list p=16F84A

#include P16F84A.inc

\_\_CONFIG \_CP\_OFF & \_WDT\_OFF & \_PWRTE\_ON & \_XT\_OSC

;\*\*\*\* --Definicion de variables-- \*\*\*\*

reset org 0x00 ;se define reset en la localidad 0

ORG 4

GOTO INTERRUPCIONWOO

CONTA\_1 equ 12 ;se declaran las variables

CONTA\_2 equ 33

CONTA\_3 equ 21

LED\_ACTIVO equ 34

PUERTOS equ 35

;CBLOCK ;Se definen las variables donde se guardarán los registros

Guarda\_W equ 36

Guarda\_STATUS equ 37

Guarda\_CONTA\_1 equ 38

Guarda\_CONTA\_2 equ 39

Guarda\_CONTA\_3 equ 40

Guardar\_PORTA equ 41

Guardar\_PORTB equ 42

;R\_ContA

;R\_ContB

;ENDC

;configuracion de puertos

;Vector de interrupción

goto inicio

org 0x05 ;se inicia el programa en la localidad 5

inicio bsf STATUS, RP0

movlw b'00000' ; se declaran salidas en el puerto A

movwf TRISA

movlw b'00000001' ; se declaran salidas en el puerto B y la entrada en el bit 0 del puerto B

movwf TRISB;

BCF OPTION\_REG,6 ; INTERRUPCION POR FLANCO DESCENDENTE (CUANDO SE APRIETA EL PB)

bcf STATUS,RP0 ;

MOVLW B'10010000' ; HABILITAMOS INTERRUPCION, QUE SEAN LAS EXTERNAS INT Y GIE

MOVWF INTCON ; CONFIGURACION REG INTCON

clrf PORTA ; se limpia puerto A

clrf PORTB ; se limpia puerto B

PUHS

;BTFSS PORTB,0 ;si el bit 0 del puerto B esta en 1, salta

GOTO SECUENCIA ;ir a SECUENCIA

;GOTO FIN ; sino terminar programa

INTERRUPCIONWOO

; call retardo\_20ms

; BTFSS PORTB,0

; return

movwf Guarda\_W ; Guarda W y STATUS.

swapf STATUS,W ; Ya que "movf STATUS,W", modifica el bit Z.

movwf Guarda\_STATUS

;bcf STATUS,RP0 ; Para asegurarse que trabaja con el banco 0.

movf CONTA\_3,W ; Guarda los registros del retando 0.5 utilizados

movwf Guarda\_CONTA\_3

movf CONTA\_2,W

movwf Guarda\_CONTA\_2

movf CONTA\_1,W

movwf Guarda\_CONTA\_1

movwf Guardar\_PORTA

movf PORTA

movwf Guardar\_PORTB

movf PORTB

MOVF LED\_ACTIVO,0

MOVWF PORTB

CALL Retardo\_UnSeg

CLRF PORTB

CALL Retardo\_UnSeg

MOVF LED\_ACTIVO,0

MOVWF PORTB

CALL Retardo\_UnSeg

CLRF PORTB

CALL Retardo\_UnSeg

MOVF LED\_ACTIVO,0

MOVWF PORTB

CALL Retardo\_UnSeg

CLRF PORTB

CALL Retardo\_UnSeg

FinInterrupcion

swapf Guarda\_STATUS,W ;Restauramos valor de STATUS.

movwf STATUS

swapf Guarda\_W,F ;Restaura W como estaba antes de producirse

swapf Guarda\_W,W ;interrupción.

movf Guarda\_CONTA\_1,W ;Restaura los registros utilizados en esta

movwf CONTA\_1 ;subrutina y también en la principal.

movf Guarda\_CONTA\_2,W

movwf CONTA\_2

movf Guarda\_CONTA\_3,W

movwf CONTA\_3

bcf INTCON,INTF ;Limpia flag de reconocimiento de la interrupción.

retfie ;Retorna y vuelve a habilitar las interrupciones.

SECUENCIA

MOVLW b'00000010'

MOVWF LED\_ACTIVO

MOVWF PORTB

CALL Retardo\_MedioSeg

MOVLW b'00000100'

MOVWF LED\_ACTIVO

MOVWF PORTB

CALL Retardo\_MedioSeg

MOVLW b'00001000'

MOVWF LED\_ACTIVO

MOVWF PORTB

CALL Retardo\_MedioSeg

MOVLW b'00010000'

MOVWF LED\_ACTIVO

MOVWF PORTB

CALL Retardo\_MedioSeg

MOVLW b'00100000'

MOVWF LED\_ACTIVO

MOVWF PORTB

CALL Retardo\_MedioSeg

MOVLW b'01000000'

MOVWF LED\_ACTIVO

MOVWF PORTB

CALL Retardo\_MedioSeg

MOVLW b'10000000'

MOVWF LED\_ACTIVO

MOVWF PORTB

CALL Retardo\_MedioSeg

VUELTA

MOVLW b'00000000'

MOVWF PORTB

CALL Retardo\_MedioSeg

MOVLW b'10000000'

MOVWF LED\_ACTIVO

MOVWF PORTB

CALL Retardo\_MedioSeg

MOVLW b'01000000'

MOVWF LED\_ACTIVO

MOVWF PORTB

CALL Retardo\_MedioSeg

MOVLW b'00100000'

MOVWF LED\_ACTIVO

MOVWF PORTB

CALL Retardo\_MedioSeg

MOVLW b'00010000'

MOVWF LED\_ACTIVO

MOVWF PORTB

CALL Retardo\_MedioSeg

MOVLW b'00001000'

MOVWF LED\_ACTIVO

MOVWF PORTB

CALL Retardo\_MedioSeg

MOVLW b'00000100'

MOVWF LED\_ACTIVO

MOVWF PORTB

CALL Retardo\_MedioSeg

MOVLW b'00000010'

MOVWF LED\_ACTIVO

MOVWF PORTB

CALL Retardo\_MedioSeg

GOTO PUHS ;regresa a la subrutina PUHS

Retardo\_MedioSeg ;etiqueta Retardo\_MedioSeg (Retardo\_MedioSegardo de 0.5 segundos)

movlw d'6';movlw d'6' ;se da el valor de 6 al registro w

movwf CONTA\_3 ;se mueve lo que hay en el registro w a CONTA\_3

d movlw d'101' ;se da el valor de 101 al registro w

movwf CONTA\_2 ;se mueve el registro w a CONTA\_2

c movlw d'250' ;se da el valor de 250 a w

movwf CONTA\_1 ;se mueve el registro w a CONTA\_1

a nop ;no opera

decfsz CONTA\_1 ;CONTA\_1 <- CONTA\_1 - 1 y salta si llego a 0 (249 veces)

goto $-.2 ;no es 0, entonces salta a nop

decfsz CONTA\_2 ;si es 0 entonces CONTA\_2 <- CONTA\_2 -1 y salta si llega a 0 (repite 100 veces)

goto $-.6 ;no es 0, entonces a caragar CONTA\_1

decfsz CONTA\_3 ; si es 0, entonces CONTA\_3 <- CONTA\_ y salta si llega a 0

goto $-.10 ;no es 0, entonces regresa a cargar CONTA\_3(repite 5 veces)

return ;si es 0, entonces fin del retrdo y regresa

Retardo\_UnSeg

movlw d'11' ;etiqueta RET3 (retardo de 1 segundo)

movwf CONTA\_3 ;se da el valor de 6 al registro w

movlw d'101' ;se mueve lo que hay en el registro w a CONTA\_3

movwf CONTA\_2 ;se da el valor de 101 al registro w

movlw d'250' ;se da el valor de 250 a w

movwf CONTA\_1 ;se mueve el registro w a CONTA\_1

nop ;no opera

decfsz CONTA\_1 ;CONTA\_1 <- CONTA\_1 - 1 y salta si llego a 0 (249 veces)

goto $-.2 ;no es 0, entonces salta a nop

decfsz CONTA\_2 ;si es 0 entonces CONTA\_2 <- CONTA\_2 -1 y salta si llega a 0 (repite 100 veces)

goto $-.6 ;no es 0, entonces a caragar CONTA\_1

decfsz CONTA\_3 ; si es 0, entonces CONTA\_3 <- CONTA\_ y salta si llega a 0

goto $-.10 ;no es 0, entonces regresa a cargar CONTA\_3(repite 5 veces)

return ;si es 0, entonces fin del retrdo y regresa

;

retardo\_20ms

; movlw d'11' ;etiqueta RET3 (retardo de 1 segundo)

; movwf CONTA\_3 ;se da el valor de 6 al registro w

; movlw d'101' ;se mueve lo que hay en el registro w a CONTA\_3

; movwf CONTA\_2 ;se da el valor de 101 al registro w

movlw d'25' ;se da el valor de 250 a w

movwf CONTA\_1 ;se mueve el registro w a CONTA\_1

nop ;no opera

decfsz CONTA\_1 ;CONTA\_1 <- CONTA\_1 - 1 y salta si llego a 0 (249 veces)

goto $-.2 ;no es 0, entonces salta a nop

; decfsz CONTA\_2 ;si es 0 entonces CONTA\_2 <- CONTA\_2 -1 y salta si llega a 0 (repite 100 veces)

; goto $-.6 ;no es 0, entonces a caragar CONTA\_1

; decfsz CONTA\_3 ; si es 0, entonces CONTA\_3 <- CONTA\_ y salta si llega a 0

; goto $-.10 ;no es 0, entonces regresa a cargar CONTA\_3(repite 5 veces)

return

FIN BTFSC PORTB,0 ;etiqueta fin, si el bit 0 del puerto B esta en 0, salta.

;GOTO PUHS

END ;terminar programa

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente