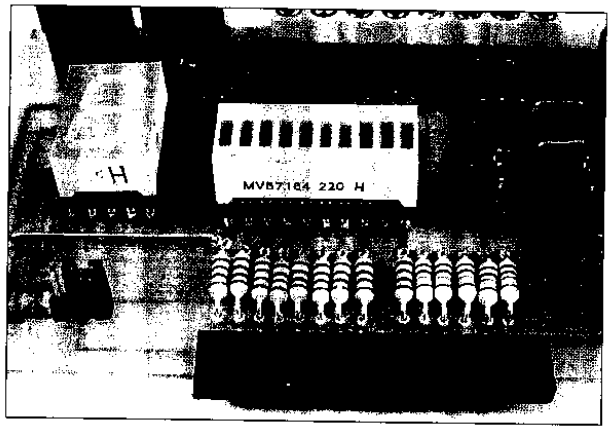
CAPITULO-3 Nº 163 - Noviembre 1997

CURSO
µC PIC
DE APLICACIONES

Curso de aplicaciones con microcontroladores PIC

Dr. Eugenio Martín Cuenca
Facultad de Ciencias. Universidad de Granada.
E-mail:emartin@goliat.ugr.es



SISTOR

CIRCUITERÍA DE RESET

El PIC16F84 dispone de un solo pin de reset, la patita MCLR. Incorpora internamente circuitería de reset que entra en funcionamiento de forma automática una vez se conecta la alimentación. El PIC16F84 hay que distinguir varios tipos de reset:

Por conexión de la alimentación VDD. (Power-on Reset : POR).

Reset durante el funcionamiento normal al activar MCLR.

Reset durante modo de reposo SLEEP al activar MCLR.

Reset al sobrepasar el contador del "Perro Guardian" - WDT.

Detección de fallo en la alimentación (Brown-Out)

Si no se necesita circuitería de reset, la patita MCLR se conecta directamente a VDD, por ejemplo cuando la alimentación alcanza una velocidad de crecimiento de la tensión rápida mayor de 0,05 V / ms. El temporizador Power-up Timer proporciona un retardo fijo de 72 ms durante el encendido. Este diseño mantiene el dispositivo en reset hasta que se estabiliza la alimentación.

La figura 1 muestra un diagrama de bloques simplificado de la circuiteria de reset incluida en el microcontrolador. La patita MCLR dispone de un filtro antiruido que ignora pulsos pequeños. Algunos registros no son afectados por ninguna condición de reset, su estado es desconocido durante el reset POR y no varían con cualquier otro reset. Muchos otros registros son afectados por el reset POR, MCLR o WDT en funcionamiento normal o

El PIC16F84 dispone de un solo pin de reset, la patita MCLR. Incorpora internamente circuitería de reset que entra en funcionamiento de forma automática una vez se conecta la alimentación. El PIC16F84 hay que distinguir varios tipos de reset:

R17

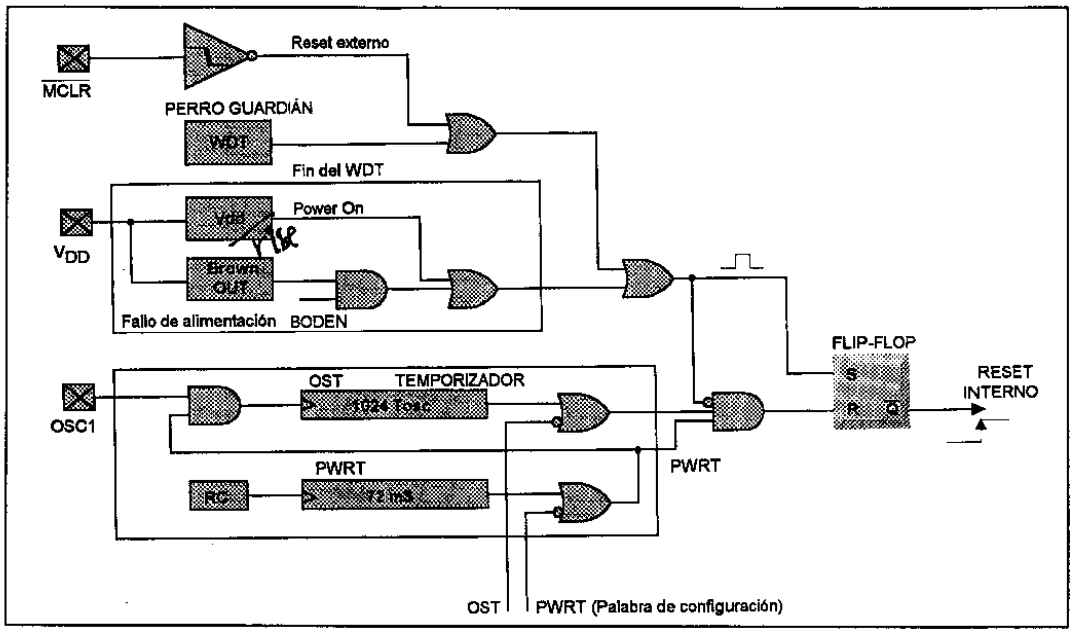


Figura 1.- Diferentes formas de conexionado del oscilador.(A) Tipo RC. (B) Tipos LP, XT y HS.

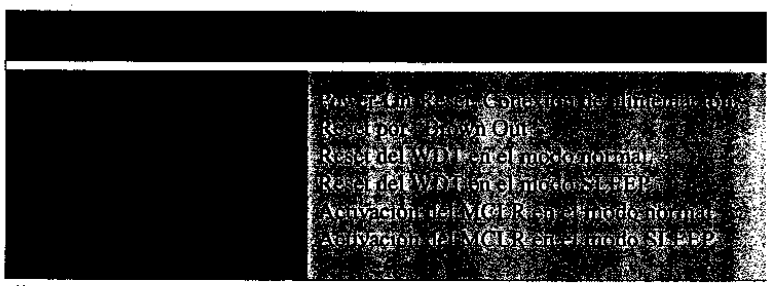


Tabla 1.- Descripción del estado de los bits de STATUS y su significado.

MCLR durante el modo reposo SLEEP.

Power-on Reset (POR). Se genera un pulso de reset POR cuando se detecta la subida de VDD entre 1.2V a 1.7. Para disponer de esta ventaja es necesario llevar la patita MCLR a VDD directamente o a través de una resistencia. Esto elimina los componentes RC externos usualmente necesarios para el reset.

Power-up Timer (PWRT): Este temporizador proporciona un tiempo de 72 ms para POR. Opera mediante un oscilador RC interno. El chip se mantiene en reset mientras PWRT esté activo. Esto permite que VDD suba a un nivel aceptable. Este temporizador puede activarse o desactivarse durante la grabación del microcontrolador mediante el bit PWRTE (Ver palabra de configuración).

Si no se necesita circuitería de reset, la patita MCLR se conecta directamente a VDD,

Oscillatro Satrt-up Timer (OST): El oscilador Start-up proporciona un retardo de 1024 ciclos de reloj de OSC1 después de la finalización del retardo de PWRT. Esto asegura que el cristal de cuarzo o el resonador ha arrancado y se ha estabilizado. Este temporizador solo actúa si se usan los modos XT, LP y HS y solo en el Reset Power-on o en wake-up en modo SLEEP. Cuando VDD sube muy despacio, es posible que TPWRT time-out y TOST terminen antes de que VDD haya tomado su valor final. En este caso sería nece-

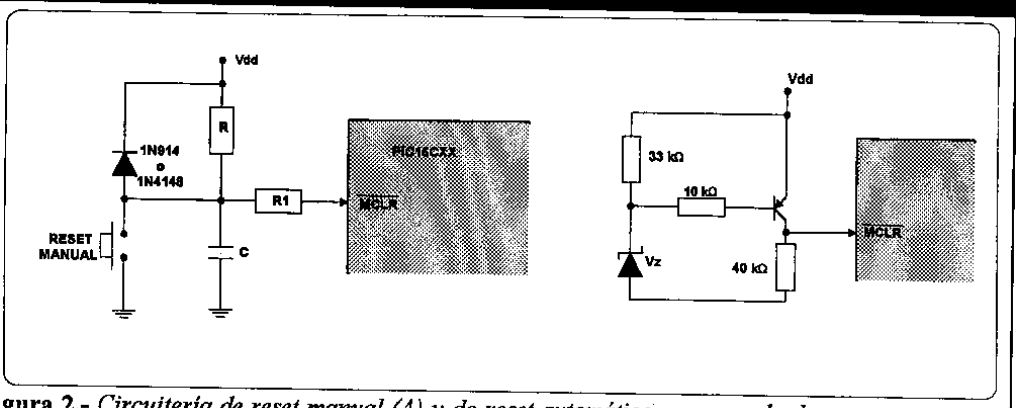


Figura 2.- *Circuitería de reset manual (A) y de reset automático en caso de descenso de la tensión de alimentación (B).*

sario una circuitería externa de reset power-on.

Time Out secuence y Power-Doan status bits (TO/PD) : En Power-up la secuencia de time-out se invoca después de que ha finalizado POR. Entonces se activa OST. El tiempo total de time-out varia dependiendo de la configuración del oscilador y la del bit PWRTE.

Cuando se necesita control de reset externo, puede emplearse el esquema mostrado en la figura 2-A. También es posible conectar una circuitería de reset sensible a las variaciones de la tensión de alimentación, que sea capaz de activarse si esta desciende por debajo de un determinado valor umbral (Brown-Out) figura 2-B. Reset por descenso de la tensión de alimentación.

Reset en Brown-Out : Esta es una condición donde la alimentación del dispositivo VDD toma un valor inferior del mínimo, pero sin llegar a cero y luego se recupera. El microcontrolador entra en reset en ese momento. Para que este se produzca es necesario añadir una circuitería externa de reset por descenso de la alimentación como la presentada en la figura 2-B.

Bits 4-13 CP **Code Protection**	**1 : Protección de código off** **0 : Toda la memoria con código protegido**
Bit 3 PWRTE **Power-up Timer**	**1 : Power-up timer desactivado** **0 : Power-up timer activado**
Bit 2 WDTE **Watchdog Timer Enable**	**1 : WDT "Perro Guardian " activado** **0 : WDT "Perro Guardian " desactivado**
Bits 1 - 0 FOCS1 - FOSC0. **Oscillator Selection**	**11 : Oscilador tipo RC** **10 : Oscilador tipo HS** **01 : Oscilador tipo XT** **00 : Oscilador tipo LP**

13													0
CP	CP	CP	CP	CP	CP	CP	CP	CP	CP	WRITE	WDTE	FOSC1	FOSC0

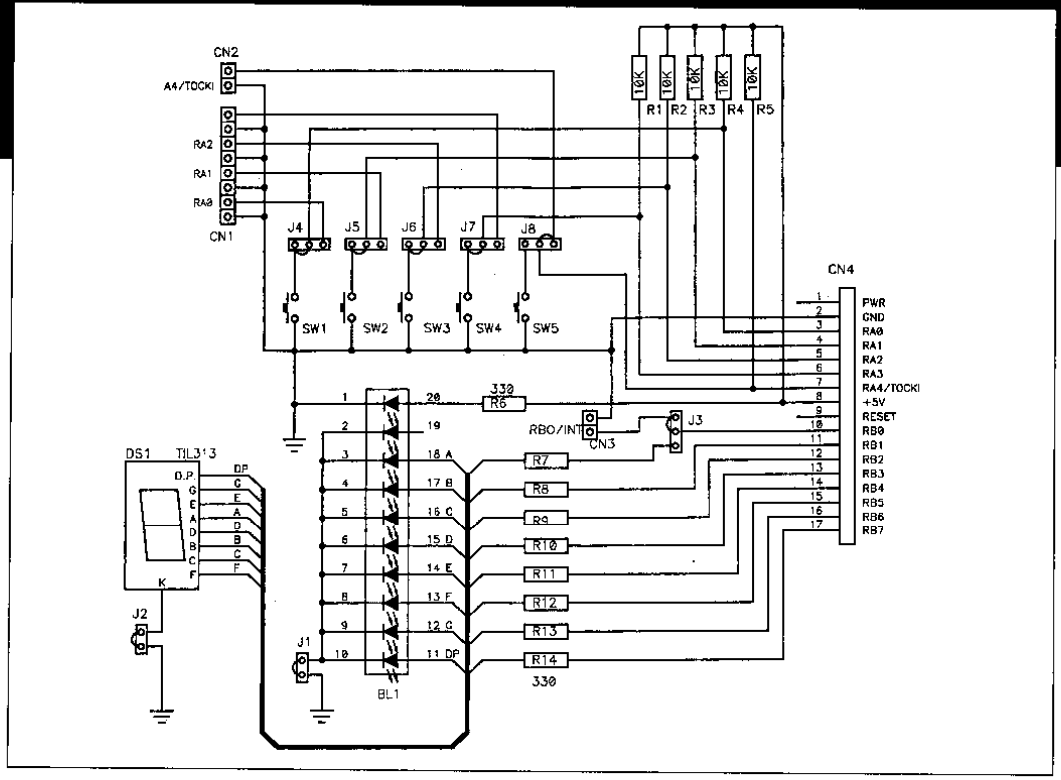
Figura 3.- función de la palabra de configuración.

La palabra de configuración de la figura 3 es válida para el PIC16F84, pero en el caso del PIC16C84 los bits PWRTE están invertidos.

PALABRA DE CONFIGURACIÓN

Se encuentra en la posición $2007 de la memoria de programa, dirección a la que únicamente puede accederse durante la grabación de microcontrolador. Al ser una posición de la memoria de programa se dispone por tanto de 14 bits cuya función es la reflejada en la figura 3:

La palabra de configuración de la figura 3 es válida para el PIC16F84, pero en el caso del PIC16C84 los bits PWRTE están invertidos. Esta es una da las escasas diferencias entre el modelo C84 y F84. Desde el punto de vista del usuario, no es de gran importancia, pues esto se ha tenido en cuenta en el diseño



del programa PIC84. EXE que controla el grabador (Kit Resistor). Solo es necesario seleccionar el modelo de PIC a grabar y si solo cuando se realice protección de código.

Practicas (III)

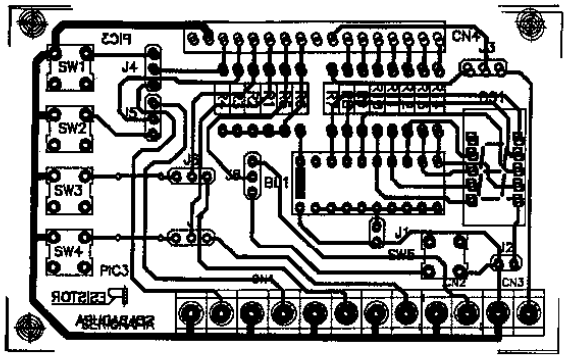
MÓDULO - 02

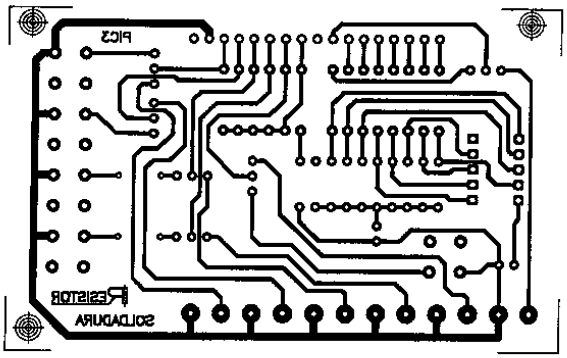
Con el programador, el modulo numero 1 y este que presentamos, puede realizar el 90 % de las practicas incluidas en el libro.

En algunos casos como para el uso de interrupciones deberá ajustar los diferentes jumpers. En la configuración por defecto,

	Indirecto	* Power-On-Reset	* MCLR en modo normal * MCLR en modo SLEEP * RESET provocado por el WDT * Brown-Out	* Final de SLEEP provocado por interrupción * Final de SLEEP provocado por rebasamiento del WDT
W	—	xxxx xxxx	uuuu uuuu	uuuu uuuu
INDF	00h	—	—	—
TMR0	01h	xxxx xxxx	uuuu uuuu	uuuu uuuu
PCL	02h	0000 0000	0000 0000	PC + 1 (3)
STATUS	03h	0001 1xxx	000? ?uuu (4)	uuu? ?uuu (4)
FSR	04h	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTA	05h	—x xxxx	—u uuuu	—u uuuu
PORTB	06h	xxxx xxxx	uuuu uuuu	uuuu uuuu
EEDATA	08h	xxxx xxxx	uuuu uuuu	uuuu uuuu
PCLATH	0Ah	—0 0000	—0 0000	—u uuuu
INTCON	0Bh	0000 000x	0000 000x	uuuu uuuu (2)
INDF	80h	— —	— —	— —
OPTION	81h	1111 1111	1111 1111	uuuu uuuu
TRISA	85h	—1 1111	—1 1111	—u uuuu
TRISB	86h	1111 1111	1111 1111	uuuu uuuu
EEADR	09h	xxxx xxxx	uuuu uuuu	uuuu uuuu
EECON1	88h	—0 x000	—0 q000	—0 uuuu
EECON2	89h	— —	— —	— —

Tabla 2.- Condición en reset de algunos registros.





las patillas de la Puerta A están conectadas a cinco pulsadores (A0 - A4) y las patillas de la Puerta B están conectadas a una barra de diodos electroluminiscentes y a un display de 7 segmentos. Aunque pueden funcionar simultáneamente la barra de LEDs y el display de 7 segmentos, los jumpers JP1 y JP2 permite desconectar uno u otro elemento.

Si se desean realizar experimentos de interrupción, el jumper JP3 permite llevar la línea RB0/INT a 1 conector CN3 donde queda disponible. Para el uso del TMR0 a través de la entrada de la patita A4, el jumper JP4 lleva esta línea al conector CN2.

Se ha separado exprofeso la línea A4 al tener la característica adicional de ser la entrada de contador del Timer0.

Por último, los jumpers JP5-JP8 pueden llevar cualquiera de las líneas RA0-RA3 al conector CN1, lo que permite el empleo de cualquiera de estas líneas en funciones de entrada y salida diferentes a las de su uso mediante los pulsadores A0-A4.

La línea de alimentación se ha conectado a uno de los diodos LEDs de la barra cuya función es indicar la presencia de suministro eléctrico.

Así por lo tanto, en este momento se ha suministrado al lector material suficiente para poder realizar sus diseños en la mayoría de los casos.

Se ha separado exprofeso la línea A4 al tener la característica adicional de ser la entrada de contador del Timer0. Con respecto a las líneas del Puerto B se ha llevado a una barra de diodos LEDs y a un display de 7 segmentos. Esto permite realizar las practicas de salidas de información. También se dispone de dos jumpers que permiten desactivar ambos elementos.

EXPERIENCIA (2)

La segunda experiencia consiste en la escritura de dos programas para el aprendizaje usando el Módulo-01 y el Módulo-02. Conecte los dos módulos tal y como se ve en la fotografía. Ya se ha indicado repetidas veces que con esta combinación pueden realizarse la mayoría de las experiencias del libro. Aquí realizaremos dos a título de ejemplo, aunque antes de seguir adelante debemos recordar que el programa CONTAR.ASM del capítulo anterior puede ejecutarse aquí perfectamente.

El primer programa consiste en imaginarse un sistema de alarma. Supongamos que se dispone de un sistema de alarma que controla cuatro posibles entradas a nuestro hogar, por ejemplo cuatro ventanas y la puerta. La activación de los sensores colocados en las mismas, activaran la alarma, encendiendo un número en un display se 7 segmentos, corres-

pondiente a la entrada activada. Así si la puerta es la número 1, se encenderá el display con el número 1, si se activa la ventana número tres, en el display aparecerá el número 3. Para todo lo anterior, el cierre de los contactos de los pulsadores A0-A4 actuará como indicación de que se ha actuado sobre una de las entradas. Si esto ocurre se saca al display de 7 segmentos el número correspondiente un número comprendido entre 0 y 4.

BIBLIOGRAFÍA

Angulo J.M., Martín Cuenca, E. y Angulo, I. (1997). Microcontroladores PIC. La solución en un CHIP. Paraninfo-ITP.

LISTA DE COMPONENTES

RESISTENCIAS:

9	220Ω	R6,R7,R8,R9, R10,R11,R12, R13,R14
5	10KΩ	R1,R2,R3, R4,R5

SEMICONDUCTORES:

1	Barra leds	BL1
1	TIL313	DS1

OTROS:

6	Bornas/2c	CN1,CN2,CN3
5	Teclas	SW1,SW2, SW3,SW4, SW5
1	Tira 17 pins	CN4
6	Jumper 3 pins	J3,J4,J5, J6,J7,J8
2	Jumper 2 pins	J1,J2

```
;*****************************************************************
*******************************
; Programa RESIS000.ASM        Fecha : 10 - Septiembre - 97
*
; Este programa saca al display de 7 segmentos un número comprendido
*
; entre 0 y 4, dependiendo de la tecla pulsada.
;
;
; Revisión : 0.0               Programa para PIC16C84 y PIC16F84
*
; Velocidad del Reloj: 4 MHz   Reloj Instrucción: 1 MHz = 1 uS
*
; Perro Guardián :  deshabilitado   Tipo de Reloj : XT
*
; Protección del código : OFF
*
;*****************************************************************
*******************************
 IGUALDADES
*****************************************************
; ********** Igualdades que designa los destinos
*********************************
w       EQU  0      ;El resultado se guarda en w
f       EQU  1      ;El resultado se guarda en el registro
; ********** Igualdades de la CPU y del mapa de memoria
**********************
PORTA   EQU  05h  ; Puerta A
PORTB   EQU  06h  ; Puerta B
TRISA   EQU  05h  ; Registro Triestado Puerta A
TRISB   EQU  06h  ; Registro Triestado Puerta B
STATUS  EQU  03h  ; Registro Status
RP0     EQU  05h  ; Bit RP0 del registro de STATUS
cero    EQU  05fh   ;salida al puerto B para mostrar 0
uno     EQU  06h  ;salida al puerto B para mostrar 1
dos     EQU  03bh  ;salida al puerto B para mostrar 2
tres    EQU  02fh  ;salida al puerto B para mostrar 3
cuatro  EQU  066h  ;salida al puerto B para mostrar 4
;
*****************************************************************
*************************
```

```
                ORG   00h             ; Dirección del vector de Reset
                GOTO  Inicio          ; Comienza el programa después
                                      ; del vector de interrupción
                ORG   05h             ; Una posición detrás vector de Int.
Inicio          BSF   STATUS,RP0      ; Selecciona la página 1 de la memoria
                clrf  TRISB           ; puerto B como salida
                movlw 0ffh            ; coloca en el acumulador el valor 255
                movwf TRISA           ; puerto A como entrada
                bcf   STATUS,RP0      ; seleccionamos el banco de registros 0
bucle           btfss PORTA,0         ; ¿Es 1 el bit 0 de la puerta A? Si -> salta
                goto  muestra0        ; el bit 0 de la puerta A es 0 -> pinta 0
                btfss PORTA,1         ; ¿Es 1 el bit 1 de la puerta A? Si -> salta
                goto  muestra1        ; el bit 1 de la puerta A es 0 -> pinta 1
                btfss PORTA,2         ; ¿Es 1 el bit 2 de la puerta A? Si -> salta
                goto  muestra2        ; el bit 2 de la puerta A es 0 -> pinta 2
                btfss PORTA,3         ; ¿Es 1 el bit 3 de la puerta A? Si -> salta
                goto  muestra3        ; el bit 3 de la puerta A es 0 -> pinta 3
                btfss PORTA,4         ; ¿Es 1 el bit 4 de la puerta A? Si -> salta
                goto  muestra4        ; el bit 4 de la puerta A es 0 -> pinta 4
                clrf  PORTB           ; no se ha pulsado ningún botón, no hay
                                      ; salida
                goto  bucle           ; empezamos otra vez el test de los
                                      ; botones
muestra0        movlw cero            ; pone en el acumulador la definición
                                      ; de 0
                goto  muestra         ; actualiza la salida al display
muestra1        movlw uno             ; pone en el acumulador la definición
                                      ; de 1
                goto  muestra         ; actualiza la salida al display
muestra2        movlw dos             ; pone en el acumulador la definición
                                      ; de 2
                goto  muestra         ; actualiza la salida al display
muestra3        movlw tres            ; pone en el acumulador la definición
                                      ; de 3
                goto  muestra         ; actualiza la salida al display
muestra4        movlw cuatro          ; pone en el acumulador la definición
                                      ; de 4
muestra         movwf PORTB           ; pasa el contenido del acumulador
                                      ; al puerto B
                goto  bucle           ; inicia el bucle de test
                end                   ; fin de programa
```

El segundo experimento consiste en la escritura de un programa cuya misión, es sacar al display de 7 segmentos un número aleatorio comprendido entre 0 y 9, cuando se pulse la tecla A0.

```
;*****************************************************************************
*****
; Programa RESIS001.ASM        Fecha : 10 - Septiembre - 97          *
; Este programa saca un número aleatorio  de 0 a 9 al display        *
; de 7 segmentos conectado a  la PuertaB                             *
; Revisión : 0.0               Programa para PIC16C84 y PIC16F84     *
; Velocidad del Reloj: 4 MHz        Reloj Instrucción: 1 MHz = 1 uS   *
; Perro Guardián :  deshabilitado   Tipo de Reloj : XT                *
; Protección del código : OFF                                        *
;*****************************************************************************
;************* IGUALDADES ****************************************************
; ********** Igualdades que designa los destinos *********************************
w           EQU   0              ; El resultado se guarda en w
f           EQU   1              ; El resultado se guarda en el registro
; *********** Igualdades de la CPU y del mapa de memoria **********************
PORTA       EQU   05h    ; Puerta A
PORTB       EQU   06h    ; Puerta B
TRISA       EQU   05h    ; Registro Triestado Puerta A
TRISB       EQU   06h    ; Registro Triestado Puerta B
STATUS      EQU   03h    ; Registro Status
RP0         EQU   05h    ; Bit RP0 del registro de STATUS
Z           EQU   02h    ; Bit Z del registro de estado
cero        EQU   05fh          ; salida al puerto B para mostrar 0
uno         EQU   06h    ; salida al puerto B para mostrar 1
dos         EQU   03bh   ; salida al puerto B para mostrar 2
tres        EQU   02fh   ; salida al puerto B para mostrar 3
cuatro      EQU   066h   ; salida al puerto B para mostrar 4
cinco       EQU   06dh          ; salida al puerto B para mostrar 5
seis        EQU   07ch          ; salida al puerto B para mostrar 6
siete       EQU   07h           ; salida al puerto B para mostrar 7
ocho        EQU   07fh          ; salida al puerto B para mostrar 8
nueve       EQU   067h          ; salida al puerto B para mostrar 9
numero      EQU   0ch           ; donde se almacena el número aleatorio
;
*****************************************************************************
            ORG   00h           ; Dirección del vector de Reset
            GOTO  Inicio        ; Comienza el programa después
                                ; del vector de interrupción
            ORG   05h           ; Una posición detrás vector de Int.
```

```
;*****************************************************************************
***
;Rutina que proporciona el valor de salida para el puerto B, para la          *
;codificación de los números del 0 al 9.                                      *
;entrada: en el acumulador (w) un número de 0 a 9                             *
;salida:  en el acumulador (w) se pasa la salida para el puerto B             *
;*****************************************************************************
****
codigo      addwf  PCL,f          ; suma al contador bajo de programa
                                  ; el acumulador
            retlw  cero           ; sale y guarda en w el código del cero
            retlw  uno            ; sale y guarda en w el código del uno
            retlw  dos            ; sale y guarda en w el código del dos
            retlw  tres           ; sale y guarda en w el código de tres
            retlw  cuatro         ; sale y guarda en w el código de cuatro
            retlw  cinco          ; sale y guarda en w el código de cinco
            retlw  seis           ; sale y guarda en w el código de seis
            retlw  siete          ; sale y guarda en w el código de siete
            retlw  ocho           ; sale y guarda en w el código de ocho
            retlw  nueve                  ; sale y guarda en w el código de nueve
Inicio      bsf    STATUS,RP0     ; Selecciona la página 1 de la memoria
            clrf   TRISB                  ; puerto B como salida
            movlw  0ffh                   ; coloca en el acumulador el valor 255
            movwf  TRISA                  ; puerto A como entrada
            bcf    STATUS,RP0             ;seleccionamos el banco de registros 0
            clrf   numero                 ; pone a cero la variable
bucle       btfss  PORTA,0                ; ¿Es 1 el bit 0 de la puerta A? Si -> salta
            goto   muestra                ; el bit 0 puerta A es 0->pinta numero
            incf   numero,f               ; se incrementa el número aleatorio
            movf   numero,w               ; se pasa al acumulador el número
            sublw  0Ah                    ; se compara con 10
            btfsc  STATUS,Z               ; si no son iguales -> salta
            clrf   numero                 ; si son iguales ->borramos el número
            goto   bucle          ; empezamos otra vez el bucle
muestra     movf   numero,0               ; pasamos el número al acumulador (w)
            call   codigo         ; llamamos a la rutina código
            movwf  PORTB                  ; el código que corresponda al número
                                          ; se muestra
            goto   bucle          ; vamos al bucle principal
            end                           ; fin de programa
```