SISTEMAS ELECTRÓNICOS Y AUTOMÁTICOS
PRACTICAS DE MICROCONTROLADORES PIC

PRÁCTICA 9:

INTERRUPCIONES (I)

- ***Importancia de las interrupciones***
- ***Ejercicios y ejemplos de las interrupciones: INT y RBI***

---

1. Importancia de las interrupciones

Una interrupción consiste en un mecanismo por el cual un evento interno o externo puede interrumpir la ejecución de un programa en cualquier momento. A partir de entonces se produce automáticamente un salto a una subrutina de atención a la interrupción, ésta atiende inmediatamente el evento y retoma luego la ejecución del programa exactamente donde estaba en el momento de ser interrumpido, continuando su tarea justo donde la dejó. La interrupción tiene la característica de la inmediatez, nace de la necesidad de ejecutar una subrutina en el instante preciso y, por tanto, se considera su intervención urgente.

Las interrupciones constituyen el mecanismo más importante para la conexión del microcontrolador con el exterior ya que sincroniza la ejecución de programas con los acontecimientos externos. Esto es muy útil, por ejemplo, para el manejo de dispositivos de entrada que requieren de una atención inmediata, tales como detección de pulsos externos, recepción de datos, activación de pulsadores, etc.

El funcionamiento de las interrupciones es similar al de las subrutinas, de las cuales se diferencian, principalmente, en los procedimientos que las ponen en marcha. Así como las subrutinas se ejecutan cada vez que en el programa aparece una instrucción CALL, las interrupciones se ponen en marcha al aparecer en cualquier instante un evento externo al programa, por lo tanto, una interrupción se activa por un mecanismo hardware.

El PIC16F84 dispone de 4 posibles fuentes de interrupción:

- **Interrupción INT**. Por activación del pin RB0/INT.
- **Interrupción RBI**. Por cambio de estado en una o varias de las 4 líneas de más peso del puerto B (RB7:RB4).
- **Interrupción T0I**. Por desbordamiento del Timer 0.
- **Interrupción EEI.** Por la finalización de la escritura en la EEPROM de datos

El PIC16F877A también dispone de estas fuentes de interrupción y su configuración y funcionamiento es igual a la del PIC16F84A.

Cuando se produce cualquiera de los sucesos indicados anteriormente se origina una petición de interrupción que, si se acepta, origina el siguiente mecanismo hardware:

1º Salva el valor actual del PC guardando su contenido en la pila
2º El bit GIE del registro INTCON es puesto a cero, lo que prohíbe cualquier otra interrupción.
3º El PC se carga con el valor 0004h, que es la posición de vector de interrupción.
4º Comienza a ejecutarse el programa de atención a la interrupción que se encuentra a partir de la dirección 0004h.

En el PIC 6F84, los bits de control localizados en el registro **INTCON** habilitan y configura las interrupciones. Cada causa de interrupción actúa con dos flags (banderas):

- un flag indica si se ha producido o no la interrupción: **T0IF**, **INTF**, **RBIF** y **EEIF**.
- y el otro flag funciona como permiso o prohibición de la interrupción en sí: **T0IE**, **INTE**, **RBIE**, **EEIE** y **GIE**.

Hay un único **vector de interrupción** en la dirección **0004h.**

La instrucción **RETFIE** utilizada al final de la subrutina de interrupción es idéntica a un retorno de subrutina **RETURN**, pero además, coloca automáticamente a "**1**" el bit **GIE**, volviendo a habilitar las interrupciones.

2. Interrupción externa INT

La fuente de interrupciones externa INT es muy importante para atender eventos externos en tiempo real. Cuando la línea RB0/INT se hace una petición de interrupción el bit INTF del registro INTCON se pone a "1" de forma automática y, si el bit GIE está a "1", se pone en marcha el mecanismo ya comentado de la interrupción.

Mediante el bit INTDEG del registro OPTION es seleccionado el flanco activo de RB0/INT, ya que con éste puesto a "1" el flanco activo es el ascendente y cuando está a "0" el flanco activo es el descendente.

El programa de atención a la interrupción antes de regresar al programa principal debe borrar el flag INTF, puesto que en caso contrario al ejecutar la instrucción de retorno de interrupción RETFIE se volverá a desarrollar el mismo proceso de interrupción.

A continuación se expone un programa ejemplo (*Int_INT_02.asm*) para la tarjeta EASYPIC, cuyo pulsador (RB0) producirá una interrupción externa INT al actuar sobre él. Es un ejemplo típico de lectura de entradas digitales aplicable a multitud de proyectos. En el programa ejemplo (*Int_INT_02.asm*), cada vez que presiona el pulsador conectado al pin RB0/INT se incrementa un contador que es visualizado en el módulo LCD. La lectura del pulsador se realiza mediante el uso de la interrupción INT.

El ejemplo es idéntico al denominado igual en el libro[1], pero con la diferencia que, para su funcionamiento sobre la EasyPIC, se considera que el pulsador conectado al pin RB0 es activo a nivel alto, es decir, cuando está pulsado RB0=5V.

EJERCICIO 1:

a) Comprueba el funcionamiento del programa ***Int_INT_02.asm*** en la tarjeta EasyPIC.

EJERCICIO 2:

a) Modifica el programa ***Int_INT_02.asm*** de modo que en la pantalla del módulo LCD se visualice un mensaje en movimiento cada vez que se presione el pulsador del pin RB0. Para ello utiliza la subrutina "**LCD_Movimiento**" que visualiza mensajes en movimiento, de la librería LCD_MENS.INC.
b) Comprueba su funcionamiento en la tarjeta EasyPIC.

[1] "Microcontrolador PIC16F84, Desarrollo de proyectos", E. Palacios, F. Remiro, L.J. López. Ra-Ma, 2004

```
;********************************** Int_INT_02.asm************************************
;
;
; Cada vez que presiona el pulsador conectado al pin RB0/INT se incrementa un contador
; que es visualizado en el módulo LCD. La lectura del pulsador se hará mediante
; interrupciones.
;
; Se supone que el pulsador conectado al pin RB0 en la EasyPIC es activo a nivel alto,
;es decir, cuando está pulsado RB0=5V.
;
; ZONA DE DATOS *****************************************************************

       __CONFIG   _CP_OFF &  _WDT_OFF & _PWRTE_ON & _HS_OSC
       LIST      P=16F877A
       INCLUDE  <P16F877A.INC>

       CBLOCK  0x20
       Contador                        ; El contador a visualizar.
       ENDC

#DEFINE  Pulsador     PORTB,0       ; Línea donde se conecta el pulsador.

; ZONA DE CÓDIGOS ***************************************************************

       ORG    0
       goto   Inicio
       ORG    4                     ; Vector de interrupción.
       goto   ServicioInterrupcion
Inicio
       call   LCD_Inicializa
       bsf    STATUS,RP0            ; Acceso al Banco 1.
       bsf    Pulsador              ; La línea RB0/INT se configura como entrada.
       bsf    OPTION_REG,INTEDG     ; Interrupción INT se activa por flanco de SUBIDA.
       bcf    STATUS,RP0            ; Acceso al Banco 0.
       clrf   Contador                    ; Inicializa el contador y lo visualiza.
       call   VisualizaContador
       movlw  b'10010000'           ; Habilita la interrupción INT y la general.
       movwf  INTCON

; La sección "Principal" es de mantenimiento. Sólo espera las interrupciones
; en modo de bajo consumo.

Principal
       sleep                        ; Pasa a modo de bajo consumo o reposo.
       goto   Principal

; Subrutina "ServicioInterrupcion" ----------------------------------------------------
-
;
; Subrutina de servicio a la interrupción. Incrementa un contador y lo visualiza.
;
ServicioInterrupcion
       call   Retardo_20ms          ; Espera a que se estabilice el nivel de tensión.
       btfss  Pulsador                    ; Comprueba si es un rebote.
       goto   FinInterrupcion             ; Era un rebote y por tanto sale.
       incf   Contador,F            ; Incrementa el contador y lo visualiza.
VisualizaContador
       call   LCD_Linea1
       movf   Contador,W
       call   BIN_a_BCD             ; Se debe visualizar en BCD.
       call   LCD_Byte
FinInterrupcion
       bcf    INTCON,INTF           ; Limpia flag de reconocimiento (INTF).
       retfie                       ; Retorna y rehabilita las interrupciones (GIE=1).

       INCLUDE   <RETARDOS.INC>
       INCLUDE   <BIN_BCD.INC>
       INCLUDE   <LCD_EASY.INC>
       END
```

EJERCICIO 3:

a) El programa ***Int_INT_07.asm*** realiza la siguiente tarea:

"Cada vez que se presiona el pulsador conectado a la línea RB0/INT conmutará el estado de los LEDs conectados a PORTC. Al mismo tiempo en el módulo LCD se visualizará un mensaje desplazándose por la pantalla."

Para ello el programa principal se encarga de enviar el mensaje al módulo LCD, mientras que la subrutina de interrupción se encarga de conmutar el estado de los LEDs.

b) Comprueba su funcionamiento en la tarjeta EasyPIC.

```
;********************************** Int_INT_07.asm************************************
;
;
;
;
; Comprueba el funcionamiento de la interrupción por activación del pin RB0/INT y analiza
; cómo deben guardarse los datos que se corrompen durante el proceso de la llamada a subrutina.
;
; Cada vez que presione el pulsador conectado al pin RB0/INT conmutará el estado de los LEDs
; conectados a PORTC. Al mismo tiempo en el módulo LCD se visualizará un mensaje
; desplazándose por pantalla.
;
;
;
; ZONA DE DATOS ********************************************************************

       __CONFIG   _CP_OFF &  _WDT_OFF & _PWRTE_ON & _HS_OSC

       LIST      P=16F877A
       INCLUDE  <P16F877A.INC>

       CBLOCK  0x20
       ENDC

#DEFINE  Pulsador  PORTB,0          ; Línea donde se conecta el pulsador.
#DEFINE  LUCES    PORTC             ; LEDS EN EL PUERTO PORTC
#DEFINE  LED      PORTC,0

; ZONA DE CÓDIGOS ******************************************************************

       ORG    0
       goto   Inicio
       ORG    4                    ; Vector de interrupción.
       goto   ServicioInterrupcion
Inicio
       call   LCD_Inicializa
       bsf    STATUS,RP0           ; Acceso al Banco 1.
       bsf    Pulsador             ; La línea RB0/INT se configura como entrada.
       clrf LUCES                  ; Se configura como salida.
       bsf    OPTION_REG,INTEDG    ; Interrupción INT se activa por flanco de SUBIDA
       bcf    STATUS,RP0           ; Acceso al Banco 0.
       movlw  b'10010000'          ; Habilita la interrupción INT y la general.
       movwf  INTCON
Principal
       movlw  MensajeLargo         ; Visualiza el mensaje desplazándose por la
       call   LCD_MensajeMovimiento ; pantalla.
       goto   Principal
```

```
; Subrutina "ServicioInterrupcion" ----------------------------------------------------
-
;
; Subrutina de atención a la interrupción. Conmuta el estado del LED.

; Como esta subrutina altera los valores del registro de trabajo W, del STATUS, y de los
; registros R_ContA y R_ContB utilizados en los retardos, habrá que preservar su valor
; previo y después restaurarlo al final.

       CBLOCK
       Guarda_W
       Guarda_STATUS
       Guarda_R_ContA
       Guarda_R_ContB
       ENDC

ServicioInterrupcion
       movwf   Guarda_W              ; Guarda W y STATUS.
       swapf   STATUS,W              ; Ya que "movf STATUS,W" corrompe el bit Z.
       movwf   Guarda_STATUS
       bcf     STATUS,RP0            ; Para asegurarsede que trabaja con el Banco 0.
       movf    R_ContA,W             ; Guarda los registros utilizados en esta
       movwf   Guarda_R_ContA        ; subrutina y también en la principal.
       movf    R_ContB,W
       movwf   Guarda_R_ContB
;
       call    Retardo_20ms
       btfss   Pulsador              ; Comprueba si es un rebote.
       goto    FinInterrupcion              ; Era un rebote y por tanto sale.
       btfsc   LED                   ; Testea el último estado de las luces (en concreto
el  LED de RC0).
       goto    EstabaEncendido
EstabaApagado
       movlw   0xFF
       movwf   LUCES                 ; Estaba apagado y lo enciende.
       goto    FinInterrupcion
EstabaEncendido
       clrf    LUCES                 ; Estaba encendido y lo apaga.
FinInterrupcion
       swapf   Guarda_STATUS,W              ; Restaura el STATUS.
       movwf   STATUS
       swapf   Guarda_W,F            ; Restaura W como estaba antes de producirse
       swapf   Guarda_W,W            ; interrupción.
       movf    Guarda_R_ContA,W      ; Restaura los registros utilizados en esta
       movwf   R_ContA               ; subrutina y también en la principal.
       movf    Guarda_R_ContB,W
       movwf   R_ContB
       bcf     INTCON,INTF           ; Limpia flag de reconocimiento de la interrupción.
       retfie                        ; Retorna y rehabilita las interrupciones.

; "Mensajes" ---------------------------------------------------------------------------
-

Mensajes
       addwf   PCL,F
MensajeLargo
       DT "               "
       DT "Me gusta desarrollar proyectos con PICs."
       DT "                ", 0x00

       INCLUDE   <LCD_MENS.INC>
       INCLUDE   <LCD_EASY.INC>
       INCLUDE   <RETARDOS.INC>
       END
```

2. Interrupción “RBI” por cambio en las líneas RB7:RB4.

Para activar la interrupción por cambio de nivel en los pines RB7:RB4 los bits RBIE y GIE del registro INTCON deben de estar a “1”, en estas condiciones cuando se produce un cambio de nivel en cualquiera de las líneas RB7 a RB4 se activa el flag RBIF del registro INTCON.

A continuación se expone un programa ejemplo (*Int_RBI_01.asm*) para la tarjeta EASYPIC. Este programa comprueba el funcionamiento de la interrupción por cambio de estado de una línea de la parte alta del Puerto B, por ejemplo la RB6. Cada vez que presiona el pulsador conectado al pin RB6 se incrementará un contador que se visualiza en el módulo LCD.

El ejemplo es idéntico al denominado igual en el libro[2], pero con la diferencia que, para su funcionamiento sobre la EasyPIC, se considera que el pulsador conectado al pin RB6 es activo a nivel alto, es decir, cuando está pulsado RB6=5V.

EJERCICIO 4:

a) Comprueba el funcionamiento del programa ***Int_RBI_01.asm*** en la tarjeta EasyPIC.

[2] "Microcontrolador PIC16F84, Desarrollo de proyectos", E. Palacios, F. Remiro, L.J. López. Ra-Ma, 2004

```
;********************************** Int_RBI_01.asm ***********************************
;
; Este programa comprueba el funcionamiento de la interrupción por cambio de estado de una
; línea de la parte alta del Puerto B, por ejemplo la RB6. Cada vez que presiona el pulsador
; conectado al pin RB6 se incrementará un contador que se visualiza en el módulo LCD.
;
; Se supone que el pulsador conectado al pin RB6 en la EasyPIC es activo a nivel alto, es decir,
; cuando está pulsado RB6=5V.

; ZONA DE DATOS *****************************************************************

       __CONFIG   _CP_OFF &  _WDT_OFF & _PWRTE_ON & _HS_OSC
       LIST      P=16F877A
       INCLUDE  <P16F877A.INC>

       CBLOCK   0x20
       Contador
       ENDC

#DEFINE   Pulsador    PORTB,6       ; Línea donde se conecta el pulsador.

; ZONA DE CÓDIGOS ***************************************************************

       ORG    0
       goto   Inicio
       ORG    4
       goto   ServicioInterrupcion
Inicio
       call   LCD_Inicializa
       bsf    STATUS,RP0
       bsf    Pulsador              ; La línea se configura como entrada.
       bcf    STATUS,RP0
       clrf   Contador              ; Inicializa el contador y lo visualiza.
       call   VisualizaContador
       movlw  b'10001000'           ; Habilita la interrupción RBI y la general.
       movwf  INTCON
Principal
       sleep                        ; Pasa a modo de bajo consumo y espera las
       goto   Principal             ; interrupciones.

; Subrutina "ServicioInterrupcion" ----------------------------------------------
;
; Subrutina de atención a la interrupción. Incrementa el contador y lo visualiza.
;
ServicioInterrupcion
       call   Retardo_20ms          ; Para salvaguardar de los rebotes.
       btfss  Pulsador              ; Comprueba si es un rebote.
       goto   FinInterrupcion               ; Era un rebote y por tanto sale.
       incf   Contador,F            ; Incrementa el contador y lo visualiza.
VisualizaContador
       call   LCD_Linea1            ; Se pone al principio de la línea 1.
       movf   Contador,W
       call   BIN_a_BCD             ; Se debe visualizar en BCD.
       call   LCD_Byte
EsperaDejePulsar
       btfsc  Pulsador              ; Para que no interrumpa también en el flanco
       goto   EsperaDejePulsar      ; de bajada cuando suelte el pulsador.
FinInterrupcion
       bcf    INTCON,RBIF           ; Limpia flag de reconocimiento RBIF.
       retfie

       INCLUDE   <RETARDOS.INC>
       INCLUDE   <BIN_BCD.INC>
       INCLUDE   <LCD_EASY.INC>
       END
```

EJERCICIO 5:

a) El programa ***Int_RBI_05.asm*** realiza la siguiente tarea:

"Cada vez que se presionan los pulsadores conectados a las líneas RB7 y RB6 se produce una interrupción en el PIC. En el módulo LCD se visualizará el nombre del pulsador activado: "RB7" o "RB6"

b) Comprueba su funcionamiento en la tarjeta EasyPIC.

EJERCICIO 6[3]:

a) A partir del ejemplo anterior, escribe un programa que realice la siguiente tarea mediante el uso de las interrupciones de cambio de nivel en los pines RB7:RB4:

Cada vez que se presione alguno de los pulsadores conectados al nibble alto de ***PORTB*** *se visualizará un mensaje diferente de más de 16 caracteres que se irá desplazando a lo largo de la pantalla.*

b) Comprueba su funcionamiento en la tarjeta EasyPIC.

EJERCICIO 7:

a) Escribe un programa que realice la siguiente tarea mediante el uso de las interrupciones INT y RBI:

Cada vez que se presione el pulsador conectado a RB0, conmutará el estado de los LEDs conectados a PORTC. Además, cada vez que se presione alguno de los pulsadores conectados al las líneas RB7:RB4, en el módulo LCD se visualizará el nombre del pulsador activado: "RB7", "RB6", "RB5" o "RB4".

b) Comprueba su funcionamiento en la tarjeta EasyPIC.

[3] **Este ejercicio pide realizar la misma tarea que el EJEMPLO 8 de la Práctica 8, pero esta vez, utilizando las interrupciones RBI.**

```
;*********************************** Int_RBI_05.asm *********************************
;
; A las líneas RB7 y RB6 se conectan dos pulsadores que producen una interrupción cada vez
; que se pulsan. En el módulo LCD se visualizará el nombre del pulsador activado: "RB7" o "RB6".
;
; ZONA DE DATOS ******************************************************************

        __CONFIG   _CP_OFF &  _WDT_OFF & _PWRTE_ON & _HS_OSC
        LIST       P=16F877A
        INCLUDE  <P16F877A.INC>

        CBLOCK  0x20
        Contador
        ENDC

#DEFINE  EntradaRB7       PORTB,7
#DEFINE  EntradaRB6       PORTB,6

; ZONA DE CÓDIGOS ****************************************************************

        ORG     0
        goto    Inicio
        ORG     4                       ; Vector de interrupción.
        goto    ServicioInterrupcion
Inicio
        call    LCD_Inicializa
        movlw   MensajeInicial
        call    LCD_Mensaje             ; Visualiza el mensaje inicial.
        bsf     STATUS,RP0              ; Acceso al Banco 1.
        bsf     EntradaRB7              ; Las líneas se configuran como entrada.
        bsf     EntradaRB6
        bcf     STATUS,RP0              ; Acceso al Banco 0.
        movlw   b'10001000'             ; Activa interrupción  por cambio en las
        movwf   INTCON                  ; líneas del Puerto B (RBIE) y la general (GIE).
Principal
        sleep                           ; Pasa a modo bajo consumo esperando las
        goto    Principal               ; interrupciones.

; Subrutina "ServicioInterrupcion" -----------------------------------------------
;
; Subrutina de atención a la interrupción. Detecta qué ha producido la interrupción y
; ejecuta la subrutina correspondiente.

ServicioInterrupcion
        call    Retardo_20ms            ; Espera se estabilicen niveles.
        btfsc   EntradaRB7              ; ¿Está presionado el pulsador RB7?
        call    VisualizaRB7
        btfsc   EntradaRB6              ; ¿Está presionado el pulsador RB6?
        call    VisualizaRB6
        bcf     INTCON,RBIF
        retfie                          ; Retorna y rehabilita las interrupciones, GIE=1.

; Subrutinas "VisualizaRB7" y "VisualizaRB6" -------------------------------------------

VisualizaRB7
        call    LCD_Borra
        movlw   MensajeRB7              ; Visualiza el mensaje para RB7.
        call    LCD_Mensaje
        return

VisualizaRB6
        call    LCD_Borra
        movlw   MensajeRB6              ; Visualiza el mensaje para RB6.
        call    LCD_Mensaje
        return

; "Mensajes" ---------------------------------------------------------------------
;
Mensajes
        addwf   PCL,F
MensajeInicial
        DT "Editorial  Ra-Ma", 0x00
MensajeRB7
        DT "  Pulsador RB7", 0x00
MensajeRB6
        DT "   Ahora RB6  ", 0x00

        INCLUDE   <LCD_EASY.INC>
        INCLUDE   <LCD_MENS.INC>
        INCLUDE   <RETARDOS.INC>
        END
```