

**UNIVERSIDAD CATOLICA ANDRES BELLO  
FACULTAD DE INGENIERIA  
ESCUELA INFORMATICA  
CATEDRA DE ARQUITECTURA DEL COMPUTADOR**

**PRACTICA #9**

***Microcontroladores PIC 16F84 (Manejo las interrupciones del microcontrolador).***

Objetivos:

- 1.- Realizar un programa donde se configure la interrupción externa.
- 2.- Manejo de los vectores de interrupción y de las banderas (flags) del microcontrolador.

Materiales:

- 1 Protoboard.
- 1 Fuente de poder
- 1 Multímetro Digital
- 1 PIC16F84
- 2 Resistencia de  $150\Omega$ ,  $\frac{1}{4}$  W
- 3 Resistencias de 1K
- 2 diodos LED (colores verdes y rojo)
- 2 Pulsador

**DESARROLLO**

En el PIC 16f84 hay 4 fuentes de interrupciones, veamos cuales son...

- Por el pin RB0/INT, que regresa al PIC del modo SLEEP (interrupción externa).
- Por los pines RB4 a RB7, configurados como entrada y en caso de que alguno de ellos cambie de estado (interrupción externa).
- Por desbordamiento del registro TMR0, cuando este registro pasa de 255 a 0 en decimal ó 0xFF a 0x00 en hexa (interrupción interna).
- Al completar la escritura de la EEPROM de datos (interrupción interna).

Debe haber algo que nos indique la fuente de interrupción que se ha producido, y estas son las banderas de interrupciones, cada interrupción tiene su propia bandera y es un bit del registro INTCON, que cambia de estado de **0** a **1** cuando se produce la interrupción, salvo la última que se encuentra en el registro EECON1

Ahora veamos cuales son esas banderas...

- Para **PB0/INT** la bandera es **INTF** (Bit1)
- Para los pines **PB4** a **PB7**, es **PBIF** (Bit0)
- Para **TMR0**, es **T0IF** (Bit2)
- Para la **EEPROM**, es **EEIF** (Bit4) en el registro **EECON1**.

Si prestas atención, todas estas banderas terminan en **F** es decir **FLAG's**, Bien, ahora veremos todo esto en el registro INTCON:

REGISTRO INTCON							
<b>GIE</b>	<b>EEIE</b>	<b>TOIE</b>	<b>INTE</b>	<b>RBIE</b>	<b>TOIF</b>	<b>INTF</b>	<b>RBIF</b>

El Bit GIE habilita todas las interrupciones, Los Bit's de fondo gris son las banderas, y los BIT's que se corresponden con cada flag son la habilitación de la fuente de interrupción para que esta cambie, recuerda que el flag de EEIE se encuentra en el registro EECON1.

Y ahora los detalles de cada Bit del registro INTCON.

BIT's	L ó E	Reset	Descripción
Bit 7: <b>GIE</b> Habilitación General.	L/E	0	1 = Todas las Interrupciones activadas 0 = Todas las Interrupciones desactivadas
Bit 6: <b>EEIE</b> Int. de Periféricos	L/E	0	1 = Activada 0 = Desactivada
Bit 5: <b>TOIE</b> Int. del TMR0	L/E	0	1 = Activada 0 = Desactivada
Bit 4: <b>INTE</b> Int. Externa	L/E	0	1 = Activada 0 = Desactivada
Bit 3: <b>RBIE</b> Int. por PORTB	L/E	0	1 = Activada 0 = Desactivada
Bit 2: <b>TOIF</b> Bandera del TMR0.	L/E	0	1 = TMR0 desbordado. 0 = No se ha desbordado
Bit 1: <b>INTF</b> Bandera - RB0/INT	L/E	0	1 = Ocurrió una interrupción externa 0 = No ha ocurrido interrupción externa
Bit 0: <b>RBIF</b> Bandera - RB4:RB7	L/E	x	1 = Al menos un pin cambio de estado 0 = Ningún pin ha cambiado de estado.

En la tabla, los valores de **L ó E** son para que sepas si el bit es de lectura o escritura, los valores de **Reset** son el estado de cada Bit después de producirse un reset o cuando se inicia el micro.

Por cierto y antes de que lo olvide, si bien cada flag cambia o se pone a 1 al producirse una interrupción, es tarea de usted borrarlo o ponerlo a cero nuevamente, ya que si no lo haces el micro estará siempre interrumpido o lo que es lo mismo, creerá que la interrupción se está produciendo continuamente.

Lo primero que debes saber, es que cuando una interrupción se produce, sea cual fuere la fuente de interrupción, el micro deja todo y salta a la dirección 04H, éste es el vector de interrupción, en esta dirección es donde escribiremos la rutina que dé servicio a todas las interrupciones, o bien haremos un salto a donde se encuentre ese trozo de código, el cual se conoce como **ISR** (Rutina de Servicio de Interrupción)

El Tiempo de Procesamiento de la ISR debe ser lo más breve posible, para dar lugar a que se ejecuten las otras interrupciones, ya que puedes haber habilitado más de una de ellas.

Lo más crítico de una interrupción es tener que guardar todos los registros importantes con sus respectivos valores, para luego restaurarlos, y así el micro pueda continuar con la tarea que estaba realizando cuando fue interrumpido.

Vamos a realizar un programa que una vez generado una interrupción no lo indicara con el encendido de un diodo led, mientras no se ejecute esta interrupción el programa se encontrara realizando una rutina en la cual haremos que se muestre en un display una letra (La letra P), pero cuando llegue la interrupción mostrara otra letra (La letra H). Para que salgamos de la interrupción debemos de presionar otro pulsador el cual será el apagara el flag y dará fin a la interrupción. Utilizaremos la interrupción del puerto PB0.

Únicamente del registro OPTION, tomaremos el Bit 6 (INTEDG) el cual nos indicara de que forma se activara la interrupción externa:

Si el valor es 1 entonces será por flanco de subida

Si el valor es 0 entonces será por flanco de bajada

A continuación tenemos el programa propuesto para esta práctica

```
INTER - Blocco note
File Modifica Formato Visualizza ?

LIST P=16F84

OPT EQU 01H
STATUS EQU 03H
PA EQU 05H
PB EQU 06H
TRISA EQU 05H
TRISB EQU 06H
INTCON EQU 0BH

ORG 0
GOTO INICIO ;SALTO A DONDE VAMOS A INICIAR EL PROGRAMA PARA NO SOLAPAR CON EN VECTOR DE INTERRUPCION

INICIO ORG 10

BSF STATUS,5; CBIAMOS AL BANCO DE MEMORIA 1
BSF OPT,6 ; DECIMOS QUE EL BIT INTEDG VALE 1 PARA QUE LA INTERRUPCION EXTERNA SUCEDA EN EL FLANCO DE SUBIDA
CLRF TRISA ; CONFIGURAMOS TODO EL PUERTO A COMO SALIDA
MOVLW H'FF' ; MOVEMOS EL VALOR DE 11111111 AL REGISTRO W PARA LUEGO PODER CONFIGURAR LA PUERTA B
MOVWF TRISB ; CONFIGURAMOS LA PUERTA B COMO SALIDA
BCF STATUS,5; CAMBIAMOS AL BANCO DE MEMORIA 0

MOVLW B'10010000' ; MOVEMOS EL VALOR A W PARA DESPUES CONFIGURA EL REGISTRO DE INTERRUPCIONES
MOVWF INTCON ;ACTVAMOS EL HABILITADOR PRINCIPAL, Y ACTIVAMOS UNICAMENTE A LA INTERRUPCION EXTERNA POR EL PUERTO PB0

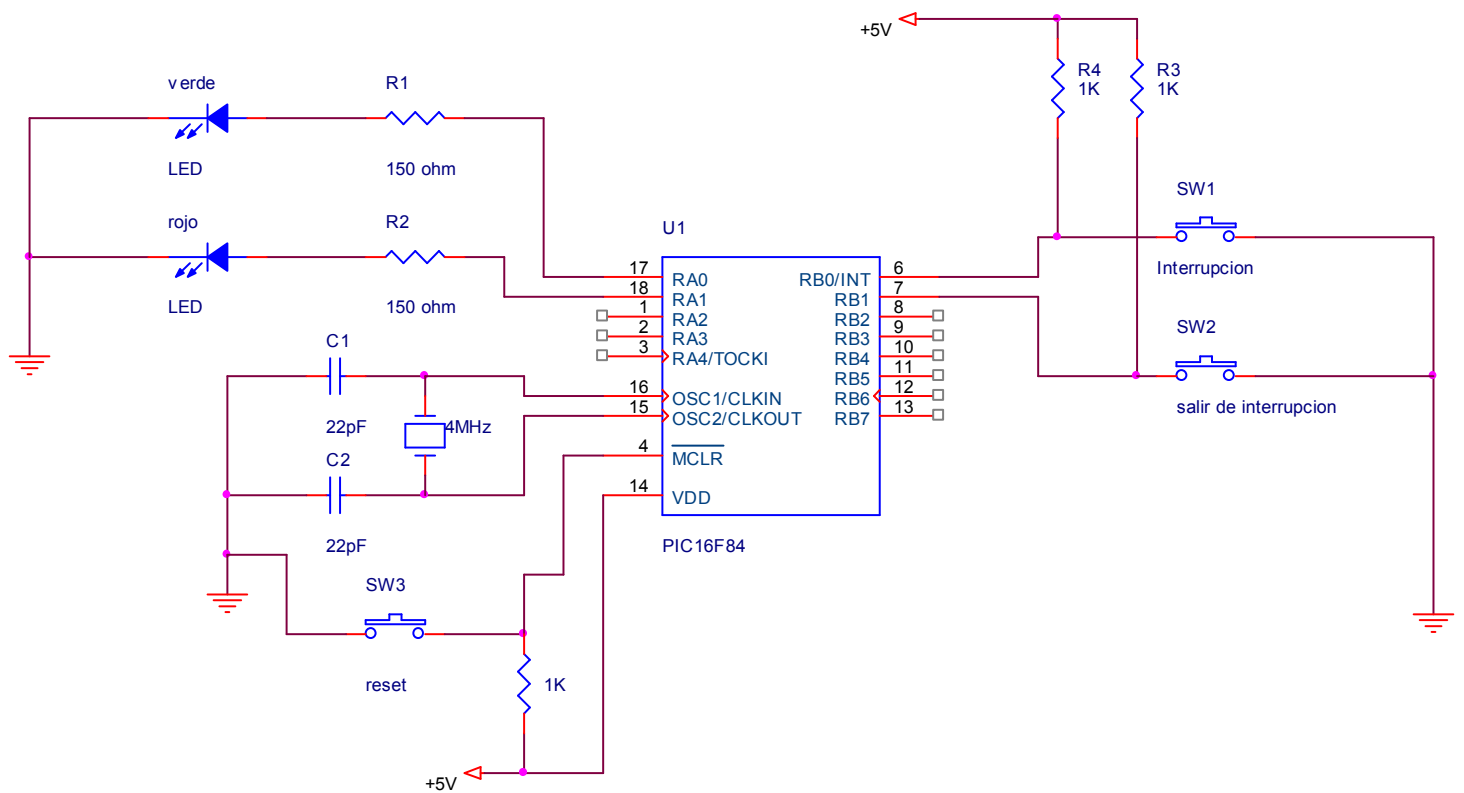
E1 MOVLW B'00000001' ; MOVEMOS EL VALOR BINARIO A W
MOVWF PA ; MOVEMOS EL VALOR AL PUERTO A, PARA ENCENDER EL DIODO LED QUE INDICA FUNCIONAMIENTO NORMAL
GOTO E1 ; VOLVEMOS A LA ETIQUETA E1, PARA TENER UN PROCESO CICLICO

ORG 4
GOTO INTER ;SALTAMOS A UNA ETIQUETA DONDE VAMOS A CORRER EL PROGRAMA DE LA INTERRUPCION

INTER ORG 50
E2 MOVLW B'00000010' ; MOVEMOS EL VALOR BINARIO A W
MOVWF PA ; MOVEMOS EL VALOR AL PUERTO A, PARA ENCENDER EL LED QUE INDICA QUE SE PRESENTO LA INTERRUPCION
BTFSB PB,1 ; CHEQUEAMOS SI EL PUERTO PB1 VALE 0, SI ES ASI SALTAMOS Y BORRAMOS LA BANDERA, DE LO CONTRARIO HACEMOS IN CICLO
GOTO E2
BCF INTCON,1 ;BORRAMOS LA BANDERA DE LA INTERRUPCION
RETFIE ; SALIMOS DE LA INTERRUPCION

END
```

Una vez terminado el programa deberá de simularlo (puede usar la herramienta de sim84) y luego construir el siguiente circuito:



Una vez construido ensamble el programa y descárguelo al microcontrolador con el uso del programador, luego pruébelo y emita sus conclusiones.