

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

PRACTICA #10

Microcontroladores PIC 16F84 (Manejo del temporizador y la interrupción interna).

Objetivos:

- 1.- Realizar un programa donde se configure el temporizador interno.
- 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 150Ω, ¼ W
- 3 Resistencias de 1K
- 2 diodos LED (color verde y rojo)
- 1 Pulsador

DESARROLLO

El temporizador/contador TMR0 es un registro de 8 bits, es decir, un particular tipo de registro cuyo contenido es incrementado con una cadencia regular y programable directamente por el hardware del PIC. Como es de 8 bits, el máximo de la cuenta está en 255.

El TMR0 tiene las siguientes características:

- Temporizador/Contador de 8 bits.
- Divisor de 8 bits programable por software.
- Selección de reloj interno y externo.
- Interrupción por desbordamiento.
- Selección del flanco del reloj externo.

Este registro puede usarse para contar eventos externos por medio de un pin de entrada especial (modo contador) o para contar pulsos internos de reloj de frecuencia constante (modo temporizador). Además, en cualquiera de los dos modos, se puede insertar un prescaler, es decir un divisor de frecuencia programable que puede dividir por 2, 4, 8, 16, 32, 64, 128 o 256. Este divisor puede ser utilizado alternativamente como prescaler o del TMR0 o como postscaler del Watch Dog Timer, según se lo programe.

En la práctica, a diferencia de los otros registros, el TMR0 no mantiene inalterado el valor que tiene memorizado, sino que lo incrementa continuamente.

Si por ejemplo escribimos en él el valor 10, después de un tiempo igual a cuatro ciclos máquina, el contenido del registro comienza a ser incrementado a 11, 12, 13 y así

sucesivamente con una cadencia constante y totalmente independiente de la ejecución del resto del programa.

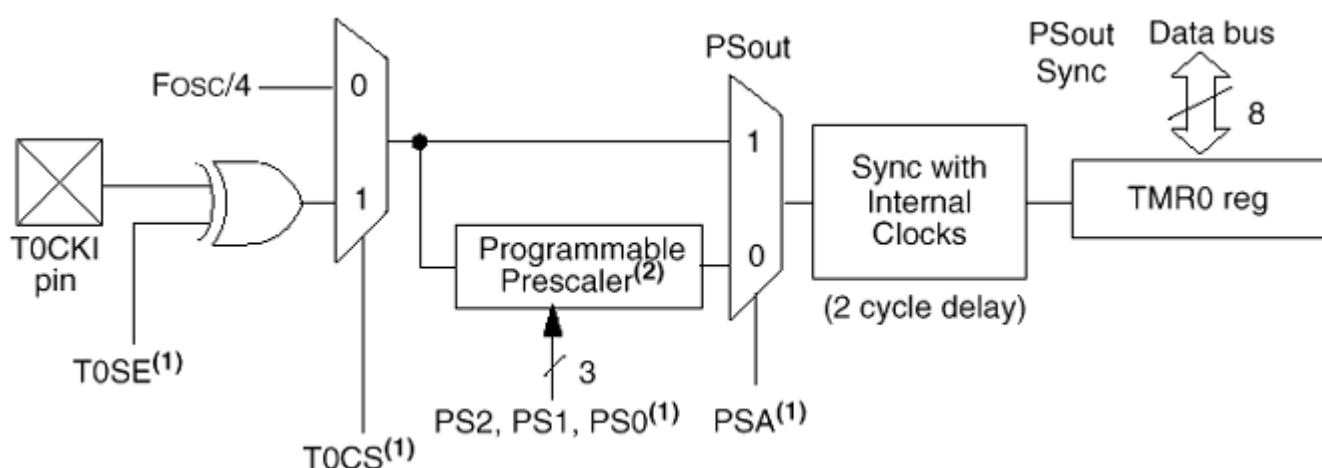
Una vez alcanzado el valor 255, el registro TMR0 es puesto a cero automáticamente comenzando entonces a contar desde cero y no desde el valor originalmente cargado. La frecuencia de conteo es directamente proporcional a la frecuencia de reloj aplicada al PIC y puede ser modificada programando adecuadamente algunos bits de configuración.

El modo temporizador se selecciona colocando a 0 el bit T0CS del registro OPTION. En este modo, el Timer0 se incrementa con cada ciclo de instrucción (con el divisor desactivado).

El modo contador se selecciona colocando a 1 el bit T0CS del registro OPTION. En este modo, el contador se incrementará con cada flanco de subida o de bajada del pin RA4/T0CKI. El flanco se selecciona con el bit T0SE del registro OPTION.

Se generará una interrupción cuando el registro TMR0 se desborda de FFh a 00h. Este desbordamiento se indicará el bit T0IF del registro INTCON.

En la siguiente figura está representado el esquema de bloques internos del PIC que determinan el funcionamiento del registro TMR0:



Fosc/4 y T0CKI representan las dos posibles fuentes de señal de reloj, para el contador TMR0.

Fosc/4 es una señal generada internamente por el PIC tomada del circuito de reloj y que es igual a la frecuencia del oscilador dividida por cuatro.

T0CKI es una señal generada por un posible circuito externo y aplicada al pin T0CKI correspondiente al pin 3 del PIC16F84.

Los bloques T0CS y PSA son dos selectores de señal (multiplexores) en cuya salida se presenta una de las dos señales de entrada en función del valor de los bits T0CS y PSA del registro OPTION.

El bloque PRESCALER ó Predivisor es un divisor programable cuyo función es dividir la frecuencia de conteo, interna ó externa.

El Prescaler

El PRESCALER consiste en un divisor programable de 8 bits a utilizar en el caso de que la frecuencia de conteo enviada al contador TMR0 sea demasiado elevada para nuestros propósitos. Se configura a través de los bits PS0, PS1 y PS2 del registro OPTION.

La frecuencia $F_{osc}/4$ es una cuarta parte de la frecuencia de reloj. Utilizando un cristal de 4Mhz tendremos una $F_{osc}/4$ igual a 1 MHz, con lo que la cadencia de conteo que se obtiene provoca en TMR0 1 millón de incrementos por segundo (1/1.000.000 seg. =1MHz), que para muchas aplicaciones podría resultar demasiado elevada.

Con el uso del PRESCALER podemos dividir la frecuencia $F_{osc}/4$ configurando oportunamente los bits PS0, PS1 y PS2 del registro OPTION según la siguiente tabla:

PS2	PS1	PS0	DIVISOR	FRECUENCIA DE SALIDA PREESCALER (Hz.)
0	0	0	2	2.000.000 frecuencia base 4.000.000
0	0	1	4	1.000.000 frecuencia base 4.000.000
0	1	0	8	500.00 frecuencia base 4.000.000
0	1	1	16	250.00 frecuencia base 4.000.000
1	0	0	32	125.00 frecuencia base 4.000.000
1	0	1	64	62.500 frecuencia base 4.000.000
1	1	0	128	31.250 frecuencia base 4.000.000
1	1	1	256	15.625 frecuencia base 4.000.000

Ejemplo práctico: Introducir un retardo igual a un segundo utilizando el registro TMR0.

Debemos programar el bit T0CS a 0 para seleccionar como fuente de conteo el reloj del PIC, el bit PSA también debe estar a 0 para asignar el PRESCALER al registro TMR0 en lugar de al Watch Dog Timer y los bits de configuración del PRESCALER a 111 para obtener una frecuencia de división igual a 1:256. La frecuencia que obtendremos en TMR0 será igual a:

$$F_{osc} = 4MHz$$

$$T_{osc} = \frac{1}{4MHz} = 0,25mseg$$

$$Tiempo = 4 \times T_{osc} (256 - TMR0) \times Divisor$$

El valor del tiempo a elegir podría ser 10 milisegundos, con lo cual la interrupción se ejecutara cada 10 mseg. Para conseguir el segundo bastara únicamente con que coloquemos una variable la cual se usara como contador descendiente desde 100 (Recordemos que 1000mseg es 1seg) y cuando llegue a cero habremos completado un tiempo igual a 1 segundo.

$$10\text{mseg} = 4 \times 0.25\mu\text{seg}(256 - \text{TMR0}) \times 256$$

$$256 - \frac{10\text{mseg}}{4 \times 0.25\mu\text{seg} \times 256} = \text{TMR0}$$

$$\text{TMR0} = 216,93 \Rightarrow 217$$

Copie el siguiente programa para descargarlo en el microcontrolador:

```

SEGUNDO - Blocco note
File Modifica Formato Visualiza
LIST P=16F84

OP EQU 01H ;DECLARACION DEL REGISTRO OPCION
TMR0 EQU 01H ;DECLARACION DEL TIMER 0
INTCON EQU 0BH ;DECLARACION DEL REGISTRO INTCON
TRISA EQU 05H ;DECLARACION DEL REGISTRO TRISA (CONFIGURACION DEL PUERTO A)
PA EQU 05H ;DECLARACION DEL PUERTO A
STATUS EQU 03H ;DECLARACION DEL REGISTRO STATUS
VAR EQU 10H ;DECLARACION DEL REGISTRO VAR (UTILIZADO COMO CONTADOR)

ORG 0 ;VECTOR DE INTERRUPCION DEL RESET
GOTO INICIO ;SALTO A INICIO DEL PROGRAMA

INICIO ORG 10
BSF STATUS,5 ;CAMBIO DE BANCO DE MEMORIA (BANCO 1)
CLRF TRISA ;DECLARACION TODOS LOS PINES DEL PUERTO A COMO SALIDA
MOVLW B'00000111' ;MOVIENDO EL VALOR DE 00000111 A W
MOVWF OP ;ASIGNADO EL DIVISOR A 256, ASIGNADO AL TIMER, UTILIZANDO LA SENAL INTERNA DEL MICROCONTROLADOR
BCF STATUS,5 ;CAMBIO DE BANCO DE MEMORIA (BANCO 0)

MOVLW B'10100000' ;MOVIENDO EL VALOR DE 10100000 A W
MOVWF INTCON ;HABILITANDO LAS INTERRUPCIONES Y DANDO PERMISO A LA INTERRUPCION DEL TIMER

MOVLW d'217' ;MOVIENDO EL VALOR DE 217 A W
MOVWF TMR0 ;PRECARGANDO EL TIMER EN 217
MOVLW d'100' ;MOVIENDO EL VALOR DE 100 A W
MOVWF VAR ;CARGANDO EN LA VARIABLE DE CONTEO EL VALOR DE 100

N1 NOP ;NO REALIZA NINGUNA OPERACION PERO GASTA UN CICLO DE MAQUINA
NOP
GOTO N1 ;SALTA A LA ETIQUETA N1

ORG 4 ;VECTOR DE INTERRUPCION
GOTO INTER ;SALTA A LA RUTINA A EJECUTAR CUANDO LLEGA LA INTERRUPCION

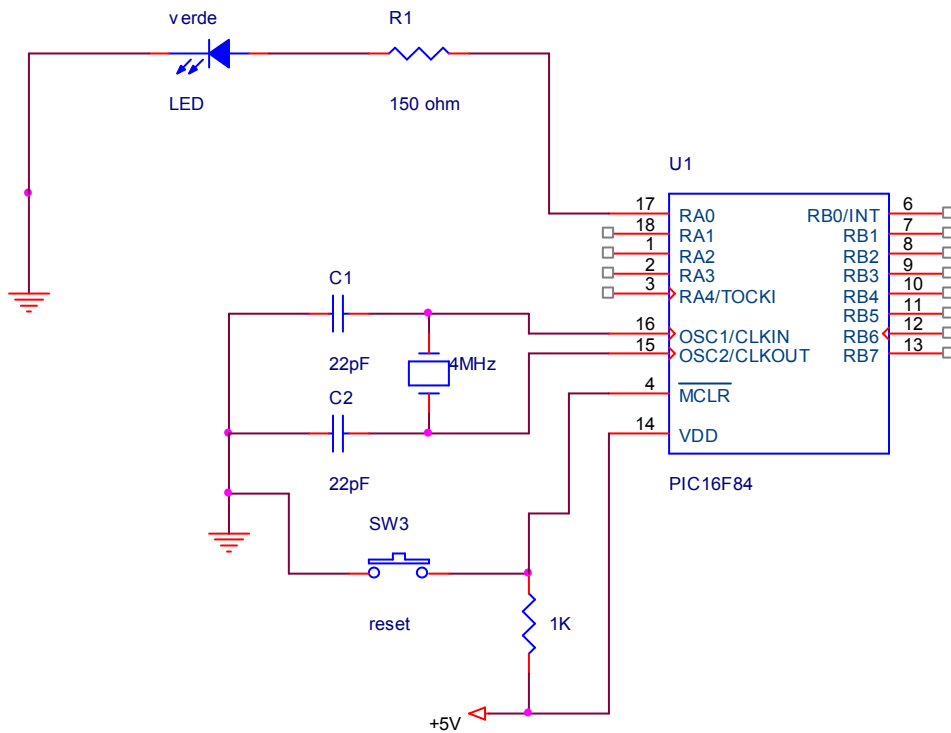
INTER ORG 50
DECFSZ VAR,1 ;DECREMENTA EN 1 LA VARIABLE DE CONTEO (VAR) Y SI EL RESULTADO ES CERO SALTA UNA LINEA
GOTO N3 ;EL RESULTADO DE LA VARIABLE ES DISTINTO DE CERO
SEG ;EL RESULTADO DE LA RESTA ES CERO Y SI ES ASI ACABA DE PASAR UN SEGUNDO
N3 MOVLW d'217' ;MOVIENDO EL VALOR DE 217 A W
MOVWF TMR0 ;PRECARGANDO EL TIMER EN 217
BCF INTCON,2 ;APAGANDO LA BANDERA DE LA INTERRUPCION DE DESBORDAMIENTO DEL TIMER
RETFIE ;SALIENDO DE LA INTERRUPCION

SEG BTFSZ PA,0 ;PREGUNTANDO SI EN BITO DEL PUERTO A ES CERO (SI ES CIERTO SALTA UNA LINEA)
GOTO OFF ;COMO EL BITO ES 1 ENTONCES SALTAMOS A LA ETIQUETA OFF
BSF PA,0 ;COMO EL BITO ERA 0 ENTONCES LO COLOCAMOS EN 1
GOTO N2 ;SALTAMOS A LA ETIQUETA N2
OFF BCF PA,0 ;EN ESTA INSTRUCCION PONEMOS A 0 AL BITO DEL PUERTO A
N2 MOVLW d'100' ;MOVIENDO EL VALOR DE 100 A W
MOVWF VAR ;CARGANDO LA VARIABLE DE CONTEO A 100
GOTO N3 ;SALTANDO A LA ETIQUETA N3

END

```

Luego de ensamblado, descárguelo en el microcontrolador y construya el siguiente circuito:



Utilice un cronometro y mida el tiempo de encendido y apagado del diodo led, el cual debería de ser 1 segundo en cada caso, recuerde que puede haber un pequeño error debido a la velocidad de arranque y parada sobre el cronometro (el cual debe ser de algunas milésimas de segundo)