Laboratoire d'actionneurs intégrés (LAI)
Commande embarquée de moteurs

Laboratoire 3 : Pont en H

Objectifs : Maîtrise d'un pont en H, PWM double, PWM simple.

Etapes préliminaires :

- Régler la source de tension d'alimentation sur 12V (limitée à 1A)
- Faire une copie de sauvegarde de votre dossier contenant le code du labo 1 avant de continuer votre travail
- Alternativement, vous pouvez créer un nouveau projet et y recopier les fichiers labo1.ioc et main.c

Alimenter le moteur en PWM double :

Activer le channel 2 :

La première étape consiste à dédoubler le code de démarrage des PWM afin de générer des signaux de PWM sur chacune des branches. Pour ce faire, on utilise les fonctions HAL_TIM_PWM_Start et HAL_TIMEx_PWMN_Start sur le channel 2 du timer 1. Vérifier qu'aucun de ces appels de fonction n'est commenté pour le channel 1.
Il faut également dédoubler l'affectation du niveau de comparaison dans l'interruption à l'aide du registre TIM1->CCR2.
Recopier le bout de programme de la routine d'interruption et commenter-le avant de passer à la suite (il sera utile pour la dernière partie du labo)

Choisir les transistors qui vont commuter :

A ce moment on a des signaux identiques au résultat du labo1 sur les deux branches du pont. On va les modifier en jouant sur un registre qui gère les sorties de chacun des channels. Ce registre est TIM1->CCER et les bits qui gèrent les sorties sont les bits 0, 2, 4 et 6 pour les sorties 1, 1N, 2 et 2N respectivement. Des constantes (#define) permettent de rendre le code plus lisible (voir exemple ci-dessous). Elles sont déjà déclarées par ST dans le ficher stm32f401xe.h.

La démarche pour faire un enable des sorties 1 et 2N est la suivante (à programmer dans l'interruption):

- Copier le registre dans une variable temporaire *tmpccer* (à déclarer de type uint32_t)
- Mettre à zéro les bits 0,2,4 et 6 de cette dernière variable (en utilisant tmpccer&=0xFFAA)
- Mettre à un les bits d'une paire de transistors opposés : par exemple pour les transistors 1 et 4 :
  tmpccer|=TIM_CCER_CC1E|TIM_CCER_CC2NE
- Affecter tmpccer au registre TIM1->CCER

Le moteur tourne maintenant en PWM double.

Faire tourner le moteur dans les deux sens (si vous avez le temps):
Pour le faire tourner dans les deux sens il s'agit d'ajouter une détection de seuil sur la valeur de la tension sur le potentiomètre de manière à avoir la position moteur à l'arrêt au milieu de la course du potentiomètre. Il faut ensuite calculer le taux à appliquer sur les registres CCR1 et CCR2 à partir de la tension mesurée (déterminer le calcul à faire avant le labo). Et mettre à jour CCER pour activer les bonnes sorties.

Pwm simple :
On souhaite n'avoir plus qu'un transistor qui fait la modulation (celui du haut). La manière la plus simple de le faire est de jouer sur les niveaux de comparaison à l'aide des registres CCR1 et CCR2 en affectant celui commandant le transistor du bas à PERIPWM (vérifier que c'est bien le cas).

Pwm double croisé:
Dans ce cas, il faut reprendre le code de l'interruption original du début du labo, puis changer la polarité des sorties dans l'interface graphique en passant de PWM mode1 à PWM mode 2 pour la configuration du channel 2 du timer 1**.** La gestion des CCER est inutile dans ce mode et le code est trivial !
Avant de mettre la tension, vérifier à l'aide de l'oscilloscope que vous avez un temps d'anti-chevauchement entre les commandes des transistors (1 et 2) et (3 et 4).
Il est particulièrement intéressant de visualiser ce qui se passe lorsque le rapport cyclique est aux alentours de 50%.