTAT				
15h	CCPR1L	95h					
16h	CCPR1H	96h					
17h	CCP1CON	97h					
18h	RCSTA	98h	TXSTA				
19h	TXREG	99h	SPBRG				
1Ah	RCREG	9Ah					
1Bh	CCPR2L	9Bh					
1Ch	CCPR2H	9Ch					
1Dh	CCP2CON	9Dh					
1Eh	ADRESH	9Eh	ADRESL				
1Fh	ADCON0	9Fh	ADCON1				

* PORTD et PORTE existent seulement dans le 16F877 et le 16F874

Tableau I.2 : Registres de configuration avec leurs adresses

Les registres de configuration avec leurs états après un RESET

STATUS	IRP	RP1	RP0	TO	PD	Z	DC	C	0001 1xxx
OPTION_REG	RBPUP	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111
INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x
PIE1	PSPIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000
PIR1	PSPIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000
PIE2	N.I.	Réservé	N.I.	EEIE	BCLIE	N.I.	N.I.	CCP2IE	-r-0 0-0
PIR2	N.I.	Réservé	N.I.	EEIF	BCLIF	N.I.	N.I.	CCP2IF	-r-0 0-0
EECON1	EEPGRD	—	—	—	WRERR	WREN	WR	RD	x--- x000
TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x
CCPxCON	—	—	DCxB1	DCxB0	CCPxM3	CCPxM2	CCPxM1	CCPxM0	--00 0000
T1CON	—	—	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON	--00 0000
T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000
SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000
SSPCON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	0000 0000
SSPSTAT	SMP	CKE	D/A	P	S	R/W	UA	BF	0000 0000
CCP1CON	—	—	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0	--00 0000
TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x
CCP2CON	—	—	CCP2X	CCP2Y	CCP2M3	CCP2M2	CCP2M1	CCP2M0	--00 0000
ADCON0	ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/DONE	—	ADON	0000 00-0
ADCON1	ADFM	—	—	—	PCFG3	PCFG2	PCFG1	PCFG0	0--- 0000
TRISx									1111 1111

Tableau I.3 : détail des registres SFR et leurs états au démarrage

3- Les registres internes

STATUS REGISTER : (h'03' ou h'83' ou h'103 ou h'183').

Bit 7

Bit 0

IRP	RP1	RP0	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	C
-----	-----	-----	------------------------	------------------------	---	----	---

Au reset : STATUS = 00011XXX

IRP = permet la sélection des pages en adressage indirect.

RP1 et RP0 = permettent la sélection des pages en adressage direct.

RP1	RP0		Page sélectionnée
0	0	PAGE 0	de 00 à 7F
0	1	PAGE 1	de 80 à FF
1	0	PAGE 2	de 100 à 17F
1	1	PAGE 3	de 180 à 1FF

Bit 4 : **TO** = Time Out bit.

Bit en lecture seulement.

1 = Après une mise sous tension, après une RAZ du watchdog (CLRWDT) ou bien après l'instruction SLEEP.

0 = Signifie qu'un Time Out du timer de watchdog est survenu.

Bit 3 : **PD** = Power Down bit.

1 = Après une mise sous tension ou bien après une RAZ du Watchdog.

0 = Après l'instruction SLEEP.

Bit 2 : **Z** = Zero bit.

1 = Le résultat d'une opération arithmétique ou logique est zéro.

0 = Le résultat d'une opération arithmétique ou logique est différent de zéro.

Bit 1 : **DC** = Digit Carry bit.

1 = Une retenue sur le 4eme bit des poids faible est survenue après les instructions : ADDWF et ADDLW.

0 = Pas de retenue sur le 4eme bit des poids faible.

Bit 0 : **C** = Carry bit.

1 = Une retenue sur le bit MSB est survenue après les instructions ADDWF et ADDLW.

0 = Pas de retenue sur le bit MSB.

OPTION REGISTER : (h'81' ou h'181').

Ce registre en lecture écriture permet de configurer les prédiviseurs du Timer et du Watchdog, la source du Timer, le front des interruptions et le choix du Pull up sur le Port B..

Bit 7

Bit 0

$\overline{\text{RBPU}}$	INTEDG	TOCS	TOSE	PSA	PS2	PS1	PS0
--------------------------	--------	------	------	-----	-----	-----	-----

Au reset : OPTION = 11111111

Bit 7 : RBPU = Pull up Enable bit on Port B.

1 = Pull up désactivé sur le Port B.

0 = Pull up activé.

Bit 6 : INTEDG = Interrupt Edge select bit.

1 = Interruption si front montant sur la broche PB0/IRQ (pin 6).

0 = Interruption si front descendant sur PB0/IRQ.

Bit 5 : TOCS = Timer TMR0 Clock Source select bit.

1 = L'horloge du Timer est l'entrée PA4/Clk (pin 3).

0 = Le Timer utilise l'horloge interne du PIC.

Bit 4 : TOSE = Timer TMR0 Source Edge select bit.

1 = Le Timer s'incrémente à chaque front montant de la broche PA4/Clk.

0 = Le Timer s'incrémente à chaque front descendant de la broche PA4/Clk.

Bit 3 : PSA = Prescaler Assignment bit.

1 = Le prédiviseur est affecté au watchdog..

0 = Le prédiviseur est affecté au Timer TMR0.

Bits 2 à 0 : PS2 PS1 PS0 = Prescaler Rate Select bits.

PS2	PS1	PS0	Prédiv Timer	Prédiv Watchdog
0	0	0	2	1
0	0	1	4	2
0	1	0	8	4
0	1	1	16	8
1	0	0	32	16
1	0	1	64	32
1	1	0	128	64
1	1	1	256	128

Quand le pré diviseur est affecté au Watchdog (PSA=1), TMR0 est pré divisé par 1.

INTCON REGISTER : (h'0B' ou h'8B' ou h'10B' ou h'18B').

Ce registre en lecture écriture permet de configurer les différentes sources d'interruption.

Bit 7

Bit 0

GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF
-----	------	------	------	------	------	------	------

Au reset : INTCON = 0000000X

Bit 7 : GIE = Global Interrupt Enable bit

1 = Autorise toutes les interruptions non masquées.

0 = Désactive toutes les interruptions.

Bit 6 : PEIE = Peripheral Interrupt Enable bit.

1 = Autorise les interruptions causées par les périphériques.

0 = Désactive les interruptions causées par le périphériques.

Bit 5 : TOIE = Timer TMR0 Overflow Interrup Enable bit.

1 = Autorise les interruptions du Timer TMR0.

0 = Désactive les interruptions du Timer TMR0.

Bit 4 : **INTE** = RB0/Int Interrup Enable bit.

1 = Autorise les interruptions sur la broche : PB0/IRQ (pin6).

0 = Désactive les interruptions sur la broche : PB0/IRQ (pin6).

Bit 3 : **RBIE** = RB Port Change Interrup Enable bit.

1 = Autorise les interruptions par changement d'état du Port B (PB4 à PB7).

0 = Désactive les interruptions par changement d'état du Port B (PB4 à PB7).

Bit 2 : **TOIF** = Timer TMR0 Overflow Interrup Flag bit.

1 = Le Timer à débordé. Ce flag doit être remis à zéro par programme.

0 = Le Timer n'a pas débordé.

Bit 1 : **INTF** = RB0/Int Interrup Flag bit.

1 = Une interruption sur la broche PB0/IRQ (pin 6) est survenue.

0 = Pas d' interruption sur la broche PB0/IRQ (pin 6).

Bit 0 : **RBIF** = RB Port Change Interrup Flag bit. Ce flag doit être remis à zéro par programme.

1 = Quand au moins une entrée du port B (de PB4 à PB7) a changé d'état.

0 = Aucune entrée de PB4 à PB7 n'a changé d'état.

La directive **ORG**, définit la position dans la mémoire programme à partir de laquelle seront inscrites les instructions suivantes.

CONFIG : permet de définir les 14 bits (fusibles ou *switch*) de configuration qui seront copiés dans l'EEPROM de configuration (adresse 2007h) lors de l'implantation du programme dans le PIC.

CP1	CP0	DEBUG	—	WRT	CPD	LVP	BODEN	CP1	CP0	PWRTE	WDTE	FOSC1	FOSC0
-----	-----	-------	---	-----	-----	-----	-------	-----	-----	-------	------	-------	-------

CP1/CP0 : bits 13/12 ; Déterminent quelle zone de la mémoire programme sera protégée contre la lecture externe (via ICSP) ou l'écriture par programme conformément à l'état du bit 9 (WRT). On peut choisir de protéger la totalité de la mémoire ou seulement une partie.

Les différentes zones pouvant être protégées sont les suivantes :

1 1 : Aucune protection (`_CP_OFF`)

1 0 : Protection de la zone 0x1F00 à 0x1FFF (`_CP_UPPER_256`)

0 1 : Protection de la zone 0x1000 à 0x1FFF (`_CP_HALF`)

0 0 : Protection de l'intégralité de la mémoire (`_CP_ALL`) 20

DEBUG : bit 11 : Debuggage sur circuit. Permet de dédicacer RB7 et RB6 à la communication avec un debugger.

1 : RB6 et RB7 sont des I/O ordinaires (`_DEBUG_OFF`)

0 : RB6 et RB7 sont utilisés pour le debuggage sur circuit (`_DEBUG_ON`)

WRT : bit 9 : Autorisation d'écriture en flash

1 : Le programme peut écrire dans les zones non protégées par les bits CP1/CP0 (`_WRT_ENABLE_ON`)

0 : Le programme ne peut pas écrire en mémoire flash (`_WRT_ENABLE_OFF`)

CPD : bit 8 : Protection en lecture de la mémoire EEPROM de données.

1 : mémoire EEPROM non protégée (`_CPD_OFF`)

0 : mémoire EEPROM protégée contre la lecture externe via ICSP (`_CPD_ON`)

LVP : bit 7 : Utilisation de la pin RB3/PGM comme broche de programmation 5V

1 : La pin RB3 permet la programmation du circuit sous tension de 5V (`_LVP_ON`)

0 : La pin RB3 est utilisée comme I/O standard (`_LVP_OFF`)

BODEN : bit 6 : provoque le reset du PIC en cas de chute de tension (surveillance de la tension d'alimentation)

1 : En service (`_BODEN_ON`)

0 : hors service (`_BODEN_OFF`)

PWRTE : bit 3 : Délai de démarrage à la mise en service. Attention, est automatiquement mis en service si le bit BODEN est positionné.

1 : délai hors service (sauf si BODEN = 1) (`_PWRTE_OFF`)

0 : délai en service (`_PWRTE_ON`)

WDTE : bit 2 : Validation du Watchdog timer

1 : WDT en service (`_WDT_ON`)

0 : WDT hors service (`_WDT_OFF`)

FOSC1/FOSC0 : bits 1/0 : sélection du type d'oscillateur

11 : Oscillateur de type RC (`_RC_OSC`) (3K < R < 100k, C > 20 pF)

10 : Oscillateur haute vitesse (`_HS_OSC`) (4 Mhz à 20 Mhz)

01 : Oscillateur basse vitesse (`_XT_OSC`) (200 kHz à 4 Mhz)

00 : Oscillateur faible consommation (`_LP_OSC`) (32 k à 200 kHz)

Remarque : Attention, les microcontrôleurs avec l'extension « A » comme 16F876A ou 16F877A ont une disposition des bits de configuration légèrement différente, consultez le datasheet.

4- Le module de conversion A/N

Ce module est constitué d'un convertisseur Analogique Numérique 10 bits dont l'entrée analogique peut être connectée sur l'une des 8 (5 pour 16F876) entrées analogiques externes.

Les entrées analogiques doivent être configurées en entrée à l'aide des registres TRISA et/ou TRISE. L'échantillonneur bloqueur est intégré, il est constitué d'un interrupteur d'échantillonnage et d'une capacité de blocage de 120 pF.

Les tensions de références permettant de fixer la dynamique du convertisseur. Elles peuvent être choisies parmi Vdd, Vss, Vref+ ou Vref-

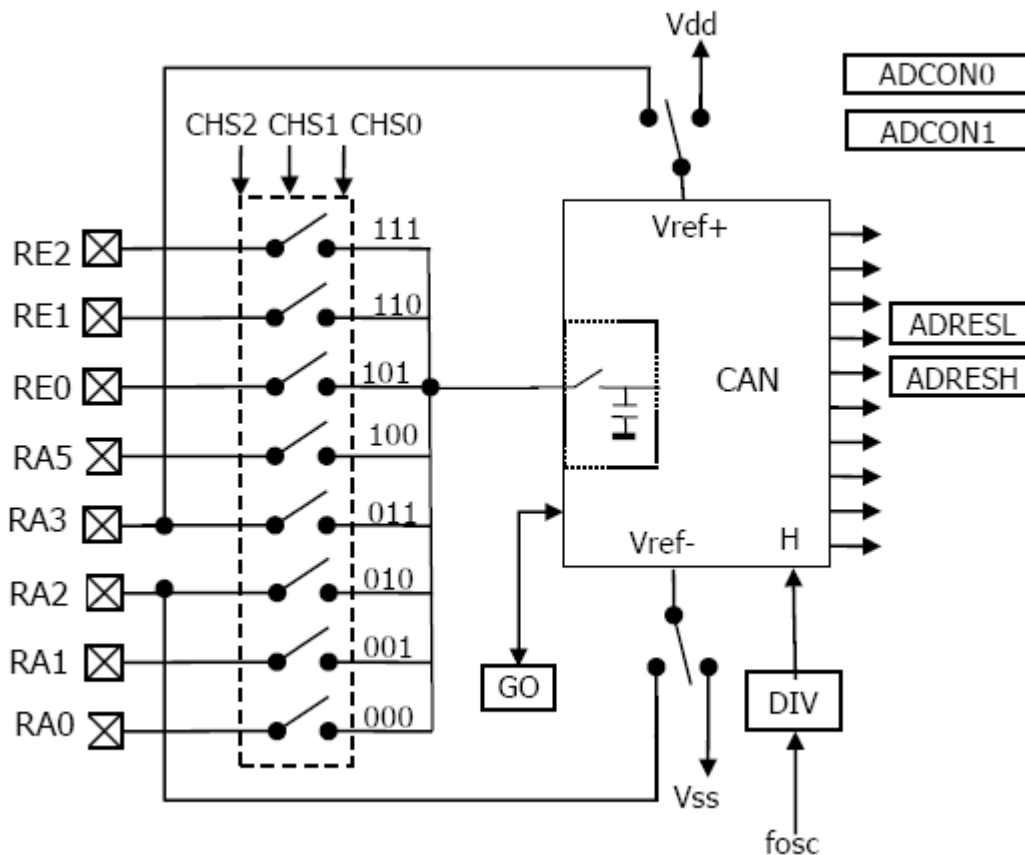


Figure I-6 : module CAN du pic 16F877

Le control du module se fait par les deux registres ADCON0 et ADCON1.

ADCON0	ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/DONE	—	ADON
---------------	-------	-------	------	------	------	---------	---	------

ADCS1:ADCS0 : Choix de l'horloge de conversion donc du temps de conversion

00 : Fosc/2

01 : Fosc/8

10 : Fosc/32

11 : Oscillateur RC dédié au CAN

CHS2:CHS0 : choix de l'entrée analogique

000 = channel 0, (RA0)

001 = channel 1, (RA1)

010 = channel 2, (RA2)
 011 = channel 3, (RA3)
 100 = channel 4, (RA5)
 101 = channel 5, (RE0)
 110 = channel 6, (RE1)
 111 = channel 7, (RE2)

GO/DONE : Une conversion démarre quand on place ce bit à 1. A la fin de la conversion, il est remis automatiquement à zéro. Ce bit peut aussi être positionné automatiquement par le module CCP2.

ADON : Ce bit permet de mettre le module AN en service

ADCON1	ADFM	—	—	—	PCFG3	PCFG2	PCFG1	PCFG0
---------------	------	---	---	---	-------	-------	-------	-------

ADFM : justification à droite ou à gauche du résultat dans les registre ADRESH et ADRESL

	ADRESH	ADRESL
1 : justifié à droite	0 0 0 0 0 XX	XXXXXXXXX
0 : justifié à gauche	XXXXXXXXX	XX 0 0 0 0 0 0

PCFG3:PCFG0 : configuration des E/S et des tensions de références.

Les 5 broches de PORTA et les 3 de PORTE peuvent être configurés soit en E/S digitales, soit en entrées analogiques. RA2 et RA3 peuvent aussi être configurées en entrée de référence.

PCFG3:PCFG0	RE2	RE1	RE0	RA5	RA3	RA2	RA1	RA0	VREF+	Vref-	A/R/N
0000	A	A	A	A	A	A	A	A	VDD	VSS	8/0/0
0001	A	A	A	A	Vref+	A	A	A	RA3	VSS	7/1/0
0010	N	N	N	A	A	A	A	A	VDD	VSS	5/0/3
0011	N	N	N	A	VREF+	A	A	A	RA3	VSS	4/1/3
0100	N	N	N	N	A	N	A	A	VDD	VSS	3/0/5
0101	N	N	N	N	VREF+	N	A	A	RA3	VSS	2/1/5
011x	N	N	N	N	N	N	N	N	VDD	VSS	0/0/8
1000	A	A	A	A	VREF+	VREF-	A	A	RA3	RA2	6/2/0
1001	N	N	A	A	A	A	A	A	VDD	VSS	6/0/2
1010	N	N	A	A	VREF+	A	A	A	RA3	VSS	5/1/2
1011	N	N	A	A	VREF+	VREF-	A	A	RA3	RA2	4/2/2
1100	N	N	N	A	VREF+	VREF-	A	A	RA3	RA2	3/2/3
1101	N	N	N	N	VREF+	VREF-	A	A	RA3	RA2	2/2/4
1110	N	N	N	N	N	N	N	A	VDD	VSS	1/0/7
1111	N	N	N	N	VREF+	VREF-	N	A	RA3	RA2	1/2/5

Tableau I.4 : tableau de configuration des E/S et des tensions de référence du pic 16F877

Pour les utiliser en E/S numériques, il faut écrire '00000110' dans le registre ADCON1.

5- Les ports d'E/S

Le port d' E/S PORTA

Le port A désigné par PORTA est un port de 6 bits (RA0 à RA5). RA6 et RA7 ne sont pas accessibles.

La configuration de direction se fait à l'aide du registre TRISA :

Bit i de TRISA = 0 \rightarrow bit i de PORTA configuré en **sortie**

Bit i de TRISA = 1 \rightarrow bit i de PORTA configuré en **entrée**

En entrée, la broche RA4 peut être utilisée soit comme E/S numérique normale, soit comme entrée horloge pour le Timer TMR0.

En sortie, RA4 est une E/S à drain ouvert, pour l'utiliser comme sortie logique, il faut ajouter une résistance de pull-up externe. Le schéma (Fig. I.7) illustre le principe d'une sortie drain ouvert (ou collecteur ouvert) :

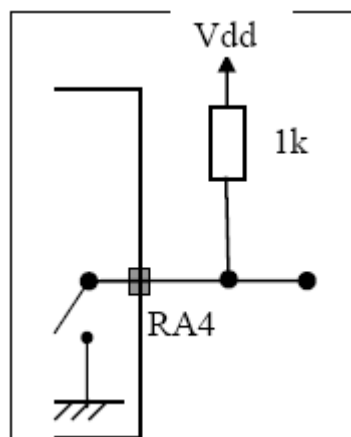


Fig. I.7 : résistance de pull-up

Si RA4 est positionnée à 0, l'interrupteur est fermé, la sortie est reliée à la masse, c'est un niveau bas. Si RA4 est placée à 1, l'interrupteur est ouvert, la sortie est positionnée en niveau haut grâce à la résistance externe qui place la sortie au niveau haut.

Si on veut utiliser RA4 pour allumer une LED, on peut utiliser le schéma de Fig. I.8. Il faut juste remarquer que la logique est inversée, si on envoie 0 sur RA4, l'interrupteur se ferme et la LED s'allume. Si on envoie 1, l'interrupteur s'ouvre et la LED s'éteint.

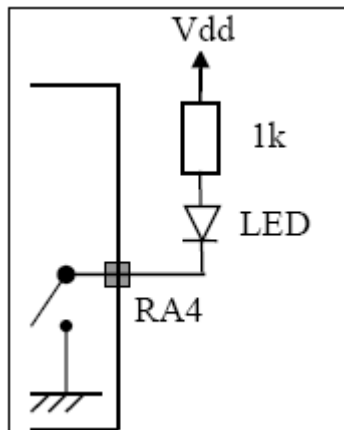


Fig. I.8 : LED sur RA4

Les autres broches de PORTA

Les autres broches (RA0, RA1, RA2, RA3 et RA5) peuvent être utilisées soit comme E/S numériques soit comme entrées analogiques.

Au RESET, ces E/S sont configurées en entrées **analogiques**. Pour les utiliser en E/S numériques, il faut écrire '00000110' dans le registre ADCON1.

Pour utiliser PORTA en port Numérique (normal), il faut placer 06h dans le registre ADCON1 (bank1)

Quelque soit le mode (Analogique ou Numérique), il faut utiliser le registre TRISA pour configurer la direction des E/S :

Le port d' E/S PORTB

Il comporte 8 bits. Le registre de direction correspondant est TRISB.

Si on écrit un "1" dans le registre TRISB, le driver de sortie correspondant passe en haute impédance. Si on écrit un "0", le contenu du Latch de sortie correspondant est recopié sur la broche de sortie.

Chaque broche du PORT B est munie d'un tirage au +VDD que l'on peut mettre ou non en service en mode entrée uniquement. On active cette fonction par la mise à "0" du bit 7 dans le registre OPTION en h'81'.

Au reset, le tirage est désactivé. Il est inactif quand le port est configuré en sortie.

Les 4 broches PB7 PB6 PB5 et PB4 provoquent une interruption sur un changement d'état si elles sont configurées en ENTREE.

On doit remettre à zéro le Flag de cette interruption (bit 0 du registre INTCON en h'0B') dans le programme d'interruption.

Cette possibilité d'interruption sur un changement d'état associé à la fonction de tirage configurable sur ces 4 broches, permet l'interfaçage facile avec un clavier. Cela rend possible le réveil du PIC en mode SLEEP par un appui sur une touche du clavier.

Le bit 0 du PORT B peut également être utilisé comme entrée d'interruption externe. Le choix du front de déclenchement se fait en configurant le bit 6 du registre OPTION.

Le port d' E/S PORTC

Il s'agit d'un PORT 8 bits bidirectionnel.

- Il est partagé avec le module de transmission synchrone I2C et l'USART.
- La configuration de direction se fait à l'aide du registre TRISC, positionner un bit de TRISC à 1 configure la broche correspondante de PORTC en entrée et inversement. Au départ toutes les broches sont configurées en entrée.

Le port d' E/S PORTD

- Le port D désigné par PORTD est un port bidirectionnel de 8 bits (RD0 à RD7). Toutes les broches sont compatibles TTL et ont la fonction trigger de Schmitt en entrée.
- Chaque broche est configurable en entrée ou en sortie à l'aide du registre TRISD. Pour configurer une broche en entrée, on positionne le bit correspondant dans TRISD à 1 et inversement.
- PORTD n'est pas implémenté sur 16F876, il est disponible sur le 16F877.
- PORTD peut être utilisé dans un mode particulier appelé *parallel slave port*, pour cela il faut placer le bit PSPMODE (bit 4) de TRISE à 1. Dans ce cas les 3 bits de PORTE deviennent les entrées de control de ce port (RE, WE et CS)

Pour utiliser PORTD en mode normal, il faut placer le bit PSPMODE de TRISE à 0

Le port d' E/S PORTE

- PORTE contient seulement 3 bits RE0, RE1 et RE2. Les 3 sont configurables en entrée ou en sortie à l'aide des bits 0, 1 ou 2 du registre TRISE.
- PORTE n'est pas implémenté sur 16F876, il est disponible sur le 16F877.
- Les 3 bits de PORTE peuvent être utilisés soit comme E/S numérique soit comme entrées analogiques du CAN. La configuration se fait à l'aide du registre ADCON1.
- Si le bit PSPMODE de TRISE est placé à 1, Les trois bits de PORTE deviennent les entrées de control du PORTD qui (dans ce cas) fonctionne en mode *parallel Slave mode*
- A la mise sous tension (RESET), les 3 broches de PORTE sont configurées comme entrées analogiques.

Pour utiliser les broches de PORTE en E/S numériques normales :

- ***Placer 06h dans ADCON1***
- ***Placer le bit PSPMODE de TRISE à 0***

6- Les interruptions

Une interruption provoque l'arrêt du programme principal pour aller exécuter une procédure d'interruption. A la fin de cette procédure, le microcontrôleur reprend le programme principal à l'endroit où il l'a laissé. A chaque interruption sont associés deux bits, un bit de validation et un drapeau. Le premier permet d'autoriser ou non l'interruption, le second permet au programmeur de savoir de quelle interruption il s'agit. Sur le 16F876/877, les interruptions sont classées en deux catégories, les interruptions primaires et les interruptions périphériques. Elles sont gérées par les registres :

INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF
PIE1 (bk1)	PSPIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE
PIR1 (bk0)	PSPIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF
PIE2 (bk0)	-	-	-	EEIE	BCLIE	-	-	CCP2IE
PIR2 (bk1)	-	-	-	EEIF	BCLIF	-	-	CCP2IF
OPTION_REG(bk1)		INTEDG						

- Toutes les interruptions peuvent être validées/interdites par le bit INTCON.GIE
- Toutes les interruptions périphériques peuvent être validées/interdites par le bit INTCON.PEIE
- Chaque interruption peut être validée/interdite par son bit de validation individuel

En résumé, pour valider une interruption périphérique (par exemple), il faut positionner 3 bits, GIE, PEIE et le bit individuel de l'interruption.

Les sources d'interruption

Interruption : Source d'interruption	Validation	Flag	PEIE
T0I : Débordement Timer 0	INTCON,T0IE	INTCON,T0IF	non
INT : Front sur RB0/INT	INTCON,INTE	INTCON,INTF	non
RBI : Front sur RB4-RB7	INTCON,RBIE	INTCON,RBIF	non
ADI : Fin de conversion A/N	PIE1,ADIE	PIR1,ADIF	oui
RCI : Un Octet est reçu sur l'USART	PIE1,RCIE	PIR1,RCIF	oui
TXI : Fin transmission d'un octet sur l'USART	PIE1,TXIE	PIR1,TXIF	oui
SSPI : Caractère émis/reçu sur port série synchrone	PIE1,SSPIE	PIR1,SSPIF	oui
TMR1I : Débordement de Timer 1	PIE1,TMR1IE	PIR1,TMR1IF	oui
TMR2I : Timer 2 a atteint la valeur programmée	PIE1,TMR2IE	PIR1,TMR2IF	oui
PSPI : Lecture/écriture terminée sur Port parallèle (16F877)	PIE1,PSPIE	PIR1,PSPIF	oui
CCP1I : Capture/comparaison de TMR1 avec module CCP1	PIE1,CCP1IE	PIR1,CCP1IF	oui
CCP2I : Capture/comparaison de TMR1 avec module CCP2	PIE2,CCP2IE	PIR2,CCP2IF	oui
EEI : Fin d'écriture en EEPROM	PIE2,EEIE	PIR2,EEIF	oui
BCLI : Collision sur bus SSP en mode I2C	PIE2,BCLIE	PIR2,BCLIF	oui

7- Les Timers

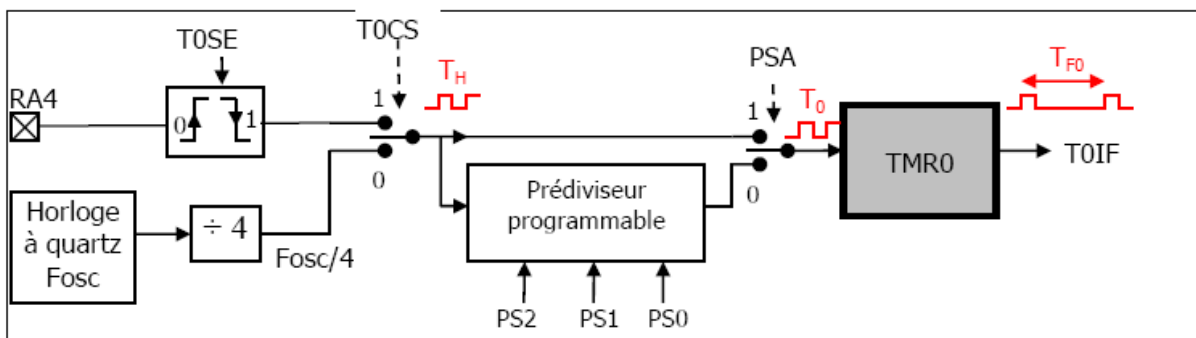
Le Timer TMR0

C'est un compteur 8 bits ayant les caractéristiques suivantes :

- Il est incrémenté en permanence soit par l'horloge interne $F_{osc}/4$ (mode timer) soit par une horloge externe appliquée à la broche RA4 du port A (mode compteur). Le choix de l'horloge se fait à l'aide du bit T0CS du registre OPTION_REG
 - T0CS = 0 → horloge interne
 - T0CS = 1 → horloge externe appliquée à RA4
- Dans le cas de l'horloge externe, Le bit T0SE du registre OPTION_REG permet de choisir le front sur lequel le TIMER s'incrémente.
 - T0SE = 0 → incrémentation sur fronts montants
 - T0SE = 1 → incrémentation sur fronts descendants
- Quelque soit l'horloge choisie, on peut la passer dans un diviseur de fréquence programmable (prescaler) dont le rapport DIV est fixés par les bits PS0, PS1 et PS2 du registre OPTION_REG

PS2	PS1	PS0	Div
0	0	0	2
0	0	1	4
0	1	0	8
0	1	1	16
1	0	0	32
1	0	1	64
1	1	0	128
1	1	1	256

- L'affectation ou non du prédiviseur se fait à l'aide du bit PSA du registre OPTION_REG
 - PSA = 0 → on utilise le prédiviseur
 - PSA = 1 → pas de prédiviseur (affecté au chien de garde)



En résumé, chaque fois que le compteur complète un tour, le drapeau T0IF se lève. Si on note TH la période de l'horloge source, T0 l'horloge de TMR0 et TF0 le temps qui sépare 2 levés de drapeau successifs :

- **Sans prédiviseur : $TF0 = 256 T0 = 256 TH$**
- **Avec prédiviseur : $TF0 = 256 T0 = 256 \times (DIV \times TH)$**
- **Avec prédiviseur et compteur N dans le programme : $TF0 = N \times 256 \times (DIV \times TH)$**

Le Watchdog Timer WDT (Chien de garde)

C'est un compteur 8 bits incrémenté en permanence (même si le μC est en mode sleep) par une horloge RC intégrée indépendante de l'horloge système. Lorsqu'il déborde, (WDT TimeOut), deux situations sont possibles :

- Si le μC est en fonctionnement normal, le WDT time-out provoque un RESET. Ceci permet d'éviter de rester planté en cas de blocage du microcontrôleur par un processus indésirable non contrôlé
- Si le μC est en mode SLEEP, le WDT time-out provoque un WAKE-UP, l'exécution du programme continue normalement là où elle s'est arrêtée avant de rentrer en mode SLEEP. Cette situation est souvent exploitée pour réaliser des temporisations

L'horloge du WDT a une période voisine de 70 μs ce donne un Time-Out toutes les 18 ms. Il est cependant possible d'augmenter cette durée en faisant passer le signal Time-Out dans un prédiviseur programmable (partagé avec le timer TMR0). L'affectation se fait à l'aide du bit PSA du registre OPTION_REG

Le rapport du prédiviseur est fixé par les bits PS0, PS1 et PS2 du registre OPTION_REG (voir tableau ci-contre)

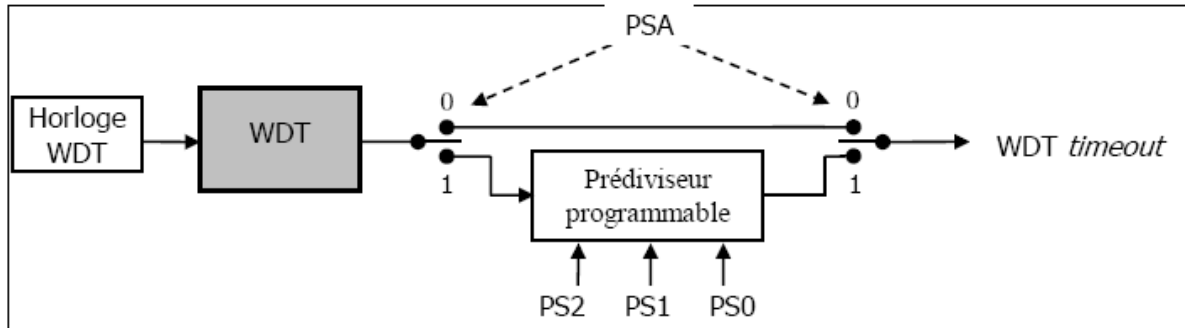
- PSA = 1 → on utilise le prédiviseur
- PSA = 0 → pas de prédiviseur (affecté à TMR0)

Le rapport du prédiviseur est fixé par les bits PS0, PS1 et PS2 du registre OPTION_REG (voir tableau ci-dessous)

PS2	PS1	PS0	Div
0	0	0	1
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

L'utilisation du WDT doit se faire avec précaution pour éviter la réinitialisation (inattendue) répétée du programme. Pour éviter un WDT timeOut lors de l'exécution d'un programme, on a deux possibilités :

- Inhiber le WDT d'une façon permanente en mettant à 0 le bit WDTE dans l'EEPROM de configuration
- Remettre le WDT à 0 périodiquement dans le programme à l'aide de l'instruction CLRWDT pour éviter qu'il ne déborde



Le Timer TMR1

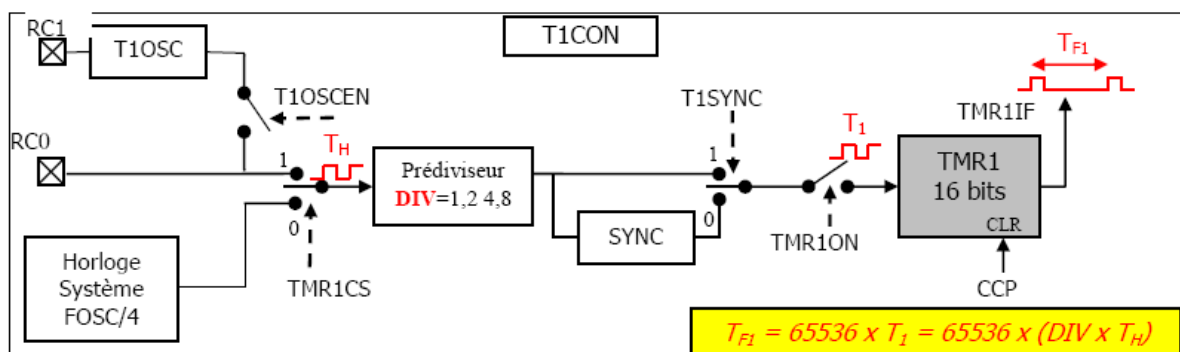
TMR1 est un Timer/Compteur 16 bits accessible en lecture/écriture par l'intermédiaire des registres 8 bits TMR1H (bank0) et TMR1L (bank0) qui constituent sa partie haute et sa partie basse.

Le registre TMR1 (constitué de TMR1H et TMR1L) s'incrémente de h'0000' jusqu'à h'FFFF' et repasse ensuite à h'0000' pour continuer le comptage.

Ce module peut fonctionner en mode TIMER, quand il s'incrémente à chaque cycle instruction ($F_{osc}/4$ avec le pré diviseur considéré à "1") ou en mode compteur, quand il s'incrémente à chaque front montant de l'horloge externe appliquée sur le Port C0.

L'horloge externe peut également être l'oscillateur interne, dont la fréquence est fixée par un quartz externe branché entre la broche Port C0 et la broche Port C1.

Le contrôle du TIMER 1 se fait par le registre T1CON en h'10' (bank0).



Le registre de control de T1CON

—	—	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON
---	---	---------	---------	---------	--------	--------	--------

T1CKPS1, T1CKPS0 : Control du prescaler

00 : division par 1

01 : division par 2

10 : division par 4

11 : division par 8

T1OSCEN : Validation de l'Oscillateur associé à TMR1

0 : Oscillateur arrêté

1 : Oscillateur activé

T1SYNC : Synchronisation de l'horloge externe (ignoré en mode timer)

0 : Synchronisation

1 : pas de synchronisation

TMR1CS : Choix de l'horloge du Timer

0 : horloge système (Fosc/4) : mode timer

1 : Horloge externe : mode compteur

TMR1ON : Démarrer arrêter le timer

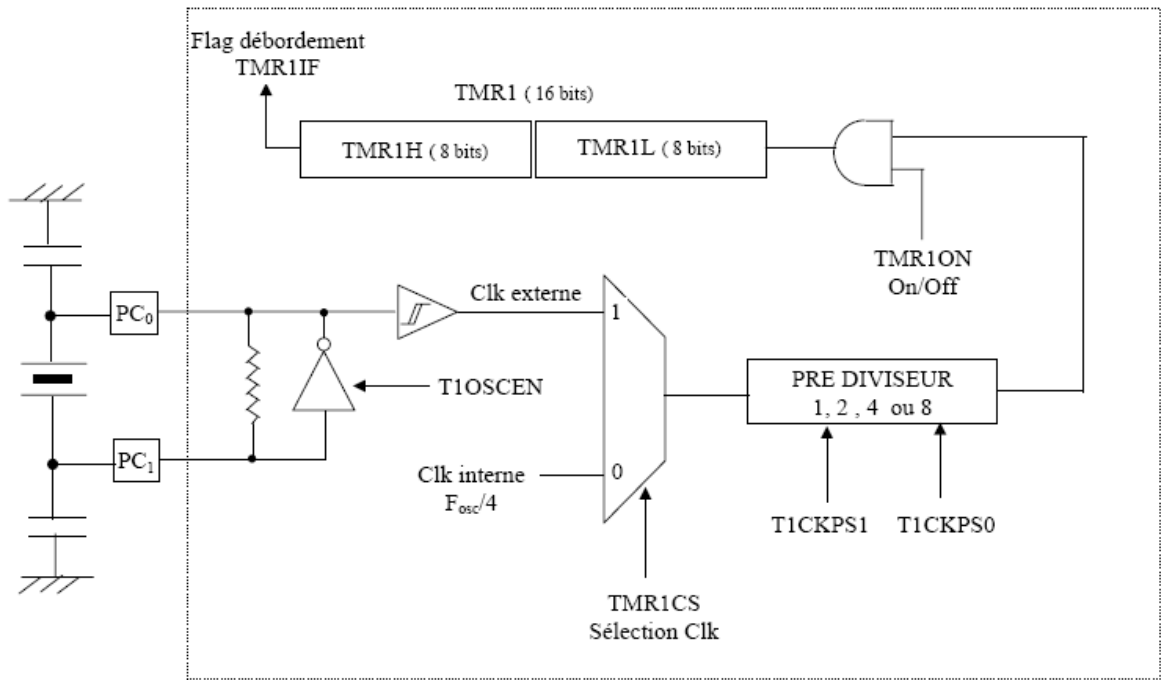
0 : Timer stoppé

1 : Timer en fonctionnement

OSCILLATEUR INTERNE du TIMER 1 :

Un oscillateur à quartz a été embarqué sur le chip. Il est branché entre les broches PC0 (oscillateur out) et PC1 (oscillateur in). Il est mis en service par la mise à "1" du bit T1OSCEN. Cet oscillateur à faible consommation est limité à 200 KHz. Il continue à osciller en mode SLEEP du PIC.

Il est principalement destiné pour générer un événement temps réel toutes les secondes par utilisation d'un quartz 32,768 KHz.



MODULE TIMER 2 :

Le module Timer 2 est un compteur 8 bits avec pré diviseur et post diviseur.

Ce compteur TMR2 en h'11 ' page 0 est un registre en lecture ou écriture. Il possède un registre 8 bits pour la période : PR2 en h'92' page 1. Le compteur s'incrémente de h'00' jusqu'à la valeur contenue par PR2 et repasse ensuite à "0" pour continuer le comptage. Au reset PR2 est initialisé à "FF".

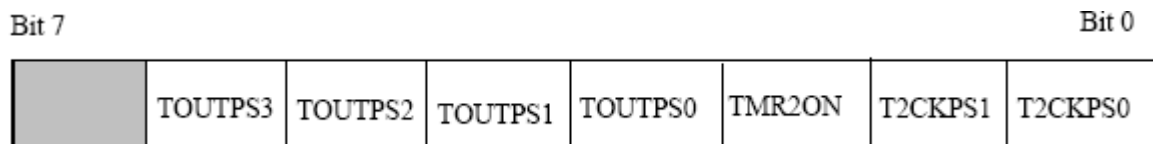
L'entrée du compteur est l'horloge cycle interne : Fosc/4 qui passe à travers un pré diviseur programmable par 1, 4 ou 16.

La sortie du compteur passe dans un post diviseur programmable sur 4 bits entre 1 et 16.

Quand la sortie du compteur passe par la valeur programmée dans PR2, il y a génération d'une interruption (si elle a été autorisée par TMR2IE=1) et le flag TMR2IF est positionné à "1". Ceci bien entendu en considérant le post diviseur programmé à "1".

Le contrôle du Timer 2 se fait par le registre T2CON en h'12' page 0.

T2CON :



Au reset : T2CON = 00000000

bit 7 : bit non implémenté.

bit 6 à bit 3 : TOUTPS : Programmation du Post diviseur.

0 0 0 0 = post divise par 1.

0 0 0 1 = post divise par 2.

0 0 1 0 = post divise par 3.

.....

....
 1 1 1 1 = post divise par 16.
bit 2 : **TMR2ON** : mise en service du Timer 2.
 1= Timer 2 : On.
 0= Timer 2 : Off.
bit 1 et bit0 : **T2CKPS** : Programmation du pré diviseur.
 0 0= pré divise par 1.
 0 1= pré divise par 4.
 1 X= pré divise par 16.

MODULE CCP : CAPTURE COMPARE et PWM

Il y a deux modules identiques CCP1 et CCP2 composés chacun d'un registre 16 bits. Ils peuvent opérer soit comme un registre 16 bits de capture, soit comme un registre 16 bits de comparaison, soit enfin comme un registre 8 bits pour générer du PWM.

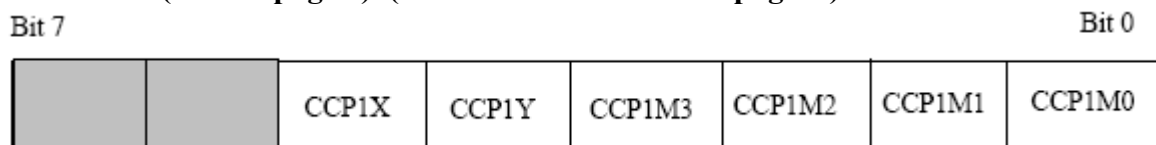
Le module CCP1 est constitué de deux registres de 8 bits : CCPR1L en h'15' page 0 et CCPR1H en h'16' page 0. Ce module est contrôlé par le registre CCP1CON en h'17' page 0. La sortie en mode COMPARE ou mode PWM et l'entrée en mode CAPTURE se font par la broche PC2.

Le module CCP2 est constitué de deux registres de 8 bits : CCPR2L en h'1B' page 0 et CCPR2H en h'1C' page 0. Ce module est contrôlé par le registre CCP2CON en h'1D' page 0. La sortie en mode COMPARE ou mode PWM et l'entrée en mode CAPTURE se font par la broche PC1.

En mode COMPARE ou CAPTURE, les modules utilisent le TIMER 1. En mode PWM, ils utilisent le TIMER 2.

Les registres de contrôles CCP1CON et CCP2CON sont identiques. On ne décrira que CCP1CON.

CCP1CON : (h'17' : page 0). (et CCP2CON en h'1D' : page 0.)



Au reset : CCP1CON = 00000000

bit 7 et bit 6 : **bits non implémentés.**

bit 5 et bit 4 : **CCP1X et CCP1Y :**

Bits non utilisés en modes Compare et Capture.

Ce sont les 2 bits LSB pour le Duty cycle en mode PWM. Les 8 bits MSB sont dans le registre CCPR1L en h'15' page 0.

bit 3 à bit 0 : **CCP1M3 à CCP1M0** : bits de sélection du mode.

0 0 0 0 = Module CCP stoppé.

0 1 0 0 = Mode Capture à chaque front descendant.

0 1 0 1 = Mode Capture à chaque front montant.

0 1 1 0 = Mode Capture tous les 4 fronts montants.

0 1 1 1 = Mode Capture tous les 16 fronts montants.

1 0 0 0 = Mode Compare. Pin de sortie mise à "1" et Flag CCP1IF = 1 à l'égalité.

1 0 0 1 = Mode Compare. Pin de sortie mise à "0" et Flag CCP1IF = 1 à l'égalité.

1 0 1 0 = Mode Compare. Génération d'une Interrup. et Flag CCP1IF = 1 à l'égalité.
1 0 1 1 = Mode Compare. Evénement spécial généré et Flag CCP1IF = 1 à l'égalité.
1 1 x x = Mode PWM.

MODE COMPARE :

Les deux modules CCP étant identiques on ne décrira que le module 1.

Les 16 bits des registres CCPR1 (CCPR1H et CCPR1L) sont constamment comparés avec le valeur sur 16 bits des registres du Timer 1 (TMR1H et TMR1L). Quand il y a égalité, la broche préalablement programmée en sortie PC2, passe soit à "1" soit à "0" suivant la configuration des 4 bits CCP1M du registre CCP1CON. Au même instant le Flag CCP1IF est mis à "1".

En mode Compare, les événements spéciaux générés quand il y a égalité sont:

- Pour CCP1: reset du Timer 1.
- Pour CCP2 : reset du Timer 1 et démarrage d'une conversion A/D.

Dans ce cas la broche de sortie n'est pas affectée, mais le Flag CCP1IF est mis à "1". Il est rappelé que ce Flag doit être remis à "0" par soft.

MODE CAPTURE :

Quand un événement extérieur apparaît sur la broche préalablement programmée en entrée PC2, la valeur des 16 bits des registres du Timer 1 (TMR1L et TMR1H) est recopiée dans les registres CCPR1 (CCPR1H et CCPR1L). Cet événement est programmable par les 4 bits CCP1M du registre CCP1CON. La capture peut avoir lieu à chaque front descendant, à chaque front montant, tous les 4 ou tous les 16 fronts montants.

Quand la capture a eu lieu, le flag CCP1IF est mis à "1". Ce bit doit être remis à "0" par soft.

Si une nouvelle capture survient alors que la valeur dans CCPR1 n'a pas été lue, l'ancienne valeur est perdue.

Les fonctions de Capture et de Compare sur le Timer 1 par les modules CCP1 et CCP2 peuvent générer une interruption quand le Flag CCP1IF passe à "1" si le bit d'autorisation CCP1IE du registre PIE1 est mis à "1".

MODE PWM :

Un signal PWM est caractérisé par une période, et un temps de travail ou le signal est à "1". Ce temps est appelé DUTY CYCLE.

8- L'USART

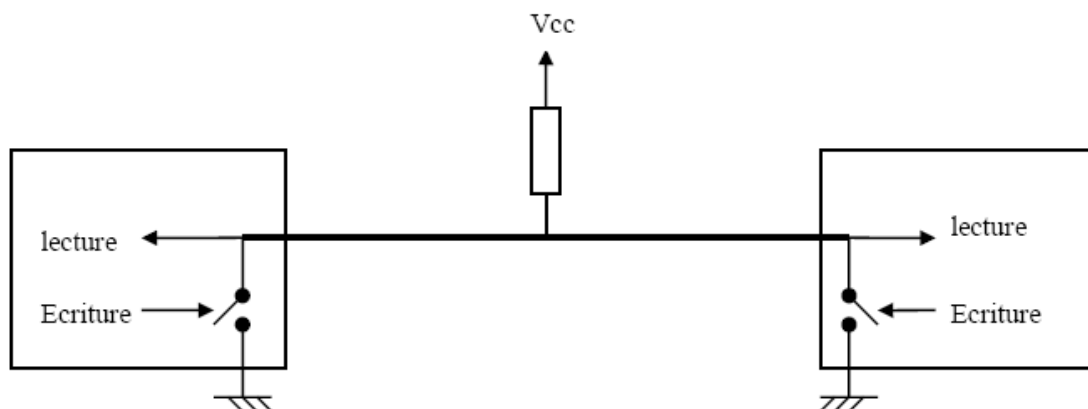
L'USART (Universal Synchronous Asynchronous Receiver Transmitter) est l'un des deux modules de communication série dont dispose le PIC 16F876/877. L'USART peut être configuré comme système de communication asynchrone full duplex ou comme système synchrone half duplex (non étudié). La communication se fait sur les deux broches RC6/TX et RC7/RX qui doivent être configurés toutes les deux en ENTREE par TRISC.

9- Le module MSSP (Master Synchronous Serial Port)

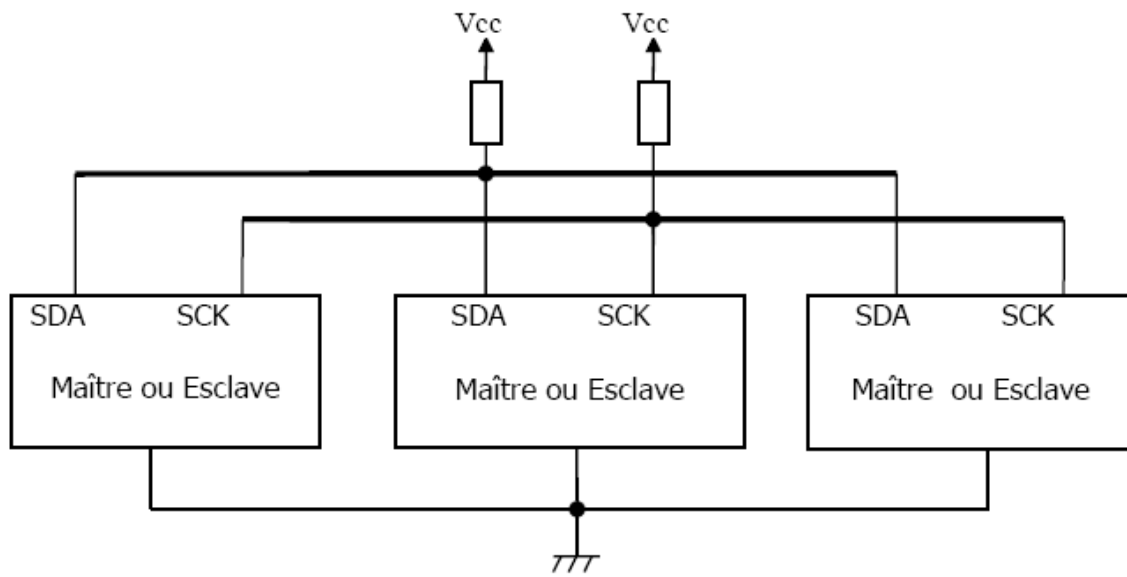
Le MSSP est une des deux modules de communication série du PIC 16F876/877. Il permet d'échanger des données en mode synchrone avec d'autres circuits qui peuvent être des microcontrôleurs, des mémoires EEPROM série, des convertisseurs A/N, des modules d'affichage . . . Il peut fonctionner selon deux modes : le mode SPI (Serial Peripheral Interface) et le mode I2C (Inter-Integrated Circuit)

Introduction au bus I2C

Avant de parler du module MSSP en mode I2C du PIC, introduisons très brièvement Le standard I2C Le bus I²C permet de faire communiquer entre eux des composants électroniques très divers grâce à seulement 3 fils : Un signal de donnée (SDA), un signal d'horloge (SCL), et un signal de référence électrique (Masse). Comme les lignes SDA et SCK sont utilisées dans les deux sens par les deux circuits qui communiquent, on peut avoir un circuit qui place la ligne à 1 (Vcc) et l'autre qui la place à 0 (masse) ce qui correspond à un court circuit qui peut détruire les deux composants. Pour éviter ce problème, les E/S SDA et SCK fonctionnent en mode collecteur ouvert (ou drain ouvert) de sorte qu'un circuit ne peut imposer que le niveau bas ou ouvrir la ligne, le niveau haut est obtenu par une résistance de tirage externe. Ainsi une ligne est à 0 quand un des deux circuits impose le 0. Elle passe à 1 quand les deux circuits imposent le 1 (circuit ouvert). Le protocole I2C jongle avec cette situation pour organiser l'échange des données entre les deux composants.



Un bus I2C peut être relié à plusieurs circuits, mais pendant une communication, un des circuits est le maître, c'est lui génère l'horloge et initie les séquences de transmission, l'autre est l'esclave, il subit l'horloge du maître sur la ligne SCK mais il peut tout de même recevoir et émettre des données sur la ligne SDA. Chaque esclave a une adresse, au début d'une séquence de communication, le maître qui initie la séquence envoie l'adresse du slave avec lequel il désire communiquer, celui-ci reconnaît son adresse et répond, les autres slaves (s'il y en a) restent muets.



Certains circuits sont fabriqués pour être des masters, d'autres des slaves et d'autres peuvent être soit l'un soit l'autre. Pour prendre le contrôle du bus, il faut que celui-ci soit au repos (SDA et SCL à '1'). Lorsqu'un circuit prend le contrôle du bus, il en devient le maître. C'est lui qui génère le signal d'horloge et c'est lui qui initie les séquences d'échange.