



T.P. N°3 microprocesseurs

Gestion des interruptions du PIC 16F877

I. Aperçu

Une interruption est un mécanisme dont dispose le microprocesseur pour répondre à la demande des périphériques afin d'initier un transfert de données par exemple. Le microprocesseur arrête alors l'exécution du **programme principal** et exécute un programme spécial appelé **programme de gestion d'interruption**.

Le PIC 16F877 dispose de plusieurs sources d'interruption mais ne possède qu'un seul vecteur d'interruption c'est-à-dire un seul programme de gestion. Dans ce TP il s'agit d'étudier l'interruption externe, sur la ligne RB0/INT.

Comme son nom l'indique, la ligne RB0/INT peut être utilisée soit comme ligne numérique d'entrée/Sortie Standard ou comme entrée d'interruption. Si cette ligne est utilisée comme entrée d'interruption il faut alors d'une part l'activer et d'autre part écrire une routine de gestion qui exécutera le programme y afférent.

I.1 Le registre INTCON (INTerrupt CONtrol) (adresse : 0x0B)

Le registre INTCON est un registre situé en mémoire données et contient les bits qui permettent d'activer ou de désactiver les interruptions ainsi que le bit d'état de l'interruption sur la ligne RB0/INT.

7	6	5	4	3	2	1	0
GIE			INTE			INTF	

GIE (General Interrupt Enable)

0 : masque le déclenchement de toutes les interruptions

1 : active le déclenchement des interruptions

INTE (INTerrupt Enable)

0 : masque le déclenchement de l'interruption sur la ligne RB0/INT

1 : active le déclenchement de l'interruption sur la ligne RB0/INT

INTF (Interrupt Flag)

0 : pas d'interruption sur la ligne RB0/INT

1 : une interruption est déclenchée par la ligne RB0/INT

Pour permettre donc au PIC16F877 de prendre en compte l'interruption externe (RB0/INT), il faut obligatoirement mettre les bits d'activation **GIE** et **INTE** à 1. Le bit d'état **INTF** est alors mis à 1 automatiquement quand une interruption vient de la ligne RB0/INT et la procédure de gestion de l'interruption commence. **Ce bit doit être remis à 0 avant de quitter le programme d'interruption.**



I.2. Procédure de gestion d'interruption

Quand une interruption survient une procédure spéciale est exécutée automatiquement par le PIC16F877 :

- a- L'instruction en cours d'exécution est terminée,
- b- Les interruptions sont masquées,
- c- Le contenu du registre PC est stocké dans la pile,
- d- 0x0004 est mis dans le registre PC.

Après cette procédure, le PIC16F877 se branche au programme de gestion d'interruption écrits par l'utilisateur.

À la fin du programme de gestion d'interruption, le PIC récupère le contenu stocké précédemment (étape c) dans la pile et retourne au programme principal. Il doit alors retrouver tous les registres utilisés dans le même état où ils étaient avant d'avoir quitté le programme principal.

I.3. La fonction de gestion d'interruption

Pour gérer les interruptions, l'utilisateur doit écrire un programme qui doit respecter certaines conditions.

- a- Il doit stocker obligatoirement les deux registres w et STATUS en mémoire ainsi que tout registre dont le contenu risque d'être modifié par le programme d'interruption,
- b- Il doit identifier la source de l'interruption en scrutant le bit d'activation et le bit d'état de chaque interruption prise en charge,
- c- Réaliser la tâche demandée par l'interruption
- d- Remettre à 0 le bit d'état de chaque interruption traitée,
- e- Restaurer les deux registres w et STATUS et tout registre stockée précédemment
- f- Terminer le programme par l'instruction: RETFIE

II. travail à faire

On veut réaliser un dispositif qui permet à l'aide d'un bouton de lancer le clignotement d'une led ou de l'éteindre. Le fonctionnement du système doit être le suivant :

- Au démarrage du système la led est éteinte,
- Si on appuie sur le bouton, la led change d'état (si elle était éteint, alors elle se met à clignoter et inversement).

Pour cela on utilise la ligne RB1 pour connecter la led et la ligne RB0/INT pour connecter le bouton (Fig.). On doit donc configurer la ligne RB1 comme sortie et la ligne RB0/INT comme entrée d'interruption.

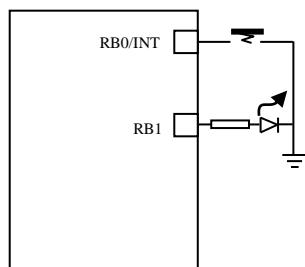


Fig. Schéma électronique de principe du système



II. Travail à faire

1. Créer un projet Mplab ou ouvrir un projet existant.
2. Créer un nouveau fichier (Menu **File/New**).
3. Enregistrer le fichier (Menu **File/Save As...**) dans le dossier c:\Mplab, lui donner le nom **TP3.asm**.
4. Saisir le programme ci-dessous, le compiler (Menu **Project/Build All**), corriger les éventuelles erreurs
5. Céer un "break point" au niveau de l'instruction **GOTO INT**

```
#include <P16F877.inc>

;::::::::::::::::::::::::::::::::::;;
;      PROGRAMME PRINCIPAL      ;
;::::::::::::::::::::::::::::::::::;;
ORG    0X000
GOTO  MAIN
ORG 0X004
GOTO  INT

; CONFIGURATION DES ENTREE/SORTIES
MAIN    BSF    STATUS,RP0
        BCF    TRISB,0
        BCF    TRISB,1
        BCF    STATUS,RP0
        BSF    PORTB,0
        BSF    STATUS,RP0
        BSF    TRISB,0
        CLRF   OPTION_REG
        BCF    STATUS,RP0
        BCF    PORTB,1

; CONFIGURATION DES INTERRUPTIONS
        MOVLW  0x10
        MOVWF  INTCON
        BSF    INTCON,7
        CLRF   0X72
; ATTENTE D'UNE INTERRUPTION
INIT    BTFSS  0X72,0
        GOTO  INIT
; CLIGNOTEMENT
        BSF    PORTB,1
ATT1    DECFSZ 0X73,F
        GOTO  ATT1
        BCF    PORTB,1
ATT2    DECFSZ 0X73,F
        GOTO  ATT2
CONT   GOTO  INIT

;::::::::::::::::::::::::::;;
;      FONCTION D'INTERRUPTION      ;
;::::::::::::::::::::::::::;;

INT     MOVWF  0X70
        SWAPF  STATUS,W
        MOVWF  0X71
FIN
        BTFSS  INTCON,INTE
        GOTO  FIN
        BTFSS  INTCON,INTF
        GOTO  FIN
        COMF   0X72,F
        SWAPF  0X71,W
        MOVWF  STATUS
        SWAPF  0X70,F
        SWAPF  0X70,W
        BCF    INTCON,INTF
        RETFIE
END
```