

Explication du programme

Programme d'interruption :

Si l'inter « SE / ET » est positionné sur « SE » Le programme d'interruption devient prioritaire. La commande d'interruption est appliquée sur la broche RB0. A l'état bas ce bit entraîne l'arrêt du programme principal et stop toutes les actions en cours.

RB5 passe à 1 pour actionner RE3 afin de stopper l'électrovanne de sécurité.

RB3 et RB4 passe à 1 pour commuter T2 et T3 afin d'annuler le déplacement du vérin

La sortie RB7 visualise sur la LED « DL9 » la mise en sécurité du système.

Seul le retour de l'inter « SE / ET » sur la position « ET » permet par la broche RA1 d'annuler la fonction d'interruption.

Cette action sur RA1 va permettre de rejoindre le programme principal.

Programme principal :

Le programme principal a pour but de gérer :

- 1) le comptage
- 2) la fin de cycle
- 3) l'inter « SE / ET » pour définir si le programme doit se dérouler avec le programme d'interruption ou pas.

On vérifie sur RA1 la position de l'inter « SE / ET »

Si RA1 = 1 aucune fonction de sécurité ne sera possible.

Si RA1 = 0 les fonctions de sécurité seront possibles par le programme d'interruption.

L'info comptage est appliquée sur la broche RA0.

A chaque état bas une impulsion de 200ms est envoyée sur la broche RB6 pour incrémenté le compteur.

Si un nombre de cycle a été défini sur le compteur la broche RA2 reçoit cette info et bascule ce bit à 0 une routine de temporisation va permettre de confirmer que cette info est vrai. Cette condition remplie le programme valide les infos suivantes :

RB5 passe à 1 pour actionner RE3 afin de stopper l'électrovanne de sécurité.

RB3 et RB4 passe à 1 pour commuter T2 et T3 afin d'annuler le déplacement du vérin.

La sortie RA3 visualise sur la LED « DL10 » la fin du programme.

L'arrêt total du système est nécessaire pour redémarrer un nouveau cycle.

J.P. Fruchard
Janvier 2007

FICHER SOURCE

```
; TITRE: Programme d'asservissement des pilotages vérins
; DATE: 27/01/2007
; AUTEUR: J.P. FRUCHARD
; PIC UTILISE: PIC16F84A
; Le programme d'interruption est défini sur la broche RB0
; L'horloge est de 4MHz
; RA0 = entrée comptage
; RA1 = Inter "SE/ET"
; RA2 = entrée "Fin de cycle"
; RA3 = Led "Fin de cycle"
; RA4 = Non utilisé
; RBO = Interruption
; RB1 = Led "ET"
; RB2 = Led "SE"
; RB3 = Asservissement vérin
; RB4 = Signal E1
; RB5 = Relais électro sécurité
; RB6 = Compteur d'impulsion
; RB7 = Led "défaut"
;-----Directives d'assemblage-----
List p=16f84A
#include p16f84A.inc
__config H'3FF9'
;-----Déclarations des variables utilisées-----
RETARD1      EQU  0X0C  ; Le registre temporaire retard1 se trouve à l'adresse 0C
RETARD2      EQU  0X0F  ; Le registre temporaire retard1 se trouve à l'adresse 0F
RETARD3      EQU  0X10  ; Le registre temporaire retard1 se trouve à l'adresse 10
;-----PROGRAMME PRINCIPAL-----

ORG  0          ; Adresse de reset
    GOTO  INIT

;-----Programme d'interruption-----

ORG  04         ; Adresse d'interruption
    BTFSS PORTB,0 ; Sécurités enclenchées? si oui on saute la prochaine instruction
    GOTO  DEBUT  ; Si non on va à DEBUT
    CALL  TEMPO1 ; On appelle la TEMPO1
    BTFSS PORTB,0 ; Confirmé? si oui on saute la prochaine instruction
    GOTO  DEBUT  ; Si non on va à DEBUT

STOP

    MOVLW B'00111100' ; On charge la valeur dans le registre W
    MOVWF PORTB       ; On transfère W sur le port B (RE3 activé vérin stoppé)

LED

    CALL  TEMPO1      ; On appelle la TEMPO1
    MOVLW B'10111100' ; On charge la valeur dans le registre W
    MOVWF PORTB      ; On met W dans le port B (Led DL9 allumée)
    CALL  TEMPO1      ; On appelle la TEMPO1
    MOVLW B'00111100' ; On charge la valeur dans le registre W
    MOVWF PORTB      ; On met W dans le port B (Led DL9 éteinte)
    BTFSS PORTA,1    ; Inter sécurité basculé? si oui on saute la prochaine instruction
    GOTO  LED        ; Si non on va à LED
    CALL  TEMPO1      ; On appelle TEMPO
    BTFSS PORTA,1    ; Inter sécurité basculé? si oui on saute la prochaine instruction
    GOTO  LED        ; Si non on va à LED
    MOVLW B'00000010' ; On charge la valeur dans le registre W
    MOVWF PORTB      ; On transfère W sur le port B (RE3 désactivé vérin libéré)
    BCF   INTCON,1    ; On met à 0 le 1er bit du registre INTCON
    BSF   INTCON,4    ; On met à 1 le 4ème bit du registre INTCON
    BSF   INTCON,7    ; On met à 1 le 7ème bit du registre INTCON pour autoriser toutes les interruptions
    GOTO  INIT        ; On saute à INIT
```

```

;-----Initialisation des ports A et B-----
INIT
BSF   STATUS,5   ; On met à 1 le 5e bit du registre status pour accéder à la 2er page mémoire
MOVLW B'00000001' ; On met 1 dans le registre W
MOVWF TRISB      ; On met W dans le port B qui est programmé en sortie de RB1 à RB7
MOVLW B'00000111' ; On met 7 dans le registre W
MOVWF TRISA      ; On met W dans le port A programmé en entrée de RA0 à RA2
BCF   STATUS,5   ; On remet à 0 le 5e bit du registre status pour accéder à la 1er page mémoire
BCF   INTCON,4   ; On met à 0 le 4ème bit du registre INTCON pour interdire l'interruption sur RB0
BCF   INTCON,7   ; On met à 0 le 7ème bit du registre INTCON pour interdire toutes les interruptions

;-----RAZ du port B-----
CLRF  PORTA      ; RAZ du portA
CLRF  PORTB      ; RAZ du portB

;-----PROGRAMME PRINCIPAL-----
DEBUT
BTFSS PORTA,1    ; Inter sécu = 1? si oui on saute la prochaine instruction
GOTO  SECU_ON    ; Sinon on va à SECU_ON
GOTO  SECU_OFF   ; On saute à SECU_OFF

SECU_ON
BCF   INTCON,1   ; On met à 0 le 1er bit du registre INTCON pour réinitialiser l'interruption
BSF   INTCON,4   ; On met à 1 le 4ème bit du registre INTCON pour autoriser l'interruption sur RB0
BSF   INTCON,7   ; On met à 1 le 7ème bit du registre INTCON pour autoriser toutes les interruptions
MOVLW B'00000100' ; On charge la valeur dans le registre W
MOVWF PORTB     ; On transfère W sur le port B (sécurité activée DL7 allumée)
GOTO  TEST      ; On saute à TEST

SECU_OFF
BCF   INTCON,4   ; On met à 0 le 4ème bit du registre INTCON pour interdire l'interruption sur RB0
BCF   INTCON,7   ; On met à 0 le 7ème bit du registre INTCON pour interdire toutes les interruptions
MOVLW B'00000010' ; On charge la valeur dans le registre W
MOVWF PORTB     ; On transfère W sur le port B (sécurité désactivée DL8 allumée)
GOTO  TEST      ; On saute à TEST

TEST
BTFSS PORTA,0    ; Incrément = 1? si oui on saute la prochaine instruction
GOTO  COMPTE     ; Sinon on va à COMPTE
BTFSS PORTA,2    ; Fin de cycle = 1? si oui on saute la prochaine instruction
GOTO  ARRET      ; Sinon on va à ARRET
GOTO  DEBUT     ; Retour à DEBUT

COMPTE
BSF   PORTB,6    ; On met RB6 à 1
CALL  TEMPO     ; On appelle la temporisation
BCF   PORTB,6    ; On met RB6 à 0

BOOT
BTFSS PORTA,0    ; Incrément = 1? si oui on saute la prochaine instruction
GOTO  BOOT      ; On boucle sur BOOT
GOTO  DEBUT     ; Retour à DEBUT

ARRET
CALL  TEMPO     ; On appelle la temporisation
BTFSS PORTA,2    ; Confirme = 0? si oui on saute la prochaine instruction
GOTO  TEST      ; Sinon retour à TEST
MOVLW B'00111100' ; On charge la valeur dans le registre W
MOVWF PORTB     ; On transfère W sur le port B (RE3 activé vérin stoppé)

FIN
BSF   PORTA,3    ; On met RA3 à 1 (Led DL10 allumée)
GOTO  FIN       ; On boucle sur FIN

```

-----Programme de temporisation -----

TEMPO

```
MOVLW 0X50      ; On charge la valeur dans le registre W
MOVWF RETARD1   ; On met W dans le registre retard1
MOVWF RETARD2   ; On met W dans le registre retard2
MOVLW 0X12      ; On charge la valeur dans le registre W
MOVWF RETARD3   ; On met W dans le registre retard3
```

ATTENTE2

```
DECFSZ RETARD1,F ; On décrémente retard1 et on saute la prochaine instruction si le registre retard1 = 0
GOTO ATTENTE2    ; Sinon retour à attente2
MOVLW 0X50      ; On charge la valeur dans le registre W
MOVWF RETARD1   ; On met W dans retard1
DECFSZ RETARD2,F ; On décrémente retard2 et on saute la prochaine instruction si le registre retard2 = 0
GOTO ATTENTE2    ; Sinon retour à attente2
MOVLW 0X12      ; On charge la valeur dans le registre W
MOVWF RETARD2   ; On met W dans retard2
DECFSZ RETARD3,F ; On décrémente retard3 et on saute la prochaine instruction si le registre retard3 = 0
GOTO ATTENTE2    ; Sinon retour à attente2
RETURN          ; Retour au programme principal
```

-----Programme de temporisation 1-----

TEMPO1

```
MOVLW 0X80      ; On charge la valeur dans le registre W
MOVWF RETARD1   ; On met W dans le registre retard1
MOVWF RETARD2   ; On met W dans le registre retard2
MOVLW 0X80      ; On charge la valeur dans le registre W
MOVWF RETARD3   ; On met W dans le registre retard3
```

ATTENTE3

```
DECFSZ RETARD1,F ; On décrémente retard1 et on saute la prochaine instruction si le registre retard1 = 0
GOTO ATTENTE3    ; Sinon retour à attente3
MOVLW 0X30      ; On charge la valeur dans le registre W
MOVWF RETARD1   ; On met W dans retard1
DECFSZ RETARD2,F ; On décrémente retard2 et on saute la prochaine instruction si le registre retard2 = 0
GOTO ATTENTE3    ; Sinon retour à attente3
MOVLW 0X12      ; On charge la valeur dans le registre W
MOVWF RETARD2   ; On met W dans retard2
DECFSZ RETARD3,F ; On décrémente retard3 et on saute la prochaine instruction si le registre retard3 = 0
GOTO ATTENTE3    ; Sinon retour à attente3
RETURN          ; Retour au programme principal
```

END