

## TP : accéléromètre et afficheur LCD

L'objectif de ce TP est d'utiliser l'accéléromètre et l'afficheur LCD.

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

### 1) Utilisation de bibliothèques

Importez le projet Accel-F5529 (comme expliqué dans la vidéo : File, Import, **déployer Code Composer Studio, CCS project, Next, Select Archive File**, choisir le fichier accel-F5529.zip, Ouvrir, Finish).

Prenez connaissance du programme **main.c**.

Remarquez les **#include** au début.

Regardez le contenu des fichiers **PYR\_CarteBlanche.h** et **PYR\_CarteBlanche.c**.

Regardez le contenu des fichiers **PYR\_5PWM.h** et **PRY\_5PWM.c**.

Compilez **main.c**. Placez votre carte sur une surface plane et exécutez le programme. Faites bouger ensuite votre carte et observez les valeurs des accélérations sur les PWM des LED : l'axe X sur LED5 (valeurs positives) et sur LED6 (valeurs négatives), l'axe Y sur LED 7 et LED8, la valeur absolue de l'axe Z sur LED4, signe de l'axe Z sur LED1 et LED3.

Observez les courbes X et Y qui apparaissent sur l'afficheur LCD.

Regardez de détail le programme. Il commence par les initialisations (carte, PWM, accéléromètre, et afficheur). Laissez pour le moment la procédure `Cma3000_calibre()`;

La boucle principale affiche si nécessaire les axes sur l'afficheur, puis se synchronise avec le début du cycle du PWM et fait l'acquisition des accélérations : `Cma3000_readAccel()`; Laissez de côté pour le moment le calcul de `accX`, `accY` et `accZ`.

Regardez ensuite le calcul des PWM et des valeurs des LED.

Regardez finalement l'affichage tous les 50 cycles des composantes X et Y de l'accélération, pour former deux courbes.

### 2) Accélérations relatives

Bien que la bibliothèque `HAL_Cma3000` offre la possibilité de placer des offsets sur les accélérations des trois axes, j'ai programmé cette fonctionnalité dans le programme principal.

Regardez la procédure de calibration `Cma3000_calibre()`. Elle commence par attendre 100 cycles du PWM, ce qui semble stabiliser les valeurs des accélérations après l'initialisation du circuit. Ensuite, elle lit 16 fois les accélérations et mémorise les moyennes des valeurs en X, Y et Z dans `offsetX`, `offsetY` et `offsetZ`.

Dans la boucle principale, après la lecture des accélérations, ces offsets sont soustraits aux valeurs lues pour donner les valeurs `accX`, `accY` et `accZ`, qui deviennent alors des variations par rapport à la position initiale de la carte (horizontale et sans mouvement).

### 3) Balle sur un plan à inclinaison variable

Modifiez le programme pour qu'il affiche une balle (procédure `Dogs102x6_circleDraw`) qui se déplace sur l'afficheur en fonction de l'inclinaison donnée à la carte (accélérations X et Y).

*Bonus* : vous pouvez ajouter un obstacle graphique à franchir, ou programmer un labyrinthe...

**Envoyez votre code source à [pyr@pyr.ch](mailto:pyr@pyr.ch)**

(envoyez seulement le `main.c`)

**Bon TP à distance !**