

## TP Découverte des Interruptions

### A partir de

DataSheet P18Fxx20  
 Cours MCC18  
 Doc ressource data sheet PIC18F  
 Programmes : demo\_it\_rb0.c  
 Carte PICDEM2+

### Découverte des interruptions

Créer un projet MPLAB (si nécessaire) avec comme seul fichier source : demo\_it\_rb0.c  
 Tester et analyser ce programme, pour cela :

- a) En mode « run », constatez que la LED sur PRB3 bascule à chaque appui sur le bouton relié à INT0
- b) Arrêter le programme et placer un point d'arrêt dans le sous-programme d'interruption, sur la ligne `INTCONbits.INT0IF=0;`  
 Effectuer une avance en « pas à pas » que constatez-vous ? Conclure quant à l'avantage de la programmation en interruption par rapport à la programmation par scrutation.  
 Lancer le programme, constatez que le point d'arrêt est attend lors de l'appui sur le bouton  
 Visualiser en pas à pas dans une fenêtre « watch » la variable `cpt` et vérifier qu'elle est incrémentée à chaque appui sur le bouton.
- c) A l'aide du data sheet du PIC utilisé, indiquer le rôle des bits et la page de référence correspondant dans le datasheet :
 

```
TRISBbits.TRISB0
INTCONbits.INT0IE
INTCONbits.GIE
INTCONbits.INT0IF
```
- d) Expliquez pour quelle raison le programme d'interruption `void une_fonction(void)` renvoie l'exécution de l'interruption sur la fonction `void it_sur_rb0(void)`
- e) Donner la signification des « pragmas » :
 

```
#pragma interrupt it_sur_rb0
#pragma code vecteur_d_IT=0x08
#pragma code
```
- f) Dessiner l'algorithme du programme en faisant apparaitre le programme principal et le sous-programme d'interruption.
- g) L'interruption a lieu lors du relâchement du bouton, modifier le programme de manière à déclencher l'interruption lors de l'appui sur le bouton, consulter pour cela le data sheet et rechercher des informations sur le bit `INTEDG0` du registre `INTCON2`
- h) Modifier le programme de manière à déclencher l'interruption lors du relâchement et de l'enfoncement du bouton
- i) Modifier le programme initial de manière à utiliser l'interruption `INT1` à la place de `INT0`, quelle modification matérielle cela entraine t il ? Faire l'essai sur le simulateur VSM
- j) Modifier le programme initial de manière à incrémenter `cpt` sur l'IT `INT0` et décrémenter `cpt` sur l'IT `INT1`

## demo\_it\_rb0.c

**Fonctionne sur carte PICDEM2+ avec oscillateur externe**

```
#include <p18f4620.h>

#pragma config PBADEN = OFF

unsigned int cpt=0;    // compteur d'interruption

#pragma interrupt it_sur_rb0
void it_sur_rb0(void)
{
    if (INTCONbits.INT0IF)
    {
        cpt++;
        PORTBbits.RB3= ! PORTBbits.RB3;
        INTCONbits.INT0IF=0;
    }
}

#pragma code vecteur_d_IT=0x08
void une_fonction(void)
{
    _asm
        goto it_sur_rb0
    _endasm
}
#pragma code

void main (void)
{
    TRISBbits.TRISB0=1;
    TRISBbits.TRISB3=0;
    INTCONbits.INT0IE=1;
    INTCONbits.GIE=1;
    while(1);
}
```

**IMPORTANT** : Le PORTB n'est pas automatiquement configure en mode numérique, il peut être nécessaire d'ajouter la directive « #pragma config PBADEN = OFF » au programme.