

TP : Interruptions

Le but de ce TP est de mettre en œuvre des interruptions sur un MSP430.

1) Interruption sur une entrée

Prenez le programme *Inter.c*. Examinez-le !

Testez-le. L'interruption se fait avec le poussoir 1. Les rebonds de contact sont parfois visibles : la LED change alors d'état aussi au relâchement du poussoir.

a) Modifiez le programme pour qu'il réagisse au fronts montant du signal (relâchement du poussoir).

b) Modifiez-le ensuite pour qu'il réagisse aux deux fronts.

Idée : changer le mode à chaque interruption.

2) Interruption sur deux entrées

a) Modifiez le programme pour avoir une interruption sur les deux poussoirs 1 et 2, le poussoir 1 agissant sur la LED 1 et le poussoir 2 sur la LED 2.

Comme les poussoirs ne sont pas sur le même port, il sera nécessaire d'avoir deux routines d'interruptions, contrairement à l'exemple du cours où deux entrées du même port se partagent un vecteur d'interruption.

b) Modifiez le programme pour qu'il compte (poussoir 1) et décompte (poussoir 2) les interruptions et qu'il affiche la valeur en binaire sur les LED 4 à 8 (LED bleues).

La variable de comptage devra être une variable globale, vu qu'elle sera initialisée par le programme principal, incrémentée dans la routine d'interruption du poussoir 1 et décrétementée dans la procédure d'interruption du poussoir 2.

La procédure `AfficheLedBleues(uint16_t val)` affiche la valeur en binaire sur 5 bits.

3) Encodeurs par interruptions

Écrivez un programme qui gère les signaux des encodeurs par interruption. L'algorithme devient particulièrement simple : la partie qui était nécessaire pour la détection des changements est naturellement réalisée par les deux routines d'interruption, il ne reste plus que la discrimination du sens.

Testez votre programme avec un moteur Logidule. Affichez la position avec la procédure *`AfficheLedBleues`*.