

Chapitre 1 – GENERALITES

1 – DEFINITION

Un microcontrôleur est un microprocesseur RISC (Reduced Instruction Set Computer) :

Le nombre d'instructions est réduit : 35 instructions pour le 16F62X

Il dispose :

- de ports d'entrée/sortie pour communiquer avec l'environnement extérieur
- de registres internes
- de mémoire
- d'une horloge interne ou externe.

Fabriqués par la société Microchip

Autres μ C : Motorola (68HC11 par exemple)
ARMLtd microcontrolleurs ARM (ARM7)
STMicroelectronics (STM32)
Intel (8085)
Atmel (AVR, AT91)
Etc ...

2 – INTERET

Faible coût

Composant programmable et effaçable.

Utilisé de façon autonome : plus besoin d'ordinateur une fois le composant programmé.

Simplification de l'électronique : gain de temps et de coût.

Les principaux domaines d'utilisation sont la robotique, la domotique, l'industrie.

3 – LES OUTILS POUR REALISER UNE APPLICATION

Pour développer une application fonctionnant à l'aide d'un microcontrôleur, il faut disposer de :

Editeur :

Ecriture du programme
en respectant la syntaxe
d'un langage

Compilateur :

Fabrique le programme
(Soft)

Programmateur :

Transfert le programme
du PC au PIC
(soft+hard)

Le compilateur : Logiciel traduisant un programme écrit dans un langage donné (C, basic, assembleur) en langage machine.

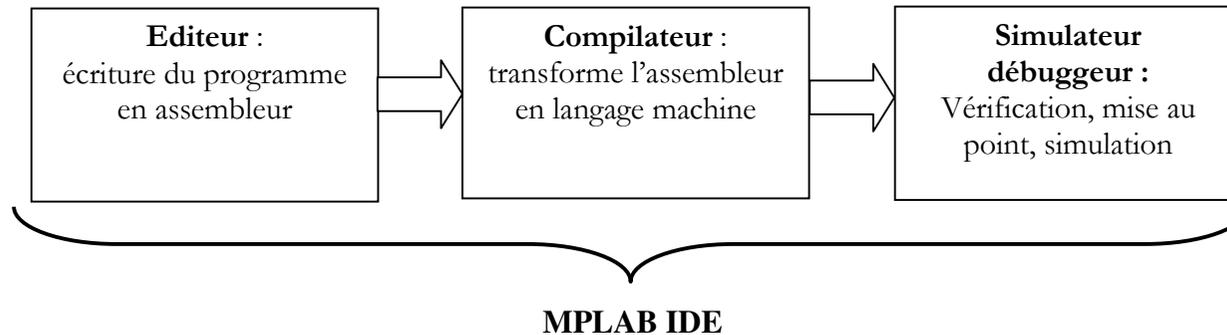
Le programmeur : Transfert le programme compilé (langage machine) dans la mémoire du microcontrôleur.

Il est constitué d'un circuit branché sur le port série du PC, sur lequel on implante le PIC, et d'un logiciel permettant d'assurer le transfert. Il existe différents logiciels, nous utiliserons Icprog.

Les microcontrôleurs PIC utilisent la plate-forme logiciel de développement MPLAB IDE.
(gratuite téléchargeable sur www.microchip.com)

IDE : Integrated Development Environment => Environnement de Développement Intégré.

Environnement comportant : Un éditeur de texte, un compilateur, un débogueur.



4 - LANGAGE DE PROGRAMMATION UTILISE

Dans l'environnement MPLAB, Le programme doit être écrit en assembleur, langage proche de la machine et donc nécessitant un long apprentissage.

Nous avons donc préféré un langage de programmation évolué : basic ou c.
Notre choix se porte sur le langage c étudié par ailleurs en cours d'informatique d'instrumentation.

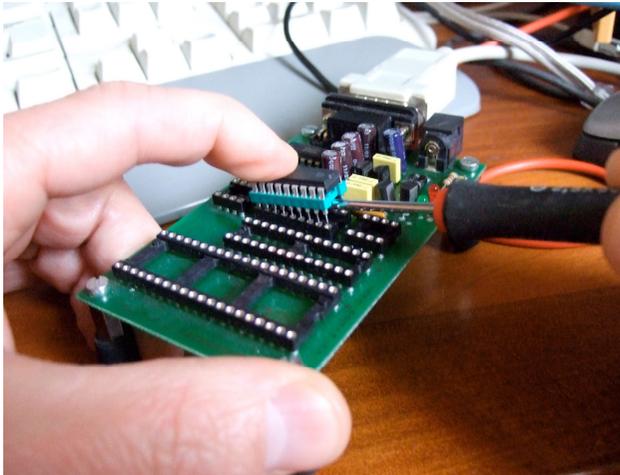
Le code source écrit en langage c doit donc être compilé en assembleur à l'aide d'un compilateur c.

On utilisera le compilateur CC5X dans sa version gratuite téléchargeable sur www.bknd.com. Cette version gratuite permet d'écrire environ 1ko de programme.

On peut alors intégrer CC5X dans l'environnement MPLAB. Ainsi CC5X devient un outil de MPLAB dans lequel l'écriture, la simulation et le debugging du programme en c devient alors possible.

5 – PRESENTATION

5.1 – BOITIER



Boîtier Plastique DIL (Dual in Line) à 18 pattes

Le boîtier intègre un μ C :

- Un processeur
- Des périphériques

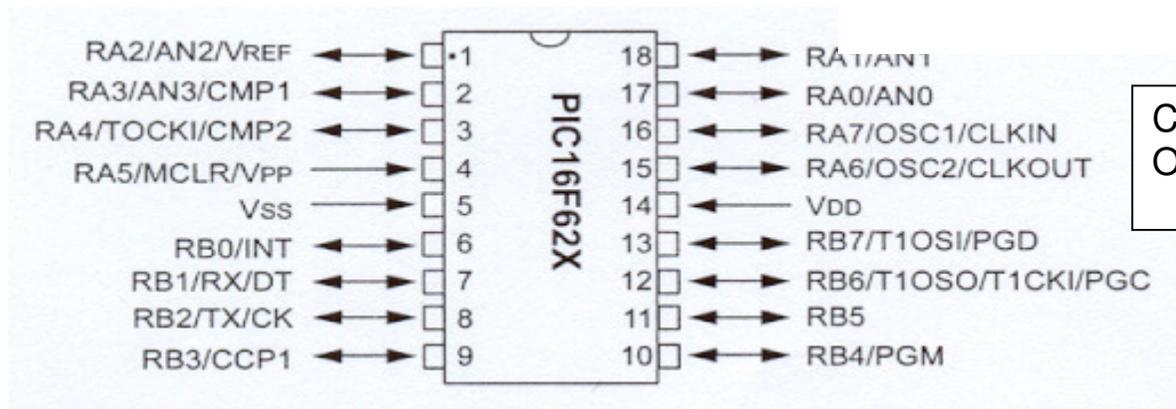
Technologie CMOS

Cadencé par une horloge :

- 4 MHz interne
- 4 à 20 MHz externe (à Quartz ou Circuit RC)

Alimentation continue : 3,5 à 5,5 V

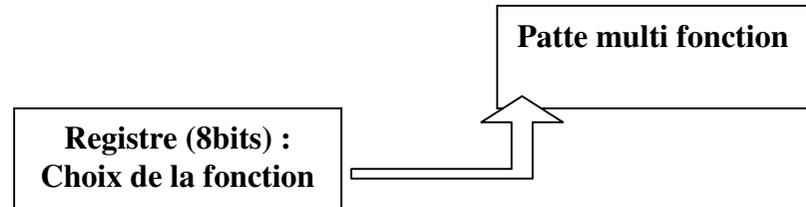
5.2 - BROCHAGE



Certaines pattes ont plusieurs fonctions :
On dit que les fonctions sont multiplexées

5.3 – REGISTRE

N. B : La fonction que l'on choisit d'utiliser pour une patte est définie par programmation en donnant une valeur précise à un registre spécifiquement prévu à cet usage.



*Lire la doc ...
en anglais !*

Un registre est un emplacement de mémoire interne au μ P

Exemple : Le Registre TRISA permet de définir si les différentes pattes du port A fonctionnent en entrée ou en sortie.

5.4 - MEMOIRE

L'ensemble de la mémoire est composée de :

Mémoire programme (type FLASH) : 2048 mots

Mémoire FLASH : Mémoire de type EEPROM => Electrically Erasable Programmable Read Only Memory

C'est une EEPROM rapide
Effaçable par secteurs entiers
Utilisée aussi pour les BIOS des cartes mères, le
stockage des adresses mac (media access control),
les cartes SD, USB, Compact flash ...

EPROM effaçable par uv

EEPROM : Effaçable électriquement in situ

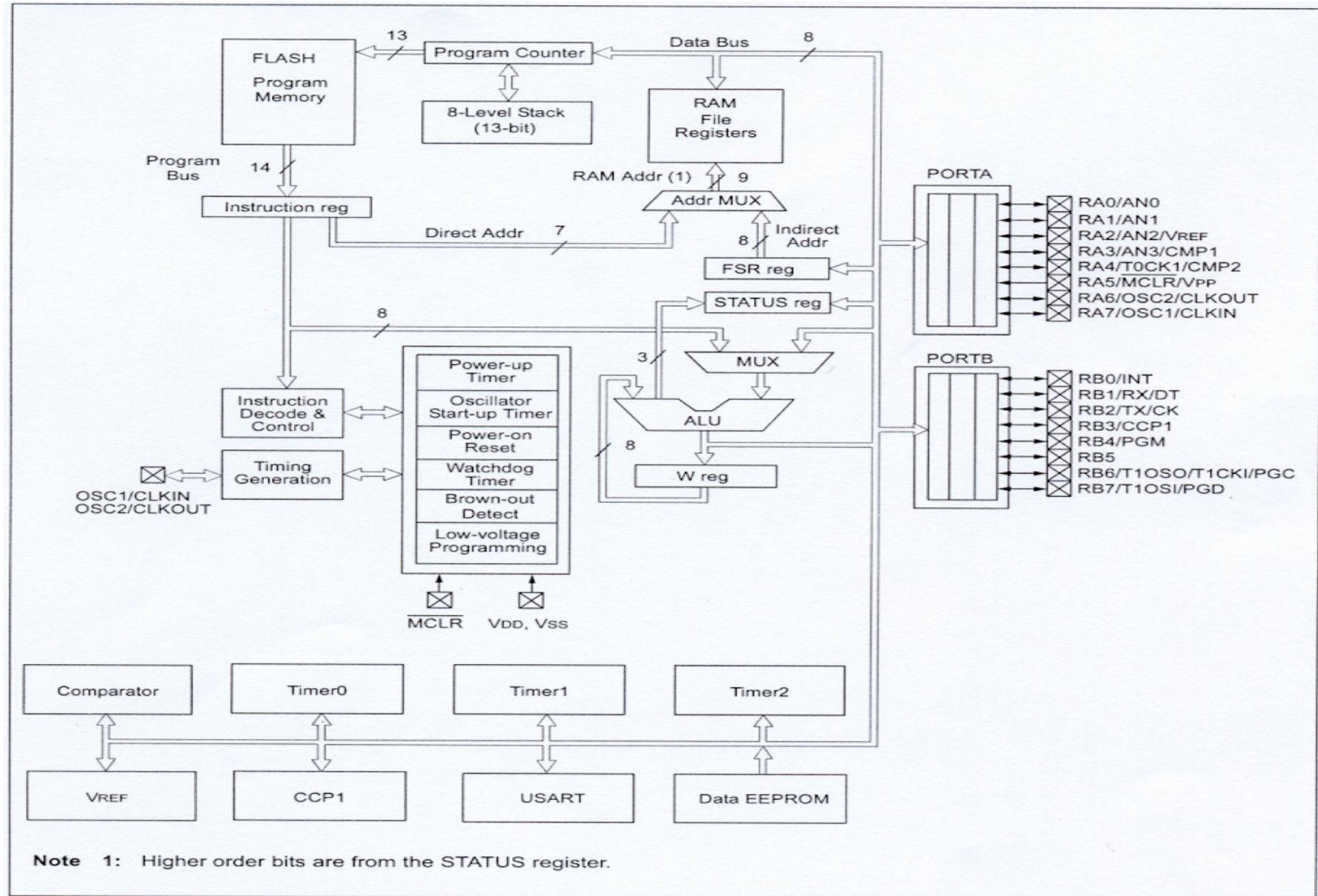
Mémoire RAM de données : 224 octets

Mémoire EEPROM de données : 128 octets

5.5 - PERIPHERIQUES

- 3 modules timers
- 2 comparateurs
- 1 périphérique Capture/Compare/PWM
- 1 USART (Universal Synchronous & Asynchronous Receiver Transmitter)
- 1 Tension de référence interne

5.6 – STRUCTURE INTERNE (Pour Information)



Chapitre 2 – PORTS D'ENTREES/SORTIES : Utilisation en Entrée.

1 – DESCRIPTION

Ce microcontrôleur dispose de deux ports bidirectionnels d'E/S : PORTA et PORTB de 8 bits

Ces ports peuvent servir d'E/S standard ou d'E/S de périphériques.

En effet, certaines pattes de ces ports sont multiplexées avec d'autres fonctions de périphériques internes (comparateur et référence de tension par exemple).

Chaque borne du port a plusieurs fonctions possibles.

Les registres de configuration associés permettent de définir les fonctions souhaitées.

Quand le périphérique est activé, la borne ne peut plus être utilisée en E/S.

2 – PORT A

2.1 – FONCTION MULTIPLEXES

Le tableau ci-dessous décrit les différentes fonctions multiplexées sur le port A.

Name	Function	Input Type	Output Type	Description
RA0/AN0	RA0	ST	CMOS	Bi-directional I/O port
	AN0	AN	—	Analog comparator input
RA1/AN1	RA1	ST	CMOS	Bi-directional I/O port
	AN1	AN	—	Analog comparator input
RA2/AN2/VREF	RA2	ST	CMOS	Bi-directional I/O port
	AN2	AN	—	Analog comparator input
	VREF	—	AN	VREF output
RA3/AN3/CMP1	RA3	ST	CMOS	Bi-directional I/O port
	AN3	AN	—	Analog comparator input
	CMP1	—	CMOS	Comparator 1 output
RA4/T0CKI/CMP2	RA4	ST	OD	Bi-directional I/O port
	T0CKI	ST	—	External clock input for TMR0 or comparator output. Output is open drain type
	CMP2	—	OD	Comparator 2 output
RA5/ $\overline{\text{MCLR}}$ /VPP	RA5	ST	—	Input port
	$\overline{\text{MCLR}}$	ST	—	Master clear
	VPP	HV	—	Programming voltage input. When configured as $\overline{\text{MCLR}}$, this pin is an active low RESET to the device. Voltage on $\overline{\text{MCLR}}$ /VPP must not exceed VDD during normal device operation
RA6/OSC2/CLKOUT	RA6	ST	CMOS	Bi-directional I/O port.
	OSC2	XTAL	—	Oscillator crystal output. Connects to crystal resonator in Crystal Oscillator mode.
	CLKOUT	—	CMOS	In ER/INTRC mode, OSC2 pin can output CLKOUT, which has 1/4 the frequency of OSC1
RA7/OSC1/CLKIN	RA7	ST	CMOS	Bi-directional I/O port
	OSC1	XTAL	—	Oscillator crystal input
	CLKIN	ST	—	External clock source input. ER biasing pin.

Legend: ST = Schmitt Trigger input HV = High Voltage OD = Open Drain AN = Analog

Certaines de ces fonctions seront examinées dans les paragraphes suivants.

2.2 - UTILISATION EN ENTREES SORTIES NUMERIQUES

a) Registre CMCON

Les pattes du port A étant multiplexées avec les entrées du comparateur, il convient de définir leur rôle grâce au registre CMCON (Comparator Control Register) registre de contrôle du comparateur.

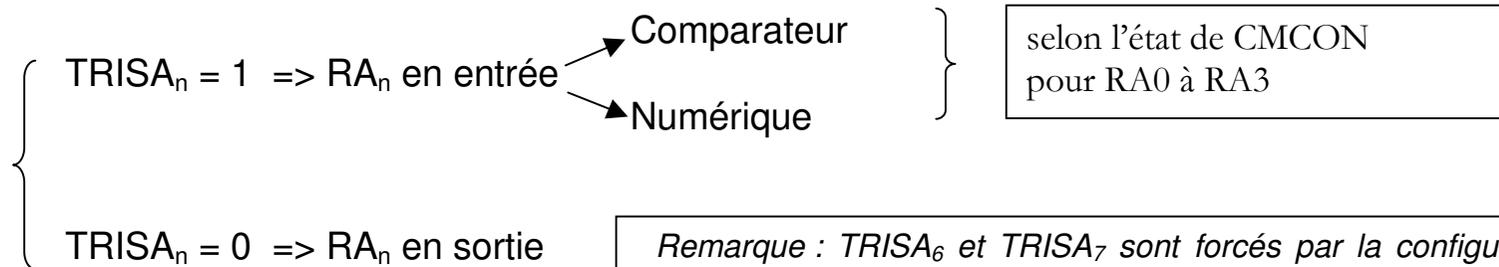
On doit avoir : $CMCON = 0b00000111 = 0x07 = 7$ pour forcer toutes les entrées en entrées numériques.

b) Registre TRISA

Ce registre permet de définir si la patte considérée fonctionne en entrée ou en sortie.

Un « 1 » dans un bit du registre TRISA met la sortie correspondante en haute impédance, elle peut ainsi servir d'entrée.

Un « 0 » dans un bit de ce registre transfère le contenu de la sortie de la bascule D sur la sortie correspondante.



Remarque : $TRISA_6$ et $TRISA_7$ sont forcés par la configuration de l'oscillateur. Dans ce cas la donnée lue est « 0 » et ces deux bits sont alors ignorés.

Remarque : Toute opération d'écriture sur une des sorties est précédée d'une lecture de la patte correspondante.

3 – PORT B

3.1 – FONCTION MULTIPLEXES

Name	Function	Input Type	Output Type	Description
RB0/INT	RB0	TTL	CMOS	Bi-directional I/O port. Can be software programmed for internal weak pull-up.
	INT	ST	—	External interrupt.
RB1/RX/DT	RB1	TTL	CMOS	Bi-directional I/O port. Can be software programmed for internal weak pull-up.
	RX	ST	—	USART Receive Pin
	DT	ST	CMOS	Synchronous data I/O
RB2/TX/CK	RB2	TTL	CMOS	Bi-directional I/O port
	TX	—	CMOS	USART Transmit Pin
	CK	ST	CMOS	Synchronous Clock I/O. Can be software programmed for internal weak pull-up.
RB3/CCP1	RB3	TTL	CMOS	Bi-directional I/O port. Can be software programmed for internal weak pull-up.
	CCP1	ST	CMOS	Capture/Compare/PWM I/O
RB4/PGM	RB4	TTL	CMOS	Bi-directional I/O port. Can be software programmed for internal weak pull-up.
	PGM	ST	—	Low voltage programming input pin. Interrupt-on-pin change. When low voltage programming is enabled, the interrupt-on-pin change and weak pull-up resistor are disabled.
RB5	RB5	TTL	CMOS	Bi-directional I/O port. Interrupt-on-pin change. Can be software programmed for internal weak pull-up.
RB6/T1OSO/T1CKI/PGC	RB6	TTL	CMOS	Bi-directional I/O port. Interrupt-on-pin change. Can be software programmed for internal weak pull-up.
	T1OSO	—	XTAL	Timer1 Oscillator Output
	T1CKI	ST	—	Timer1 Clock Input
	PGC	ST	—	ICSP Programming Clock
RB7/T1OSI/PGD	RB7	TTL	CMOS	Bi-directional I/O port. Interrupt-on-pin change. Can be software programmed for internal weak pull-up.
	T1OSI	XTAL	—	Timer1 Oscillator Input
	PGD	ST	CMOS	ICSP Data I/O

Legend: O = Output
 — = Not used
 TTL = TTL Input

CMOS = CMOS Output
 I = Input
 OD = Open Drain Output

P = Power
 ST = Schmitt Trigger Input
 AN = Analog

Le tableau ci-contre décrit les différentes fonctions multiplexées sur le port B.

Certaines de ces fonctions seront examinées dans les paragraphes suivants.

3.2 - UTILISATION EN ENTREES SORTIES NUMERIQUES

a) Multiplexage sur le port B

Le port B est multiplexé avec :

- interruption externe
- USART
- CCP module
- TMR1 clock in/out

b) Registre TRISB

Ce registre permet de définir si la patte considérée fonctionne en entrée ou en sortie.

Un « 1 » dans un bit du registre TRISB met la sortie correspondante en haute impédance, elle peut ainsi servir d'entrée.

Un « 0 » dans un bit de ce registre transfère le contenu de la sortie de la bascule D sur la sortie correspondante.

$$\left\{ \begin{array}{l} \text{TRISB}_n = 1 \Rightarrow \text{RB}_n \text{ en entrée} \\ \text{TRISB}_n = 0 \Rightarrow \text{RB}_n \text{ en sortie} \end{array} \right.$$

Remarque : Toute opération d'écriture sur une des sorties est précédée d'une lecture de la patte correspondante.

Chapitre 3 – UTILISATION DU TIMER 0

Le composant dispose de 3 timers : timer0 (TMR0), timer1 (TMR1) et timer2 (TMR2)

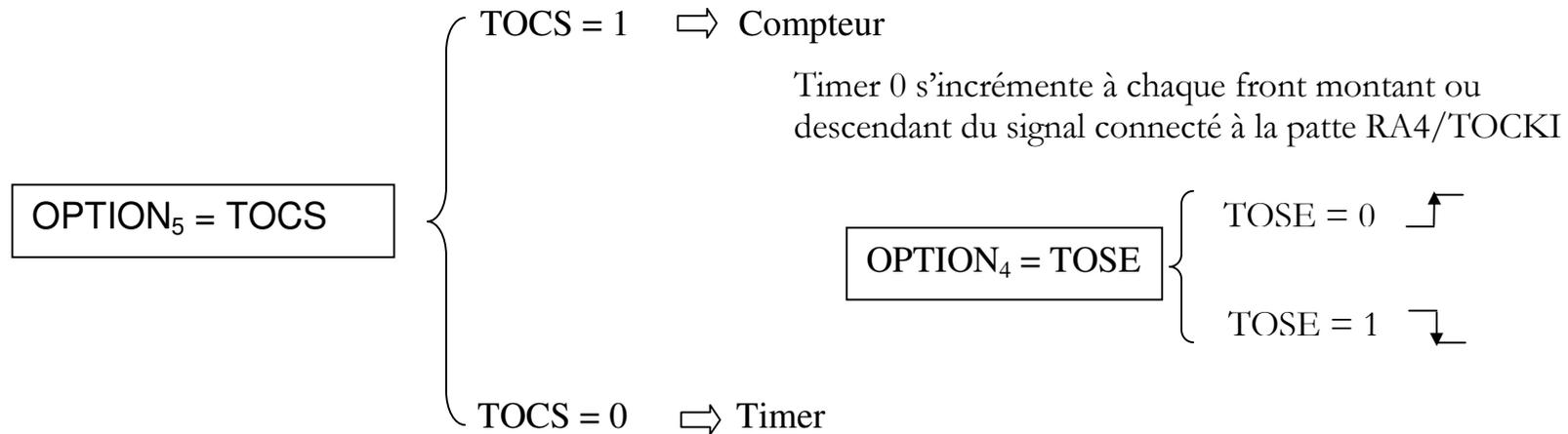
1 – CARACTERISTIQUES DU TIMER 0

Le module Timer0 a les caractéristiques suivantes :

- Timer ou compteur 8 bits
- Utilisable en lecture ou écriture
- Pré diviseur 8 bits programmable
- Sélection de l'horloge interne ou externe
- Interruption sur dépassement
- Sélection du front montant ou descendant pour l'horloge externe

2 - SELECTION DU MODE TIMER OU COMPTEUR

Cette sélection s'opère grâce au 5^{ème} bit TOCS du registre OPTION



Timer 0 s'incrémente à chaque cycle s'il n'y a pas de prédiviseur.
En cas d'opération d'écriture dans le timer, l'incrémentation est inhibée pour les 2 cycles suivants.

Avec une horloge à 4 MHz,
l'incrémentation se fait toutes les 1 µs
(4 cycles par période d'horloge)

3 – REGISTRE OPTION

OPTION REGISTER (ADDRESS: 81h, 181h)

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
$\overline{\text{RBP}}\text{U}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
bit 7							bit 0

- bit 7 **RBP**U: PORTB Pull-up Enable bit
1 = PORTB pull-ups are disabled
0 = PORTB pull-ups are enabled by individual port latch values
- bit 6 **INTEDG**: Interrupt Edge Select bit
1 = Interrupt on rising edge of RB0/INT pin
0 = Interrupt on falling edge of RB0/INT pin
- bit 5 **T0CS**: TMR0 Clock Source Select bit
1 = Transition on RA4/T0CKI pin
0 = Internal instruction cycle clock (CLKOUT)
- bit 4 **T0SE**: TMR0 Source Edge Select bit
1 = Increment on high-to-low transition on RA4/T0CKI pin
0 = Increment on low-to-high transition on RA4/T0CKI pin
- bit 3 **PSA**: Prescaler Assignment bit
1 = Prescaler is assigned to the WDT
0 = Prescaler is assigned to the Timer0 module
- bit 2-0 **PS2:PS0**: Prescaler Rate Select bits

Bit Value	TMR0 Rate	WDT Rate
000	1 : 2	1 : 1
001	1 : 4	1 : 2
010	1 : 8	1 : 4
011	1 : 16	1 : 8
100	1 : 32	1 : 16
101	1 : 64	1 : 32
110	1 : 128	1 : 64
111	1 : 256	1 : 128

Choix du taux de division

Legend:

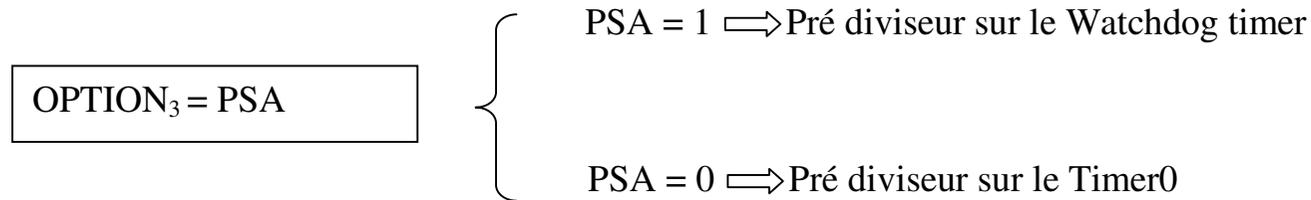
R = Readable bit
-n = Value at POR

W = Writable bit
'1' = Bit is set

U = Unimplemented bit, read as '0'
'0' = Bit is cleared
x = Bit is unknown

4 - PREDIVISEUR

Le pré-diviseur est partagé entre Timer0 et le chien de Garde (Watchdog Timer) ; Ce choix s'opère par l'état du bit 3 PSA du registre OPTION.



Le taux de division est alors réglable par les bits PS0, PS1 et PS2 du registre OPTION (Voir ci dessus § 3.3)

5 – EXEMPLE : Temporisation

Pour utiliser le TIMER0 avec pré-division par 256, il faudra une ligne de code dans le programme :

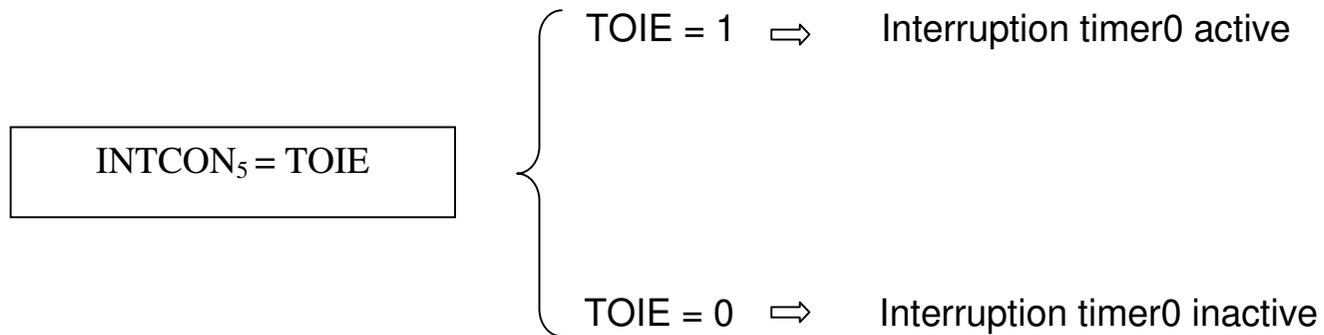
<pre>OPTION=0b00000111 ; Ou OPTION=0x07 ; Ou OPTION=7 ;</pre>	<pre>TMR0 = 0 ; do { } while (TMR0 < 240);</pre>
---	---

6 – INTERRUPTION DU TIMER0

Une interruption est générée par le timer0 si le timer ou le compteur passe de xFF à x00 (en hexadécimal).

Ce dépassement met à 1 le bit TOIF, bit 2 du registre INTCON.

On peut activer ou pas cette interruption par le bit TOIE, bit 5 du registre INTCON.

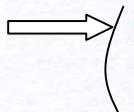
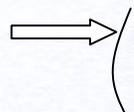


Le taux de division est alors réglable par les bits PS0, PS1 et PS2 du registre OPTION (Voir ci dessus § 3.3)

INTCON REGISTER (ADDRESS: 0Bh, 8Bh, 10Bh, 18Bh)

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

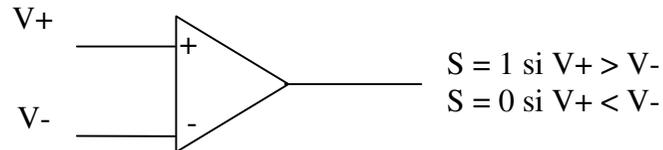
- bit 7 **GIE:** Global Interrupt Enable bit
1 = Enables all unmasked interrupts
0 = Disables all interrupts
- bit 6 **PEIE:** Peripheral Interrupt Enable bit
1 = Enables all unmasked peripheral interrupts
0 = Disables all peripheral interrupts
- bit 5 **TOIE:** TMR0 Overflow Interrupt Enable bit
1 = Enables the TMR0 interrupt
0 = Disables the TMR0 interrupt
- bit 4 **INTE:** RB0/INT External Interrupt Enable bit
1 = Enables the RB0/INT external interrupt
0 = Disables the RB0/INT external interrupt
- bit 3 **RBIE:** RB Port Change Interrupt Enable bit
1 = Enables the RB port change interrupt
0 = Disables the RB port change interrupt
- bit 2 **TOIF:** TMR0 Overflow Interrupt Flag bit
1 = TMR0 register has overflowed (must be cleared in software)
0 = TMR0 register did not overflow
- bit 1 **INTF:** RB0/INT External Interrupt Flag bit
1 = The RB0/INT external interrupt occurred (must be cleared in software)
0 = The RB0/INT external interrupt did not occur
- bit 0 **RBIF:** RB Port Change Interrupt Flag bit
1 = When at least one of the RB7:RB4 pins changed state (must be cleared in software)
0 = None of the RB7:RB4 pins have changed state



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

Chapitre 4 - UTILISATION DU COMPAREUR

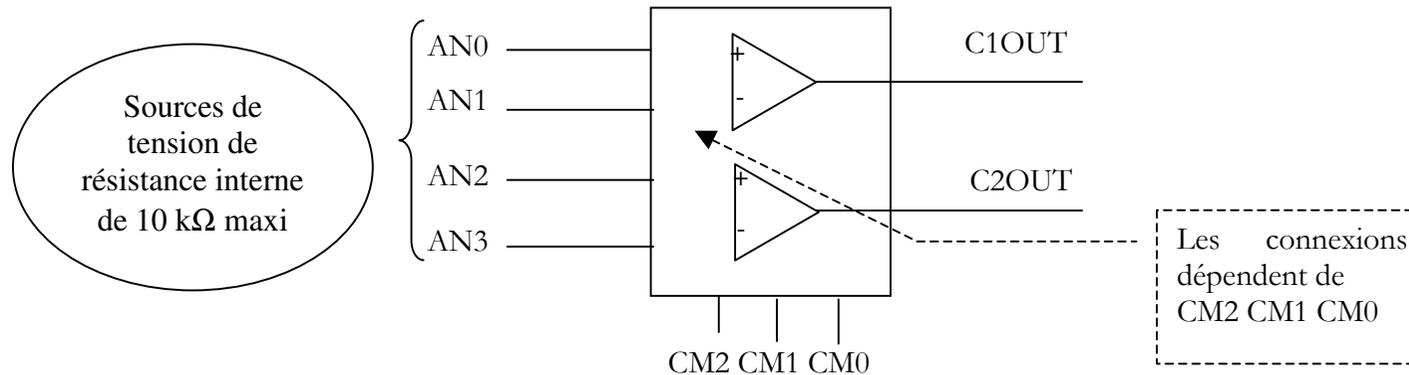
1 – RAPPEL



2 – LES COMPAREURS

Le microcontrôleur comporte 2 comparateurs donc 4 entrées et 2 sorties.

Ces entrées sorties de comparateur sont multiplexées avec les entrées sorties numériques du port A comme déjà signalé.



Les bits CM2 CM1 et CM0 du registre CMCON (Cf § 4.3) définissent les fonctions de comparaison sélectionnées.

Le registre TRISA contrôle la direction des données (E/S) pour chaque entrée/sortie du port A même en mode comparateur. Il convient donc d'initialiser correctement TRISA.

3 – DIFFERENTS MODES DE COMPARAISON

Le tableau ci-dessous récapitule les différentes configurations possibles.

Exemple :

Si l'on souhaite utiliser un seul comparateur, les bits CM2 CM1 et CM0 du registre CMCON sont respectivement 101.

Le comparateur est alors constitué des deux entrées RA1/AN1 (patte 18) et RA2/AN2 (patte 1). Le résultat de la comparaison est disponible sur les bits 6 et 7 du registre CMCON.

Ces deux bornes appartiennent au port A qui doit être configuré correctement à l'aide de TRISA : RA1/AN1 et RA2/AN2 en entrées

On a alors TRISA = 0bxxxxx**11**x.

4 – SORTIES DU COMPAREUR

Les résultats des deux comparaisons sont disponibles sur les bits 6 et 7 du registre CMCON. (Ces deux bits sont en lecture seule).

Ils peuvent aussi être transmis sur les sorties RA3/AN3/CMP1 (patte 2) et RA4/TOCKI/CMP2 (patte3). Pour cela il faut être dans la configuration où CM2 :CM0=110.

Ces deux bornes appartenant au port A, il faut les configurer en sortie à l'aide de TRISA : TRISA = 0bxxx**00**xxx

5 – COMPLEMENTS SUR LE REGISTRE CMCON

CMCON REGISTER (ADDRESS: 01Fh)

R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
C2OUT	C1OUT	C2INV	C1INV	CIS	CM2	CM1	CM0
bit 7				bit 0			

bit 7
C2OUT: Comparator 2 Output
When C2INV = 0:
 1 = C2 VIN+ > C2 VIN-
 0 = C2 VIN+ < C2 VIN-

When C2INV = 1:
 1 = C2 VIN+ < C2 VIN-
 0 = C2 VIN+ > C2 VIN-

bit 6
C1OUT: Comparator 1 Output
When C1INV = 0:
 1 = C1 VIN+ > C1 VIN-
 0 = C1 VIN+ < C1 VIN-

When C1INV = 1:
 1 = C1 VIN+ < C1 VIN-
 0 = C1 VIN+ > C1 VIN-

bit 5
C2INV: Comparator 2 Output Inversion
 1 = C2 Output inverted
 0 = C2 Output not inverted

bit 4
C1INV: Comparator 1 Output Inversion
 1 = C1 Output inverted
 0 = C1 Output not inverted

bit 3
CIS: Comparator Input Switch
When CM2:CM0 = 001
 Then:
 1 = C1 VIN- connects to RA3
 0 = C1 VIN- connects to RA0

When CM2:CM0 = 010
 Then:
 1 = C1 VIN- connects to RA3
 C2 VIN- connects to RA2
 0 = C1 VIN- connects to RA0
 C2 VIN- connects to RA1

bit 2-0
CM2:CM0: Comparator Mode
 Figure 9-1 shows the Comparator modes and CM2:CM0 bit settings

Inversion des sorties
du comparateur

Bit de contrôle du multiplexage
des entrées des comparateurs

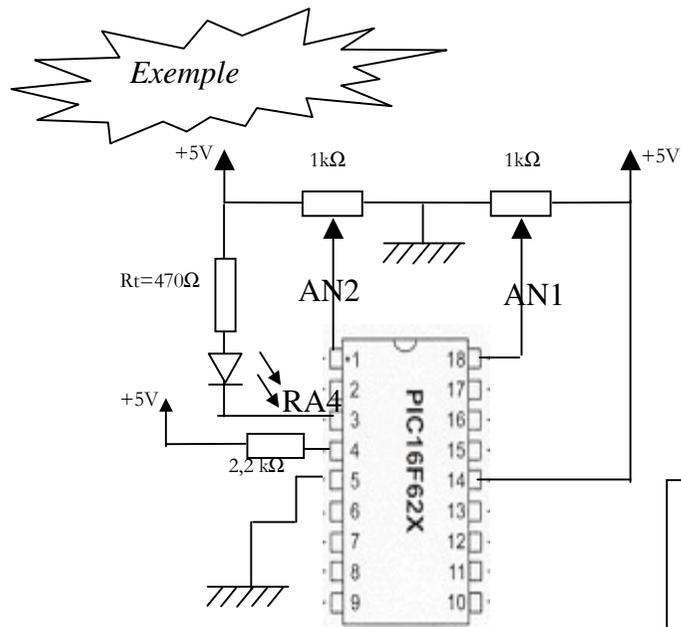
Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

6 – EXEMPLE

Le programme suivant permet de faire fonctionner le pic en comparateur, comme un simple ampli op en boucle ouverte.

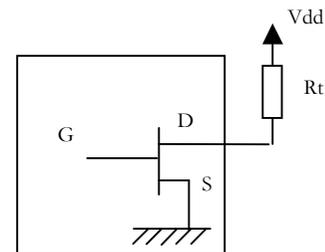
```
void main(void)
{
    CMCON=0b00000110 ; /*choix du mode de
comparaison Cf tableau §4.3 */
    TRISA=0b00000110 ; /*RA1 et RA2 entrées, RA4 en
sortie */
}
```

Après avoir compilé le programme source et programmé le composant, on peut câbler le montage suivant pour tester le fonctionnement :



La résistance R_t est une résistance dite de tirage (pull up).

Elle est nécessaire car la sortie RA4 en mode comparateur est à drain ouvert :



Si $G = 1 \Rightarrow V_{DS} = 0$ et $V_{Rt} = V_{dd}$
 Si $G = 0 \Rightarrow V_{DS} = V_{dd}$ et $V_{Rt} = 0$

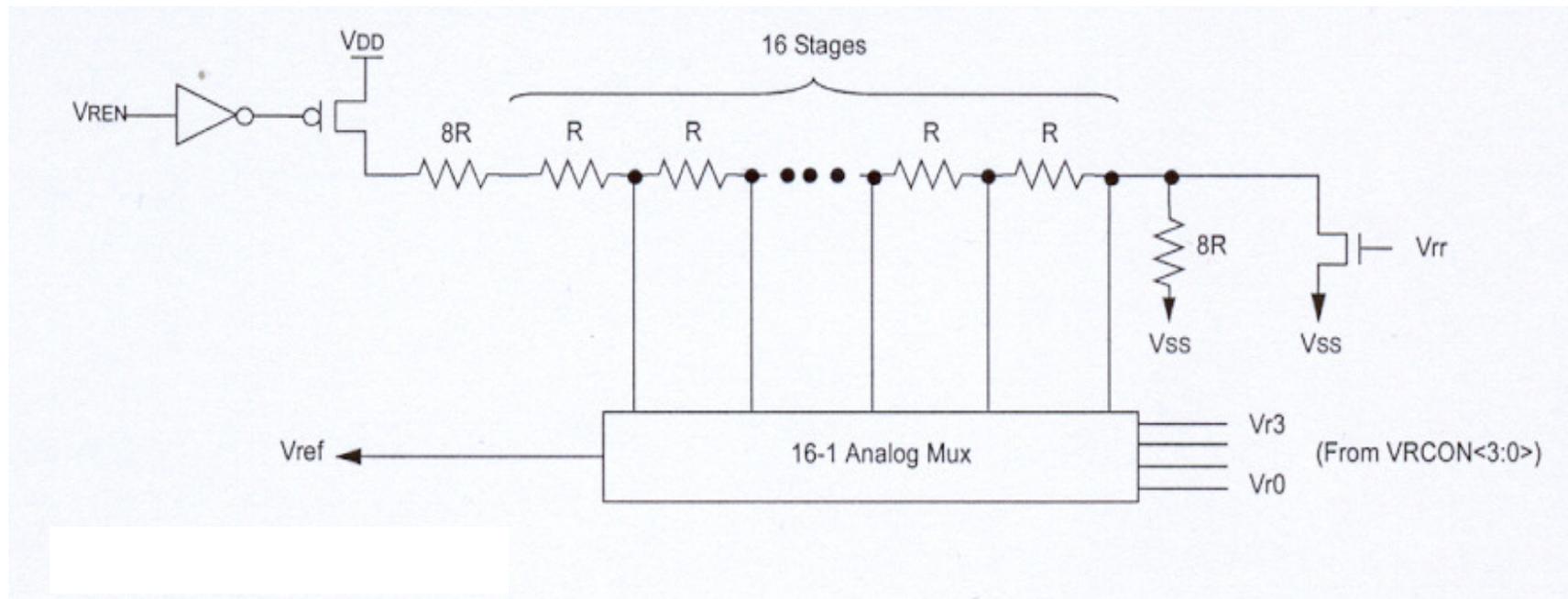
Si $AN2 > AN1 \Rightarrow RA4 = 1 \Rightarrow$ Led éteinte

Si $AN2 < AN1 \Rightarrow RA4 = 0 \Rightarrow$ Led allumée

7 – UTILISATION DE TENSIONS DE REFERENCE

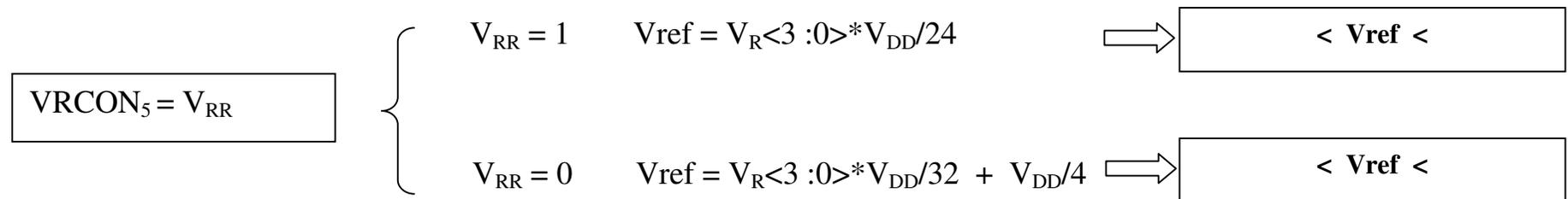
7.1 – REALISATION DE LA TENSION DE REFERENCE

Le module est constitué d'un réseau de résistances en échelle permettant de fabriquer une tension de référence V_{Ref} .



7.2 – VALEUR DE LA TENSION DE REFERENCE

On dispose de deux gammes dont le choix s'opère par V_{RR} le bit 5 du registre VRCON.
La valeur exacte dans la gamme est commandée par V_R les bits 3 à 0 du registre VRCON.



$V_{R<3:0>}$ est la valeur décimale du mot binaire constitué par les bits V_{R3} V_{R2} V_{R1} V_{R0} du registre VRCON.

Calculer les valeurs extrêmes de chaque gamme

7.3 – TRANSMISSION DE LA TENSION DE REFERENCE SUR LA SORTIE

La tension de référence doit être activée par le bit 7 mis à 1 dans le registre VRCON.

La tension de référence élaborée est envoyée sur la sortie RA2/AN2/ V_{REF} du port A si V_{ROE} le bit 6 du registre VRCON est à 1. Sinon, la tension de référence est déconnectée.

Ainsi on doit avoir $VRCON = 0b1110\ 0110 \Rightarrow V_{ref} = 6 * V_{DD}/24 = 1,25\ V$ envoyé sur la sortie RA2.

En même temps, **la sortie RA2 du port A doit être configurée en entrée** par l'intermédiaire du registre TRISA : $TRISA = 0bxxxxx1xx$

```
void main(void)
{
    VRCON=0b11100110 ;
    TRISA=0b00000100 ;
}
```

7.4 – REGISTRE VRCON

VRCON REGISTER (ADDRESS: 9Fh)

R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
VREN	VROE	VRR	—	VR3	VR2	VR1	VR0
bit 7							bit 0

- bit 7 **VREN:** VREF Enable
 1 = VREF circuit powered on
 0 = VREF circuit powered down, no IDD drain
- bit 6 **VROE:** VREF Output Enable
 1 = VREF is output on RA2 pin
 0 = VREF is disconnected from RA2 pin
- bit 5 **VRR:** VREF Range selection
 1 = Low Range
 0 = High Range
- bit 4 **Unimplemented:** Read as '0'
- bit 3-0 **VR<3:0>:** VREF value selection $0 \leq VR [3:0] \leq 15$
 When VRR = 1: $VREF = (VR<3:0>/ 24) * VDD$
 When VRR = 0: $VREF = 1/4 * VDD + (VR<3:0>/ 32) * VDD$

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

7.5 – UTILISATION EN CNA

La sortie Vref programmée par l'intermédiaire d'un mot binaire $V_{R<3:0>}$ constitue un Convertisseur Numérique Analogique. Cependant, cette sortie ne peut être chargée sans l'utilisation d'un étage suiveur.

Chapitre 5 - LE TIMER 1

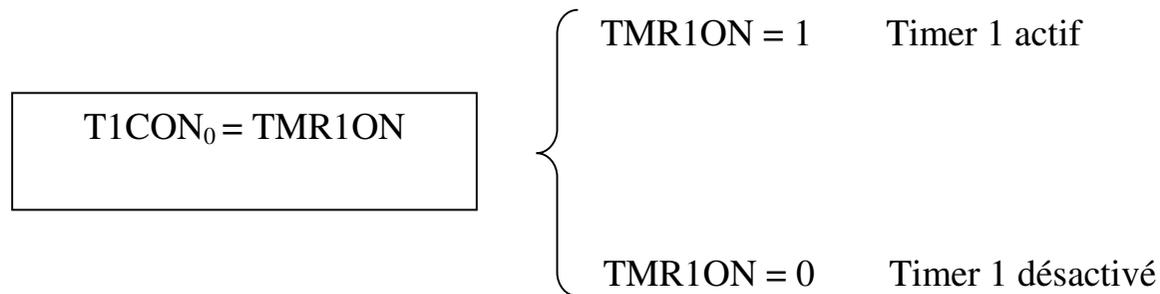
1 – DESCRIPTION DU TIMER 1

Le module Timer 1 est un timer/compteur 16 bits constitué de deux registres 8 bits TMR1H et TMR1L pouvant être en lecture ou écriture.

Le Timer 1 s'incrémente donc de 0x0000 à 0xFFFF puis repasse à 0x0000. Le dépassement est signalé par le bit TMR1IF du registre PIR1.

2 – REGISTRE T1CON

Le bit 0 de ce registre permet d'activer ou désactiver le timer1.



Les bits 4 et 5 permettent de choisir le taux de division de la fréquence d'horloge interne ou externe.

T1CON: TIMER1 CONTROL REGISTER (ADDRESS: 10h)

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{T1SYNC}$	TMR1CS	TMR1ON

bit 7

bit 0

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

bit 5-4 **T1CKPS1:T1CKPS0:** Timer1 Input Clock Prescale Select bits

11 = 1:8 Prescale value

10 = 1:4 Prescale value

01 = 1:2 Prescale value

00 = 1:1 Prescale value

bit 3 **T1OSCEN:** Timer1 Oscillator Enable Control bit

1 = Oscillator is enabled

0 = Oscillator is shut off⁽¹⁾

bit 2 **T1SYNC:** Timer1 External Clock Input Synchronization Control bit

TMR1CS = 1

1 = Do not synchronize external clock input

0 = Synchronize external clock input

TMR1CS = 0

This bit is ignored. Timer1 uses the internal clock when TMR1CS = 0.

bit 1 **TMR1CS:** Timer1 Clock Source Select bit

1 = External clock from pin RB6/T1OSO/T1CKI (on the rising edge)

0 = Internal clock (Fosc/4)

bit 0 **TMR1ON:** Timer1 On bit

1 = ^{Enable} Timer1

0 = Stops Timer1

Note 1: The oscillator inverter and feedback resistor are turned off to eliminate power drain.

Legend:

R = Readable bit

W = Writable bit

U = Unimplemented bit, read as '0'

-n = Value at POR

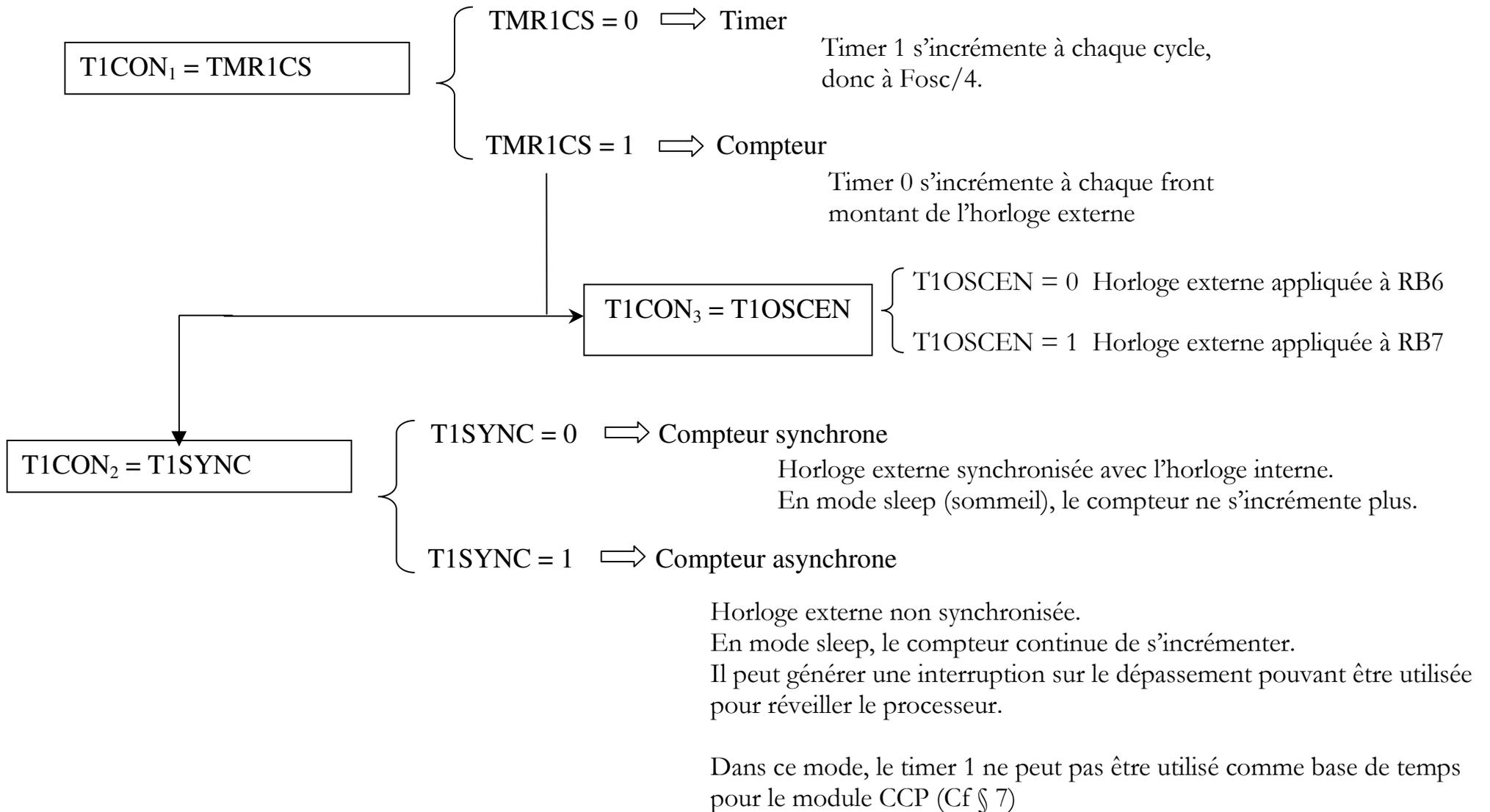
'1' = Bit is set

'0' = Bit is cleared

x = Bit is unknown

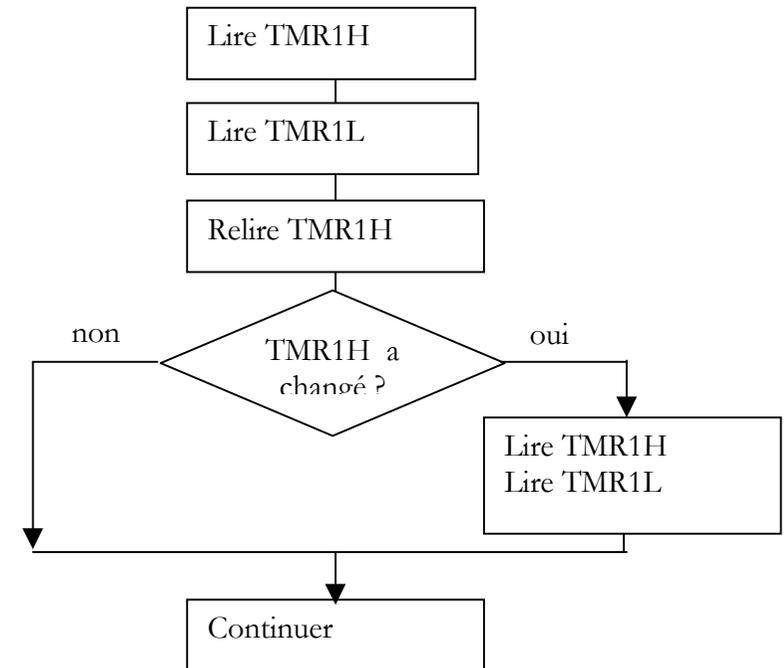
3 – SELECTION DU MODE TIMER OU COMPTEUR

Cette sélection s'effectue grâce au bit TMRCS du registre T1CON.

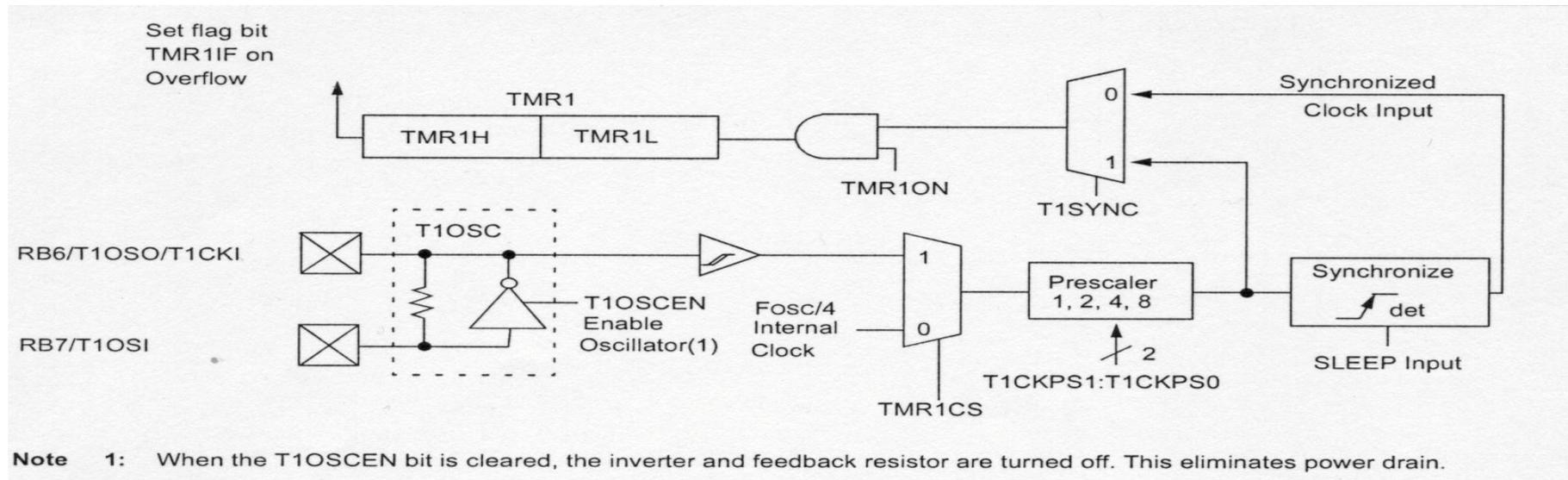


La lecture et l'écriture du timer se font en 2 opérations du fait de la séparation du timer 1 en deux registres TMR1H et TMR1L. Un dépassement peut intervenir entre temps.

- ⇒ Pour l'écriture, il est préférable d'arrêter le timer auparavant.
- ⇒ Pour la lecture, il faut réaliser la procédure suivante :



4 – BLOC DIAGRAMME DU TIMER 1



Chapitre 6 – MODULE CCP

Ce module contient un registre 16 bit constitué en réalité de deux registres 8 bits : CCPR1H et CCPR1L.

Les opérations de ce module sont contrôlées par le registre CCP1CON.

Ce module peut fonctionner de trois façons :

- mode capture, en relation avec le timer 1
- mode compare, en relation avec le timer 1
- mode PWM, en relation avec le timer 2.

1 – REGISTRE CCP1CON

CCP1CON REGISTER (ADDRESS: 17h)

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	
—	—	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0	
bit 7								bit 0

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

bit 5-4 **CCP1X:CCP1Y:** 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 **CCP1M3:CCP1M0:** CCPx Mode Select bits

0000 = Capture/Compare/PWM off (resets CCP1 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 (CCP1IF bit is set)

1001 = Compare mode, clear output on match (CCP1IF bit is set)

1010 = Compare mode, generate software interrupt on match (CCP1IF bit is set, CCP1 pin is unaffected)

1011 = Compare mode, trigger special event (CCP1IF bit is set; CCP1 resets TMR1)

11xx = PWM mode

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 – MODE CAPTURE

Ce mode permet de transférer la valeur (16 bits) du timer 1 dans les deux registres 8 bits CCPR1H et CCPR1L, lorsqu'un événement se produit sur le port RB3 d'entrée.

Le timer 1 doit alors être en mode timer ou compteur synchrone.

RB3 doit bien sûr être configuré en entrée $TRISB_3 = 1$.

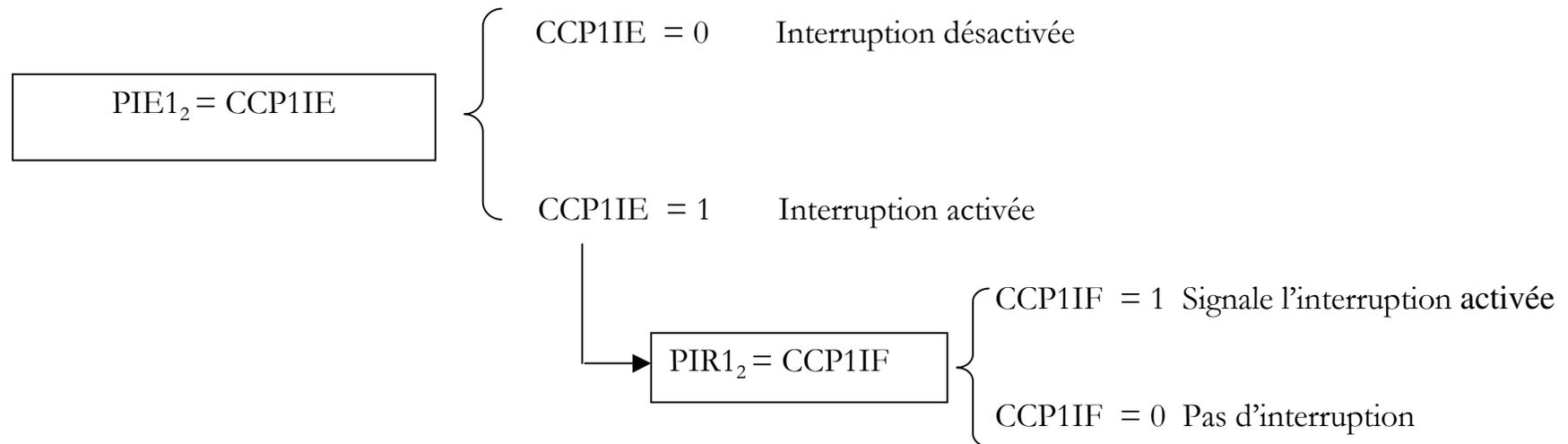
L'événement déclenchant le transfert est sélectionné par les bits 0 à 3 du registre CCP1CON. (Cf §7.1) :

- Sur chaque front descendant du signal connecté sur RB3
- Sur chaque front montant du signal connecté sur RB3
- Tous les 4 fronts montants du signal connecté sur RB3
- Tous les 16 fronts montants du signal connecté sur RB3

Les bits 0 à 3 du registre CCP1CON permettent donc de régler un pré-diviseur agissant sur le signal appliqué à RB3.

La réalisation de la capture est signalée par la mise à 1 du drapeau correspondant : bit 2 CCP1IF du registre PIR1. Ce bit doit alors être remis à 0 dans le programme.

Le contrôle de l'interruption est réalisé par le bit 2 CCP1IE du registre PIE1.



Attention : le changement du mode de capture entraîne une fausse interruption.
Avant toute modification du mode, il faut désactiver l'interruption et remettre à zéro le drapeau CCP1IF.

3 – REGISTRES PIR1 ET PIE1

Le registre PIR1 est un registre regroupant les drapeaux d'interruption c'est à dire des bits signalant qu'un événement déclenchant une interruption s'est produit.

Le registre PIE1 est le registre qui active ou pas les interruptions.

PIE1 REGISTER (ADDRESS: 8Ch)

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
EEIE	CMIE	RCIE	TXIE	—	CCP1IE	TMR2IE	TMR1IE

bit 7

bit 0

- bit 7 **EEIE:** EE Write Complete Interrupt Enable Bit
 1 = Enables the EE write complete interrupt
 0 = Disables the EE write complete interrupt
- bit 6 **CMIE:** Comparator Interrupt Enable bit
 1 = Enables the comparator interrupt
 0 = Disables the comparator interrupt
- bit 5 **RCIE:** USART Receive Interrupt Enable bit
 1 = Enables the USART receive interrupt
 0 = Disables the USART receive interrupt
- bit 4 **TXIE:** USART Transmit Interrupt Enable bit
 1 = Enables the USART transmit interrupt
 0 = Disables the USART transmit interrupt
- bit 3 **Unimplemented:** Read as '0'
- bit 2 **CCP1IE:** CCP1 Interrupt Enable bit
 1 = Enables the CCP1 interrupt
 0 = Disables the CCP1 interrupt
- bit 1 **TMR2IE:** TMR2 to PR2 Match Interrupt Enable bit
 1 = Enables the TMR2 to PR2 match interrupt
 0 = Disables the TMR2 to PR2 match interrupt
- bit 0 **TMR1IE:** TMR1 Overflow Interrupt Enable bit
 1 = Enables the TMR1 overflow interrupt
 0 = Disables the TMR1 overflow interrupt

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

4 – MODE COMPARE

Dans ce mode, le registre 16 bits CCPR1 (CCPR1H et CCPR1L) est constamment comparé au timer 1.

Quand ils sont égaux, cela est signalé sur la sortie RB3 :

- Soit par un état haut
- Soit par un état bas
- Soit par un maintien de la valeur présente.

L'action sur la patte RB3 est choisie par les bits 0 à 3 du registre CCP1CON. (Cf § 7.1).

RB3 doit bien sûr être configuré en sortie $TRISB_3 = 0$.

Le timer 1 doit alors être en mode timer ou compteur synchrone.

Comme dans le mode capture, la réalisation de l'égalité est signalée par la mise à 1 du drapeau correspondant : bit 2 CCP1IF du registre PIR1. Ce bit doit alors être remis à 0 dans le programme.

Le contrôle de cette interruption est réalisé par le bit 2 CCP1IE du registre PIE1.

Si $CCP1CON \langle 3 : 0 \rangle = 1010$ alors la réalisation de l'égalité entraîne une interruption logiciel sans affecter RB3.

Si $CCP1CON \langle 3 : 0 \rangle = 1011$ alors la réalisation de l'égalité remet à zéro le timer 1. On a ainsi un timer 1 dont la période est programmable par le registre CCP1.

5 – UTILISATION DU TIMER 2

Le timer 2 est un timer 8 bits avec pré et post-diviseurs programmables par l'intermédiaire du registre T2CON.

Il peut être utilisé en lecture et écriture et est remis à zéro par le reset du microcontrôleur.

Il est activé ou désactivé par le bit 2 du registre T2CON.

Il sert de base de temps pour le mode PWM du module CCP : timer 2 s'incrémente de 0x00 jusqu'à ce qu'il atteigne la valeur du registre PR2 puis repasse à 0x00 lors du cycle suivant. (fonctionnant en lecture et écriture).

Le bit 1 TMR2IF du registre PIR1 signale l'égalité timer 2 = PR2 en passant à 1. Ce drapeau doit alors être remis à zéro dans le programme.

T2CON: TIMER CONTROL REGISTER (ADDRESS: 12h)

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0
bit 7							bit 0

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

bit 6-3 **TOUTPS3:TOUTPS0:** Timer2 Output Postscale Select bits

0000 = 1:1 Postscale Value

0001 = 1:2 Postscale Value

•
•
•

1111 = 1:16 Postscale

bit 2 **TMR2ON:** Timer2 On bit

1 = Timer2 is on

0 = Timer2 is off

bit 1-0 **T2CKPS1:T2CKPS0:** Timer2 Clock Prescale Select bits

00 = 1:1 Prescaler Value

01 = 1:4 Prescaler Value

1x = 1:16 Prescaler Value

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

6 – MODE PWM DU MODULE CCP

Rappel : PWM signifie Pulse Width Modulation. En Français MLI, modulation de largeur d'impulsion. Il s'agit donc de réaliser un signal dont la largeur de l'impulsion et donc le rapport cyclique est programmable.

Ce mode permet de délivrer sur la patte RB3 utilisée en sortie un signal PWM grâce à l'utilisation du timer 2.

RB3 doit donc être configuré en sortie $TRISB_3 = 0$.
Le timer 2 doit être activé (registre T2CON).

Dans ce mode, le registre 8 bits CCPR1L auquel s'ajoutent les bits 4 et 5 du registre CCP1CON constitue un mot de 10 bits correspondant à la largeur de l'impulsion :

$$\text{Largeur de l'impulsion} = \text{CCPR1L} : \text{CCP1CON}\langle 5 : 4 \rangle \cdot T_{osc} \cdot (\text{valeur du pré-diviseur du timer 2})$$

La période du signal PWM dépend d'une valeur devant être écrite dans le registre PR2.

Elle est donnée par la relation :

$$\text{Période PWM} = (\text{PR2} + 1) \cdot 4 T_{osc} \cdot (\text{valeur du pré-diviseur du Timer 2})$$

La période PWM doit être supérieure à la durée de l'impulsion.

Quand $\text{TMR2} = \text{PR2}$ cela entraîne au cycle suivant :

- La remise à zéro du timer 2
- La mise à 1 de la sortie RB3
- Le rapport cyclique est transmis de CCPR1L à CCPR1H.

Le bit 1 TMR2IF du registre PIR1 signale l'égalité timer 2 = PR2 en passant à 1. Ce drapeau doit alors être remis à zéro dans le programme.

EXERCICES

Exercice 1 : Entrées-sorties sur port A et B

1% Ecrire un programme qui lit un bouton poussoir connecté à RA0 et allume une led connectée sur RA1 si le bouton poussoir est relâché et éteint la led si le BP est enfoncé.

2% Ecrire un programme qui délivre sur les sorties RB3, RB2, RB1 et RB0 du port B du pic, les valeurs complémentées des bits RA3, RA2, RA1, RA0 du port A utilisé en entrée.

Exercice 2 : Timer 0

Ecrire un programme qui fait clignoter à la fréquence d'1 Hz, une led connectée sur la sortie RA0 du port A.

Exercice 3 : Chenillard à 4 leds

Les 4 leds seront connectées au port A.

Ecrire le programme qui permet d'allumer successivement chaque led pendant 0,5 s (l'allumage de la led suivante éteint la précédente).

Exercice 4.1

Ecrire un programme qui compare les tensions appliquées sur RA1 et RA2 et affiche le résultat sur RA0

Exercice 4.2

Ecrire un programme qui compare AN0 et AN1 à AN2 et affiche le résultat sur RB0 et RB1

Exercice 4.3 : Comparateur

Ecrire un programme délivrant sur la sortie RA4, le résultat de la comparaison de la tension V_{in} appliquée sur RA0 et de la tension de référence interne réglée à 1,25 V

Exercice 4.4 : Boucle d'attente

Ecrire un programme qui attend l'appui sur un bouton poussoir connecté sur RB4 pour allumer la led RB0.

Exercice 4.5

Ecrire un programme qui génère une tension RA2 qui augmente de 0,208 V à chaque appui sur un bouton poussoir connecté sur RB4 jusqu'à atteindre 2,5 V

Exercice 5 : Comparateur programmable

Reprendre le programme précédent et le modifier pour que la référence interne soit fonction des bits RB3 RB2 RB1 et RB0 du port B utilisé en entrée numérique.

Exercice 6 : CAN 4 bits

Ecrire un programme permettant d'utiliser un pic pour réaliser un CAN 4 bits ayant les caractéristiques suivantes :
Ve appliquée sur RA1 comprise entre 0 et 3,125 V quantum de 0,2 V.

Le mot binaire résultat de la conversion sera affiché sur 4 leds connectées sur les sorties RB3 RB2 RB1 RB0 du port B. Les leds sont allumées par un 0 sur la sortie correspondante.