Séance 3 : Programmation avec interruptions

|  |
| --- |
| A LIRE AVANT DE MANIPULER LE MATÉRIEL |
| * Déposez la carte sur une surface plate, dégagée et non métallique * Ne JAMAIS forcer en branchant des câbles/cartes * Ne JAMAIS forcer en manipulant des boutons/potentiomètres (c’est pas en appuyant plus fort que ça marchera mieux) * Eviter tant que possible de toucher les parties métalliques de la carte, en particulier l’arrière et les connecteurs, pour éviter tout risque de corrosion ou de court-circuit * Toujours remballer les différents éléments dans leur boîte avant transport |

1. Téléchargez le projet mplab contenu dans l'archive "interrupts.zip".
2. Programmez votre carte avec ce projet

* Constatez l'effet des 2 boutons.
* Remarquez la présence de deux interruptions à des fréquences différentes.
* Comment fait-on pour s'assurer que la pression sur le bouton 1 n'est prise en compte qu'une seule fois ?

1. Réécrivez les programmes de la séance précédentes en tirant parti des nouvelles possibilités qu'offrent les interruptions. Conseils :
   * Générez des interruptions 2 fois par seconde. Clignotement : leds = leds^0b0011
   * Utilisez une variable pour compter les interruptions. Gérez le clignotement en fonction de la valeur de cette variable et de la valeur sur 10 bits renvoyée par le convertisseur analogique-digital.
   * Dans le while(1) ou dans une interruption rapide ajoutez, par exemple :
     + setLeds(leds ^ (0b1100\*getButton1()) ). Pourquoi ?
2. Voir l'exemple fourni lors de cette séance
3. Idem
   * Initialisez une interruption à haute fréquence.
   * Créez un compteur d’interruption allant de 0 à 1023 (voir le compteur de centièmes de seconde dans l'exemple). À chaque passage de 1023 à 0, allumer les leds
   * Éteignez-les lorsque le compteur atteint la valeur du potentiomètre
4. On peut décaler les bits d'une variable d'une position vers la gauche (vers la droite) en la multipliant (en la divisant) par un facteur 2.
5. Comptez des interruptions à haute fréquence. La valeur de ce comptage peut être prise comme valeur 'aléatoire' lorsque l'on appuie sur le bouton.
6. Utilisez des variables d'état.