TP Moteur à courant continu (1)

Une série de travaux pratiques vont s'occuper de la commande d'un moteur à courant continu. Le moteur **Logidule**, avec ses capteurs de positions, va être utilisé.

Quelques indications vous seront données au début de la séance concernant l'utilisation des Logidules et la connexion à la carte MSP430F5529 (LaunchPad «rouge» ou carte «blanche»).

Pour la programmation des entrées-sorties, utilisez l'accès direct aux registres du microcontrôleur (P1DIR, P1OUT, P1IN, P1REN, P2...).

Utilisez (si possible) Code Composer Studio. Il faut créer un « projet CCS » et choisir le microcontrôleur MSP430F5529. Vous aurez à écrire la procédure **`int main() ...`** N'oubliez pas l'initialisation du Watchdog.

N'utilisez aucun appel Arduino. En particulier, delay() ne va plus fonctionner !

1) Récrivez le « clignotant commandé par un poussoir » en utilisant les registres du microcontrôleur (**`P1DIR, P1IN, P1OUT, P1REN,`** …) ainsi que les set-bit (|=), clear-bit (&=~) et test bit (&).

2) Observez le bloc Logidule **moteur à courant-continu**. On peut le faire tourner lentement à la main sans risque. Observez avec les lampes Logidules les signaux de **fin de course**.

Reliez les deux fins de course au MSP430 *(choix des Pins selon la carte que vous utilisez)*. N'oubliez pas de conncter la masse (Gnd, 0V) !

Les signaux de commande ***DIR*** et ***EN*** permettent de faire tourner le moteur dans les 2 sens. Reliez ces signaux au MSP430 *(choix des Pins selon la carte que vous utilisez)*.

Écrivez un programme qui donne un mouvement de **va-et-vient** entre les fins de courses lorsqu'on presse sur le bouton-poussoir Pous1.

3) Observez le signal fourni par les deux **capteurs optiques** (encodeurs).

Reliez ces signaux au MSP430 *(choix des Pins selon la carte que vous utilisez)*.

Imaginez un algorithme et écrivez le programme correspondant qui indique sur une LED le **sens de rotation** du moteur (faites tourner le moteur à la main pour la démonstration).

4) Écrivez un programme qui effectue les mouvements suivants :

- une avance d'une seconde pour un éventuel dégagement de la fin de course *(utilisez AttenteMs et non delay !)*
- un recul pour la recherche de la fin de course *(scrutation d'une fin de course)*
- une avance d'un tour de la roue dentée *(détection et comptage des fronts d'un encodeur)*