TP : lecture analogique et PWM multiple

L'objectif de ce TP est d'afficher la position du potentiomètre sur la barre des 5 LED bleues.

Matériel nécessaire : *carte MSP-EXP-430F5529 («carte blanche»).*

1) Prenez connaissance du programme ***roue.c***. Compilez-le. Des erreurs apparaissent. En effet, il nécessite l'usage d'une « librairie » HAL_Wheel. Une librairie est composée de deux fichiers :

- un ficher d'entête (header), qui indique tout ce que cette librairie offre (extension .h).
- un fichier source, qui est le programme proprement dit (extension .c).

Ajoutez les fichiers ***HAL_Wheel.h*** (en cliquant sur le nom du projet dans l'explorateur de projets, puis New, Header file) et ***HAL_Wheel.c*** (New, Source File). Le projet doit ensuite pouvoir se compiler sans erreurs.

Exécutez le programme. Observez l'affichage de la position en binaire sur les 5 LED bleues.

2) **PWM sur 5 LED**

Reprenez le programme ***inter-1pwm.c*** (corrigé question 4 du TP de la semaine 6)

Modifiez-le pour obtenir 5 signaux PWM sur les 5 LED bleues, en utilisant les 5 registres de comparaison TA0CCR0 à TA0CCR4 du timer TA0.

Utilisez 5 variables globales ***uint16_t pwm4 à pwm8*** pour les valeurs des pwm. Le programme principal pourra changer ces valeurs quand il le souhaite, mais ces valeurs seront copiées dans les registres TACCR0 à TACCR4 au moment du début du cycle (sur l'interruption overflow).

Initialisez par exemple les valeurs des PWM aux valeurs 15, 60, 3600, 21600 et 65500;

Observez les intensités des 5 LED bleues.

3) **Barre lumineuse réglable**

a) Ajoutez à votre programme la lecture de la valeur du potentiomètre dans la boucle principale. Affichez la valeur lue sur un PWM.

b) Synchronisez la lecture de la roue avec le cycle du PWM, au moyen d'une variable globale volatile. Mettez-là à 1 au début du cycle (interruption overflow). Dans la boucle principale, attendez qu'elle passe à 1 avant de lancer la lecture analogique et remettez-là immédiatement à 0. Vous éviterez ainsi des irrégularités gênantes lors de l'usage de plusieurs PWM. Il faut en effet qu'un groupe de valeurs soit appliqué au même cycle.

c) Faites que la rotation complète du potentiomètre se répercute sur les 5 LED, comme sur la vidéo de démonstration.

Astuces : J'avoue que j'ai mis du temps pour programmer la solution…

En utilisant les deux variables suivantes, le travail sera plus facile :

- uint16_t ***roueDiv*** *= roue / (MAX_ROUE/5);*

(avec une déclaration préalable #define MAX_ROUE 4095)

Comme il s'agit d'une division entière, cette variable indique sur quel cinquième de l'angle total se trouve le potentiomètre. Vous pourrez utiliser un switch…case de 0 à 4 pour gérer les 5 cinquièmes.

- uint16_t ***roueSolde*** *= ((roue % (MAX_ROUE/5))*5)<<4;*

Il s'agit du reste de la division entière. Cette valeur varie de 0 à (MAX_ROUE/5). Or le PWM sur une LED doit être donné sur 16 bits. On commence donc par multiplier cette valeur par 5, pour qu'elle varie entre 0 et MAX_ROUE (qui est la la valeur maximale du convertisseur 12 bits, donc 4095). Elle finalement décalée de 4 bits pour varier entre 0 et la valeur maximale du PWM (65'535).

Bon TP à distance !