Informe 9
Manejo de Interrupciones en PIC

Laboratorio de Arquitectura del Computador

Elaborado por:
Tomás Guzmán, 21615008

1 MARCO TEÓRICO

El PIC también es capaz de manejar interrupciones (señales ajenas al código que ejecuta el mismo) y manejar la mismas.

Para poder realizar esta tarea, va a hacer uso de dos registros especiales los cuales son INTCON y OPTION_REG (OPT en nuestro código).

1.1 INTCON

INTCON es el registro encargado de manejar interrupciones, podemos observar como funciona en el manual de PIC16F84A:

INTCON REGISTER (ADDRESS 0Bh, 8Bh)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
GIE	EEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF
bit 7							bit 0

Este registro es fundamental para el manejo de INTerrupciones. Por eso su nombre.

En este caso nos interesa el 7mo bit, GIE. El cual es el Global Interruptions Enable Bit. Activado el mismo, claramente activa las interrupciones globales.

También el bit 1, INTF (RB0/INT External Interrupt Flag bit) indica que ocurrió una interrupción en RB0/INT. Este bit debe ser apagado de forma manual (por código). Es responsabilidad del programador apagarlo al finalizar el manejo de la interrupción, de tal forma se puedan manejar las próximas.

Finalmente, nos interesa el bit 4, INTE (RB0/INT External Interrupt Enable bit), que habilita las interrupciones del puerto RB0/INT, el cual es el primer bit de PORTB, es decir PORTB<0>. Estas interrupciones serán configuradas en el próximo registro a describir.

1.2 OPTION_REG

El registro OPTION_REG ofrece una amplia funcionalidad, relacionada a muchos procesos en el PIC.

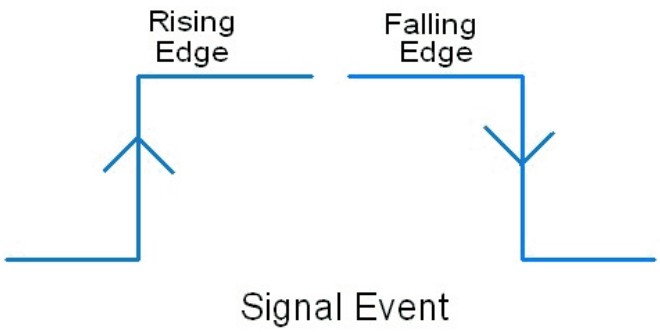
OPTION REGISTER (ADDRESS 81h)

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
bit 7							bit 0

El bit que más nos interesa será INTEDG (Interrupt Edge Select Bit) que es el OPTION_REG<6>, es decir el bit 6. Si el mismo está encendido, entonces la interrupción en el flanco de subida será detectada. De otro modo, será detectada en flanco de bajada.

Esta interrupción será detectada en el pin RB0/INT el cual es PORTB<0>.

Un flanco de subida (rising edge) es cuando se pasa de una señal LOW a una HIGH y un flanco de bajada (falling edge) es el proceso contrario.



Note que tanto INTCON como OPTION_REG son del BANK1.

2 IMPLEMENTACIÓN

```
    LIST P=16F84

OPT     EQU 01H
STATUS  EQU 03H
PORTA   EQU 05H
PORTB   EQU 06H
TRISA   EQU 05H
TRISB   EQU 06H
INTCON  EQU 0BH
```

```
#DEFINE INTEDG  OPT, 6
#DEFINE INTF  INTCON, 1

#DEFINE BANK0   BCF STATUS, 5
#DEFINE BANK1   BSF STATUS, 5

    ORG 0
    GOTO INICIO

INICIO ORG 10
    BANK1
    BSF     INTEDG      ; Manejo de interrupciones en RB0/INT en flancos d
e subida
    CLRF    TRISA       ; PORTA es de salida
    MOVLW   H'FF'       ; Queremos PORTB puerto de lectura
    MOVWF   TRISB       ; Ahora PORTB es de lectura
    BANK0

    MOVLW   B'10010000'
    MOVWF   INTCON      ; Se habilitan las interrupciones globales y de RB
0/INT

MAIN
    MOVLW   B'00000001'
    MOVWF   PORTA       ; Ahora PORTA<0> está encendido
    GOTO MAIN

    ORG     4
    GOTO    INTER       ; Si ocurre una interrupción, brincamos a INTER

INTER ORG 50

CICLO_I
    MOVLW   B'00000010'
    MOVWF   PORTA       ; Ahora PORTA<1> está encendido, indicando la inte
rrupción

    BTFSC   PORTB, 1    ; Si PORTA<1> vale 1 nos mantenemos en el ciclo
    GOTO    CICLO_I

    BCF     INTF        ; Se borrá el bit de interrupcion RB0/INT
    RETFIE              ; Salimos de la interrupción
```

```
END
```

Notamos que la directiva ORG ahora cobra más sentido. Esta determina en qué lugar se va a guardar una porción de código con la única intención de no solapar otro código. En este caso se respeta los espacios en el PIC para poder manejar el vector de interrupciones.

Este código es sencillo y cuenta de dos ciclos, el MAIN y el CICLO_I.

Sin existir la interrupción el PIC se mantiene en MAIN, en el cual constantemente se carga el valor 00000001 en PORTA, lo cual es lo mismo que activar PORTA<0> y su pin RA0.

Note que esto mismo se pudo haber obtenido haciendo uso de BSF PORTA, 0 y si se hubiese usado un #DEFINE RA0 PORTA, 0 sería tan sencillo como BSF RA0. De todas formas, ambos códigos son legibles, pero es bueno señalar las varias formas que existen de conseguir un mismo resultado.

El puerto PORTB<0>, llamado RB0/INT fue configurado para esperar la señal de algún componente externo, que en esta práctica será un botón en Multisim. Note que, esta configurado para “escuchar” interrupciones en flanco de subida, es decir, desde 0 a 1. Un botón al ser presionado hace exactamente esto.

Al recibir la interrupción, el PIC salta a CICLO_I donde se quedará hasta presionar otro botón el cual encenderá PORTB<1> o RB1 y se volverá a MAIN.

2.1 ¿SON DOS INTERRUPCIONES?

Parece que al presionar el segundo botón y activar a RB1 se está manejando una interrupción nueva. Esto no es así. En este caso solo se verifica un con un Bit Test si RB1 está activado.

Note que esto es explícito en lo señalado en rojo.

```
CICLO_I
    MOVLW   B'00000010'
    MOVWF   PORTA       ; Ahora PORTA<1> está encendido, indicando la inte
rrupción

    BTFSC   PORTB, 1    ; Si PORTA<1> vale 1 nos mantenemos en el ciclo
    GOTO    CICLO_I

    BCF     INTF        ; Se borrá el bit de interrupcion RB0/INT
    RETFIE              ; Salimos de la interrupción
```

Por otro lado, las interrupciones ocurren fuera del código. Se configuran varios registros del PIC para recibirlas, pero en ningún momento se pregunta “¿está RB0/INT encendido?”.

Además, la “pregunta” (el Bit Test) a RB1 no tiene consecuencias en otros registros. La interrupción en RB0/INT activa el flag INTF, es decir INTCON<1> el cual indica que hubo o bien un flanco de subida o uno de bajada en este pin (en este caso un flanco de subida).

2.2 COMPILACIÓN

Este programa en extensión .asm fue compilado haciendo uso de MPSAMWIN.

Este programa se incluyó en el PATH de Windows, por lo cual se puede ejecutar desde cualquier terminal.

Su sintaxis es sencilla, en este caso:

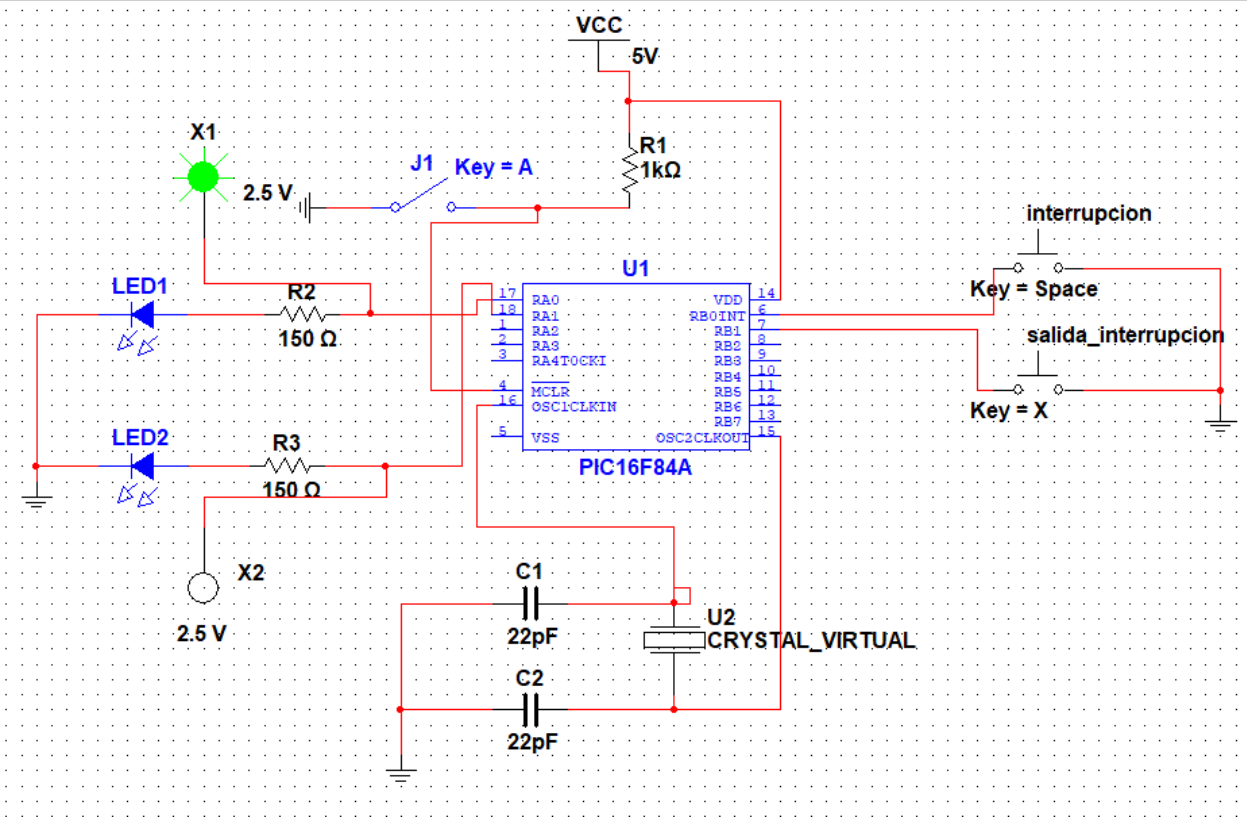
$ mpasmwin practica9.asm

Produce varios archivos, entre ellos el .hex. Este último se usará en el PIC implementado en Multisim para visualizar la funcionalidad.

3 SIMULACIÓN

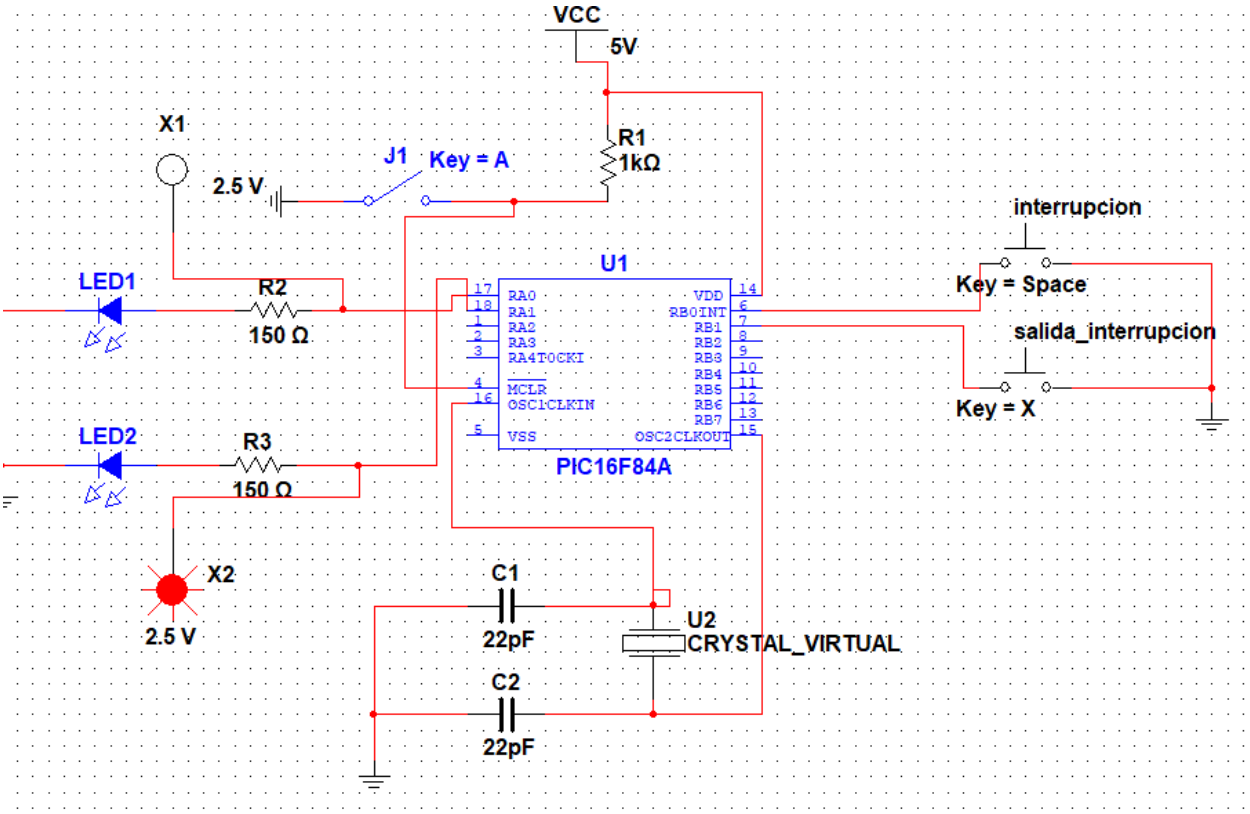
Veamos que al inicio el PIC estará ejecuntando MAIN, por lo cual PORTA<0> (pin RA0) estará activado. Este está conectado al LED verde.

Note que en Multisim los LEDs prenden por muy poco tiempo (instantes) y por eso es útil el uso de Probes, que permiten determinar si hay corriente en un cable o no.



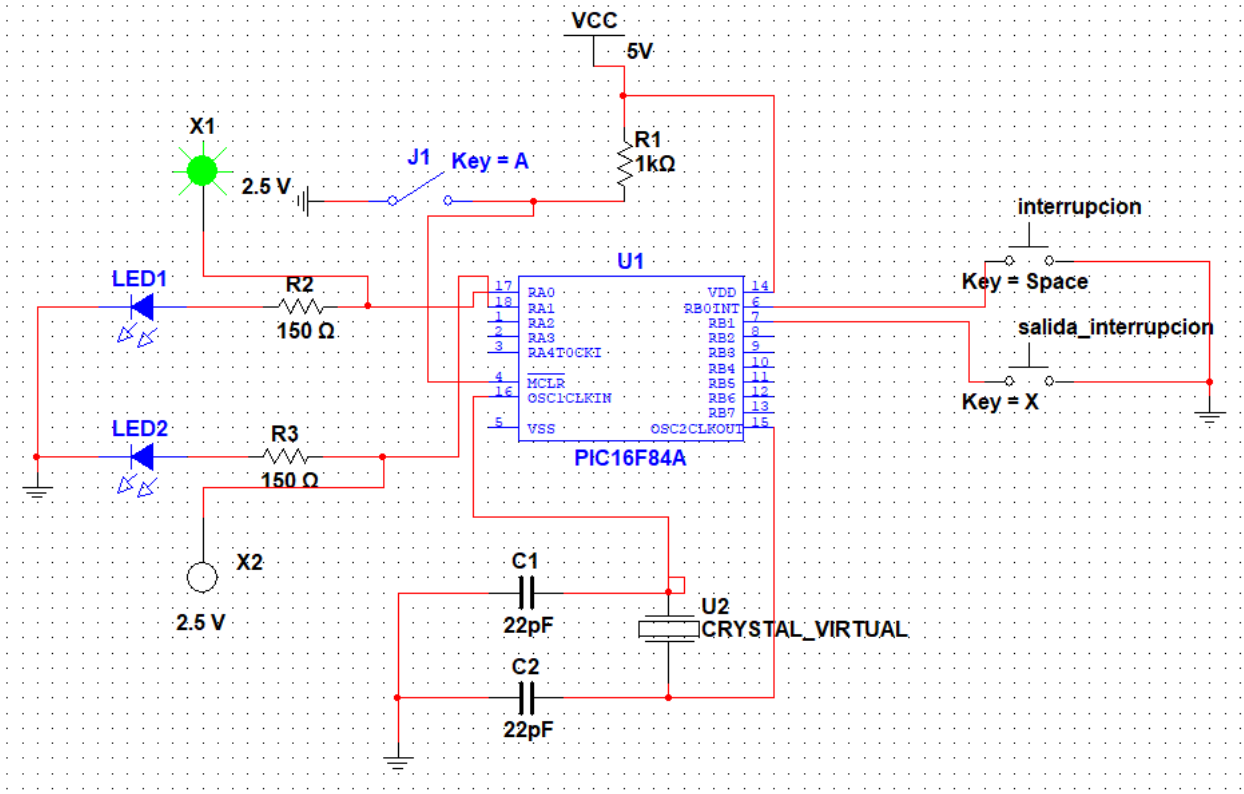
Esto ocurre al iniciar el PIC. Solo está prendido el Probe verde. Si usted corre esto en un protoboard de verdad, vería el LED verde encendido.

Una vez se presiona la tecla espacio, se manda un flanco de subida a RB0/INT, por lo cual ocurrirá una interrupción y se pasará al CICLO_I

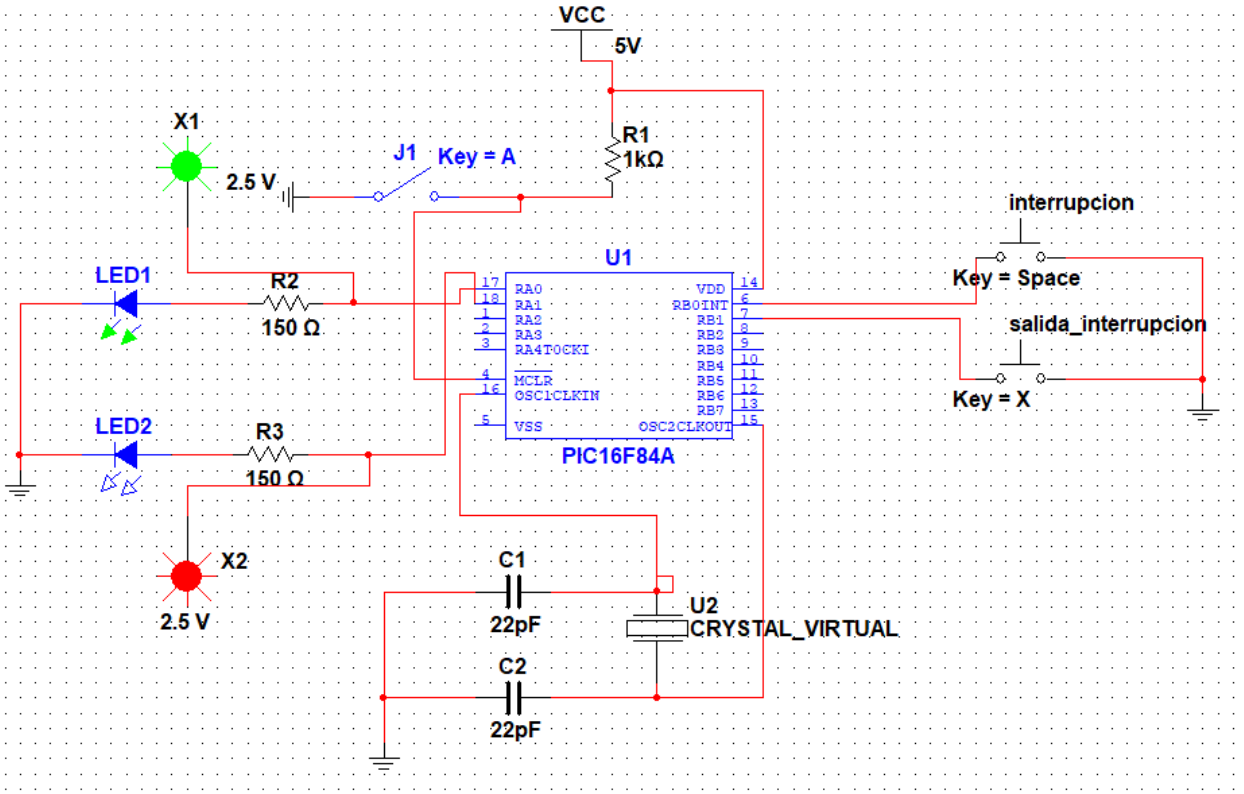


Note que ahora está encendido el LED rojo (el probe por ser Multisim). Esto significa que el PIC está ejecutando CICLO_I.

Una vez se presione X, RB1 será activado y volveremos al ciclo MAIN.



Estas fotos fueron tomadas sencillamente de Multisim, sin embargo si se logró capturar el LED encendido después de algunos reflejos altos:



4 CONCLUSIONES

- El PIC tiene un mecanismo muy sofisticado, aunque de bajo nivel, de manejar interrupciones.
- El mismo se vale de sus registros OPTION_REG e INTCON para configurar que interrupciones va a recibir. Normalmente GIE (INTCON<7>) se va a activar para recibir interrupciones. Es este bit quien representa la habilitación global de interrupciones.
- RB0/INT es un pin especial para detectar interrupciones. Los otros pines asociados a PORTB no detectan las mismas, aunque se pueda hacer Bit Test sobre los mismos.
- Un Bit Test no es lo mismo que una interrupción, aunque puede parecer que sí:
    - Un bit test no activa bits (conocidos como flags) en INTCON. Una interrupción sí
    - De una interrupción se "sale" (se regresa a la parte del código en donde ocurrió la misma) mediante el uso de RETFIE. Suponiendo que el bit test hiciera un CALL a una subrutina, de esta no se devolvería con RETFIE, sino con RETURN
- Se recalca que PORTA y PORTB pueden ser de lectura o de salida. En esta experiencia, a diferencia de las anteriores, fue PORTB quien es lectura y PORTA salida.
- Se puede resumir la operación de este PIC en
    - Si el probe verde está encendido, estamos en MAIN, no hay interrupción o ya fue manejada.
    - Si el probe rojo están encendido, estamos en CILCO_I, ocurrió una interrupción y no se ha salido de la misma.