

4 Le Timer chien de garde

Description générale

- **Date du TP** : 8ième semaine ;
- **Durée du TP** : 1,5h (1 semaines) ;
- **Matériels nécessaires** : Carte d'experimentation, microcontrôleur PIC 16f877, programmeur, des composants électroniques ;
- **Logiciels utilisés** : Complilateur C PCW, Simulateur ISIS, IC Writer ;
- **Public cible** : 4ième niveau d'études de technicien supérieur en informatique industrielle ;
- **Partie du cours en rapport avec le TP** : Classification architecturale des systèmes temps reel, les microcontrôleurs.

Objectifs

- Manipuler le chien de garde (WDT) ;
- Manipuler les interruptions ;
- Approfondir les acquis en programmation C des microcontrôleurs PIC ;
- Maîtrise des architectures des microcontrôleurs Microchip PIC16.

Pré requis

Notions générales sur :

- L'électronique ;
- Algorithmique ;
- Programmation en C ;
- Les microcontrôleurs.

4.1 Introduction

Nos connaissances sur les timers ne seraient pas complètes si nous n'évoquions le timer chien de garde ou WDT car, même si son rôle n'est pas de générer des signaux comme ses homologues, il peut se révéler un auxiliaire très précieux pour votre application. Rappelons brièvement le principe de ce dernier. Il commence à compter, au rythme d'une l'horloge interne dont on peut définir la valeur, dès qu'il est validé et, s'il atteint sa valeur maximum et donc qu'il déborde, il génère un reset du microcontrôleur. Un programme qui fait appel à ce timer doit donc venir le remettre à zéro régulièrement afin d'éviter que ce débordement puisse avoir lieu et c'est justement là que se trouve tout l'intérêt de ce timer. En effet, si le programme "se plante" pour quelque raison que ce soit, il y a une très forte probabilité pour que cette remise à zéro régulière n'ait plus lieu. Le timer chien de garde va alors générer un reset et remettre ainsi le processeur "dans le droit chemin". La gestion de ce timer en C est parfaitement possible et fait appel à trois fonctions :

- `setupwdt (valeur)` qui définit la durée de débordement du timer chien de garde ;
- `restartwdt ()` qui réinitialise le timer chien de garde ;
- `#use delay(clock = valeur, RESTART_WDT)` qui permet au compilateur d'insérer automatiquement des réinitialisations du timer chien de garde, si elles s'avèrent nécessaires, au sein des fonctions `delay_XXX()` que votre programme peut être amené à utiliser.

Le montage de la figure 4.1, que vous pouvez une fois encore réaliser, permet de mettre en évidence le fonctionnement de ce timer.

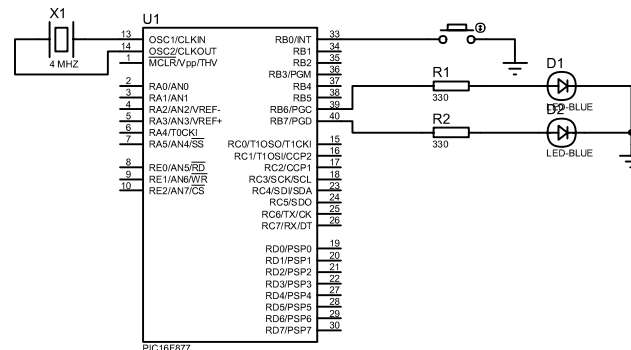


FIG. 4.1 – Schéma de l'utilisation du timer chien de garde

Dès sa mise sous tension, il allume pendant 2 secondes la LED connectée sur le port B7 puis il passe le reste de son temps à faire clignoter la LED connectée sur le port B6. Si l'on appuie sur le poussoir connecté sur le port B0, cela simule un "plantage" du programme au moyen d'une boucle d'attente infinie comme nous le verrons à l'examen du listing 5, ce qui est matérialisé par un arrêt du clignotement de la LED. Le timer chien de garde entre alors en action et réinitialise le programme, ce que l'on peut constater par le nouvel allumage, pendant 2 secondes, de la LED connectée sur B7.

```
//*****Listing 5*****  
#include <16F877.h>  
#device adc=8  
#use delay(clock=4000000, RESTART_WDT)  
#fuses WDT,XT, PUT, NOPROTECT, BROWNOUT, LVP, NOCPD, NOWRT, NODEBUG  
void main()  
{  
port_b_pullups(true);  
setup_timer_0(RTCC_INTERNAL);  
setup_wdt(WDT_288MS);  
output_b(0);  
delay_ms(2000);  
output_bit(PIN_B7,1);  
delay_ms(2000);  
output_bit(PIN_B7,0);  
  
while(true)  
{  
restart_wdt();  
while(!input(PIN_B0));  
output_bit(PIN_B6,0);  
delay_ms(500);  
output_bit(PIN_B6,1);  
delay_ms(500);  
}  
}  
//*****fin du Listing*****
```

4.2 Travail demandé

1. Compiler le programme avec PCW par la suite vérifier son bon fonctionnement à l'aide de la simulation par ISIS (pour simuler la liaison RS232 utiliser le terminal virtuelle).
2. Expliquer puis commenter le programme présenté dans les deux listing.
3. Réaliser le montage sur une plaque d'essai, programmer le microcontrôleur puis vérifier le bon fonctionnement du programme.
4. Modifier le programme de tel sorte que le Timer chien de garde déborde tout les 576 ms.