

# RENDU DU TP 3

le 19 avril 2024,  
version 1.1

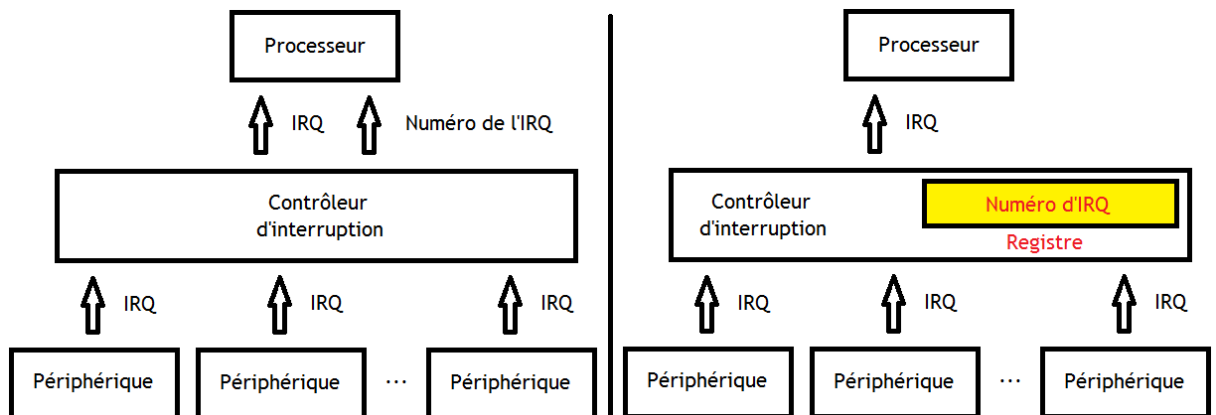
Mohamed Toujani | Taha Zougari



# TIMER ET INTERRUPTION

1. Qu'est-ce qu'une IRQ ? Proposer un schéma. Ne pas répondre uniquement Interrupt Request

Une IRQ (Interrupt Requête par le matériel) est essentiellement un signal qu'un appareil comme souris ou un clavier envoie à l'ordinateur pour dire "j'ai besoin d'attention maintenant". Ça permet à l'ordinateur de suspendre ce qu'il fait pour s'occuper de ce que le périphérique demande, comme quand tu appuies sur une touche de ton clavier et que ça apparaît à l'écran. Chaque périphérique a son propre signal d'IRQ, donc l'ordinateur sait quel périphérique veut son attention. C'est important pour que tout fonctionne bien ensemble dans ton ordinateur.

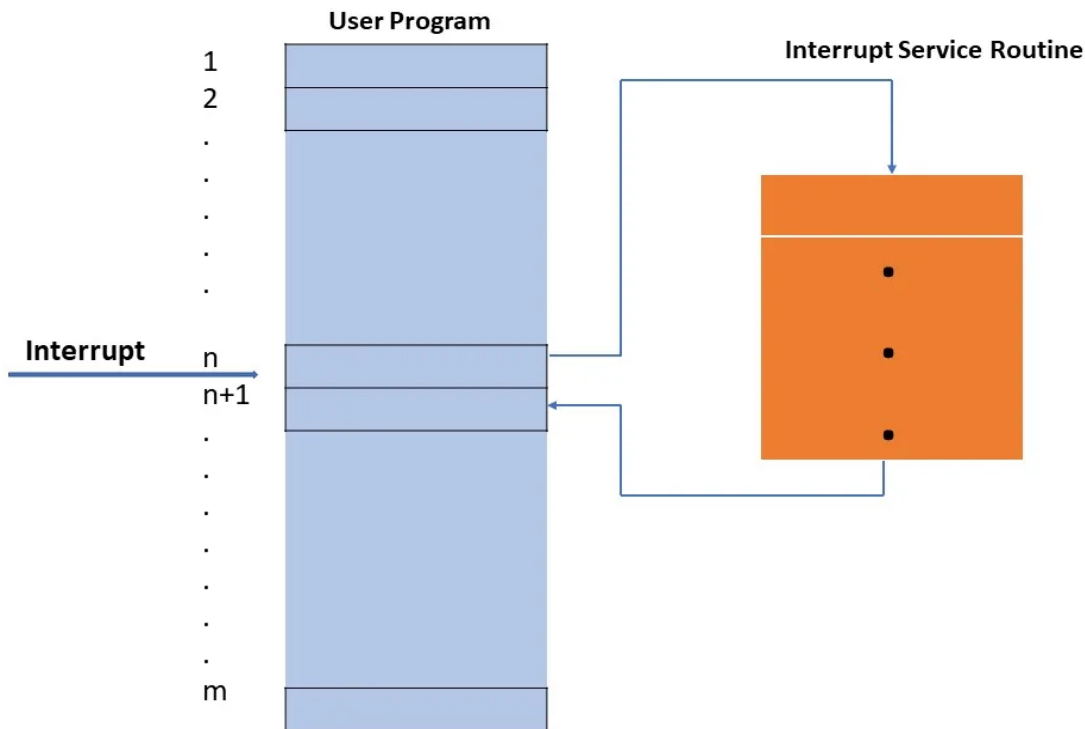


2. Qu'est-ce qu'une ISR ? Proposer un schéma. Ne pas répondre uniquement Interrupt Service Routine

Une ISR est un morceau de code qui est exécuté en réponse à une interruption matérielle, comme une IRQ. Elle est chargée de gérer cette interruption en effectuant les actions nécessaires pour répondre à la demande du périphérique. Ces actions peuvent inclure la lecture de données à partir du périphérique, la mise à jour de variables de statut, ou d'autres opérations spécifiques au périphérique.

# TIMER ET INTERRUPTION

---

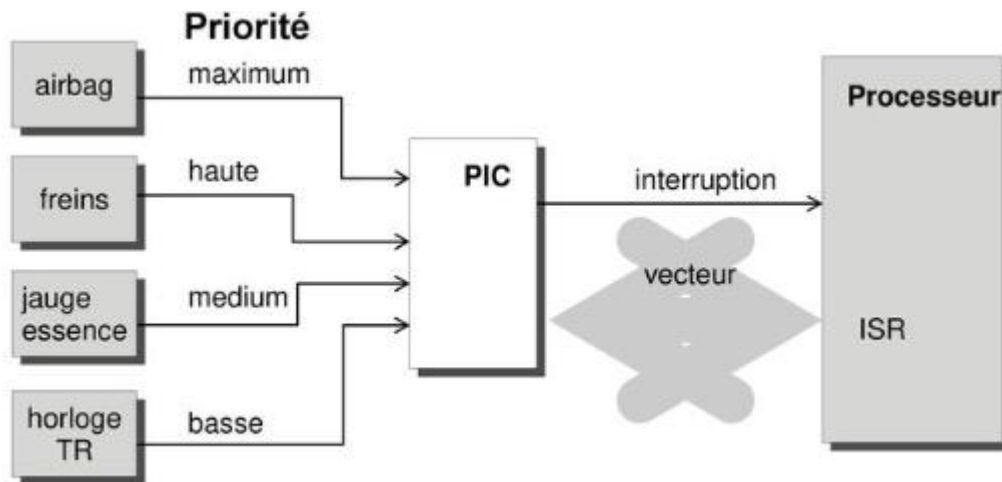


3. Qu'est-ce qu'un vecteur d'interruption et expliquer son rôle ? Proposer un schéma .

Un vecteur d'interruption, c'est un peu comme une liste de numéros de téléphone, mais pour les problèmes dans un ordinateur. Quand un problème survient, l'ordinateur regarde dans cette liste pour trouver le numéro associé à ce problème, puis il sait où chercher la solution pour résoudre ce problème. Cela aide l'ordinateur à réagir rapidement et efficacement aux différents soucis qui peuvent arriver .

Un vecteur d'interruption est moyen technique pour attacher une routine d'interruption à une requête d'interruption.

# TIMER ET INTERRUPTION



4. Proposer une configuration en assembleur permettant de configurer l'interruption de priorité haute pour le Timer0 avec démasquage global d'interruption. Répéter ce même travail pour le Timer1.

```

; Configuration de l'interruption de priorité haute pour Timer0

; Définir l'interruption de priorité haute pour Timer0
BSF INTCON2, TMR0IP ; Activer l'interruption de priorité haute pour Timer0

; Activer le Timer0
BSF TOCON, TMR0ON ; Activer le Timer0

; Activer l'interruption globale
BSF INTCON, GIE ; Activer l'interruption globale

; Configuration de l'interruption de priorité haute pour Timer1

; Définir l'interruption de priorité haute pour Timer1
BSF IPRI, TMR1IP ; Activer l'interruption de priorité haute pour Timer1

; Activer le Timer1
BSF T1CON, TMR1ON ; Activer le Timer1

; Activer l'interruption globale
BSF INTCON, GIE ; Activer l'interruption globale

```

# TIMER ET INTERRUPTION

---

5. Proposer une configuration en assembleur pour le Timer0. Nous souhaitons lever une interruption toutes les 20ms. Nous travaillerons avec une référence d'horloge interne (CPU clock 64MHz)

```
; Configuration du Timer0 pour générer une interruption toutes les 20ms
; Charger la valeur initiale dans TMR0
MOVLW 0x50
MOVWF TMR0H
MOVLW 0xC0
MOVWF TMR0L
BSF STATUS, RP0
BCF TOCON, TOCS
BSF TOCON, PSA
BCF STATUS, RP0
BSF INTCON, TMR0IE
BSF INTCON, GIE
```



Ecole Publique d'ingénieurs et d'ingénieurs en 3 ans

6 boulevard Maréchal Juin, CS 45053  
14050 CAEN cedex 04

