TD5

Sous-programme de temporisation

a/ Définition :

Un sous-programme est une séquence d'instructions chargée d'effectuer un traitement particulier.

Les sous-programmes permettent ainsi décomposer un programme en plusieurs sous-parties.

b/ structure :

programme principal

…….

…….

CALL X

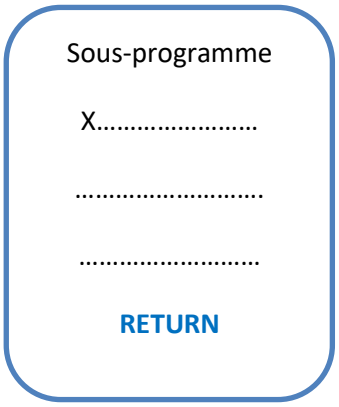
……

……

CALL X

……

……



CALL : instruction d'appel d'un sous-programme

RETURN : Instruction de retour du sous-programme

La temporisation : une opération qui consiste à **temporiser** , **retarder** un évenement ;

Il s'agit de réaliser ici une temporisation logicielle et de l'associer à un sous-programme ''***tempo***''

Il arrive souvent qu'on désire introduire des temporisations pendant l'exécution d'un programme ; Plusieurs méthodes pour réaliser des temporisations.

Module : **Microprocesseur & Microcontrôleur**

Chargée de cours : Dr K. Chaker

Chargée de TD : Dr N.Merabti

Le tableau ci-dessous récapitule les méthodes classiques

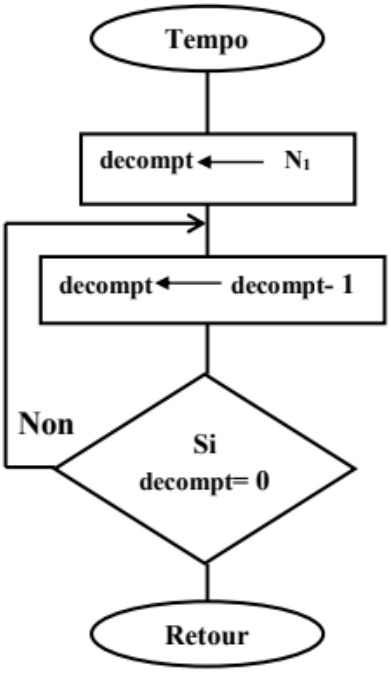
	nop	Boucle	Timer	Timer + interruption
Temporisation très courte	X			
Temporisation moyenne et longue sans le Timer		X		
Temporisation précise			X	
Temporisation précise, accompagnée de boucles longues				X

A. Temporisation avec boucle :

1 – Principe :

L'idée est d'initialiser une variable à une valeur donnée et ensuite la décrémenter en boucle jusqu'a ce qu'elle atteint 0. Connaissant le temps d'exécution de chaque instruction, on peut calculer le temps que mettra le processeur pour terminer la boucle de décrémentation.

2 - Temporisation avec une seule boucle :



Label	Instruction	Commentaire	Nombre de cycle
Tempo	MOVLW N_1	; W ← N_1	1
	MOVWF decompt	; decompt ← W	1
loop	DECFSZ decompt,f	; decompt ← decompt -1 avec test si 0	1(2)
	GOTO loop	; Boucle	2
	RETURN	; Retour	2

Calcule de la temporisation :

Tempo = [2 +1 +1 + (N1-1). (1 +2) + 2 +2] cycles = [8 + 3. (N_1-1)] cycles = [8 +3 N_1-3] cycles

Tempo = [5 +3 N_1] cycles .

Si la fréquence du quartz est égale à **4 Mhz : Fc** = 4 .10^6**/4** = **1 .10^6hz = 1 Mhz**→ Tc = 1/(1.10^6) = 10^{-6}s = 1µs .

La temporisation est maximale, pour N_{1max}= 255 = FF $_H$= 11111111_2

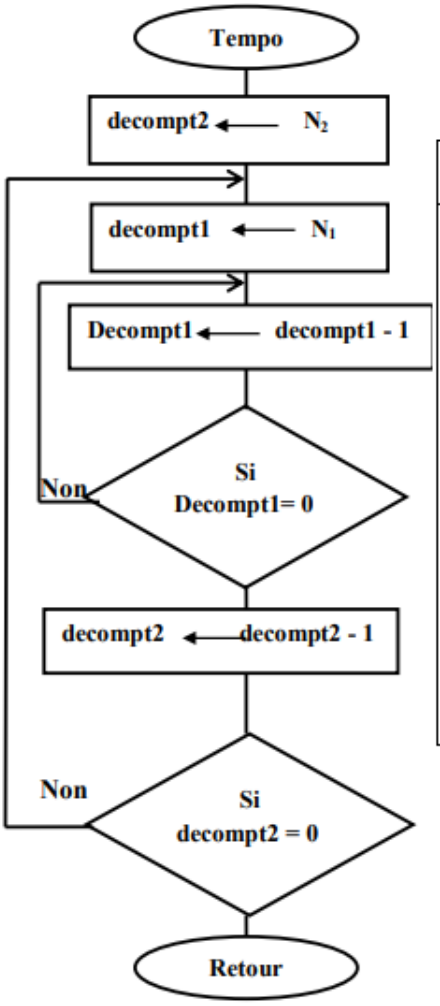
Tempo.max = [5 +3 N_{1max}] cycles = [5 +3 .255] .10^{-6}= 770µs

Remarques:

• Il faut rajouter 2 cycles pour l'instruction CALL.

• La précision peut être améliorée en y insérant l'instruction NOP.

3 - Temporisation avec deux boucles :



Label	Instruction	Commentaire	Nombre de cycle
Temp	MOVLW N_2	; W←N_2	1
	MOVWF decompt2	; decompt2←W	1
LOOP1	MOVLW N_1	; W←N_1	1
	MOVWF decompt1	; decompt1←W	1
LOOP2	DECFSZ decompt1,f	;decomp1←decomp -1 avec test si 0.	1(2)
	GOTO LOOP2	; Boucle1	2
	DECFSZ decompt2,f	; decomp2←decomp2 -1 avec test si 0.	1(2)
	GOTO LOOP1	; Boucle2	2
	RETURN	; Retour	2

Le clignotement : est obtenu par succession d'allumages et d'extinctions

L'allumage et l'extinction doivent être maintenus pendant une durée par une temporisation.

Exercice 1: Ecrire un programme qui fait clignoter une LED

Exercice 2: Ecrire un programme qui fait clignoter un ensemble de 8 LEDs montées sur le port b

Exercice 3: Ecrire un programme qui permet d'allumer successivement une led rouge, orange puis verte (l'allumage de la led suivante éteint la précédente, le programme tournera indéfiniment.