

## Curso de Aplicaciones con Microcontroladores PIC (V).

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

### Teoría (V)

Se describen en este capítulo, dos registros especiales pertenecientes a la memoria de datos, el banco de registros de funciones especiales *RFE* (ver figura 2 Capitulo 1. nº 161 y figura 2 Capitulo 4 nº 164).

#### REGISTRO DE ESTADO - STATUS

El registro de estado (*STATUS*) contiene varios bits de estado de la unidad central, el estado aritmético de la ALU, el estado del RESET y un par de bits de selección de página denominados RP1 y RP0. Ocupa la posición \$03 del Banco 0 y la \$83 del Banco 1. Este registro puede escribirse o leerse como los demás, pero algunos bits no pueden ser modificados si se intenta escribirlos. A continuación se describe el significado de los bits que lo componen:

**Bit 0 - C (Carry) - Acarreo** en el 8º bit. Es el bit de acarreo en operaciones de suma AADWF y ADDLW y sustracción SUBWF y SUBLW. También lo emplean las instrucciones RLF y RRF de rotación de bits.

1 = Acarreo en la suma en el bit de mayor peso  
0 = Acarreo en la resta y no en la suma

**Bit 1 - DC (Digit Carry) . Acarreo** en el 4º bit de menos peso. Este bit es el de acarreo de dígito. Su empleo es útil entre otros en aritmética BCD ( prácticas en este capítulo). Se pone a 1 si se produce acarreo en el 4º bit del grupo de 4 bits de menos peso.

**Bit 2 - Z ( Zero). Cero.**

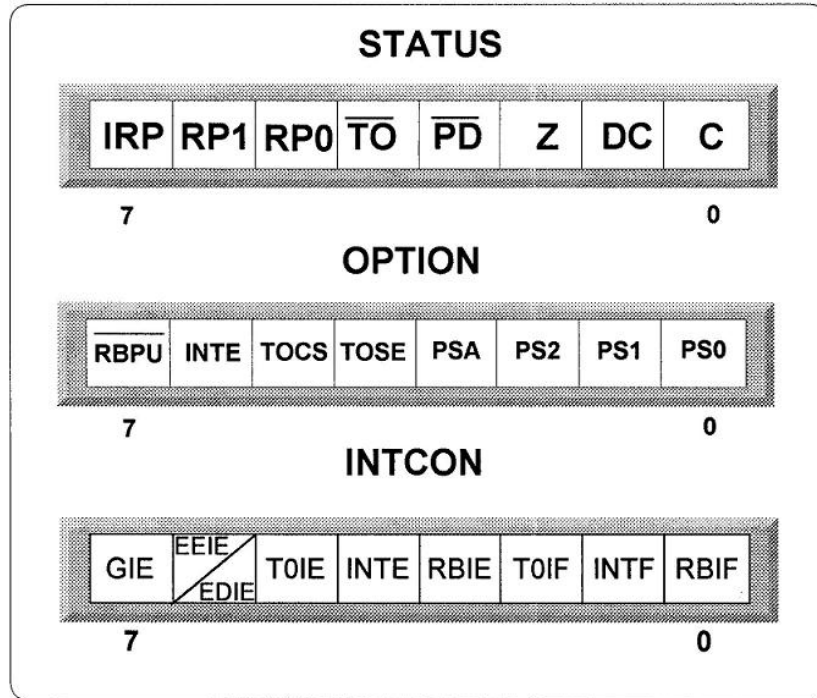
1 = El resultado de un operación aritmética o lógica es 0  
0 = El resultado es distinto de 0

**Bit 3  $\overline{PD}$  (Power Down).**

1 = Tras conectar la alimentación  $V_{DD}$  o ejecutar CLRWDT  
0 = Al ejecutar la instrucción SLEEP

**Bit 4  $\overline{TO}$  ( Time Out )**

1 = Tras conectar la alimentación VDD o ejecutar CLRWDT o SLEEP.  
0 = Desbordamiento del perro guardián WDT



**Figura 1.-** Representación en detalle varios de los registros de funciones especiales. Ver Figuras nº 2 - Capítulos 1 y 4.

**Bits 5 y 6 RP0 RP1.(Register Bank Select)** Selección de página o banco de la memoria de programa con direccionamiento directo. Cada página contiene 128 bytes. En el 16F8X solo se usa RP0, RP1 debe mantenerse a 0.

- 00 = Banco 0 ( 00h - 7Fh )
- 01 = Banco 1 ( 80h - FFh )
- 10 = Banco 2 ( 100h - 17Fh )
- 11 = Banco 3 ( 180h - 1FFh )

**Bit 7 IRP (Register Bank Select).** Selección de bancos para el direccionamiento indirecto. Está previsto para su uso en un futuro direccionamiento de página indirecto, pero no se emplea en el PIC16F84, por lo que debe estar a 0.

- 0 = Banco 0, 1 (00h - FFh)
- 1 = Banco 2, 3 (100h - 1FFh)

Ha de tener la precaución de no modificar ninguno de los bits durante la lectura del registro de estado. El empleo de las instrucciones tipo bit BTFSS ó BTFSC no los afectan, pero por ejemplo la instrucción MOVF puede afectar la bandera Z. La escritura del registro de estado mediante las instrucciones BSF, BCF o MOVE no lo afectan.

## REGISTRO DE OPCIONES : OPTION

Como indica su nombre este registro se emplea para programar las opciones del temporizador TMR0, el tipo de flanco con el que se detecta una interrupción y la activación de las resistencias de polarización del Puerto B.

Ocupa la posición \$81 de la página 1 del banco de registros. Con objeto de mantener la compatibilidad con la gama baja de PICs, debe escribirse usando la instrucción especial OPTION. Esta instrucción carga el contenido de W en el registro OPTION. Ejemplo :

```
MOVLW    11010111b ; Asignación
OPTION    ; Envía patrón de bits al registro OPTION
```

También puede emplearse las siguientes instrucciones :

```
MOVLW    11010111b ; Asignación
MOVWF    OPCION    ; Envía patrón bits al registro OPTION
```

en este último caso, se debe definir con anterioridad OPCION el las igualdades con la dirección del registro OPTION de la siguiente forma

```
OPCION    EQU    81h ; Registro de Opciones
```

**Bits 0, 1 y 2 - PS0, PS1 y PS2.** (*Prescaler Rate Select Bits*). Asigna la tasa del valor del divisor de frecuencias, y difiere dependiendo del predivisor que se haya asignado al TMR0 o al WDT.

PS2	PS1	PS0	TMR0	WDT
0	0	0	1:2	1:1
0	0	1	1:4	1:2
0	1	0	1:8	1:4
0	1	1	1:16	1:8
1	0	0	1:32	1:16
1	0	1	1:64	1:32
1	1	0	1:128	1:64
1	1	1	1:256	1:128

**Bit 3 PSA** (*PreScaler Assignment*). Como su nombre indica, se usa para la Asignación del divisor de Frecuencias, si se coloca PSA = 1, el divisor de frecuencias se le asigna al WDT, si PSA= 0 el divisor se le asigna al TMR0.

**Bit 4 T0SE** (*Timer 0 Signal Edge*). Tipo de flanco activo del T0CKI. Si este bit se coloca a 0, el TMR0 se incrementa con flanco ascendente de la señal que se aplica a la patita A4/T0CKI, y si pone a 1 el TMR0 se incrementa con el flanco descendente.

**Bit 5 T0CS (Timer 0 Signal Source).** Fuente de reloj para el TMR0. Colocando este bit a 1 TMR0 se usa en modo contador de los pulsos introducidos a través de A4/T0CKI, y si se coloca a 0 TMR0 se usa en modo temporizador haciendo uso de los pulsos de reloj internos ( $F_{osc}/4$ ).

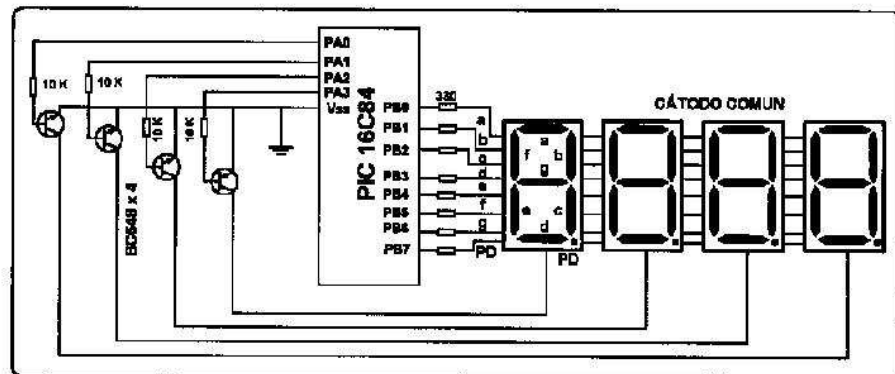
**Bit 6 INTDEG (INTerrupt EDGE).** Selecciona el tipo de Flanco para la Interrupción. Este bit indica el tipo de flanco de la señal externa que ha de provocar una interrupción en la patita RB0/INT. Si está a 1 la interrupción es producida por el flanco ascendente y si está a 0 la interrupción se produce en el flanco descendente.

**Bit 7 RBPU (RB Pull Up).** Conexión de las resistencias de polarización del Puerto B. Si se pone a 1 todas las resistencias son desconectadas, mientras que si se coloca a 0 se activan de forma individual. Se conectan todas cuando el Puerto B actúa como entrada por tanto si se pone a 0.

## PRACTICAS (IV)

### PRESENTACIÓN DE DATOS Y DISPLAYS DE 7 SEGMENTOS

Los datos obtenidos por el microcontrolador deben presentarse al usuario, y para ellos existen diversas soluciones. En este capítulo se explicará el uso de presentadores (displays) de 7 segmentos.



**Figura 2.-** Diagrama eléctrico de circuito de gobierno multiplexado de 4 displays de 7 segmentos haciendo uso de un PIC 16C84. Necesita 12 líneas de E/S.

La presentación de datos numéricos, puede realizarse fácilmente mediante el uso de displays de 7 segmentos. La solución mínima desde el punto de vista del número de componentes es la que se muestra en la figura 2. En ella solo se emplean algunos transistores para gobernar los 4 displays de 7 segmentos. Sin embargo esta solución tiene la desventaja de consumir la mayoría de las patitas de E/S del microcontrolador que pueden ser necesarias para otros menesteres. En la figura 2, se han empleado un total de 12 patitas de E/S. La programación hace uso de la técnica de multiplexado por software. Un programa para comprobar el funcionamiento del ejemplo de la figura 2 se encuentra en mi libro, reseñado en la bibliografía.

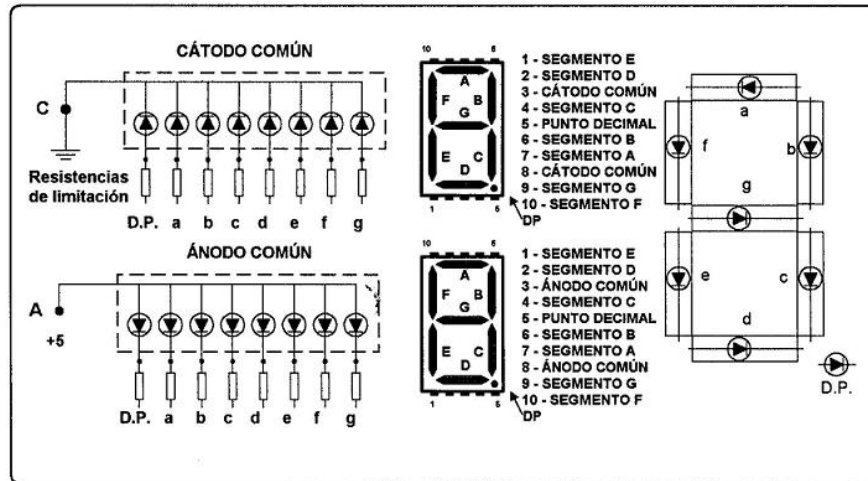


Figura 3.- Las dos diferentes disposiciones de los presentadores ( displays ) de 7 segmentos. Arriba cátodo común. Abajo ánodo común.

La activación de los displays de 7 segmentos, es igual que el encendido de los diodos LEDs, ya que como puede comprobar en la figura 3, no son mas que un determinado número de diodos LEDs colocados de una forma particular. Existen dos tipos de displays de 7 segmentos los de *ánodo común* y los de *cátodo común*. Si el display es de grandes dimensiones y su consumo es elevado existen soluciones para amplificar la señal.

Sea cual sea la solución adoptada, en todos los casos se recurre al multiplexado, en unos casos mediante software y en otros por hardware como se expondrá mas adelante.

Para tratar de emplear el menor número posible de patitas de E/S del microcontrolador pueden elegirse entre varias soluciones de las que presentamos algunas continuación y otras en próximos capítulos.

### DECODIFICADOR BCD - 7 SEGMENTOS

Para tratar de emplear un menor numero de patitas de microcontrolador podemos elegir un circuito integrado decodificador BCD (*Binary Coded Decimal - Decimal Codificado en Binario*) a 7 segmentos. Si se emplean tres o cuatro contadores binarios para representar un número decimal de tres o cuatro dígitos, cada uno debe representar los números del 0 al 9, que corresponden a los binarios 0000 a 1001. Una vez alcanzado el número 1001, la siguiente cuenta debe hacer que el contador se ponga a 0000, e incrementar en uno el siguiente contador de la cadena; es decir realizar una operación de acarreo de uno. cada número BCD es el equivalente directo de un número decimal de un solo dígito. El valor máximo del contador BCD es 1001. No hay número BCD mayores, no serian válidos ni tendrían significado.

Para representar los números del 0 al 9 por lo tanto se necesitan 4 bits de información como se ve en la Tabla.

Los principios en que se basan los indicadores de 7 segmentos (displays) son útiles para representar los números del 0 al 9 y muchas letras como A, a, C, d, E, F, H, h, Y, L, O, P, r, S, U. Permite por lo tanto la representación de números hexadecimales. Los 7 segmentos del indicador se denominan a, b, c, d, e, f, g, y d.p. (punto decimal), y su disposición se presenta en la figura.

DECIMAL	BINARIO				B.C.D.				HEXADECIMAL	BARRA ILUMINADA
	D	C	B	A	D	C	B	A		
0	0	0	0	0	0	0	0	0	0	a, b, c, d, e, f
1	0	0	0	1	0	0	0	1	1	b, c
2	0	0	1	0	0	0	1	0	2	a, b, d, e, g
3	0	0	1	1	0	0	1	1	3	a, b, c, d, g
4	0	1	0	0	0	1	0	0	4	b, c, f, g
5	0	1	0	1	0	1	0	1	5	a, c, d, f, g
6	0	1	1	0	0	1	1	0	6	a, c, d, e, f, g
7	0	1	1	1	0	1	1	1	7	a, b, c
8	1	0	0	0	1	0	0	0	8	a, b, c, d, e, f, g
9	1	0	0	1	1	0	0	1	9	a, b, c, d, f, g
10	1	0	1	0	-	-	-	-	A	a, b, c, e, f, g
11	1	0	1	1	-	-	-	-	B	c, d, e, f, g
12	1	1	0	0	-	-	-	-	C	a, d, e, f
13	1	1	0	1	-	-	-	-	D	b, c, d, e, g
14	1	1	1	0	-	-	-	-	E	a, d, e, f, g
15	1	1	1	1	-	-	-	-	F	a, e, f, g

Estos circuitos integrados convierten un dato dado en formato BCD a 7 segmentos, entre ellos encontramos el IC7447 para gobernar displays de *ánodo común* y el CD4511 que es su equivalente CMOS (de menor consumo y funcionamiento entre 3 y 15 voltios) pero para el control de displays de *cátodo común*. Estos circuitos decodifican cualquier cifra comprendida entre 0 y 9 dada en formato BCD a formato de 7 segmentos.

La descripción de su patillaje es la siguiente:

Cuatro entradas del dato numérico A, B, C, D en código BCD y siete salidas para los segmentos a, b, c, d, e, f, g.

Patita LT (3) *Test de lamparas*. Si es colocada a nivel bajo (0) enciende todos los segmentos del display para realizar su comprobación, por lo tanto indicando la cifra 8. En funcionamiento normal se coloca a nivel 1.

Patita BL (4) *Blanking*. Si es colocada a nivel 0 se apagan todos los segmentos, en funcionamiento normal debe colocarse a 1.

Patita LE (5) *Latch Enable*. Memoriza el dato, cuando la patita es colocada a nivel bajo.

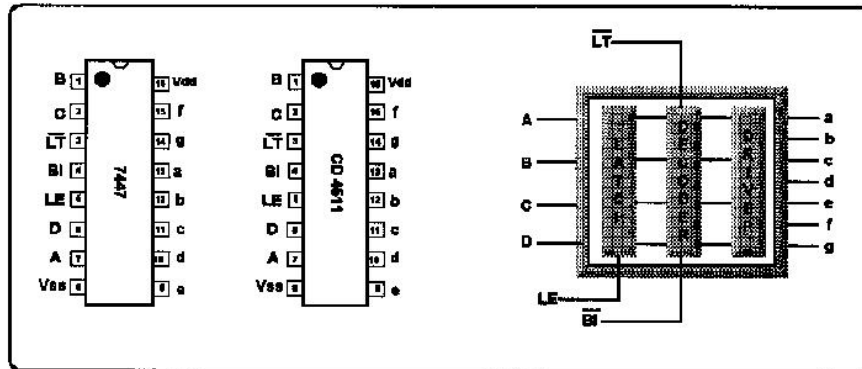


Figura 4.- Encapsulado de los decodificadores BCD - 7 segmentos. Detalles de su estructura interna.

Un pariente de estos es el CA3161E que suele hacer pareja con el conversor A/D CA3162E para la presentación de tres cifras multiplexadas.

Si elegimos esta solución, el circuito eléctrico queda como el presentado en la figura 3. En este caso el microcontrolador no entrega directamente los códigos de 7 segmentos sino que entrega datos numéricos en formato BCD. Al no gobernar directamente los displays el microcontrolador no tendrá ningún problema de corriente

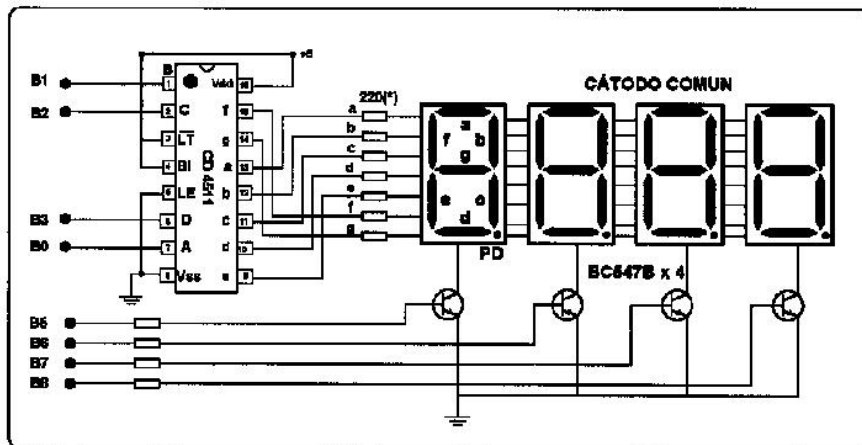


Figura 5.- Esquema eléctrico del Módulo - 03

Haciendo uso de cualquiera de estos circuitos, solo necesitamos 4 patitas para la señal BCD y otras tantas como displays de 7 segmentos deseemos activar, es decir se necesitan 7 patitas para activar 3 displays y 8 para activar 4. Aunque hemos reducido el número de patitas necesarios de 11 a 8, disponiendo ahora de tres patitas del

microcontrolador para otros usos, si es necesario un mayor ahorro de patitas puede disponerse de otras soluciones.

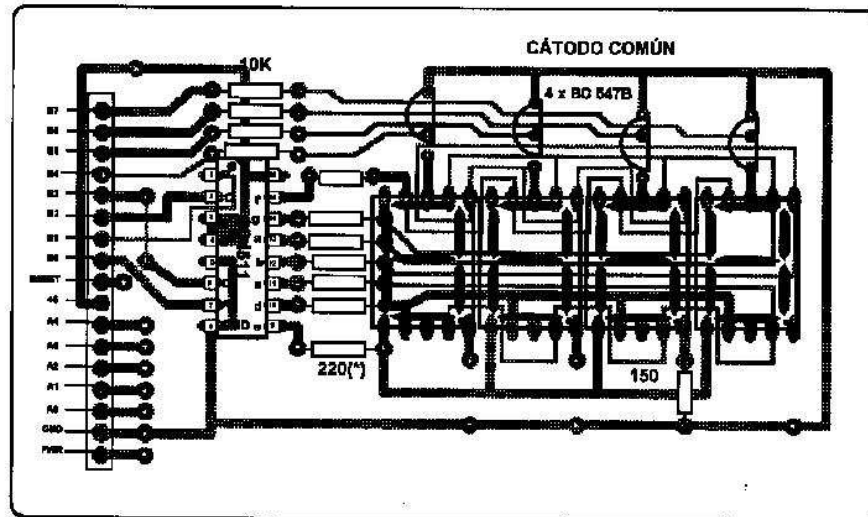


Figura 6.- Disposición de los componentes del Módulo - 03.

Ha de tener en cuenta que cada segmento consume de 10 a 20 mA, por lo que calculando una media de 15 mA, hay un consumo de 100 mA para la cifra 8. Al lado de estos consumos los consumos de los TTL 7447 o de los CMOS 4511 son bastante pequeños. Por otro lado si pensamos en que hay 4 dígitos, 8888 tendría un consumo de  $4 \times 100 \text{ mA} = 400 \text{ mA}$  !. Esto no ocurre, porque los displays están multiplexados, y solo está en realidad encendido uno cada vez, lo que ocurre es que la velocidad de encendido y apagado es tan alta que la vista no puede apreciar el truco ( se podría detectar haciendo uso de un flash estroboscópico) y siempre ve todos los displays encendidos, aunque en realidad solo hay uno encendido y el resto apagados.

La resistencias pueden tener un valor que va desde 220 ohmios hasta 50, e incluso pueden eliminarse, aunque si se realiza esto hay que prestar mucha atención al consumo. Con un valor de resistencia alto la luminosidad es baja, debido al multiplexado, que hace que cada uno de los displays esté encendido muy poco tiempo; por esta misma razón es posible hacer pasar mayor cantidad de corriente por los segmentos, es decir mas de 20 mA, dado de están continuamente apagándose y encendiéndose, esta sobrealimentación no los daña, y aumenta su luminosidad, pero también el consumo. Por tanto ha de elegir estas resistencias realizando un compromiso entre luminosidad y consumo.

Con el *Módulo - 03* se van a realizar dos experiencias.

**1ª Experiencia .-** Aquí aprenderá usar este nuevo modulo, realizando un programa que le muestra como iluminar tres dígitos. El programa se encarga de mostrar en un



marcador compuesto por tres dígitos con valores numéricos, que varían con el tiempo.

2ª Experiencia .- Como regalo navideño y para pasar el crudo invierno, con este Módulo, un par de resistencias y un par de pulsadores, les hemos diseñado un bingo electrónico, que en España también conocemos como lotería. Así podrá pasar los rigores del invierno jugando con su familia a un juego que aprendimos de nuestros abuelos.

Se presenta un número comprendido entre 0 y 90 sin repetición. Para sacar cada número, basta con pulsar el botón rojo y para iniciar una nueva partida el botón negro.

```

*****
; Programa RESIS005.ASM      Fecha : 10 - Octubre - 97      *
; Muestra en un marcador compuesto por tres dígitos      *
; valores numéricos, que varían con el tiempo.          *
;                                                           *
; Revisión : 1.0          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 *****
; ***** Direcciones de los registros de uso específico del PIC 16C/F84 *****
PORTA  EQU 05h    ; Puerta A
PORTB  EQU 06h    ; Puerta B
TRISA  EQU 05h    ; Registro Triestado Puerta A
TRISB  EQU 06h    ; Registro Triestado Puerta B
INTCON EQU 0Bh    ; Registro de Control de Interrupciones
TIMER0 EQU 01h    ; Registro Timer0
OPCION EQU 81h    ; Registro de opciones
STATUS EQU 03h    ; Registro Status
RP0    EQU 05h    ; Bit RP0 del registro de STATUS
C      EQU 00h    ; Bit de acarreo
Z      EQU 02h    ; Bit de cero
*****
; Posiciones de memoria utilizadas por el programa
; *****
ORG 0Ch
Contador1 res 1    ; Utilizada por la rutina de retardo
Contador2 res 1    ; Utilizada por la rutina de retardo
Numero    res 1    ; Guarda el número a mostrar
; *****
ORG 00h          ; Dirección del vector de Reset
GOTO Inicio     ; Comienza el programa después

```

R34

```

; del vector de interrupción
; Una posición detrás vector de Int.
ORG 05h
;*****
;Produce un retardo
;entrada nada
;salida nada
;variables utilizadas: Contador1 y Contador2
;*****
DELAY CLRF Contador1 ;Pone a cero el contenido de Contador1
MOVW 0F0h ;Almacena en W el valor 0A0h
MOVWF Contador2 ;Guarda el valor de W en Contador2
Loop1 DECFSZ Contador1,1 ;Decrementa Contador1 y salta si es cero
GOTO Loop1 ;Cerramos el primer bucle
DECFSZ Contador2,1 ;Decrementa Contador2 y salta si es cero
GOTO Loop1 ;Cerramos el segundo bucle
RETURN ;Salimos de la subrutina

; ***** Programa Principal *****
Inicio BSF STATUS,RP0 ; Selecciona la página 1 de la memoria
CLRF TRISB ;El puerto B como salida
BCF STATUS,RP0 ;Seleccionamos el primer banco
CicloNuevo CLRF Numero ;El primer número a mostrar será el 0
MOVF Numero,0 ;Carga W con el número actual
MOVWF PORTB ;Pone en el marcador el número
BSF PORTB,4 ;Activa el dígito más significativo
CALL DELAY ;Lo presenta un periodo de tiempo
BCF PORTB,4 ;Desactiva el dígito más significativo
MOVF Numero,0 ;Carga W con el número actual
MOVWF PORTB ;Pone en el marcador el número
BSF PORTB,5 ;Activa el dígito central
CALL DELAY ;Lo presenta un periodo de tiempo
BCF PORTB,5 ;Desactiva el dígito central
MOVF Numero,0 ;Carga W con el número actual
MOVWF PORTB ;Pone en el marcador el número
BSF PORTB,6 ;Activa el dígito menos significativo
CALL DELAY ;Lo presenta un periodo de tiempo
BCF PORTB,6 ;Desactiva el dígito menos significativo
INCF Numero,0 ;Incrementa el número a mostrar
ANDLW 0fh ;Nos quedamos con los 4 bits bajos
MOVWF Numero ;Actualizamos el próximo número
GOTO CicloNuevo ;Iniciamos otro ciclo

end

```

Experiencia nº 2 :

Bibliografía :

J.M. Angulo y E.M. Cuenca (1997) Microcontroladores PICs. La solución en un CHIP. ITP. Paraninfo.