;PIC16F887 Configuration Bit Settings

; Assembly source line config statements

; CONFIG1

CONFIG FOSC = INTRC\_CLKOUT ; Oscillator Selection bits (INTOSC oscillator: CLKOUT function on RA6/OSC2/CLKOUT pin, I/O function on RA7/OSC1/CLKIN)

CONFIG WDTE = OFF ; Watchdog Timer Enable bit (WDT enabled)

CONFIG PWRTE = OFF ; Power-up Timer Enable bit (PWRT disabled)

CONFIG MCLRE = ON ; RE3/MCLR pin function select bit (RE3/MCLR pin function is MCLR)

CONFIG CP = OFF ; Code Protection bit (Program memory code protection is disabled)

CONFIG CPD = OFF ; Data Code Protection bit (Data memory code protection is disabled)

CONFIG BOREN = OFF ; Brown Out Reset Selection bits (BOR disabled)

CONFIG IESO = OFF ; Internal External Switchover bit (Internal/External Switchover mode is disabled)

CONFIG FCMEN = OFF ; Fail-Safe Clock Monitor Enabled bit (Fail-Safe Clock Monitor is disabled)

CONFIG LVP = OFF ; Low Voltage Programming Enable bit (RB3 pin has digital I/O, HV on MCLR must be used for programming)

; CONFIG2

CONFIG BOR4V = BOR40V ; Brown-out Reset Selection bit (Brown-out Reset set to 4.0V)

CONFIG WRT = OFF ; Flash Program Memory Self Write Enable bits (Write protection off)

// config statements should precede project file includes.

#include <xc.inc>

;directiva de inclusión para un archivo de encabezado específico

;del compilador que proporciona definiciones y configuraciones específicas

;del microcontrolador PIC16F887.

; PIC16F877A Configuration Bit Settings

;

; Section used for main code

PSECT MainCode,global,class=CODE,delta=2

; Etiqueta MainCode, es donde se encuentra el código principal.

; Initialize the PIC hardware

;

MAIN: ;Marca el punto de inicio del programa principal.

;serie de instrucciones BANKSEL que se utilizan para seleccionar

;los bancos de registro adecuados antes de realizar operaciones

;en puertos específicos.

;Por ejemplo, BANKSEL TRISB selecciona el banco de registro correcto

;para configurar el puerto B como salida

;Las instrucciones BCF y BSF se utilizan para borrar y establecer bits

;en los puertos seleccionados.

;Por ejemplo, BCF TRISB, 0 configura el primer bit del puerto B como salida.

BANKSEL TRISA

BCF TRISA,0 ;Set RA0 to output

BCF TRISA,1 ;Set RA0 to output

BCF TRISA,2 ;Set RA0 to output

BCF TRISA,3 ;Set RA0 to output

BCF TRISA,4 ;Set RA0 to output

BCF TRISA,5 ;Set RA0 to output

BCF TRISA,6 ;Set RA0 to output

BANKSEL TRISD

BSF TRISD,0 ;Set RA0 to input

BSF TRISD,1 ;Set RA0 to input

BSF TRISD,2 ;Set RA0 to input

BANKSEL PORTD

CLRF PORTB

BANKSEL TRISC;Se definen LEDS como salida

BCF TRISC, 0

BCF TRISC, 1

BCF TRISC, 2

BCF TRISC, 3

BANKSEL PORTC

CLRF PORTC

;Después de configurar los puertos, el programa entra en un bucle principal

;etiquetado como MainLoop. Este bucle realiza una secuencia de encendido y apagado

;de un pin específico del puerto B, seguido de una llamada a la subrutina

;DELAY para crear una pausa.

NUMERO EQU 0x12 ;Crear variable NUMERO y asignarla al registro de uso general 0x12

CLRF NUMERO ;Limpio la variable NUMERO

MOVLW 0 ;Se coloca la literal 4 en el registro de trabajo

MOVWF NUMERO ;Mover el contenido del registro de trabajo W, a variable numero

CLRW ;Limpiar registro de trabajo

;BCF PORTD, 1 ; Colocar en 1 lógico el puerto D1

MainLoop:

;Inicio de sección botones y condiciones

BTFSC PORTD, 0 ;Si se presiona el botón en RD0...

GOTO INCREMENTO

BTFSC PORTD, 1; Si se presiona el botón en RD1...

GOTO DECREMENTO

/\*

BTFSC PORTD, 2 ; Si se presiona el botón en RD2...

GOTO BORRAR ; Ir a subrutina BORRAR

;Se limpia el estado de los botones

LRF PORTD

\*/

GOTO CALCULARLEDS

GOTO MainLoop ; Una vez que se completa el retraso, el programa vuelve al bucle principal y repite el proceso.

INCREMENTO:

; Comparar si NUMERO es igual a 0

MOVLW 4 ; Cargar el valor 0 en el registro W

SUBWF NUMERO, W ; Restar NUMERO de W y almacenar el resultado en W

BTFSS STATUS, 2 ; Saltar si el resultado no es igual a cero (Z = 0)

INCF NUMERO ;Si los valores no son iguales, entonces no es 4, y debe incrementar

MOVF NUMERO, 0 ;Mover el contenido de la variable NUMERO a W, para monitoreo

BCF PORTC, 0

CALL DELAY

CALL DELAY

; Llamar a la subrutina si NUMERO es igual a 0

GOTO MainLoop

DECREMENTO:

; Comparar si NUMERO es igual a 0

MOVLW 0 ; Cargar el valor 0 en el registro W

SUBWF NUMERO, W ; Restar NUMERO de W y almacenar el resultado en W

BTFSS STATUS, 2 ; Saltar si el resultado no es igual a cero (Z = 0)

DECF NUMERO

MOVF NUMERO, 0

CALL DELAY

CALL DELAY

GOTO MainLoop

; Llamar a la subrutina si NUMERO es igual a 0

CALCULARLEDS:

; Comparar si NUMERO es igual a 0

MOVLW 0 ; Cargar el valor 0 en el registro W

SUBWF NUMERO, W ; Restar NUMERO de W y almacenar el resultado en W

BTFSC STATUS, 2 ; Saltar si el resultado no es igual a cero (Z = 0)

GOTO ENCENDERLED0

MOVF PORTC, 0

; BCF PORTC, 0

; Comparar si NUMERO es igual a 0

MOVLW 1 ; Cargar el valor 0 en el registro W

SUBWF NUMERO, W ; Restar NUMERO de W y almacenar el resultado en W

BTFSC STATUS, 2 ; Saltar si el resultado no es igual a cero (Z = 0)

GOTO ENCENDERLED1

;MOVF PORTC, 0

;BCF PORTC, 0

; Comparar si NUMERO es igual a 0

MOVLW 2 ; Cargar el valor 0 en el registro W

SUBWF NUMERO, W ; Restar NUMERO de W y almacenar el resultado en W

BTFSC STATUS, 2 ; Saltar si el resultado no es igual a cero (Z = 0)

GOTO ENCENDERLED2

;MOVF PORTC, 0

;BCF PORTC, 0

; Comparar si NUMERO es igual a 0

MOVLW 3 ; Cargar el valor 0 en el registro W

SUBWF NUMERO, W ; Restar NUMERO de W y almacenar el resultado en W

BTFSC STATUS, 2 ; Saltar si el resultado no es igual a cero (Z = 0)

GOTO ENCENDERLED3

;MOVF PORTC, 0

;BCF PORTC, 0

; Comparar si NUMERO es igual a 0

MOVLW 4 ; Cargar el valor 0 en el registro W

SUBWF NUMERO, W ; Restar NUMERO de W y almacenar el resultado en W

BTFSC STATUS, 2 ; Saltar si el resultado no es igual a cero (Z = 0)

GOTO ENCENDERLED4

;MOVF PORTC, 0

;BCF PORTC, 0

GOTO MainLoop

ENCENDERLED0:

BCF PORTC,0

BCF PORTC,1

BCF PORTC,2

BCF PORTC,3

GOTO MainLoop

ENCENDERLED1:

BSF PORTC,0

BCF PORTC,1

BCF PORTC,2

BCF PORTC,3

GOTO MainLoop

ENCENDERLED2:

BCF PORTC,0

BSF PORTC,1

BCF PORTC,2

BCF PORTC,3

GOTO MainLoop

ENCENDERLED3:

BCF PORTC,0

BCF PORTC,1

BSF PORTC,2

BCF PORTC,3

GOTO MainLoop

ENCENDERLED4:

BCF PORTC,0

BCF PORTC,1

BCF PORTC,2

BSF PORTC,3

GOTO MainLoop

;BSF PORTC,4

DELAY: ;Start DELAY subroutine here

movlw 400 ;Load initial value for the delay

movwf 0x10 ;Copy the value from working reg to the file register 0x10

movwf 0x11 ;Copy the value from working reg to the file register 0x11

DELAY\_LOOP: ;Start delay loop

decfsz 0x10, F ;Decrement the f register 0x10 and check if not zero

goto DELAY\_LOOP ;If not then go to the DELAY\_LOOP labe

decfsz 0x11, F ;Else decrement the f register 0x11, check if it is not 0

goto DELAY\_LOOP ;If not then go to the DELAY\_LOOP label

retlw 0 ;Else return from the subroutine

END MAIN