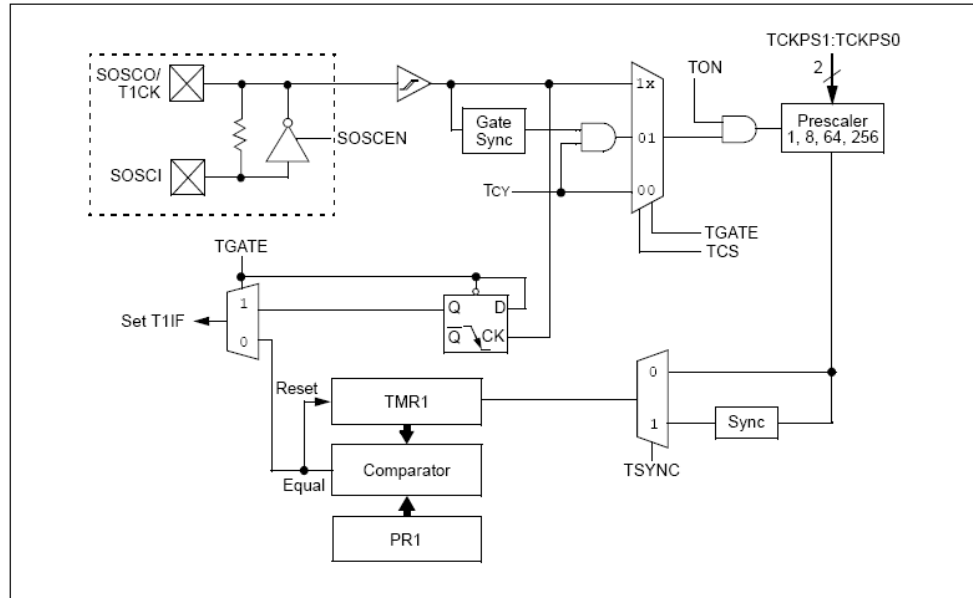


**TD 4 – Le Timer du PIC24F**

**A – Présentation des Timers**

Le PIC24F est doté de 5 Timers 16 bits, qui présentent une architecture semblable avec quelques variantes. Les timers 2-3 et 4-5 peuvent s'assembler pour former des compteurs 32 bits. Chaque timer peut fonctionner soit en "Timer" (génération de temps via une horloge interne), soit en "Counter" (comptage d'impulsions externes).

FIGURE 10-1: 16-BIT TIMER1 MODULE BLOCK DIAGRAM



- L'horloge peut se voir appliquer un facteur de division (4 choix possibles).
- Elle pilote un compteur 16 bits, accessible en lecture et écriture à travers un registre TMRx.
- Un comparateur permet d'activer un flag d'interruption s'il y a égalité entre le contenu d'un registre (PRx) et le compteur TMRx. De plus, l'activation de la sortie du comparateur provoque la RAZ du timer.
- Chaque timer est contrôlé par 3 registres, TMRx, Prx, et un registre de contrôle TxCON (x=1, 2, ... 5).

**B - Fonctionnement en scrutation (temporisation)**

On souhaite faire clignoter les leds avec une période de 1 s., en choisissant comme rapport de prescaler 256. Le cristal pilotant la carte Explorer16 est à 8 MHz. On propose le corps de programme suivant :

```
int main()
{
    TRISA = 0x00;    // _____
    PORTA = 0xf0;    // _____
    T1CON = _____; // _____

    TMR1 = 0x0000;    // RAZ initiale

    while( 1 )
    {
        if( TMR1 >= _____ )
        {
            PORTA = PORTA ^ 0xAA;
            TMR1 = 0x0000;
        }
    }
}
```

Questions préliminaires :

- Quelle sera la période du signal d'horloge pilotant TMR1 : \_\_\_\_\_
- Quelle est la durée du délai désiré : \_\_\_\_\_
- A combien de cycles-timer cela correspond-t-il : \_\_\_\_\_
- Quelle valeur faudra-t-il programmer dans T1CON : \_\_\_\_\_

**C – Fonctionnement en interruption**

On souhaite le même fonctionnement, mais en "non-bloquant". Le programme principal s'écrira :

```
while(1)
;
```

Pour avoir déclenchement d'une interruption sur l'activation du flag, il faut :

- Autoriser les interruptions du Timer 1 (registre IEC0, voir le bit TIE, p.71).
- Régler la priorité des interruptions du Timer 1 à un niveau autre que "0" (registre IPC0, p.76)

La routine (prédéclarée dans le .h) s'appelle "\_T1Interrupt()". Elle devra réinitialiser le flag T1IF , et inverser l'état des leds.

Il faudra aussi dans l'initialisation écrire dans le registre du comparateur (PR1) la valeur adéquate.

**REGISTER 10-1: T1CON: TIMER1 CONTROL REGISTER**

|                    |     |       |     |     |     |     |       |
|--------------------|-----|-------|-----|-----|-----|-----|-------|
| <b>Upper Byte:</b> |     |       |     |     |     |     |       |
| R/W-0              | U-0 | R/W-0 | U-0 | U-0 | U-0 | U-0 | U-0   |
| TON                | —   | TSIDL | —   | —   | —   | —   | —     |
| bit 15             |     |       |     |     |     |     | bit 8 |

|                    |       |        |        |       |       |       |       |
|--------------------|-------|--------|--------|-------|-------|-------|-------|
| <b>Lower Byte:</b> |       |        |        |       |       |       |       |
| U-0                | R/W-0 | R/W-0  | R/W-0  | R/W-0 | U-0   | R/W-0 | R/W-0 |
| —                  | TGATE | TCKPS1 | TCKPS0 | —     | TSYNC | TCS   | —     |
| bit 7              |       |        |        |       |       |       | bit 0 |

bit 15 **TON:** Timer1 On bit

- 1 = Starts 16-bit Timer1
- 0 = Stops 16-bit Timer1

bit 14 **Unimplemented:** Read as '0'

bit 13 **TSIDL:** Stop in Idle Mode bit

- 1 = Discontinue module operation when device enters Idle mode
- 0 = Continue module operation in Idle mode

bit 12-7 **Unimplemented:** Read as '0'

bit 6 **TGATE:** Timer1 Gated Time Accumulation Enable bit

When TCS = 1:  
This bit is ignored.

When TCS = 0:

- 1 = Gated time accumulation enabled
- 0 = Gated time accumulation disabled

bit 5-4 **TCKPS1:TCKPS0:** Timer1 Input Clock Prescale Select bits

- 11 = 1:256
- 10 = 1:64
- 01 = 1:8
- 00 = 1:1

bit 3 **Unimplemented:** Read as '0'

bit 2 **TSYNC:** Timer1 External Clock Input Synchronization Select bit

When TCS = 1:

- 1 = Synchronize external clock input
- 0 = Do not synchronize external dock input

When TCS = 0:

This bit is ignored.

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

- 1 = External clock from pin T1CK (on the rising edge)
- 0 = Internal clock (Fosc/2)

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

|                   |                  |                                    |                    |
|-------------------|------------------|------------------------------------|--------------------|
| <b>Legend:</b>    |                  |                                    |                    |
| 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 |