

TP N° 4

Gestion des interruptions via le PIC16F84A

4.1 Objectifs

- Comprendre le mécanisme des interruptions.
- Mise en œuvre des interruptions à base du microcontrôleur PIC16F84A.

4.2 Introduction

Une interruption est un événement imprévisible qui provoque l'arrêt d'un programme en cours d'exécution pour aller exécuter un autre programme appelé programme (ou routine) d'interruption. A la fin du programme d'interruption, le microcontrôleur reprend le programme principal à l'endroit où il s'est arrêté comme le montre la figure 4.1.

On distingue deux types d'interruptions :

- Les interruptions externes, qui sont déclenchées lorsqu'un événement extérieur se produit tels que le changement d'état d'une entrée destinée à l'interruption.
- Les interruptions internes, qui sont déclenchées par le déroulement du programme tel que le résultat d'un calcul ou le débordement d'un Timer.

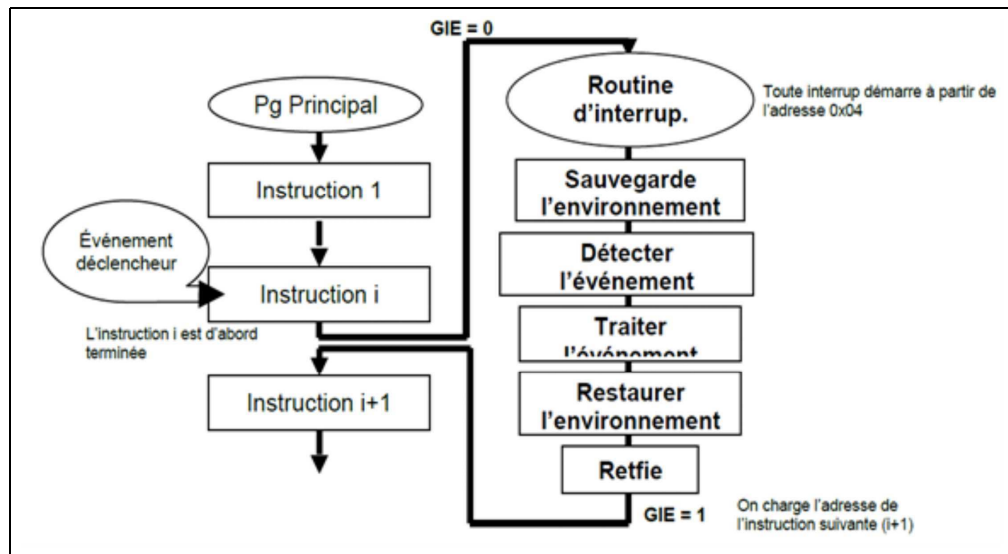


FIGURE 4.1: Séquence classique de fonctionnement d'une interruption

Dans le cas du 16F84 les différentes sources d'interruption sont au nombre de quatre :

1. **INT** : interruption externe sur la broche RB0/INT.
2. **TMR0** : interruption interne fin du comptage (débordement du registre TMR0).
3. **PORTB** : interruption externe sur changement du niveau logique d'au moins une de ces 4 broches : RB4, RB5, RB6 ou RB7 (port B).
4. **EEPROM** : interruption interne fin d'écriture en EEPROM

4.2.1 Registre de configuration des interruptions

Toute interruption est gérée à l'aide de 3 bits :

- Un bit d'activation globale (Global Enable bit). Ce bit permet d'activer ou de désactiver toutes les interruptions.
- Un bit d'activation (Enable bit). Ce bit permet d'activer ou de désactiver l'interruption correspondante.
- Un bit indicateur ou drapeau (Flag bit). Ce bit est mis à 1 lorsque l'interruption correspondante survient.

Pour le microcontrôleur PIC16F84A les bits de configuration des interruptions sont regroupés dans le registre INTCON (Figure 4.2).

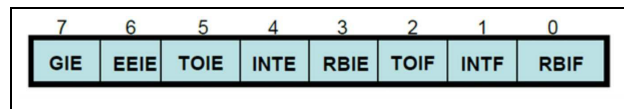


FIGURE 4.2: Structure du registre INTCON

- **GIE** : permet la validation générale des interruptions.
- **EEIE** : autorise l'interruption de fin d'écriture de l'EEPROM.
- **TOIE** : autorise l'interruption de débordement du registre TMR0.
- **INTE** : autorise l'interruption sur la broche RB0/INT.
- **RBIE** : autorise l'interruption sur les broches RB4, RB5, RB6, RB7 du port B.
- **TOIF** : drapeau (flag) mis à 1 lors du débordement du registre TMR0.
- **INTF** : drapeau (flag) mis à 1 lors d'un front (montant ou descendant selon l'état du bit INTEDG du registre OPTION_REG) sur la broche RB0/INT.
- **RBIF** : drapeau (flag) mis à 1 lors d'un changement de niveau logique d'au moins une des broches : RB4, RB5, RB6 ou RB7 (cela ne concerne que les broches configurées en entrée).

Remarque 1 : le drapeau EEIF qui signale la fin de l'écriture sur l'EEPROM se trouve dans le registre spécial EECON1 (bit 4).

Remarque 2 : Pour tous les bites de type drapeau (flag), c'est le matériel qui le met à 1 lors de la survenue de l'interruption, mais le logiciel (c'est-à-dire vous, le programmeur) qui le met à 0 à la fin de la routine d'interruption.

4.3 Travail à réaliser

Dans ce qui suit, nous allons réaliser quelques applications permettant la mise en œuvre des interruptions. Ces applications vont nous permettre de bien comprendre le mécanisme des interruptions.

4.3.1 Application N° 1 : Interruption sur la broche RB0/INT

Le but de cette première application est de programmer une interruption externe sur la broche RB0/INT permettant d'inverser l'allumage d'une LED (connectée à la broche RB0) à chaque pression sur un bouton poussoir connecté à la broche RB0.

1. Réaliser le circuit correspondant sous ISIS.
2. Afin de bien comprendre la programmation du mécanisme d'interruption du PIC16F84A, on vous donne le programme de cette première application :

```

void main() {
trisa=0x00; trisb=0x01;

INTCON.GIE=1;      //validation générale des interruptions
INTCON.INTE=1;     //autorise l'interruption sur la broche RB0/INT
OPTION_REG.INTEDG=1;

//l'interruption de la broche RB0/INT est activé sur un front montant
while(1){
// attendre le déclenchement de l'interruption pour l'allumage de la LED
}
}

void interrupt()      // routine d'interruption
{ porta.f0 = ~porta.f0;      // inverser l'état de RA0
INTCON.INTF=0;          //fin de l'interruption RB0/INT
}
    
```

Important : En MikroC, le sous-programme d'interruption est déclaré en tant que fonction avec le nom spécial « interrupt ». Cette fonction s'exécute automatiquement en réponse aux évènements déclencheurs des interruptions activées par l'utilisateur.

4.3.2 Application N°2 : Interruptions sur les broches (RB4-RB7)

Soit une LED connectée sur le pin RA0 du portA. Utiliser l'interruption externe sur les pins (RB4-RB7) pour changer la fréquence de clignotement de la LED :

1. Utiliser une fréquence de clignotement égale à 1500ms après la pression sur le bouton connecté à RB4.
2. Utiliser une fréquence de clignotement égale à 1000ms après la pression sur le bouton connecté à RB5.
3. Utiliser une fréquence de clignotement égale à 500ms après la pression sur le bouton connecté à RB6.
4. Utiliser une fréquence de clignotement égale à 200ms après la pression sur le bouton connecté à RB7.

4.3.3 Application N°3 : Chronomètre

Nous allons maintenant réaliser un chronomètre simplifié en utilisant trois afficheurs 7-segments BCD et un bouton connecté à la broche RB0. Le temps doit s'afficher sur les trois afficheurs 7-segments sous le format suivant (Figure 4.3) : « m -ss » (où m représente les minutes et s représente les secondes).

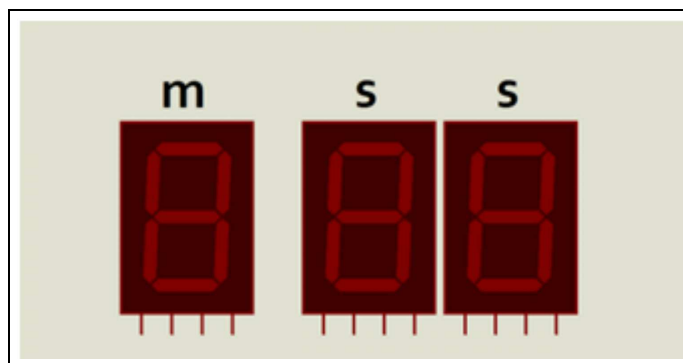


FIGURE 4.3: Format d'affichage du chronomètre

Programmer une interruption externe sur la broche RB0/INT permettant de réaliser le fonctionnement suivant :

1. A la mise sous tension, le chronomètre est initialisé à «0-00».
2. Un premier appui sur le bouton déclenche le chronomètre, le temps évolue sur les trois afficheurs.
3. Un deuxième appui sur le bouton arrête le chronomètre, le temps écoulé est affiché sur les trois afficheurs.
4. Un troisième appui sur le bouton réinitialise le chronomètre.