**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**

****

**FIEC**

**LABORATORIO DE MICROCONTROLADORES**

**PROYECTO #1**

***CONTADOR DECIMAL DE PULSOS***

**ESTUDIANTE:**

**Vicente Alexander Macas Espinosa**

**PROFESOR:**

**Dr. Carlos Valdivieso**

**GRUPO DE LABORATORIO:**

**03**

**PARALELO:**

**07**

**FECHA DE ENTREGA:**

**14/07/2011**

**I Término 2011-2012**

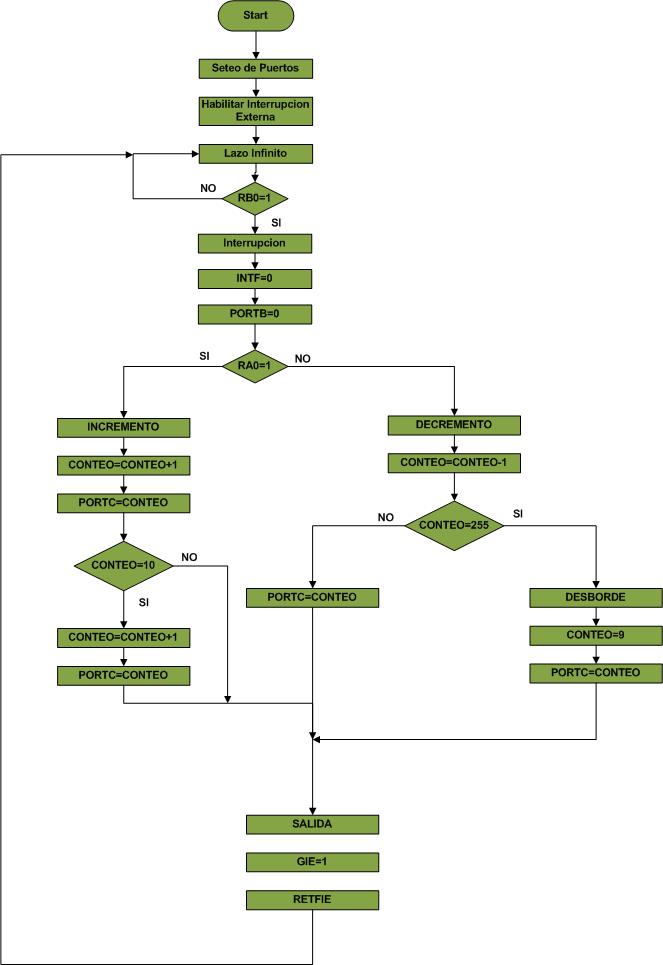
**1.) *Especificaciones técnicas del proyecto***

Este programa cuenta mediante una variable los pulsos ingresados mediante una botonera en RB0. También habrá un SWITCH en el puerto A, del cual se escogerá a RA0 que indique modo: de incremento o de decremento. Si RA0=1 entonces incrementara por cada pulso y si RA0=0 entonces disminuirá su valor con cada pulso El resultado se lo visualiza en un DISPLAYS de 7 segmentos colocado en el puerto C.

**2.) *Diagrama de bloques***



**3.) Diagrama de flujo**

****

***4.) Diagrama de Flujo funcional de las Subrutinas (No se realiza)***

***5.) Descripción del algoritmo o estrategia utilizada.***

Primeramente como en todo programa inicializamos las variables, en este caso nuestra única variable CONTEO.

Para empezar nuestro programa ubicamos en nuestra dirección de origen 0X00 el MAIN (principal) y nuestra INTERRUPCION 0X04 la interrupción que nos permitirá contar.

Ya en el MAIN realizamos el seteo de puertos, por lo que digitalizamos los puertos que contengan las entradas o salidas ANS, luego a todo PORTA lo ponemos como entrada al igual que el PORTB y al PORTC lo seteamos como salida.

El siguiente paso es habilitar la interrupción por RB0 mediante GIE e INTE.

Realizamos la inicialización en donde enceramos al PORTA, PORTB, PORTC, PORTE por precaución y CONTEO. Luego de esto nuestro display muestra cero.

En ese momento el programa entra en un lazo infinito, de donde solo puede salir si presionamos la botonera que hace de cambiar de valor a RB0, una vez presionara sale del lazo y nos dirigimos a la parte de interrupción.

En dicha interrupción primero borramos la bandera INTF, y procede a preguntar por el bit RA0, si este es uno se dirige a la sección de INCREMENTO caso contrario se dirige a la sección de DECREMENTO.

En la sección de incremento primero incrementamos la variable conteo y mediante tabla de código de 7 segmentos mostramos su valor en el display, cabe resaltar aquí si nuestra variable conteo tiene el valor de 10 en el display no se muestra nada, esto ocurrirá millonésimas de segundo asi que no se notara. Seguido realizamos una comprobación si la variable conteo es igual a 10 entonces a conteo le asignamos el valor de cero luego de esto regresa al lazo infinito en la espera de una nueva interrupción por RB0.

En la sección de decremento empieza disminuyendo CONTEO a uno, luego verifica si este valor es 255, si lo es se va a la sección DESBORDE donde le da el valor de 9 a CONTEO y lo muestra en el display, si CONTEO anteriormente no era 255 pues simplemente lo muestra en le display, luego de esto regresa al lazo infinito a la espera de la interrupción por RB0.

**6.) Listado del programa fuente en lenguaje ensamblador con comentarios en las líneas de código que considere fundamentales**

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; *CONTADOR DECIMAL DE PULSOS*

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; NOMBRE: p1\_led.asm

; FECHA: 14/07/2011

; VERSION: 1.00

; PROGRAMADOR: Vicente Alexander Macas

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; DESCRIPCION:

;Este programa cuenta mediante una variable los pulsos ingresados mediante ;una botonera en RB0. También habrá un SWITCH en el puerto A que indique ;modo: de incremento o de decremento. El resultado se lo visualiza en un ;DISPLAYS de 7 segmentos colocado en el puerto C.

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;DIRECTIVAS

LIST p=16F887 ;Tipo de microcontrolador

INCLUDE P16F887.INC ;Define los SFRs y bits del

;P16F887

\_\_CONFIG \_CONFIG1, \_CP\_OFF&\_WDT\_OFF&\_INTOSCIO&\_FCMEN\_OFF

;Setea parámetros de configuración

errorlevel -302 ;Deshabilita mensajes de

;advertencia por cambio

;bancos

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

CONTEO EQU 0X21

;PROGRAMA

ORG 0x00 ;Vector de RESET

GOTO MAIN

ORG 0x04 ;Vector de interrupción

GOTO Interrupcion ;Va a rutina de interrupción

MAIN

;SETEO DE PUERTOS

BSF STATUS,5

BSF STATUS,6 ;Selecciona el Banco 3

CLRF ANSELH ; Digitaliza puertos ANS.

CLRF ANSEL

BCF STATUS,6 ;Selecciona el Banco 1

BSF OSCCON,6

BSF OSCCON,5

BCF OSCCON,4 ; Selecciona el reloj de 8MHz

BSF OSCCON,0 ; Selecciona el oscilador de rejor interno.

MOVLW 0XFF

MOVWF TRISA ;Setea PORTA como entrada.

MOVLW 0XFF

MOVWF TRISB ;Setea PORTB como entrada.

CLRF TRISC ;Setea PORTC como salida.

MOVLW B'10010000'

MOVWF INTCON ; GIE, INTE Habilita interrupcion por RB0.

;INICIALIZACION

BCF STATUS,5 ;Selecciona el Banco 0.

CLRF PORTA ;Encera PORTA

CLRF PORTB ;Encera PORTB

CLRF PORTC ;Encera PORTCINTCON

CLRF PORTE ;Encera PORTD

CLRF CONTEO ;Encera CONTEO

MOVFW CONTEO ;Encera el registro W.

CALL TABLA

MOVWF PORTC

INICIO

NOP

GOTO INICIO ; Lazo infinito

Interrupcion

BCF INTCON,INTF ; Borra la bandera de interrupcion TMR0IF

CLRF PORTB ; Encera el PORTB.

BTFSC PORTA,0 ; Pregunta si en RA0 esta con cero.

GOTO INCREMENTO ; RA0=1 INCREMENTO

GOTO DECREMENTO ; RA0=0 DECREMENTO

INCREMENTO

INCF CONTEO ; Incrementa CONTEO

MOVFW CONTEO ; A W le asiga CONTEO

CALL TABLA

MOVWF PORTC ; PORTC se le asigna el código de 7 segmentos.

MOVFW CONTEO ; a W se le asigna CONTEO

SUBLW 0X0A ; W=10-CONTEO

BTFSS STATUS,Z ; Si CONTEO=10 salta.

GOTO SALIDA

MOVLW 0X00

MOVWF CONTEO ; CONTEO=0

MOVFW CONTEO ; W=CONTEO

CALL TABLA

MOVWF PORTC ; PORTC se carga con el código de 7 segmentos.

GOTO SALIDA

DECREMENTO

DECF CONTEO ; Decrementa CONTEO en una unidad.

MOVFW CONTEO ; W=CONTEO

SUBLW 0XFF ; W=255-CONTEO

BTFSC STATUS,Z ; Verifica si desborde mediante Z=1.

GOTO DESBORDE

MOVFW CONTEO ; W=CONTEO

CALL TABLA

MOVWF PORTC ; PORTC le asigna el código de 7 segmentos.

GOTO SALIDA

DESBORDE

MOVLW 0X09

MOVWF CONTEO ; CONTEO=9

MOVFW CONTEO

CALL TABLA

MOVWF PORTC ; PORTC se carga con 9.

SALIDA

BSF INTCON,GIE ; Se habilita la interrupcion global

RETFIE ; Regresamos al lazo infinito a espera

; de otra interrupcion.

TABLA

ADDWF PCL,F ; PCL + W -> W

; El PCL se incrementa con el

; valor de W proporcionando un

; salto

RETLW 0x3F ; Retorna con el código del 0

RETLW 0x06 ; Retorna con el código del 1

RETLW 0x5B ; Retorna con el código del 2

RETLW 0x4F ; Retorna con el código del 3

RETLW 0x66 ; Retorna con el código del 4

RETLW 0x6D ; Retorna con el código del 5

RETLW 0x7D ; Retorna con el código del 6

RETLW 0x07 ; Retorna con el código del 7

RETLW 0x7F ; Retorna con el código del 8

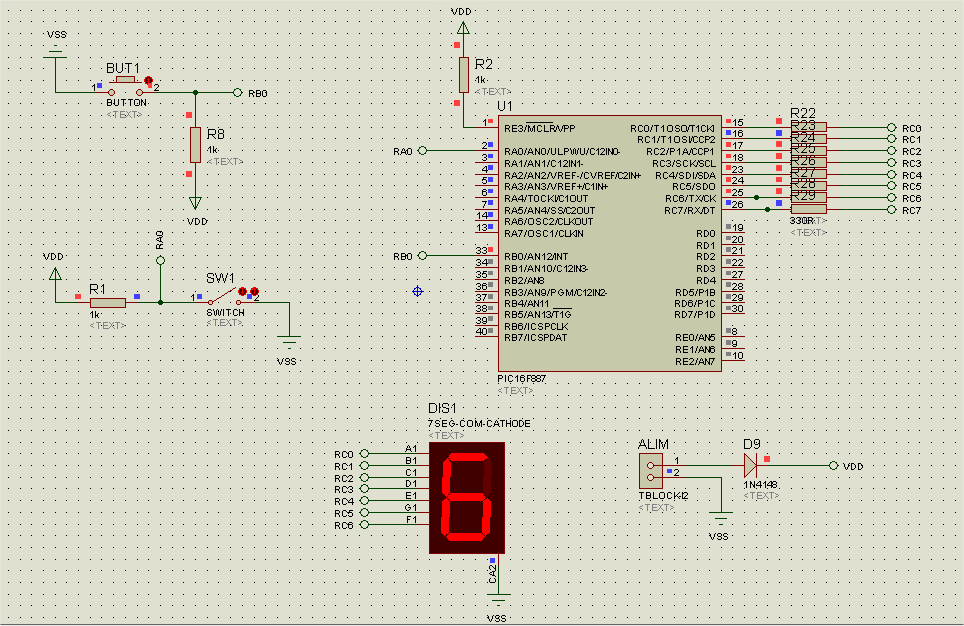
RETLW 0x67 ; Retorna con el código del 9

RETLW 0X00

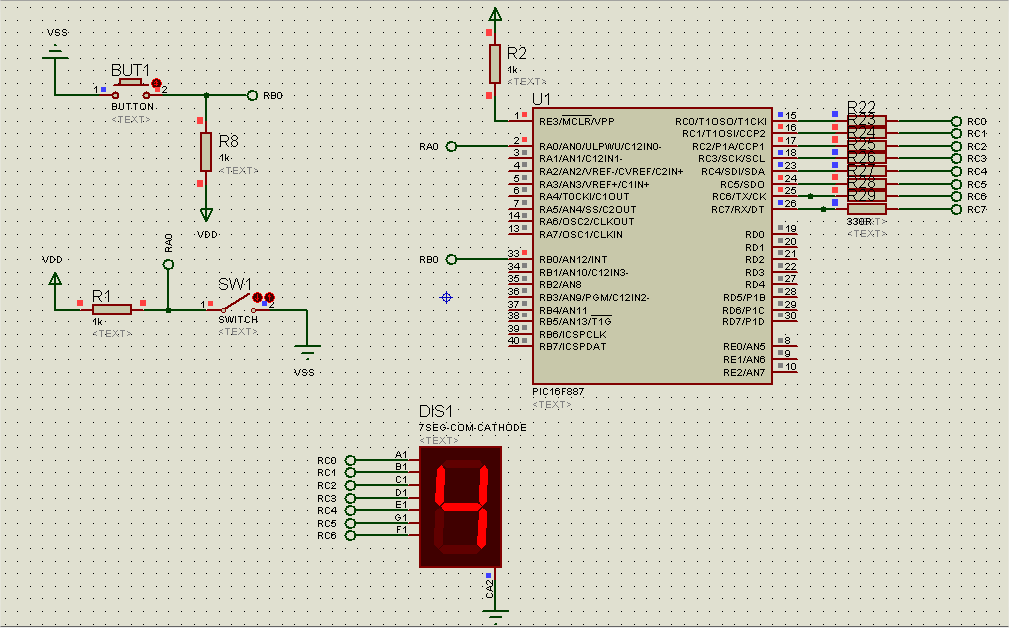
END ; Fin del programa fuente

**7.- Copia impresa del circuito armado en PROTEUS para la simulación en el momento de su ejecución.**

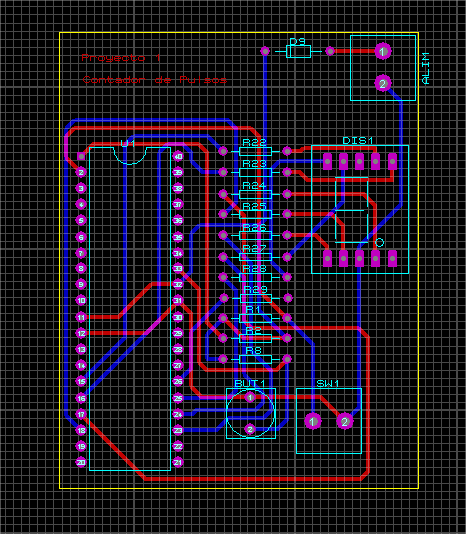
**a) RA0=0 Decremento**

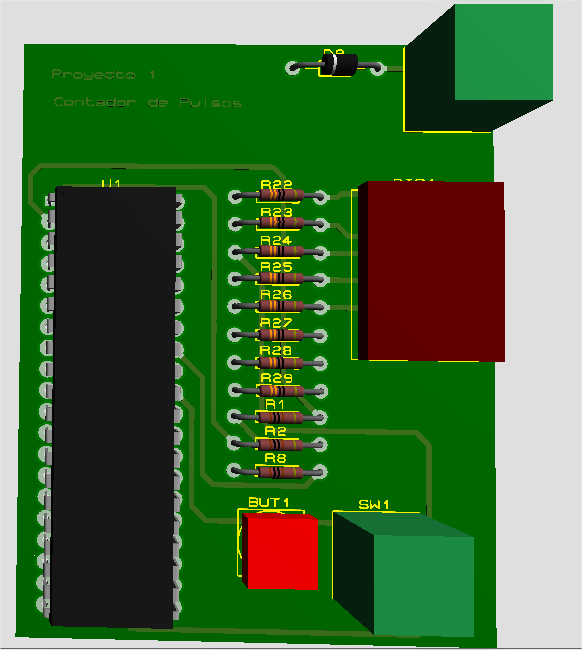
****

**a) RA0=1 Incremento**



**8.) Circuito implementado en PCB Ares (3D)**





**9.) Conclusiones**

* En el presente proyecto desarrollado pudimos poner en práctica las interrupciones externas particularmente la interrupción por RB0.
* Se logro entender cuál es el esquema básico de la programación de un microcontrolador empezando por el diagrama de bloques, luego al diagrama de flujo y finalmente traducir en un código .asm con todos sus comandos y sentencias para luego llevarlo al microcontrolador e implementarlo.
* Se pudo realizar la implementación del circuito en proteus ubicando el switch, la botonera y el display que son los que interactúan con el microcontrolador y lo hacen funcionar.
* Realizamos sumas, restas, comparaciones, interrupciones, incrementos, decrementos dentro de un mismo programa para así consolidar los estudiado en la materia teórica y experimental de microcontroladores.

**10.) Recomendaciones**

* Al usar el PROTEUS recordar que es mejor ponerle etiquetas a los pines ya que así nos ahorramos la presencia de muchos cables y la incómoda conexión. Se tendría una mejor visualización.
* Si el programa PROTEUS no compila, en algunas ocasiones se tiene que verificar que la frecuencia de trabajo sea 4Mhz.
* En PROTEUS hay que re indicar al micro controlador que el archivo .HEX se encuentra en la dirección dicha, ya que en muchas ocasiones cambian las direcciones cuando se mueve el archivo.
* Es muy importante etiquetar ordenadamente los lazos con la finalidad de tener un concepto claro del transcurso del programa.