Les interruptions

Lors des séances précédentes, la boucle de temporisation mobilisait complètement le microcontrôleur qui attendait la fin des décrémentations ou la fin du comptage des timers.
Afin de libérer le microcontrôleur de cette attente, et de lui permettre d'effectuer d'autres tâches pendant ce temps, il est possible d'utiliser des interruptions de programme.

Les interruptions du PIC16F877A

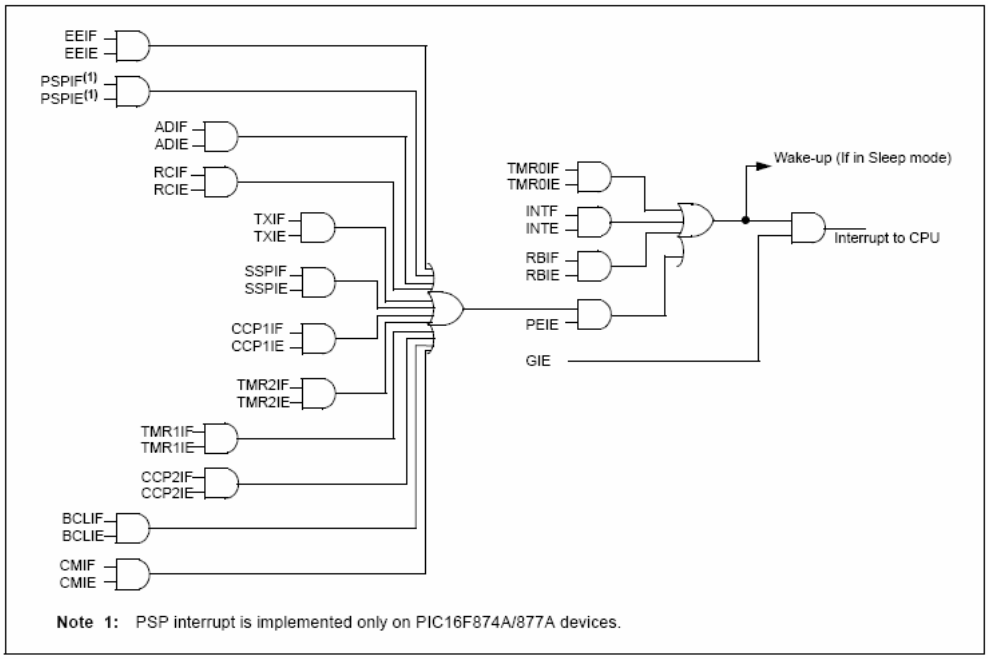
Le PIC16T877A possède 15 sources d'interruptions ; citons par exemple :

- le débordement des timers ;
- un front sur l'entrée INT (multiplexée avec PB0) ;
- un changement sur les entrées PB4 à PB7 ;
- une fin de conversion du CAN ;
- des données reçues par la liaison série etc…

Chaque source d'interruption dispose :

- d'un bit drapeau (Flag) qui passe au NL1 lorsque l'événement attendu se produit ; par exemple pour le timer 0, c'est par exemple le bit « **TMR0IF** » du registre « **IT1CON** » ;
- d'un bit autorisant ou non l'interruption, c'est par exemple le bit « **TMR0IE** » du registre « **IT1CON** » ;

Le bit « **GIE** » (General Interrupt Enable) du registre « **ITCON** » autorise ou interdit toutes les interruptions comme le montre le schéma suivant.



Excepté pour le timer 0, l'entrée INT et un changement sur PB4 à PB7, les bits drapeaux des périphériques se trouve dans les registres « **PIR1** » et « **PIR2** » tandis que les bits autorisations se trouvent dans les registres « **PEI1** » et « **PEI2** ».
Les interruptions des périphériques, hormis ceux cités plus haut, sont autorisées ou interdites par « **PEIE** » du registre « **ITCON** ».

Lorsqu'un événement susceptible de générer une interruption se produit (par exemple le débordement du timer 0) :

- le bit drapeau associé (« TMR0IF » dans notre exemple) passe au NL1 ;
- si l'interruption est autorisée (dans notre exemple si « GIE » et « TMR0IE » sont au NL1), alors une interruption est générée ;
- le compteur programme est alors chargé à 0004h qui est l'adresse du vecteur d'interruption, tandis que l'adresse actuelle du programme est sauvegardée dans la pile ;
- l'interruption interdit toute nouvelle interruption par mise au NL0 de « GEI » ;
- à l'adresse 0004h le programmateur doit placer une instruction de saut vers le sous programme d'interruption ;
- celui-ci doit commencer par sauvegarder si nécessaire le contexte avant l'interruption : registre « W », « STATUS » et « PCLATH » ;
- le sous programme d'interruption cherche ensuite la source d'interruption en testant les bits drapeau (ce n'est pas nécessaire si une seule source est autorisée) ;
- gère l'interruption concernée ;
- remet le bit drapeau à 0 pour permettre une nouvelle interruption ;
- restaure le contexte d'avant l'interruption ;
- si le retour au programme principal se fait par l'instruction « RETFIE », le bit « GEI » est automatiquement remis au NL1.

Voici la solution que propose Microchip pour sauvegarder le contexte du programme avant l'interruption et le restaurer après le traitement de l'interruption.
Les registres « **W** » (pour l'opération en cours), « **STATUS** » (pour les banques pointées en mémoire registres et données, ainsi que les indicateurs divers) et « **PCLATH** » (pour les pages pointées en mémoire programme) sont sauvegardés dans des registres temporaires par l'instruction « **SWAPF** ».
Celle-ci présente l'avantage de ne pas affecter les bits d'état.
Cette instruction inverse partie haute et basse de la donnée, mais comme l'opération est faite deux fois (lors de la sauvegarde du contexte puis lors de la restauration), cela reste transparent.

```
MOVWF   W_TEMP          ;Copy W to TEMP register
SWAPF   STATUS,W        ;Swap status to be saved into W
CLRF    STATUS          ;bank 0, regardless of current bank, Clears IRP,RP1,RP0
MOVWF   STATUS_TEMP     ;Save status to bank zero STATUS_TEMP register
MOVF    PCLATH, W       ;Only required if using pages 1, 2 and/or 3
MOVWF   PCLATH_TEMP     ;Save PCLATH into W
CLRF    PCLATH          ;Page zero, regardless of current page
:
:(ISR)                  ;(Insert user code here)
:
MOVF    PCLATH_TEMP, W  ;Restore PCLATH
MOVWF   PCLATH          ;Move W into PCLATH
SWAPF   STATUS_TEMP,W   ;Swap STATUS_TEMP register into W
                        ;(sets bank to original state)
MOVWF   STATUS          ;Move W into STATUS register
SWAPF   W_TEMP,F        ;Swap W_TEMP
SWAPF   W_TEMP,W        ;Swap W_TEMP into W
```

Programmation

Ouvrir un nouveau projet TP3_interrupt pour PIC16F877A, y inclure le fichier « Prog1.asm » du répertoire « Ressources \ TP3_interrupt », sauvegarder le fichier dans votre répertoire sous un nom différent, supprimer la version d'origine du projet.

Analyser le programme (reproduit en annexe).
Compiler, simuler et programmer.

Reprendre le programme réalisé lors de la séance précédente, utilisant le timer 1 pour la temporisation ; modifier ce programme pour gérer la temporisation par interruptions.

Annexe 1 : Programme avec gestion des interruption du timer 0

```
;*******************************************************************************
; Ce programme fait clignoter la DEL de la sortie RB0 sur la carte PICDEM2PLUS
; à une fréquence de 7,5 Hz en utilisant le timer 0 en interruption          *
;*******************************************************************************

        LIST P=16F877A              ; directive qui définit le processeur utilisé
        #include <P16F877A.INC>     ; fichier de définition des constantes

;*******************************************************************************
;               BITS DE CONFIGURATION
;*******************************************************************************
        __CONFIG  _HS_OSC & _WDT_OFF & _CP_OFF & _CPD_OFF & _LVP_OFF

;*******************************************************************************
;               DEFINITIIONS DE VARIABLE                                      *
;*******************************************************************************
bascule         EQU     0x20    ; Variable destinée à faire basculer la DEL

;*******************************************************************************
;               DEMARRAGE SUR RESET                                           *
;*******************************************************************************
        org     0x0             ; Adresse de départ après reset
        goto    debut

        org     0x4             ; Adresse du vecteur d'interruption
        goto    interrupt       ; saut au sous programme d'interruption


        org     0x10            ; adresse de début du programme
debut

;*******************************************************************************
;               PROGRAMME PRINCIPAL                                           *
;*******************************************************************************
        call init
boucle
        movf    bascule,w
        movwf   PORTB           ; allume ou éteint la LED à l'image du bit 0 de bascule
        goto    boucle          ; rebouclage
;*******************************************************************************
;               SOUS PROGRAMMES                                               *
;*******************************************************************************
;*******************************************************************************
;               INITIALISATION                                                *
;*******************************************************************************
init
                        ; initialisation du PORTB en sortie
        BANKSEL PORTB           ; passage en banque 0
        clrf    PORTB           ; RAZ des bascules D du port B
        BANKSEL TRISB           ; passage en banque 1
        movlw   b'00000000'
        movwf   TRISB           ; PORTB en sortie

                        ; initialisation du Timer 0 :division par 256 de l'horloge interne
        BANKSEL OPTION_REG      ; Accès à la BANK1
```

```
        movlw   b'00000111'             ; TOSC=0 PSA=0 PS2=1 PS1=1 PS0=1
        movwf   OPTION_REG              ; predivision TMR0 timer 1:256

                        ; initialisation des interruptions
        bsf     INTCON,T0IE             ; autorisation interruptions du timer 0
        bsf     INTCON,GIE              ; autorisation générale des interruptions

        BANKSEL PORTA                   ; retour en banque 0

        return

;*******************************************************************************
;               INTERRUPTION                                            *
;*******************************************************************************
interrupt
        movlw   0x01                    ; chargement de bascule
        xorwf   bascule,f               ; inversion de tous les bits
        bcf     INTCON,T0IF             ; prépare pour l'interruption suivante
        retfie

;*******************************************************************************
END
```

Annexe 2 : exemple de solution pour la gestion du timer 1 en interruption

Seules les parties modifiées par rapport au programme précédent ont été reproduites :

```
;*******************************************************************************
;               INITIALISATION                                                  *
;*******************************************************************************
init
                        ; initialisation du PORTB en sortie
        BANKSEL PORTB                   ; passage en banque 0
        clrf    PORTB                   ; RAZ des bascules D du port B
        BANKSEL TRISB                   ; passage en banque 1
        movlw   b'00000000'
        movwf   TRISB                   ; PORTB en sortie

                        ; initialisation du Timer 1 :division par 8 de l'horloge interne
        BANKSEL T1CON       ; passage en banque 0
        movlw   b'00110001'     ; T1CGPS1=1 T1CGPS0=1 T1OSCEN=0 TMR1CS=0 TMR1ON=1
        movwf   T1CON           ; prédivision TMR1 timer 1:8, hor ext off, hor int; timer on

                        ; initialisation des interruptions
        BANKSEL PIE1                    ; passage en banque 1
        bsf     PIE1,TMR1IE             ; autorisation interruptions du timer 1
        bsf     INTCON,PEIE             ; autorisation interruptions des périphériques
        bsf     INTCON,GIE              ; autorisation générale des interruptions

        BANKSEL PORTA                   ; retour en banque 0
        return
;*******************************************************************************
;               INTERRUPTION                                            *
;*******************************************************************************
```

```
interrupt
        movlw   0x01                    ; chargement de bascule
        xorwf   bascule,f               ; inversion de tous les bits
        bcf     PIR1,TMR1IF             ; prépare pour l'interruption suivante
        retfie

;*********************************************************************************
END
```