

COMPILACION, SIMULACION Y CREACION DE APLICACIONES BASICAS CON MICROCONTROLADORES

Jeferson Andrey López Aparicio

Escuela de Ingeniería Electrónica
Universidad Pedagógica y Tecnológica de Colombia
Sogamoso, Boyacá, Colombia
jeferson.lopez@uptc.edu.co

Elber Alexander Ponguta Fernandez

Escuela de Ingeniería Electrónica
Universidad Pedagógica y Tecnológica de Colombia
Sogamoso, Boyacá, Colombia
elber.ponguta@uptc.edu.co

Resumen—se realizó una aplicación usando un microcontrolador al finalizar la tarea principal se visualizó una secuencia, se cuenta con opción de reinicio y cargué de valor.

Palabras Clave—microcontrolador; programador; aplicacion

I. INTRODUCCION

El presente informe se refiere al montaje y simulación de una aplicación usando como base el microcontrolador PIC16F887.

Para el desarrollo de la practica el primer paso fue realizar el diagrama de flujo del programa principal y cada subprograma, luego se uso lenguaje ensamblador para la codificación y el software MPLABX para su compilacion.

Luego de terminar el proceso de codificación se realizó la simulación de este usando el software proteus, terminada esta etapa de forma satisfactoria se prosigue a programar el micro para esto se usa el programador Pickit3 y se prueba en el circuito físico.

II. OBJETIVOS

- Crear una aplicación usando las instrucciones básicas del microcontrolador.
- Crear una aplicación usando un microcontrolador.
- Comprender el funcionamiento de un microcontrolador y su programación.
- Usar métodos de simulación para el testeo de una aplicación.

III. MATERIALES Y EQUIPO UTILIZADO

- PIC16F877.
- Programador Pickit3.
- Leds.
- Resistencias

IV. PROCEDIMIENTO

A. Programa Principal

1) Diagrama de flujo

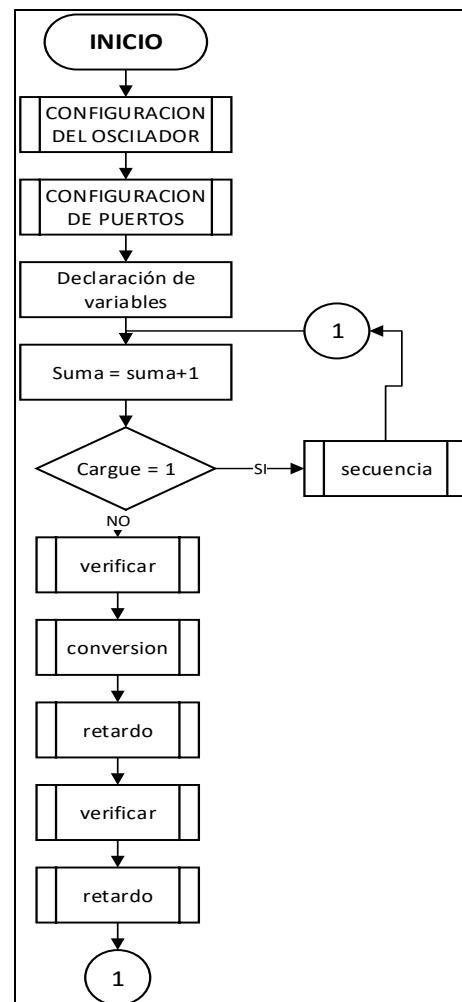


Fig 1. Diagrama de flujo programa principal.

2) Código del programa

```
incf suma,F
btfsc STATUS,Z
call secuencia
movf suma,W
movwf bin
```

```

call verificar
call conversion
call retardo_500ms
call verificar
call retardo_500ms
goto reinicio

```

Este programa hace el conteo y selecciona que tarea se realizara.

B. Verificar

1) Diagrama de flujo

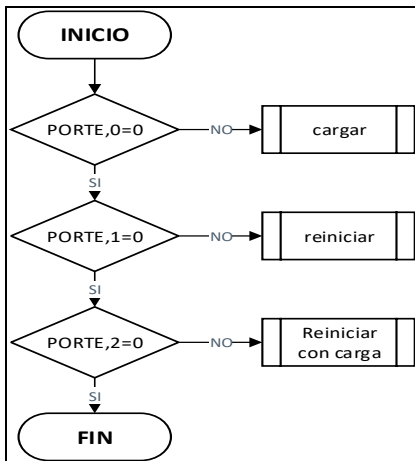


Fig 2. Diagrama de flujo de verificar.

2) Código del programa

```

btfss PORTE,0
call cargar
btfss PORTE,1
goto reiniciar
btfss PORTE,2
call reiniciar_con_carga
return

```

Este programa testea los bits del puerto E para saber si tiene que reiniciar, cargar un valor o volver al valor cargado.

C. Conversor

1) Diagrama de flujo

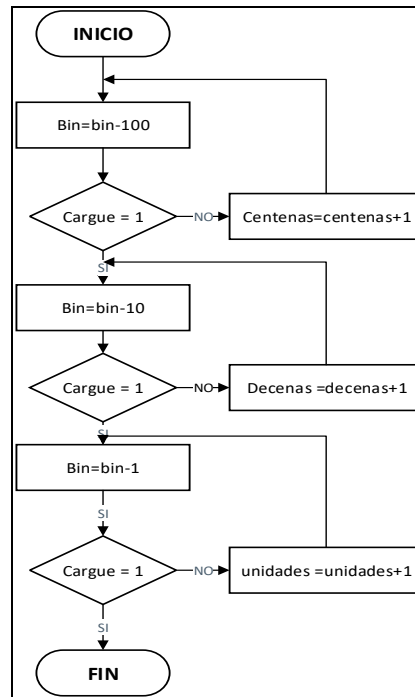


Fig 3. Diagrama de flujo del conversor bin a bcd.

2) Código del programa

conversion:

```

clrf PORTB
clrf PORTC
clrf cent
clrf uni_dec

```

bin_bcd:

```

movlw .100
subwf bin,W
btfsc STATUS,C
incf cent,F
btfsc STATUS,C
movwf bin
btfsc STATUS,C
goto bin_bcd

```

retorno:

```

movlw .10
subwf bin,W
btfsc STATUS,C
goto dec_10
movf bin,l
ret_unid:
btfss STATUS,Z
goto dec_1

```

```

movf uni_dec,W
movwf PORTB
movf cent,W
movwf PORTC
return

```

```
dec_10:
    movwf bin
    movlw 0x10
    addwf uni_dec,F
    goto retorno
```

```
dec_1:

    incf uni_dec,1
    decf bin,F
    goto ret_unid
```

Este programa consiste en tomar el número en bcd e ir restando, primero 100 para las centenas luego 10 para las decenas y por ultimo 1 para las unidades.

Si las resta es válida aumentara uno la casilla correspondiente hasta completar la conversión, para este proceso se usan variables auxiliares y cuando termina la conversión el resultado es puesto en los puertos de salida.

D. Secuencia

1) Diagrama de flujo

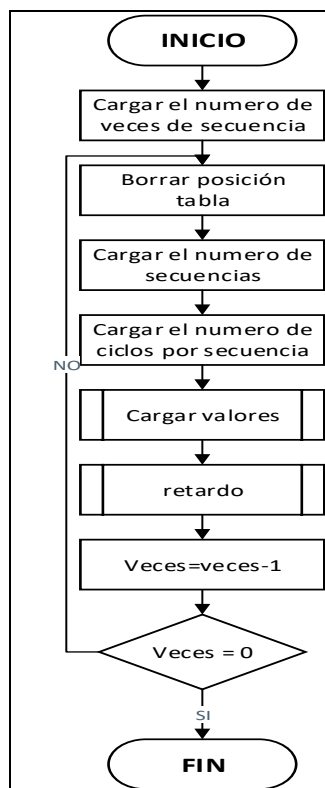


Fig 4. Diagrama de flujo de secuencia.

2) Codigo del programa

```
secuencia:
    movlw .2
    movwf repeticiones
bucle:
    clrf pos_tabla
    movlw .2
    movwf num_secuencias
bucle_2:
    movlw .11
    movwf ciclos_secuencia
bucle_3:
    call cargar_valores
    incf pos_tabla
    call retardo_500ms
    goto verificar_volver
volver_verificar:
    decfsz ciclos_secuencia
    goto bucle_3
    decfsz num_secuencias
    goto bucle_2
    decfsz repeticiones
    goto bucle
    clrf suma
    return
restaurar_secuencia:
```

```
verificar_volver:
    clrf uni_dec
    clrf cent
    movf valor_carga,W
    movwf suma
    btfss PORTE,1
    return
    btfss PORTE,2
    goto salir_carga
    goto volver_verificar
salir_carga:
    clrf suma
    return
```

```
cargar_valores:
    movf pos_tabla,W
    call tabla
    movwf PORTB
    movf pos_tabla,W
    call tabla2
    movwf PORTC
    return
```

```
tabla:
    addwf PCL,F
;primer secuencia
    retlw 0x7F
    retlw 0XFE
    retlw 0XFC
```

E. Simulacion

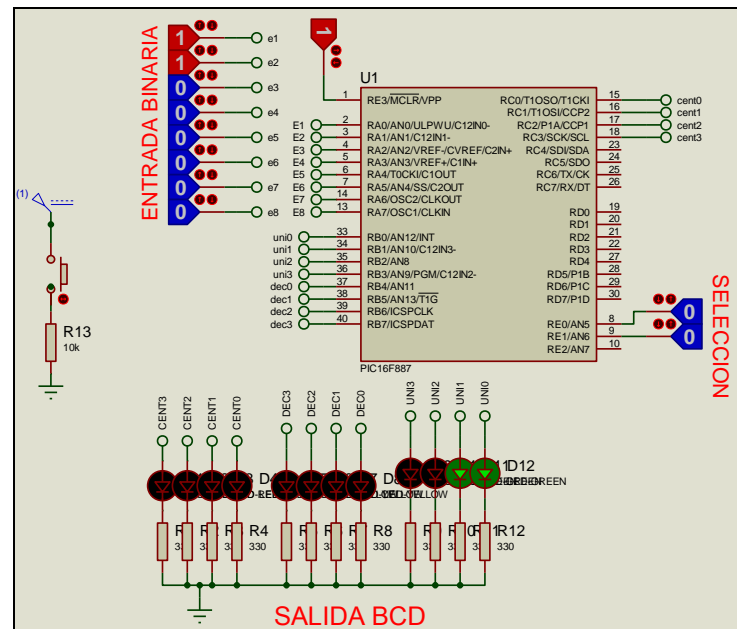


Fig 5. simulación en proteus.

F. Montaje

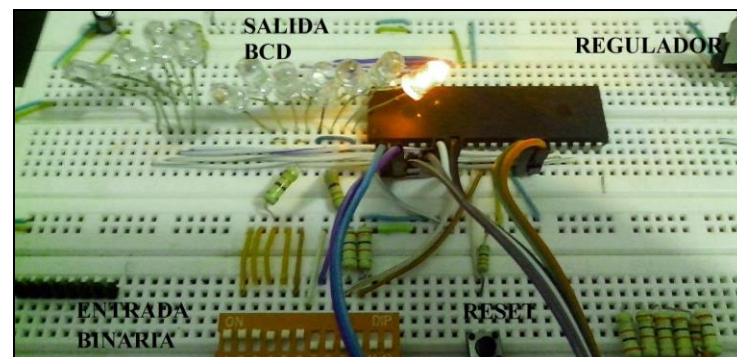


Fig 6. montaje.

CONCLUSIONES

El microcontrolador usado siendo de los más sencillos tiene bastantes instrucciones con las que se pueden hacer tareas con algún grado de complejidad.

Se pudo comprobar que los resultados obtenidos en simulación eran los mismos que en el circuito físico, llevando a esta herramienta a ser de gran ayuda en el desarrollo de aplicaciones usando microcontroladores.

Se aprendió a leer la hoja de especificaciones de un micro, como interpretar lo que esta contiene y usar al máximo las funciones del integrado.

```
retlw 0XF8
retlw 0XF0
retlw 0XE0
retlw 0XF0
retlw 0XF8
retlw 0XFC
retlw 0XFE
retlw 0X7F
;segunda secuencia
retlw 0X01
retlw 0X02
retlw 0X04
retlw 0X08
retlw 0X90
retlw 0X60
retlw 0X90
retlw 0X08
retlw 0X04
retlw 0X02
retlw 0X01
```

tabla2:

```
addwf PCL,F
;primer secuencia
retlw 0x00
retlw 0X00
retlw 0X01
retlw 0X03
retlw 0X07
retlw 0X0F
retlw 0X07
retlw 0X03
retlw 0X01
retlw 0X00
retlw 0X00
```

```
;segunda secuencia
retlw 0X08
retlw 0X04
retlw 0X02
retlw 0X01
retlw 0X00
retlw 0X00
retlw 0X00
retlw 0X01
retlw 0X02
retlw 0X04
retlw 0X08
```

La secuencia consiste en un patrón de luces el cual es repetido un número de veces.

Para realizar el patrón se usaron tablas, para la secuencia se iba tomando los valores de las tablas y luego se ponían estos valores en los puertos, finalizada la secuencia se retorna al programa principal.

REFERENCIAS

- [1] Harprit Singh Sandhu. Making Pic microcontroller instruments and controllers. McGraw-Hill, 2009.
- [2] Randall Hyde. The art of assembly language. 2da. Edicion. No starch press. 2010.
- [3] Microcontroladores PIC, la solución en un chip J. M^a. Angulo Usategui, E. Martín Cuenca y I. Angulo Martínez. Editorial Paraninfo, 2000
- [4] <http://www.microchip.com>