# **ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**



**FACULTAD DE INGENIERIA EN**

**EN ELECTRICIDAD Y TELECOMUNICACIONES**



**PROYECTO DE LABORATORIO DE MICROCONTROLADORES**

**CONTADOR ASCENDENTE/DESCENDENTE**

**Y**

**CALCULADORA SUMA Y RESTA**

***LEONARDO CUADRADO G.***

# **Paralelo 11**

# **Grupo 5**

**Ing. Carlos Valdivieso**

# **Jueves, 14 de Julio de 2011**

**CALCULADORA SUMA Y RESTA**

1. **Enunciado del Proyecto.**

Programa que realice la suma y resta de dos números binarios y que muestre la digitación de los números binarios en Leds, y el resultado de la operación en displays de 7 segmentos.

Utilizar el puerto B para botoneras de entradas de los números binarios y los botones de suma y resta, los puertos A y D para los displays de 7 segmentos y puerto C para los Leds.

1. **Diagrama de Bloques**

**PIC16F887**

**Pulsadores Entrada**

**Bloque de Visualización de Display 7Seg**

**Convertidor Display 7\_Seg**

**Convertidor Bin a BCD**

**Oscilador Interno**

**Bloque de Visualización de LEDS**

1. **Diagrama de Flujo.**

**Programa Principal**



1. Empezamos seteando los parámetros de configuración, como des habilitación del perro guardián, protección del código, entre otras configuraciones.
2. Definimos variables.
3. Ajustamos configuraciones del PIC, selección de puertos.
4. Empieza programa principal.
5. Números Binarios, cuando se pulsa cualquier botón para el ingreso del número binario, este entra a una etapa de validación donde verifica que se presionó y que se soltó la botonera, luego carga el valor del ingreso de los números binarios en los leds.
6. Botón Sumar, una vez ya ingresados los números el programa espera la pulsación del botón sumar, donde realiza la operación suma, pasa a subrutinas de conversión binario a bcd, para luego pasar a la subrutina de conversión de un número a display de 7 segmentos mostrar el los display de 7 segmentos el resultado.
7. Botón Resta, una vez ya ingresados los números el programa espera la pulsación del botón sumar, donde realiza la operación resta, pasa a subrutinas de conversión binario a bcd, para luego pasar a la subrutina de conversión de un número a display de 7 segmentos mostrar el los display de 7 segmentos el resultado.
8. Luego de terminar el proceso de las operaciones el programa vuelve al menú principal, donde después de un reset podemos cargar nuevos números para realizar una nueva operación.

**Subrutina De Tabla**



1. **Listado del Programa Fuente**

**Programa Principal**

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

; RELOJ BINARIO DE MINUTOS Y SEGUNDOS

; AJUSTE DE SEGUNDOS

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

; NOMBRE: AjusteSegundos.asm

; FECHA: 19/07/2010

; VERSION: 1.00

; PROGRAMADOR: Fernando Orbea Guerrero

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

; DESCRIPCION:

;

; Programa que realice la suma y resta de dos números binarios y que muestre la digitación de los números

;binarios en Leds, y el resultado de la operación en displays de 7 segmentos.

;Utilizar el puerto B para botoneras de entradas de los números binarios y los botones de suma y resta, los

;puertos A y D para los displays de 7 segmentos y puerto C para los Leds

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

list p=16F887

#include <P16f887.INC>

\_\_CONFIG \_CONFIG1, \_CP\_OFF&\_WDT\_OFF&\_XT\_OSC ;Setea parámetros de configuración

errorlevel -302 ;Deshabilita mensajes de advertencia

CBLOCK 0x20

var

temp

num

ENDC

#define boton\_a PORTB,0

#define boton\_b PORTB,1

#define boton\_c PORTB,2

#define boton\_d PORTB,3

#define boton\_e PORTB,4

#define boton\_f PORTB,5

#define sumar PORTB,6

#define restar PORTB,7

#define LEDS PORTC

#define Display\_1 PORTA

#define Display\_2 PORTD

org 0;inicio del programa

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

;\*\*\*\*\*\*\*\*programa principal de seteos de segundos\*\*

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

principal

call PIC\_SET ;seteo del pic 16f887

clrf LEDS

clrf Display\_1

clrf Display\_2

clrf var

clrf temp

menu\_inicio

btfss boton\_a

goto seguir\_b

goto test\_a

seguir\_b

btfss boton\_b

goto seguir\_c

goto test\_b

seguir\_c

btfss boton\_c

goto seguir\_d

goto test\_c

seguir\_d

btfss boton\_d

goto seguir\_e

goto test\_d

seguir\_e

btfss boton\_e

goto seguir\_f

goto test\_e

seguir\_f

btfss boton\_f

goto seguirs

goto test\_f

seguirs

btfss sumar

goto seguirr

goto contar

seguirr

btfss restar

goto menu\_inicio

goto decontar

test\_a

btfsc boton\_a

goto test\_a

btfsc LEDS,0

goto off\_a

movlw d'1'

addwf var

movfw var

movwf LEDS

goto menu\_inicio

off\_a

movlw d'1'

subwf var

movfw var

movwf LEDS

goto menu\_inicio

test\_b

btfsc boton\_b

goto test\_b

btfsc LEDS,1

goto off\_b

movlw d'2'

addwf var

movfw var

movwf LEDS

goto menu\_inicio

off\_b

movlw d'2'

subwf var

movfw var

movwf LEDS

goto menu\_inicio

test\_c

btfsc boton\_c

goto test\_c

btfsc LEDS,2

goto off\_c

movlw d'4'

addwf var

movfw var

movwf LEDS

goto menu\_inicio

off\_c

movlw d'4'

subwf var

movfw var

movwf LEDS

goto menu\_inicio

test\_d

btfsc boton\_d

goto test\_d

btfsc LEDS,3

goto off\_d

movlw d'8'

addwf var

movfw var

movwf LEDS

goto menu\_inicio

off\_d

movlw d'8'

subwf var

movfw var

movwf LEDS

goto menu\_inicio

test\_e

btfsc boton\_e

goto test\_e

btfsc LEDS,4

goto off\_e

movlw d'16'

addwf var

movfw var

movwf LEDS

goto menu\_inicio

off\_e

movlw d'16'

subwf var

movfw var

movwf LEDS

goto menu\_inicio

test\_f

btfsc boton\_f

goto test\_f

btfsc LEDS,5

goto off\_f

movlw d'32'

addwf var

movfw var

movwf LEDS

goto menu\_inicio

off\_f

movlw d'32'

subwf var

movfw var

movwf LEDS

goto menu\_inicio

contar

btfsc sumar

goto contar

movfw var

movwf temp

andlw d'7'

movwf num

rlf temp,1

swapf temp,0

andlw d'7'

addwf num

movfw num

movwf var

movfw var

call BIN\_a\_BCD

movwf temp

andlw d'15'

call Numero\_a\_7Segmentos

movwf Display\_2

swapf temp,0

andlw d'15'

call Numero\_a\_7Segmentos

movwf Display\_1

goto menu\_inicio

decontar

btfsc restar

goto decontar

movfw var

movwf temp

andlw d'7'

movwf num

rlf temp,1

swapf temp,0

andlw d'7'

subwf num

movfw num

movwf var

movfw var

call BIN\_a\_BCD

movwf temp

andlw d'15'

call Numero\_a\_7Segmentos

movwf Display\_2

swapf temp,0

andlw d'15'

call Numero\_a\_7Segmentos

movwf Display\_1

goto menu\_inicio

;;;;;;;;;;;;;;;;;;;;;;;;

;funciones

;;;;;;;;;;;;;;;;;;;;;;;;;

PIC\_SET

banksel ANSELH

clrf ANSELH

clrf ANSEL

banksel TRISB

bsf boton\_a

bsf boton\_b

bsf boton\_c

bsf boton\_d

bsf boton\_e

bsf boton\_f

bsf sumar

bsf restar

clrf LEDS ;salidas de leds al puerto C

clrf Display\_1

clrf Display\_2

banksel PORTA

return

**Subrutina De Conversión a Display 7Seg**

CBLOCK

Display7s\_Dato ; Aquí se reservará el valor del dato de entrada.

ENDC

Numero\_a\_7Segmentos

andlw b'00001111' ; Se queda con el nibble bajo.

addlw '0' ; Se pasa a ASCII sumándole el valor ASCII

; del "0" y ejecuta "ASCII\_7\_Segmentos".

ASCII\_a\_7Segmentos

movwf Display7s\_Dato ; Guarda el valor del carácter.

sublw ' ' ; Comprueba si es " " (espacio).

btfsc STATUS,Z ; ¿Es distinta de " " (espacio)?, ¿Z=0?

retlw 00h ; Es " ". Devuelve el código 7-Segmentos del " ".

movf Display7s\_Dato,W ; Recupera el valor del dato de entrada.

sublw 'Ñ' ; Comprueba si es "Ñ".

btfsc STATUS,Z ; ¿Es distinta de "Ñ"?, ¿Z=0?

retlw 55h ; Es "Ñ". Devuelve el código 7-Segmentos de "Ñ".

movf Display7s\_Dato,W ; Recupera el valor del dato de entrada.

sublw 'º' ; Comprueba si es "º".

btfsc STATUS,Z ; ¿Es distinta de "º"?, ¿Z=0?

retlw 63h ; Es "º". Devuelve el código 7-Segmentos de "º".

;

movf Display7s\_Dato,W ; Comprueba si el código ASCII es mayor que la "Z".

sublw 'Z' ; (W)='Z'-(Display7s\_Dato)

btfss STATUS,C ; ¿C=1?, ¿(W) positivo?, ¿'Z'>=(Display7s\_Dato)?

retlw b'10000000' ; Si el código ASCII es mayor que 'Z' es un error.

movlw '+' ; Averigua en qué orden está el carácter leído

subwf Display7s\_Dato,W ; dentro de la tabla de conversión, respecto del

btfss STATUS,C ; primero que es '+'.

retlw b'10000000' ; Si el código ASCII es menor que '+' es un error.

addwf PCL,F ; Obtiene el código 7 segmentos.

InicioTablaASCII

DT 46h, 04h, 40h, 80h, 00h, 3Fh, 06h, 5Bh, 4Fh, 66h, 6Dh ; Signos y

DT 7Dh, 07h, 7Fh, 67h, 41h, 88h, 00h, 48h, 00h, 00h, 00h ; números.

DT 77h, 7Ch, 39h, 5Eh, 79h, 71h, 6Fh, 76h, 19h, 1Eh, 7Ah, 38h, 37h ; Letras.

DT 54h, 3Fh, 73h, 67h, 50h, 6Dh, 78h, 1Ch, 3Eh, 1Dh, 70h, 6Eh, 49h

FinTablaASCII

**Subrutina de Conversión BIN a BCD**

CBLOCK ; En las subrutinas no se debe fijar la dirección

BCD\_Decenas ; de la RAM de usuario. Se toma a continuación de

BCD\_Unidades ; la última asignada.

ENDC

BIN\_a\_BCD ; Carga los registros con el resultado inicial.

clrf BCD\_Decenas ; En principio las decenas a cero.

movwf BCD\_Unidades ; Se carga el número binario a convertir.

BCD\_Resta10

movlw .10 ; A las unidades se les va restando 10 en cada

subwf BCD\_Unidades,W ; pasada. (W)=(BCD\_Unidades) -10.

btfss STATUS,C ; ¿C = 1?, ¿(W) positivo?, ¿(BCD\_Unidades)>=10?

goto BIN\_BCD\_Fin ; No, es menor de 10. Se acabó.

BCD\_IncrementaDecenas

movwf BCD\_Unidades ; Recupera lo que queda por restar.

incf BCD\_Decenas,F ; Incrementa las decenas y comprueba si ha llegado

movlw .10 ; a 10. Lo hace mediante una resta.

subwf BCD\_Decenas,W ; (W)=(BCD\_Decenas)-10).

btfss STATUS,C ; ¿C = 1?, ¿(W) positivo?, ¿(BCD\_Decenas)>=10?

goto BCD\_Resta10 ; No. Vuelve a dar otra pasada, restándole 10 a

BIN\_BCD\_Fin

swapf BCD\_Decenas,W ; En el nibble alto de (W) también las decenas.

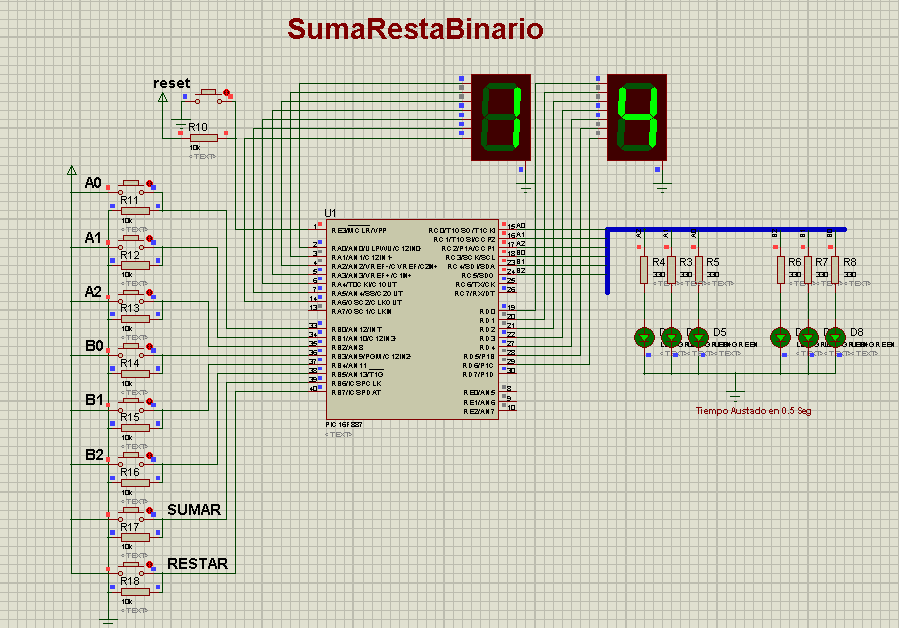
addwf BCD\_Unidades,W ; En el nibble bajo de (W) las unidades.

return ; Vuelve al programa principal.

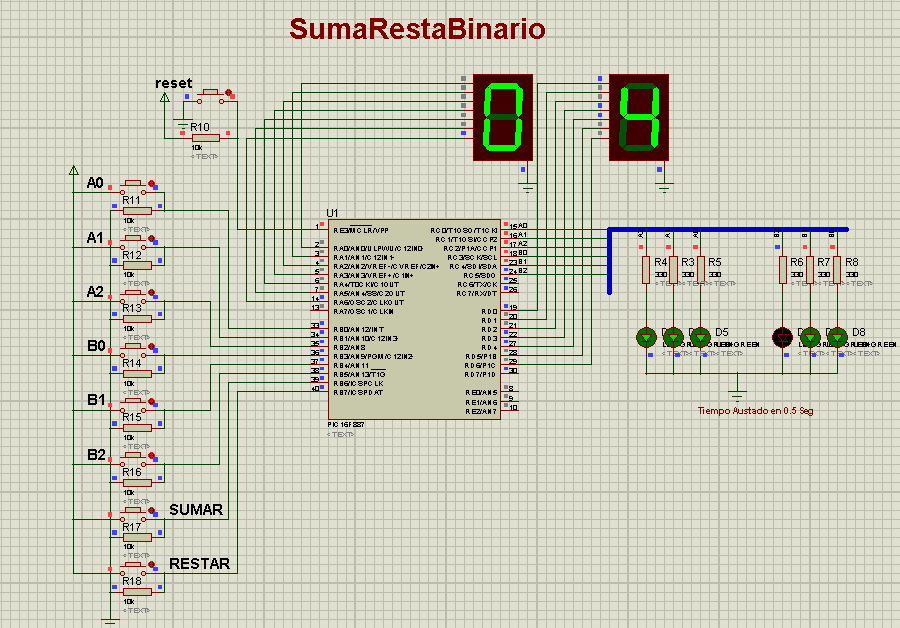
1. **Circuito Armado en Proteus**

****

Simulación donde se muestra la suma de los números



Simulación donde se muestra la resta de los números



1. **Conclusiones**

* El uso adecuado de retardos nos permitirán implementar cualquier tipo de circuito, como en el proyecto para crear retardos de 10mseg que lo usamos en el anti rebote y el de 1Seg retardo usado para contar los segundos del contador ascendente y descendente.
* Para cargar el numero en el puerto del Display hacemos uso de un convertidor a Display de 7Seg, para aquello usamos una función que ya estaba implementada y la usamos para continuar con la demás programación y ahorrar tiempo.
* Una vez culminada el proyecto pudimos darnos cuenta que el PIC 16F887 es un micro controlador muy empleado puesto que tiene mayor cantidad de recursos que el 16F84, y con el conocimiento necesario de cómo funcionan los lazos podemos modificar cualquier programa con la finalidad de obtener mayores retardos.

1. **Recomendaciones.**

* Para poder realizar la práctica y sacarle el mayor provecho es de vital importancia que el estudiante tenga el conocimiento necesario del comportamiento de los lazos anidados, de esta manera no tendrá ningún inconveniente si es instructor le pide que realice cambios en el programa para aumentar el retardo de tiempo o hacer alguna variante en ello.
* De igual forma se pide al estudiante que tenga mucho cuidado al momento de programar el micro controlador (PIC), para ello debe recordar las opciones que debe habilitar o deshabilitar para la ejecución del programa en un momento dado con el propósito de que no se vayan a presentar problemas en el proyecto y podemos llegar a una feliz culminación.
* Se recomienda al estudiante llevar adecuadamente la materia, esto es, revisar el material proporcionado en el aula de clases y revisar cada uno de los programas realizar en el laboratorio para obtener conocimientos sólidos que nos permitirán desenvolvemos de mejor manera en el laboratorio y cuando abandonemos las aulas de clases en nuestra vida laboral.