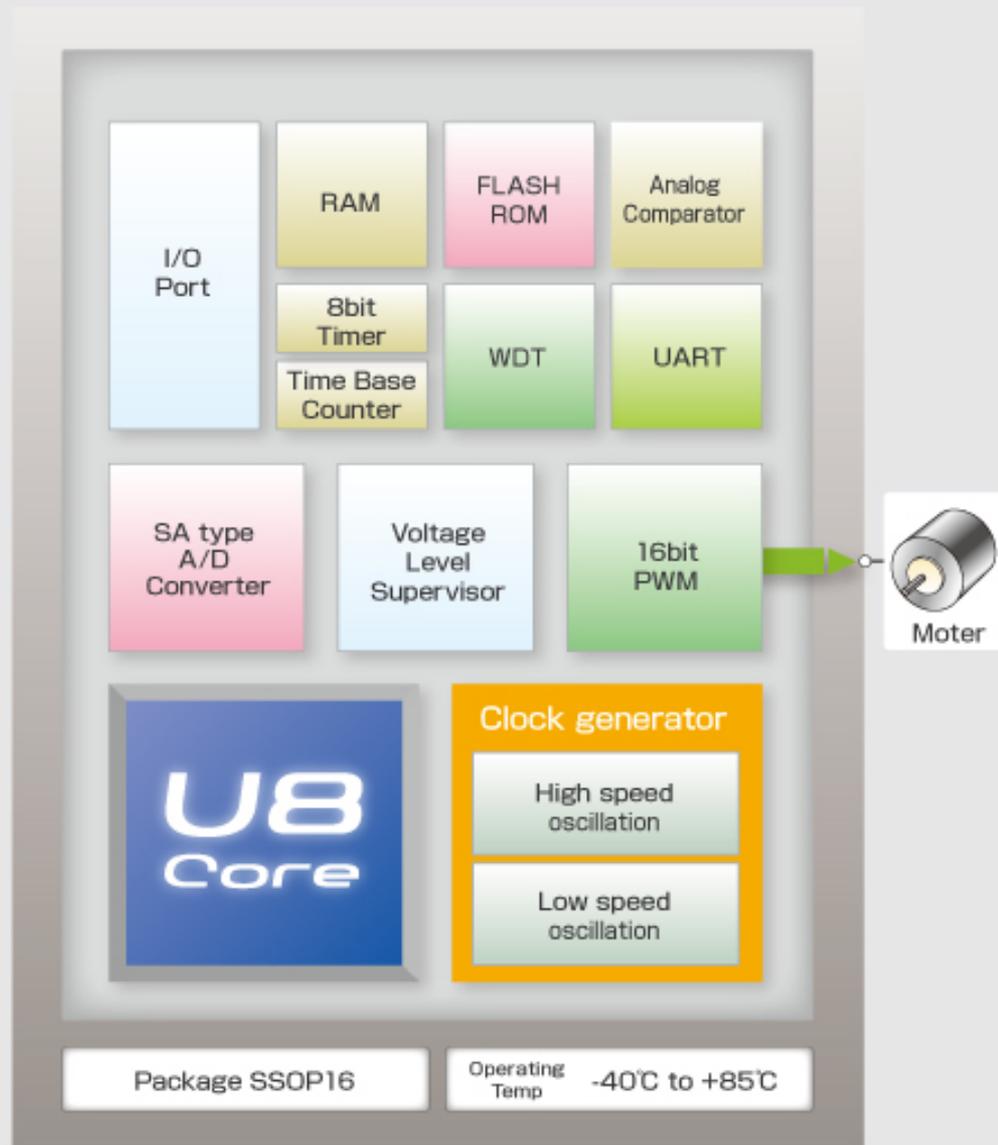


ML610Q101/102

Timer



Mesurer le Temps

Besoin de mesurer des durées, de générer des signaux bien définis temporellement.

Sans timer, cela serait bien difficile, car dépendant du temps d'exécution des instructions. Cela nécessiterait un contrôle délicat des boucles.

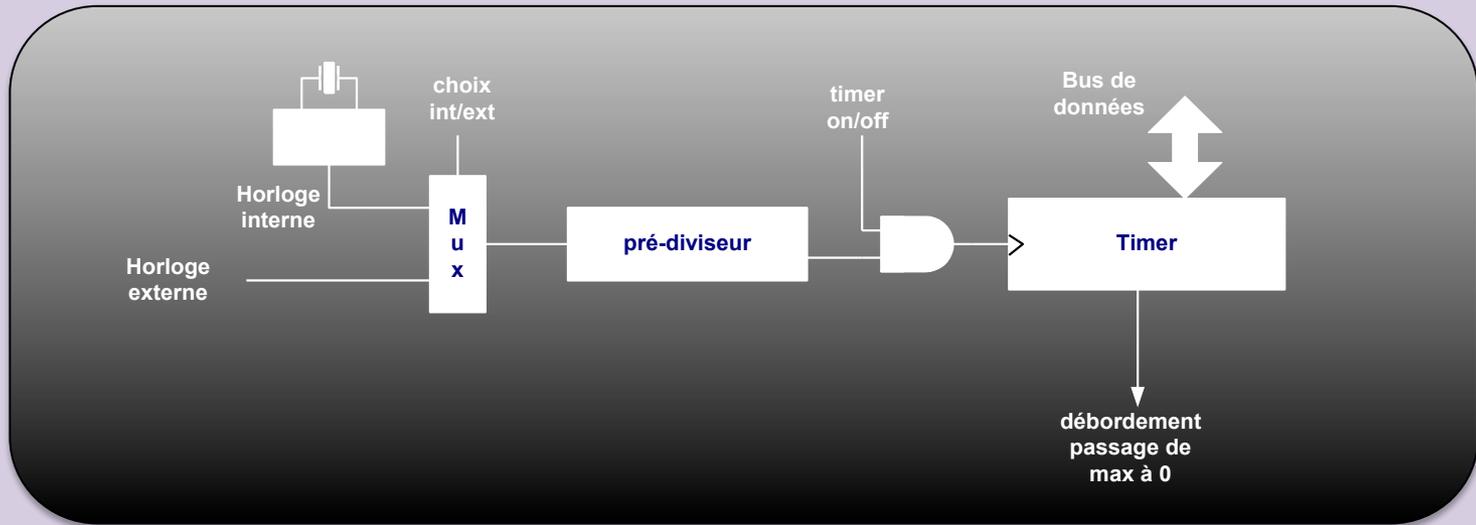
Leur usage n'est pas très compliqué, mais demande une bonne compréhension de la structure des différents timers d'un μC .

Il peut être parfois judicieux de faire un schéma fonctionnel de la configuration désirée.

Qu'est-ce qu'un timer?

Un timer est un compteur interne au μC qui peut être incrémenté par un signal d'horloge externe ou interne.

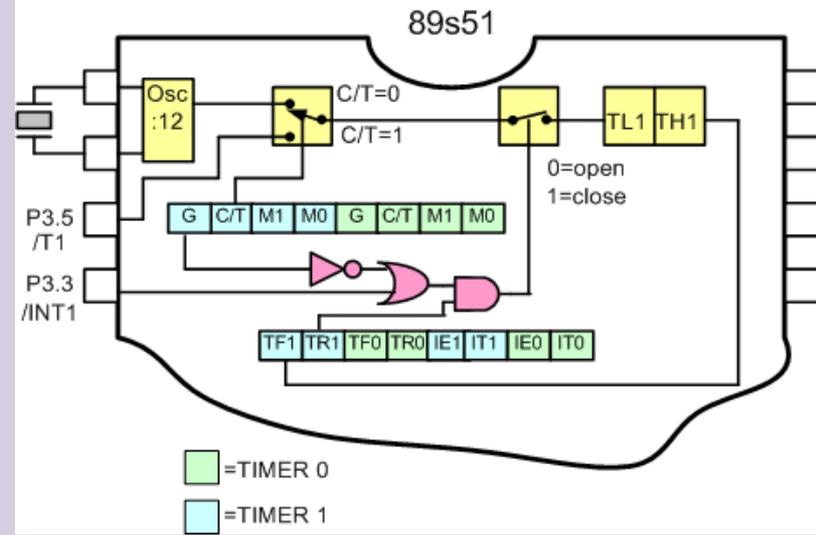
Il peut y avoir un prédiviseur avant le timer:



Un timer génère un « drapeau » de débordement à chaque passage de sa valeur max à zéro.

Certains timers peuvent être en écriture, ce qui permet de compter depuis une valeur particulière.

On peut lire à tout moment la valeur du timer.



Comptage d'évènements externes:

- Timer2 exclus d'office.
- On ne doit pas utiliser de pré-diviseur, sauf cas particulier.
- Timer0 → comptage jusqu' à 255
- Timer1 → comptage jusqu' à 65535

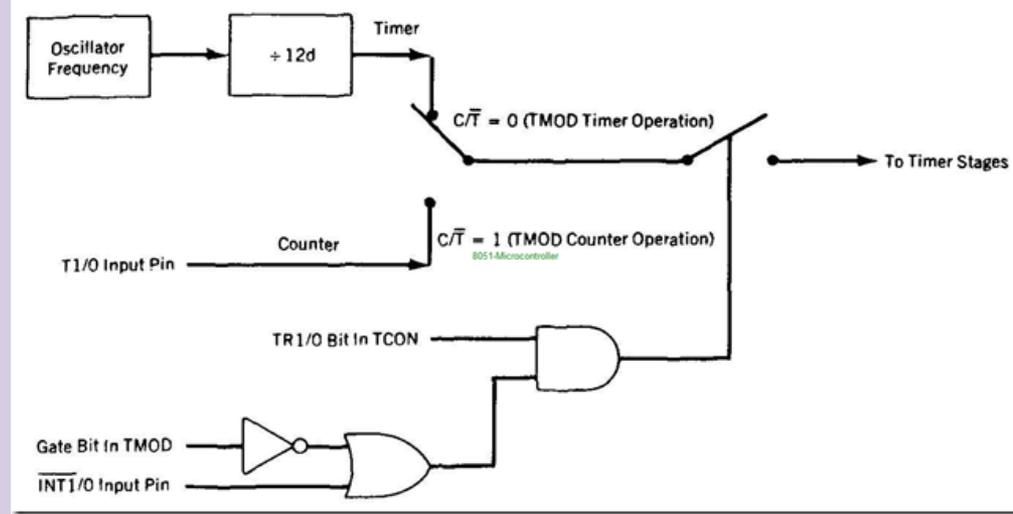
Mesure de temps:

- Tous les timers sont utilisables.
- La précision ou la capacité déterminera le Timer à utiliser

Il existe des dispositifs spécifiques internes utilisant les timers (Capture/Compare/PWM) que nous verrons plus loin pour la génération de signaux ou la mesure de période.

PIC16F87xA?

On dispose de 3 timers:



Timer0 sur 8 bits:

pré-diviseur programmable de 1:2 à 1:256,
lecture/écriture,
Horloge interne ou externe (choix du front pour cette dernière)
Interruption possible sur débordement \$FF à \$00

Timer1 sur 16 bits:

pré-diviseur programmable de 1:1 à 1:8,
Lecture/écriture,
Horloge interne ou externe (oscillateur possible pour cette dernière)
Interruption possible sur débordement \$FFFF à \$0000

Timer2 sur 8 bits :

pré-diviseur programmable 1:1, 1:4 ou 1:16,
post-diviseur programmable 1:1 à 1:16,
Lecture/écriture,
Horloge interne uniquement
comparateur entre TMR2 et PR2 qui génère un drapeau
Interruption possible sur sortie post-diviseur

C167...

- 2 Timers (GPT1 & GPT2)
 - 5 Timers (200/400ns) avec Input/Output, Reload et Capture
 - 2 Capture/Compare (CAPCOM1 & 2)
 - 4 Timers (400ns) avec Registre Reload et 32 Capture/Compare de 16 bits programmable (6 modes opératoire).
- 4 PWM de haute résolution
 - Avec base de temps indépendante each et une résolution de 50ns resolution (edge-aligned, center-aligned, burst and single-shot mode)
- ...

Utilisation du timer1:

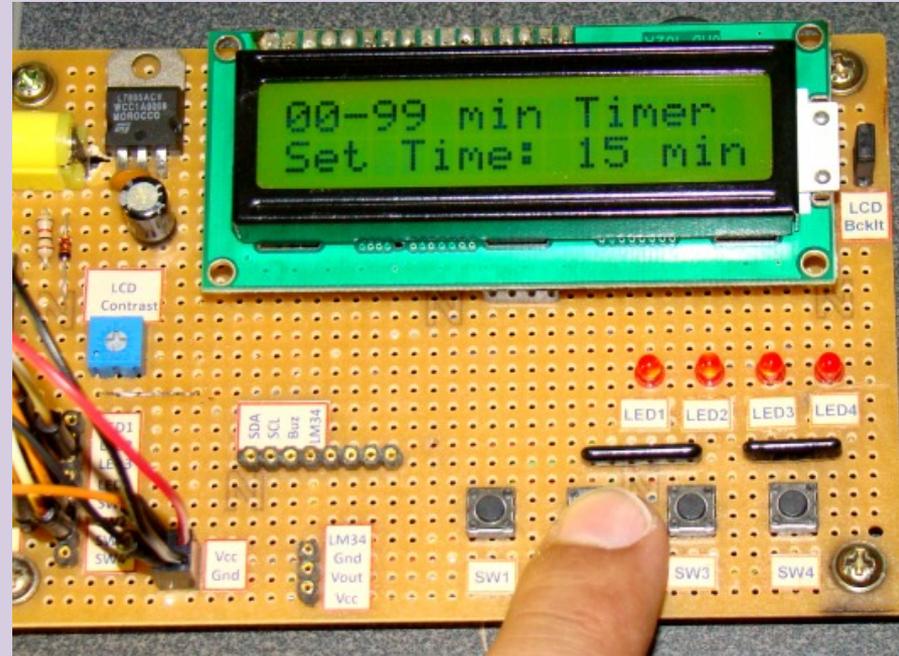
Clignotement de la LED rouge indépendamment du déroulement du programme.

Utilisation du timer0 en comptage d'événements:

Inversion de l'état de la LED tous les x appuis du bouton poussoir, indépendamment du déroulement du programme.

Utilisation conjointe Timer0 et Timer1:

Activation de la LED pendant 0,3s tous les x appuis du bouton poussoir, indépendamment du déroulement du programme.



Mesure de durée avec le module « Capture »

Ce module utilise le timer1. L'entrée de détection se fait sur l'entrée RC2/CCP1. (détection tous les 1, 4 ou 16 fronts).

A détection, le contenu du registre TMR1 est recopié dans le registre CCPR1.

Technique pour mesurer une durée:

Dans la routine d'interruption de CCP1, on mémorise l'instant du premier front ($T_1 = \text{CCPR1}$) puis au deuxième front, on mémorise cet instant ($T_2 = \text{CCPR1}$) puis on fait la différence.

Le problème, c'est que le timer1 peut être amené à déborder plusieurs fois avant que le 2^{ème} front arrive. Il faut donc dans la routine d'interruption de Timer1 compter les débordements et en tenir compte dans le calcul de durée.

FIGURE 8-1:

CAPTURE MODE OPERATION BLOCK DIAGRAM

