

Timers et PWM

cours n°8
LI326

Timer 0

- Compteur 8 bits TMR0 (ne peut être arrêté sauf si ...)
- Incrémenté par l'horloge interne (T0CS = 1)
- ou par un signal externe (T0CS = 0)
 - sur front montant (T0SE = 0)
 - ou sur front descendant (T0SE = 1)
- Le signal d'incrément peut être divisé par
 - 1 (PSA = 1)
 - 2, 4, 8, 16, 32, 64, 128 ou 256 (PSA = 0) en fonction des valeurs de PS[0:2] (resp. de 0 à 7)
- Le bit T0IF est mis à 1 lors du passage de 255 à 0
- Les bits de contrôles sont dans le registre OPTION_REG

RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
------	--------	------	------	-----	-----	-----	-----

Timers du p16f877

timer 0

8 bits
sert à fournir une base de temps aux programme

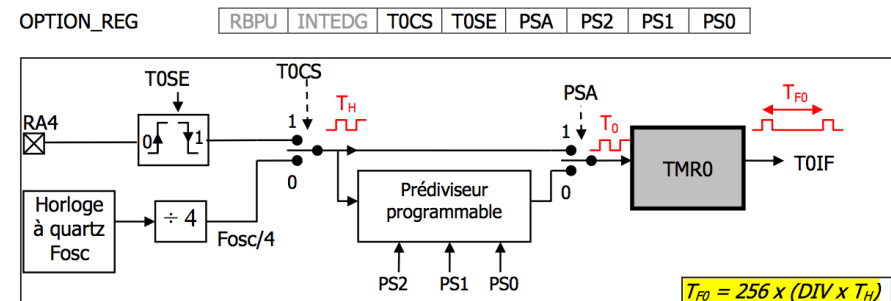
timer 1

16 bits
déclenchement de conversion AN
décodage le signaux RC5 (IR)

timer 2

8 bits
avec le timer1 permet la production de signaux PWM

Timer 0



à 20Mz

sans prédiviseur : on aura une interruption toutes les

$256 \times 4 \times 50 = 51.2$ microsecondes

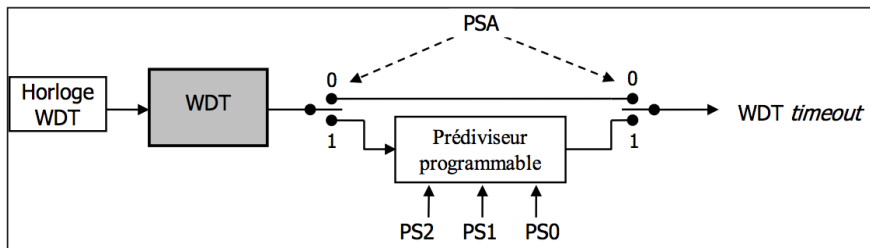
avec prédiviseur : ce sera de

103 à 131072 microsecondes (en fonction de PS[2:0])

Watch dog

- Le watch dog est un dispositif de sécurité qui provoque un reset en cas de comportement non prévu du logiciel.
- Le watch dog du p16f877 utilise le prédiviseur du timer 0.
- Le watch dog est demandé à la configuration, il ne peut ni être démarré, ni être arrêté par logiciel. Il n'est pas masquable.
- L'instruction clrwtd remet le compteur à 0.
- Le watch dog est un compteur 8 bits dont le débordement provoque un reset.
 - Si le reset intervient alors que le processeur travaille on saute à 0
 - Si le reset intervient en mode sleep alors on reprend le travail

Watch dog



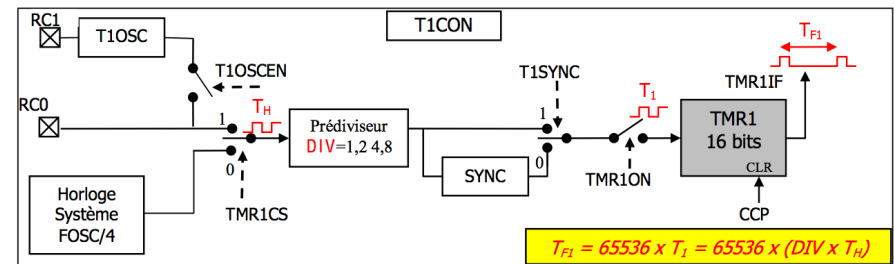
- Le prédiviseur permet de jouer sur la période de comptage
 - sans prédiviseur (PSA = 0)
 - avec prédiviseur (PSA = 1) on incrémente le WDT tous les 1, 2, 4, 8, 16, 32, 64, 128 cycles
- Le cycle du WDT est de 70 microsec provoquant un reset de $70\mu s * 256 = 18\text{ms}$ à $70\mu s * 256 * 128 = 2.3\text{secondes}$

Timer 1

- Compteur 16 bits TMR1H:TMR1L
- Incrémenté par l'horloge interne (TMR1CS = 0)
- ou par un signal externe (TMR1CS = 1)
 - signal simple (T1OSCEN = 0)
 - oscillateur externe (T1OSCEN = 1) (T1SYNC)
- Le signal d'incrément peut être divisé par
 - 1, 2, 4, 8 (PSA = 0)
 en fonction des valeurs de T1CKPS[1:0] (resp. de 0 à 3)
- Le bit T1IF est mis à 1 lors du passage de 65536 à 0
- Les bits de contrôles sont dans le registre T1CON

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

Timer 1



à 20Mz

On aura une interruption toutes les

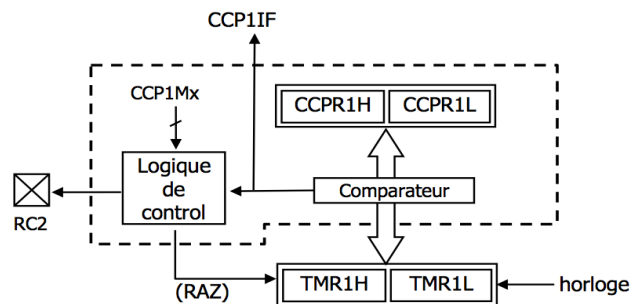
$$65536 * 4 * 50 = 131\text{ms jusqu'à } 1,048 \text{ secondes}$$

Le module CCP va permettre une remise à 0 précoce (avant débordement)

CCP : Comparaison / Capture

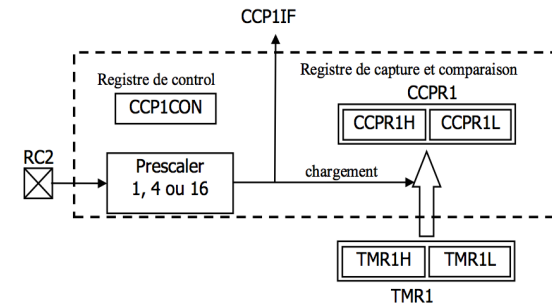
- Deux modules sont associés au TIMER1 : CCP1 / CCP2
- Chaque module dispose de deux modes :
 - mode **comparaison** permet d'attendre que le timer soit égale à la valeur d'un registre puis de déclencher une action sur d'autres périphériques et la levé d'une interruption
 - mode **capture** permet de prendre la valeur du timer lorsqu'on a détecté un événement.
- Les deux modules sont différents mais ne sont pas indépendants puisqu'ils partagent le même timer TMR1
- CCP2 peut déclencher la capture d'une valeur analogique

CCP : Comparaison



- Les registres CCPR1H et CCPR1L sont quelconques
- Lorsque $TMR1 == CCPR1$
 - on lève CCP1IF
 - en fonction de CCP1Mx on agit sur RC2 passage à 0, passage à 1, rien ou PWM

CCP : Capture



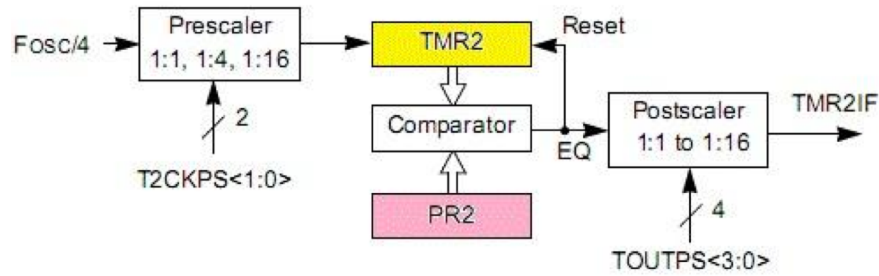
Les registres CCPR1H:CCPR1L recopient la valeur du compteur lorsque le bits RC2 à présenté

- un front montant ou un front descendant
- 4 fronts montants
- 16 fronts montants

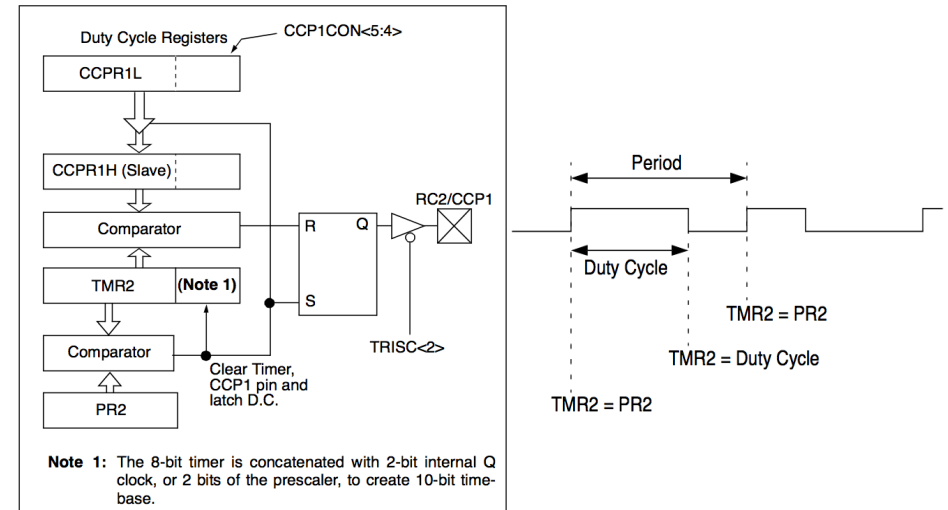
Timer 2

- Compteur 8 bits TMR2
- Incrémenté par l'horloge interne
- Le signal d'incrément peut être divisé par
 - 1, 4 ou 16 (T2CKPS[1:0])
- Le bit TMR2IF est mis à 1 lors après 1 à 16 reset
- Le reset du timer intervient lorsqu'il atteint la valeur PR2
- Le compteur peut être arrêté et redémarré.
- Le timer2 permet de fabriquer les signaux PWM

Timer 2



PWM : Pulse Width Modulation



PWM : Pulse Width Modulation

- Signal modulé en largeur d'impulsion
 - un signal avec une période fixée
 - dont la durée de l'impulsion (signal à 1) varie de 0 à 100%
- permet la commande de moteur
 - la valeur moyenne définit la vitesse de rotation
 - varie de 0 à 1 (de min à max)
- permet de produire un signal analogique basse fréquence
- Le registre PR2 définit la période du signal
- le registre CCPR1L définit la largeur de l'impulsion.

Exemples PWM



1. Clignotement progressif d'une led
2. Commande d'un servo-moteur
3. Production d'un signal analogique