CHAP 9 - INTERRUPTIONS

Informatique Embarquée

S2 2020- 2021

GEII - Toulouse III

A. Nketsa

Les interruptions

1- Définition

Une interruption est un mécanisme de prise en compte et de traitement de demande d'urgence dans un système à microprocesseur. Pour cela, le microprocesseur ou microcontrôleur dispose des entrées dédiées pour cette demande d'urgence.

2- Mécanisme

1- Un programme est en cours d'exécution,
2- une demande d'interruption survient
3- le microprocesseur termine l'instruction en cours
4- le microprocesseur sauvegarde le point de retour
4- le microprocesseur va répondre à la demande en exécutant le traitement correspondant
5- à la fin du traitement de la demande, le microprocesseur reprend éventuellement le programme interrompu en récupérant le point de retour sauvegardé.

3- Rappel du fonctionnement simplifié d'un processeur pour comprendre les interruptions

Le fonctionnement d'un processeur est basé sur l'exécution d'un programme stocké comme une succession d'ordres (instructions) dans une mémoire.
Pour réaliser ce fonctionnement, le processeur a besoin entre autres :

- ☞ d'un compteur programme (encore appelé pointeur d'instruction) qui :
  - s'incrémente pour assurer la nature séquentielle du programme
  - contient l'adresse de la case contenant le code de la prochaine instruction à exécuter à la fin de l'instruction en cours.
- ☞ d'une zone mémoire (appelée Pile) pour les sauvegardes temporaires et gérée par un registre appelé pointeur de pile qui indique l'emplacement pour cette sauvegarde.

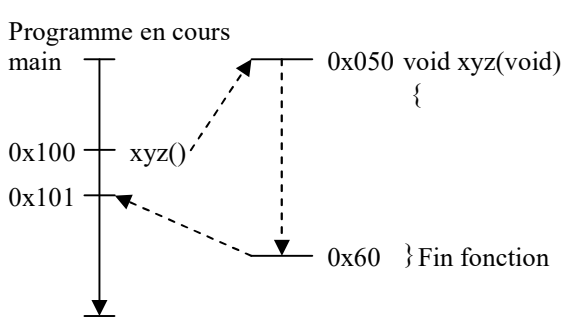
Conclusion

Application à l'exécution d'une fonction : 2 étapes : - l'appel de la fonction - le retour de la fonction	

Pour exécuter une fonction,
Au moment de l'appel de la fonction
- le processeur - sauvegarde sur la Pile l'adresse de retour
  - charge l'adresse de début de la fonction dans le compteur-programmre
- le processeur peut alors exécuter le code de la fonction
- la fin de la fonction correspond au retour au programme qui a appelé la fonction

Pour cela, le processeur charge le compteur-programme avec l'adresse de retour récupérée de la pile.

	Appel : - Sauvegarde de 0x101 sur la Pile - Chargement de 0x50 dans le compteur programme	**Fin de la fonction = Retour** - Récupération de 0x101 sur la pile - Chargement de 0x101 dans le compteur-programme - Reprise de main à la ligne 0x101

void xyz(void)
{
 	
}
 	
void main(void)
{
-
-
xyz();
-
}
		

4- Analogie de fonctionnement

Nous allons faire l'analogie entre la lecture d'un journal et la réponse à un appel téléphonique pour illustrer les interruptions :

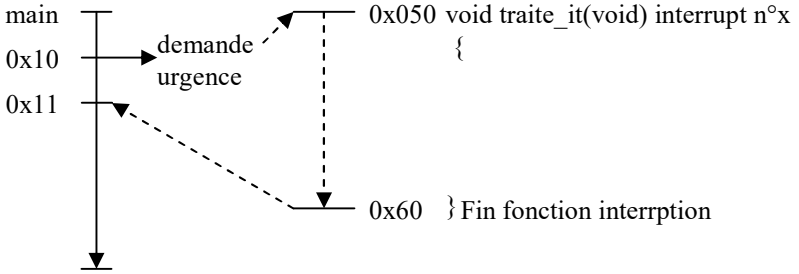
Vous	Le µP en général	Le µC C167
1- vous branchez votre téléphone	1- autoriser la prise en compte des demandes d'urgence	2 niveaux d'autorisation Plusieurs sources de demandes d'urgence (ou demandes d'IT)
2- vous lisez le journal Evolution des pages et lignes	2- un programme est en cours d'exécution ⇒ passage en revue des lignes du programme (évolution du compteur programme)	Fonctionnement normal du processeur
3- le téléphone sonne	3- une demande d'urgence survient	Une ou plusieurs source d'urgence autorisées font leur demande
4- vous terminez la lecture de la ligne en cours	4- le processeur termine l'instruction en cours	Le C167 termine l'instruction en cours
5- vous sauvegardez les numéros de la page en cours et de la ligne suivante	5- le processeur sauvegarde le numéro de la ligne (compteur programme) contenant la prochaine instruction à exécuter (On l'appellera l'adresse de retour)	Le C167 sauvegarde sur la Pile le contenu du compteur-programme.(adresse de retour)
6- vous allez répondre au téléphone	6- le processeur - interdit les demandes d'urgence - charge dans le compteur programme l'adresse de début du programme de traitement de la demande - commence l'exécution de ce programme de traitement	Le C167 : - interdit les interruptions - arbitre entre les différentes demandes - choisit la source plus prioritaire pour le traitement associé. (chargement dans le compteur_programme de l'adresse de la fonction de traitement de la source) et dévut exécution du programme de traitement.
7- conversation	7- traitement	Traitement effectif
8- à la fin de la conversation : - s'il n'y a rien de grave, vous revenez continuer la lecture en récupérant la page et la ligne mémorisées - si la situation est grave, vous partez en courant.	8- à la fin du traitement : a) si le traitement considère que le programme suspendu peut reprendre alors le processeur - récupère l'adresse de retour, - charge celle-ci dans le compteur programme et le programme suspendu peut reprendre comme si rien ne s'est passé. b) si le traitement considère que le programme suspendu ne doit pas reprendre alors le programme suspendu ne s'exécutera plus.	Retour spécial au programme suspendu - autorisation des IT - récupération de l'adresse de retour sur la pile et chargement dans le compteur-programme - reprise éventuelle du programme suspendu

5- Relation avec une fonction classique

Une fonction classique est appelée dans un programme
Une fonction pour le traitement d'une interruption n'est pas appelée dans un programme mais son appel automatique est déclenché par une demande d'urgence.

<table>
<tr>
<td>void xyz(void)
{

}

void main(void)
{
  -
  -
  -
}</td>
<td>Programme en cours
 

la fonction traite_it n'apparaît pas dans le main
C'est la demande d'urgence qui déclenche automatiquement son exécution.</td>
</tr>
</table>

6- Application aux interruptions du microcontrôleur C167

Le microcontrôleur C167 dispose de 56 sources d'interruptions (IT = interruption).

6-1 Sources des interruptions et conditions de déclenchement

Les sources peuvent être organisées de plusieurs façons.
On peut avoir :

- des sources externes. Ce sont des broches d'entrée dédiées peuvent être sont réservées pour les demandes d'interruption; Ces sources externes sont réparties aussi entre sources rapides et sources lentes. Les sources rapides sont celles pour lesquelles la prise en compte est très rapide.
- des sources internes. Ces sont des signaux produits par des composants intégrés dans le microcontrôleur comme les fin de comptage/décomptage des Timers.

Nous n'en traiterons que quelques unes. C'est le principe qui nous importe.

Pour les sources externes

Une source externe rapide déclenchée sur P2.8

Une source externe rapide déclenchée sur P2.9

Une source externe rapide déclenchée sur P2.10

La condition de déclenchement de chacune de ces sources est programmable dans le registre EXICON

Registre EXICON

Source	P2.15		P2.14		P2.13		P2.12		P2.11		P2.10		P2.9		P2.8	
Nom bit	EXI7ES		EXI6ES		EXI5ES		EXI4ES		EXI3ES		EXI2ES		EXI1ES		EXI0ES	
											F1	F0	F1	F0	F1	F0

F1	F0	
0	0	mode standard des IT pour l'entrée correspondante
0	1	front montant
1	0	front descendant
1	1	front montant et front descendant

Pour les sources internes

Fin Timer T3 T3IR

Fin Timer T6 T6IR

La condition de déclenchement de chacune des sources est un front montant du signal (TxIR) de la source correspondante.

6-2 Autorisation des interruptions

Le C167 offre 2 niveaux d'autorisation des interruptions.
Le niveau global indiqué par la valeur du bit IEN
IEN = 0 aucune interruption ne peut être prise en compte
1 les sources d'IT autorisées individuellement peuvent être prises en compte si les conditions de déclenchement sont satisfaites.

Le niveau individuel
Chaque source dispose d'un bit de la forme XXIE qui permet d'interdire ou d'autoriser l'interruption.

P2.8	bit	CC8IE
P2.9	bit	CC9IE
P2.10	bit	CC10IE
Fin Timer T3	bit	T3IE
Fin Timer T6	bit	T6IE

6-4 Notion de priorité

Puisqu'il y a plusieurs sources et que le microcontrôleur ne peut traiter qu'une seule source, s'il y a plusieurs demandes d'IT en même temps, le microcontrôleur doit choisir à traiter la source la plus prioritaire. Pour cela, on peut programmer le niveau de priorité de chaque source.

Le niveau de priorité est un nombre binaire de 6bits organisés en deux groupes :
- un groupe ILVL de 4bits (poids fort) priorité intra-groupe
- un groupe IGVL de 2bits qui définit la priorité dans un groupe ILVL (poids faible) priorité inter-groupe

Chaque source d'IT dispose d'un registre de priorrité : XXIC ayant la forme générale

Nom du registre = XXxIC

	15							8	7	6	5	4	3	2	1	0
									CCxIR	CCxIE	ILVL				GLVL	
									Mémoire demande	Autorisation/interdiction demande	Priorité de groupes				Priorité dans le groupe	

P2.8	registre de priorité	CC8IC
P2.9	registre de priorité	CC9IC
P2.10	registre de priorité	CC10IC
Fin Timer T3	registre de priorité	T3IC
Fin Timer T6	registre de priorité	T6IC

Règles de priorité :

1- Si aucune source d'IT n'est en cours de traitement, alors la source la plus prioritaire est celle ayant le nombre binaire de 6bits (ILVL - IGVL) dont la valeur décimale est la plus élevée parmi les sources ayant fait la demande en même temps.

2- Si une source est en cours de traitement alors la demande la plus prioritaire est celle ayant le plus grand ILVL la source en cours comprise. Dans ce deuxième cas, les interruptions doivent être ré-autorisées dans le programme de la source en cours.
De même, dans ce deuxième cas, les sources ayant le même ILVL ne peuvent pas s'interrompre.

6-5 Mécanisme de prise en compte des interruptions

Chaque source d'IT possède un bit de mémorisation de la demande sous la forme xxIR.
Ce bit est à mis 1 lorsque les conditions de la demande sont satisfaites.
Lorsque le µC se branche sur la source d'IT la prioritaire, il met aussi automatiquement à 0 le bit xxIR de mémorisation de la demande de la source.

6-6 Adresse des fonctions de traitement des

Table de quelques IT utiles pour les TP

La table des vecteurs d'interruption permet d'associer à chaque source d'interruption mise en oeuvre l'adresse de la procédure d'interruption correspondante.

	sources	Adresse vecteur	Numéro vecteur		Bit dem/Autoris	Registre IT ILVL,GLVL	Registre condition
Rapides	P2.8 (EX0IN)	0060H	18h		CC8IR/CC8IE	CC8IC	EXICON
	P2.9 (EX1IN)	0064H	19h		CC9IR/CC8IE	CC9IC	EXICON
Timers							
Timers GPT1	T3	008ch	23h		T3IR/T3IE	T3IC	Fin cpt/dcpt
Timers GPT2	T6	0098h	26h		T6IR/T6IE	T6IC	Fin cpt/dcpt

7- Programmation

Un programme fonctionnant sous interruptions possède :

- un programme principal
- autant de fonctions de traitement des interruptions que de sources d'IT à mettre en œuvre

7-1 Structure d'une fonction de traitement d'interruption

Remarques préliminaires :

1- Une fonction de traitement d'interruption :

- ne retourne pas de valeur
- n'a pas de paramètres formels car, l'appel est automatique et peut survenir à n'importe quel moment.

D'où la forme **void traite_interruption(void)**

2- On doit indiquer au compilateur l'adresse où la fonction doit être implantée pour être appelée automatiquement. Pour cela, chaque source possède un numéro d'IT (cf table ci-dessus) et on utilise le mot réservé **interrupt** pour signaler ce numéro

D'où la forme

```
void traite_interruption(void) interrupt  numero_IT
{  //variables locales
        //Corps du traitement d'IT
}
```

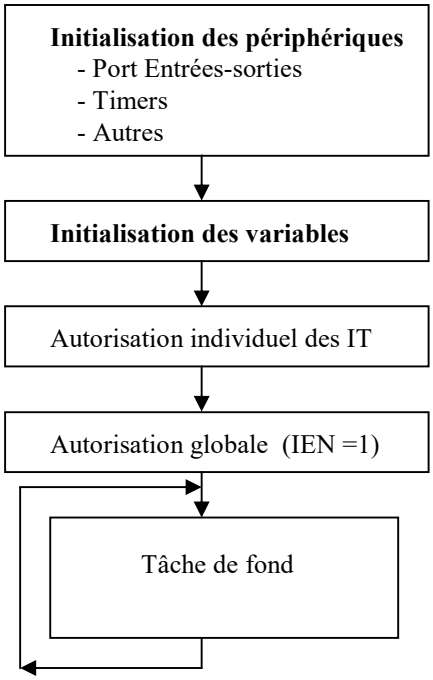
3- Le programme de traitement d'IT :

- doit être le plus court et rapide possible
- ne doit pas comporter des boucles non maîtrisées

4- Les échanges avec les autres programmes se font en utilisant des variables globales.

7-2 Structure du programme principal avec des interruptions

Nous allons donner la structure de ce type de programme principal sous forme d'organigramme



Exemple d'application

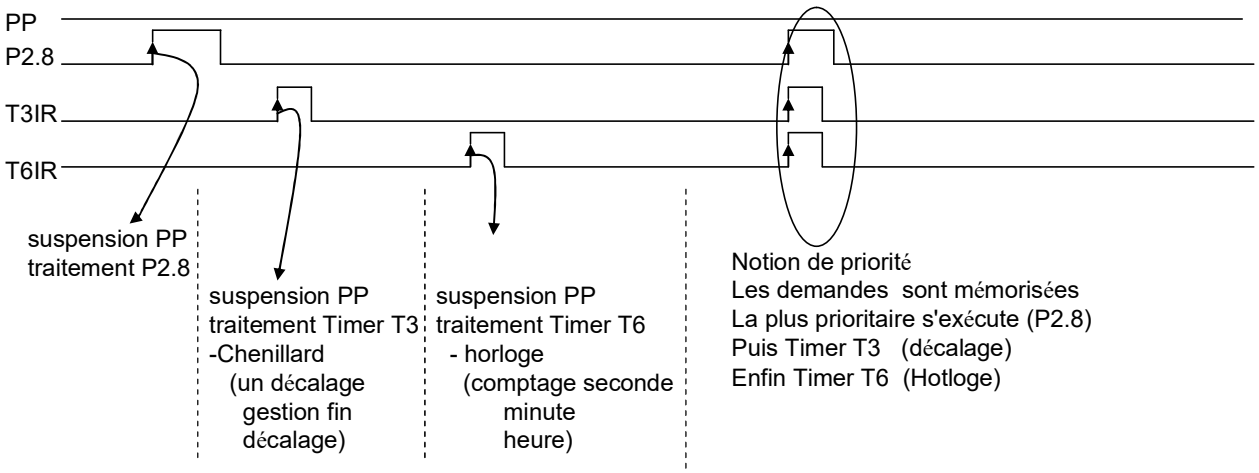
On souhaite réaliser un système permettant de faire fonctionner en même temps :

- la prise en compte par interruption rapide sur front montant d'un poussoir branché sur la broche P2.8.
  La fonction de traitement sera l'affichage du message "IT Externe rapide P2.8"
- génération d'une rampe de pente 1 sur le CNA3.
- un chenillard à vitesse variable par pas de 250ms. Le facteur de vitesse est lu sur les 8bits du port P2.
  Le chenillard est de 8bits sur le port P8
  La base de temps pour la vitesse du chenillard sera obtenue par interruption du timer T3
- une horloge avec affichage sur la console en utilisant une base de temps générée par le Timer T6

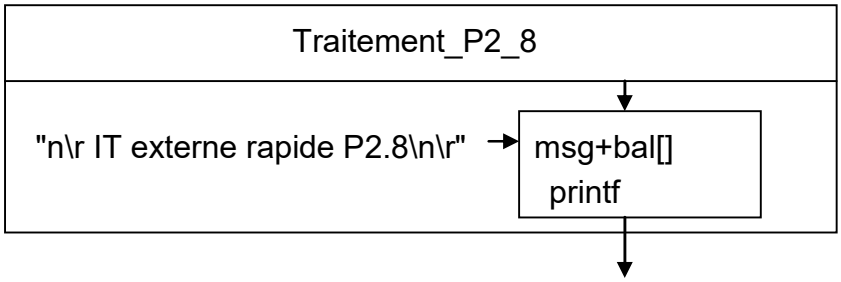
Dans le programme principal
On aura une boucle infinie dans laquelle

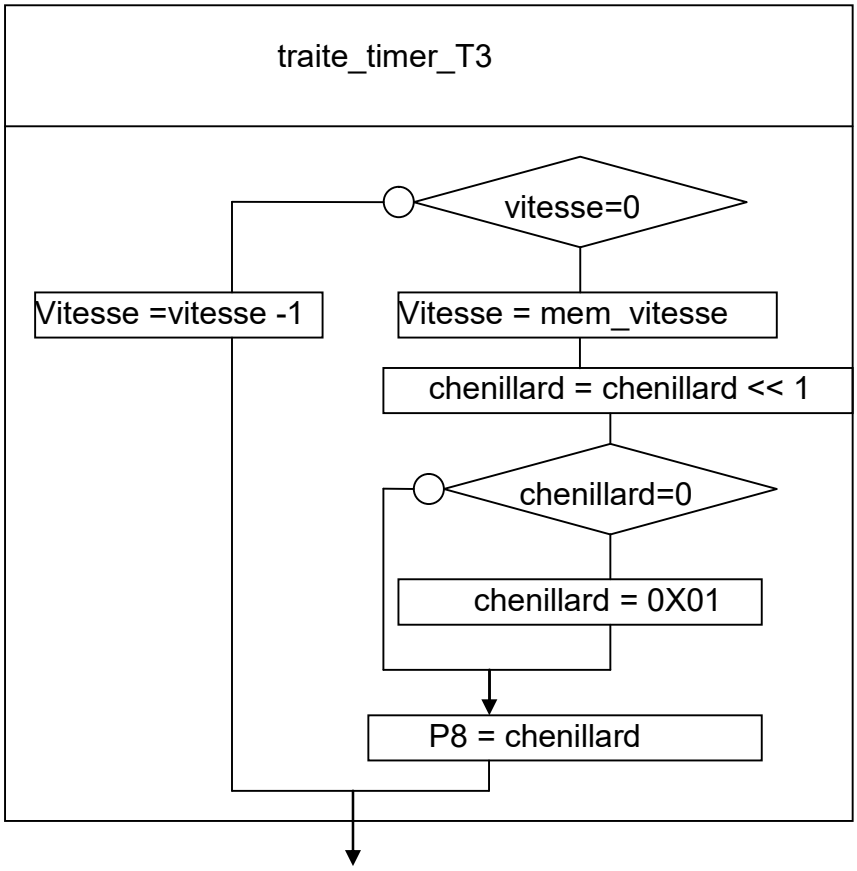
- ☞ on recopie - les 8 bits de poids faible le port P2 sur le port P7
  - le CAN0 sur le CNA1
- ☞ tous les caractères tapés au clavier sont re-affichés sur la console.

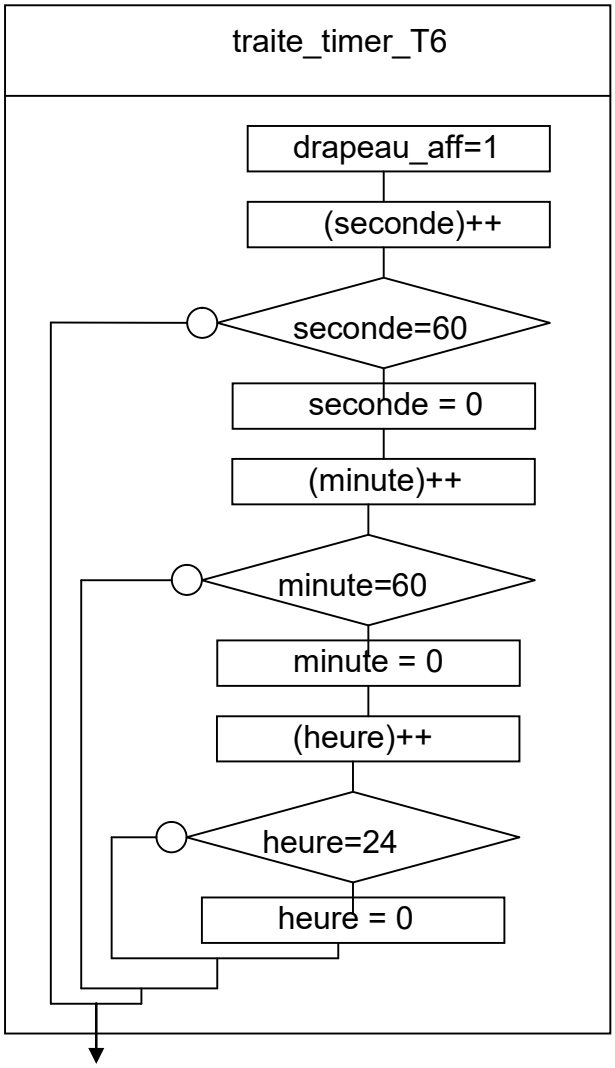
Séquence des IT



Les traitements des IT







Initialisation
Sens des Ports P2, P8

Timers T3, T6

Les variables

Interruptions
Priorité
CC8IC =
T3IC =
T6IC =

//Autorisation
// P2.8 EXICON

// Timer T3 T3IE

// TImer T6 T6IE

//Autorisation globale IEN

Programme principal

Initialisation

Lancement des Timers

Boucle infinie
{

}