TP : timers et interruptions

Ce TP a pour but de se familiariser avec les Timers, en mode de scrutation des fanions et en mode interruption.

1) **Timer en mode « polling »**

Prenez le programme ***Timer-scru.c*** . Examinez-le. Que doit-il se passer ? Testez-le !

Modifiez le programme en changeant la ligne d'initialisation du Timer :

TACTL = TASSEL_2 + ID_0 + MC_2;

- Que doit-il se passer ? Testez !

2) **Timer en mode « Interruption »**

Prenez le programme ***Timer-Inter.c*** . Examinez-le. Que doit-il se passer ? Testez-le !

3) **Registre de comparaison**

a) Ecrivez un programme qui produit un clignotement à 1 Hz en utilisant le registre de comparaison TA0CCR0 et l'interruption TIMER0_A0_VECTOR (voir l'exemple du cours).

b) Passez la fréquence du processeur à 25 MHz (procédure fournie setupDCO) et ajoutez un compteur pour ne changer l'état de la LED1 que chaque 500 interruptions. Mesurez la période du signal. Pouvez-vous en déduire que la fréquence est bien à 25 MHz ?

4) **PWM par Timer et interruptions**

Reprenez le programme ***Timer-Inter.c*** . Vous avez presque ce qu'il faut pour gérer un PWM (et même quatre PWM). Utilisez l'interruption pour le début du cycle (LED1On). Activez l'interruption sur le registre de comparaison TA0CCR1 pour la fin du cycle (LED1OFF). Initialisez-le avec la valeur 16384 (le quart de 2 ^ 16). Observez le signal. C'est du PWM, mais il est trop lent !

Augmentez la fréquence du processeur à 25 MHz et enlevez la pré-division pour observer le PWM.

5) **Commande du moteur par PWM**

Intégrez la génération du PWM avec un timer dans l'un des programme de commande du moteur (par exemple le rendu *moteur-prop-inter.c*.

Rendu : Envoyez le dernier programme que vous avez fait fonctionner à ~~pyr@pyr.ch.~~ philippe.muellhaupt@epfl.ch