

La Bouton, la LED et l'évènement

1. 1. Objectif
2. Mise en pratique

Objectif

Le but de ce TME est d'apprendre à programmer le PIC en connectant des composants logiciels connectés par des évènements. Un évènement est une extension des interruptions matérielles. Un évènement se compose d'un drapeau à deux états (un bit) associés éventuellement à un buffer pour l'échange de données.

Le principe est simple.

- Soit deux composants logiciels (des tâches) TA et TB.
- Soit un buffer de donnée BUF.
- Soit un drapeau d'évènement EVT.
 - ♦ EVT=0 signifie que BUF est vide et qu'il appartient à TA.
 - ♦ TA ne peut écrire dans EVT que si EVT contient 0, et il ne peut écrire que 1.
 - ♦ EVT=1 signifie que BUF est plein et qu'il peut le lire.
 - ♦ TB ne peut lire dans EVT que si EVT contient 1, et il ne peut écrire que 0.

Au départ on suppose que EVT=0

- TA veut envoyer une donnée à TB
 - ♦ TA écrit dans BUF
 - ♦ TA met EVT à 1
 - ♦ TA attend que EVT=0
 - ♦ [TA lit la réponse de TB] (optionnel)
- TB veut lire une donnée de TA
 - ♦ TB attend que EVT=1
 - ♦ TB lit depuis BUF
 - ♦ [TB met une réponse dans BUF] (optionnel)
 - ♦ TA met EVT à 1

Mise en pratique

L'idée est de commander 2 leds connectés sur deux broches du PORTB (RB1 et RB2) et lire deux boutons poussoirs connectés sur deux broches du portB (RB4 et et RB5). Nous n'allons pas utiliser les interruptions, ni aucun périphérique (ce serait nécessaire pour avoir des bases de temps précises mais nous le ferons plus tard.

Nous allons créé plusieurs taches communicant par événements

- **TIMER**
 - ♦ interface:
 - ◊ **e_tic1ms**: mise à 1 à chaque milliseconde
 - ♦ comportement:
 - ◊ fait une attente active d'1 milli-seconde et mets e_tic1ms à 1
- **SEQUENCEUR**
 - ♦ interface:
 - ◊ **e_tic1ms**: bit d'activité
 - ◊ **e_bp**: demande de lecture du bouton poussoir 1, mise à 1 toutes les 10ms

- ◊ **e_led**: demande de changement d'état de la led
- ◆ comportement:
 - ◊ se réveille par e_tic1ms, met à 1 l'évènement e_bp toutes les 10ms, e_led1 toutes les 125ms.
- **BP**
 - ◆ interface:
 - ◊ **e_bp**: demande de lecture des BP
 - ◊ **e_appui_bp**: mise à 1 lorsqu'un appui a été détecté sur le bouton.
 - ◆ comportement:
 - ◊ détecte l'appui du bp comme expliqué en cours
- **LED**
 - ◆ interface:
 - ◊ **e_led**: demande de changement d'état de la led
 - ◊ **e_bp**: demande d'extinction de la led.
 - ◆ comportement:
 - ◊ fait clignoter la led après un appui, l'éteint la led après un autre appui.