



Benemérita Universidad Autónoma de Puebla

Facultad de Ciencias de la Electrónica

Microcontroladores

Materia

Práctica No. 3

Camarillo Bautista Alfredo, González Escalona Miguel Ángel, Lázaro Bonilla

Ramiro, López Arce Roberto

Integrantes del equipo

Ingeniería en mecatrónica

Carrera

Ricardo Álvarez González

Docente

08 de marzo de 2020, Primavera 2021

Fecha de entrega

Contenido

Marco teórico	3
Desarrollo práctico	10
Cálculos	13
Simulación	14
Conclusiones	15
Bibliografía	15

Marco teórico

El pic 18f4550 es un microcontrolador de 8 bits de la empresa Microchip. Este microcontrolador cuenta con una gran cantidad de memoria RAM, diferentes módulos de comunicación, una gran cantidad de pines de entrada y salida y algunas otras grandes cualidades. Algunas de sus características son:

- 40 pines tipo DIP
- Interfaz USB 2.0 de alta velocidad, EEPROM 256 bytes
- Memoria RAM 2048 bytes, EEPROM 256 bytes
- Memoria de programa (memoria flash) 32 kb
- Voltaje de operación 2 a 5.5 V
- Frecuencia máxima 48 MHz
- 35 pines de entrada / salida

Las instrucciones utilizadas en la práctica se enlistarán y explicarán a continuación.

Instrucción	Sintaxis	Operación	Palabras y Ciclos	Descripción
Bcf	[label] BCF f,b[,a]	0 -> f	1 palabra 1 ciclo	El bit 'b' en el registro 'f' es limpiado. Si 'a' es 0, el Access bank será seleccionado, sobre escribiendo el valor de BSR. Si 'a' es 1, entonces el banco será seleccionado según el valor de BSR (default).
Bra	[label] BRA n	(PC) + 2 + 2n -> PC	1 palabra 2 ciclos	Suma el complemento a 2 '2n' al pc
Bsf	[label] BSF f,b[,a]	1 -> f	1 palabra 1 ciclo	El bit 'b' en el registro 'f' es colocado. Si 'a' es 0 el Access bank será seleccionado, sobre escribiendo el valor de BSR. Si 'a' es 1, entonces el banco será seleccionado según el valor de BSR (default).
Btg	[label] BTG f,b[,a]	¬(f) -> f	1 palabra 1 ciclo	El bit 'b' en la memoria de datos localizada en 'f' se invierte. Si 'a' es 0, el banco de acceso será seleccionado, sobre escribiendo el valor de BSR. Si 'a' = 1, entonces el banco será seleccionado según el valor de BSR (default).
Retfie	[label] RETFIE [s]	(TOS) → PC; 1 → GIE/GIEH or PEIE/GIEH; if s = 1 (WS) → W; (STATUS) → STATUS; (BSRS) → BSR; PCLATH, PCLATH are unchanged.	1 palabra 2 ciclos	Regresa de la interrupción. El stack se llena y la cima del stack (TOS) se carga dentro de la PC. Las interrupciones son habilitadas estableciendo el bit

				de habilitación de interrupción global de prioridad alta o baja.
Btfss	[label] BTFSS f,b[,a]	Skip if (f) = 1	0 palabras 0 ciclos	Si el bit 'b' en el registro 'f' es 1, entonces la siguiente instrucción es omitida. Si el bit 'b' es 0, entonces la siguiente instrucción traída durante la instrucción ejecutada actualmente.
Call	[label] CALL k[,s]	(PC) + 4 → TOS, k → PC<20:1>, if s = 1 (W) → WS, (STATUS) → STATUS, (BSR) → BSR	0 palabras 0 ciclos	Es un llamado de subrutina de un rango de memoria de 2 Mbytes.
Clrf	[label] CLRF f[,a]	00h → f 1 → Z	1 palabra 1 ciclo	Limpia el contenido de un registro especificado.
Decfsz	[label] DECFSZ f[,d[,a]]	(f) - 1 → dest, skip if result = 0	0 palabras 0 ciclos	Decrementa un valor al registro seleccionado, salta si es 0
Movlw	[label] MOVLW k	K → W	1 palabra 1 ciclo	Mueve el valor literal de K al working register
Movwf	[label] MOVWF f[,a]	W → f	0 palabras 0 ciclos	Mueve el valor de W al registro seleccionado
Nop	[label] NOP	Ninguna operación	1 palabra 1 ciclo	Ocupa simplemente el tiempo de 1 palabra y 1 ciclo
Return	[label] RETURN [s]	(TOS) → PC, if s = 1 (WS) → W, (STATUS) → STATUS, (BSR) → BSR, PCLATH are unchanged	0 palabras 0 ciclos	Regresa de la subrutina
Rlncf	[label] RLNCF f[,d[,a]]	(f<n>) → dest <n+1> (f<7>) → dest<0>	N, Z	El contenido del registro f rota un bit hacia la izquierda
Rrncf	[label] RRNCF f[,d[,a]]	(f<n>) → dest <n-1> (f<0>) → dest<7>	N, Z	El contenido del registro f rota un bit hacia la derecha

Directivas utilizadas

<label> EQU <value> (equate): La directiva de ensamblador **EQU** simplemente equipara un nombre simbólico a un valor numérico.

ORG <value> (origin): La directiva **ORIGIN** le dice al ensamblador donde cargar instrucciones y datos dentro de la memoria.

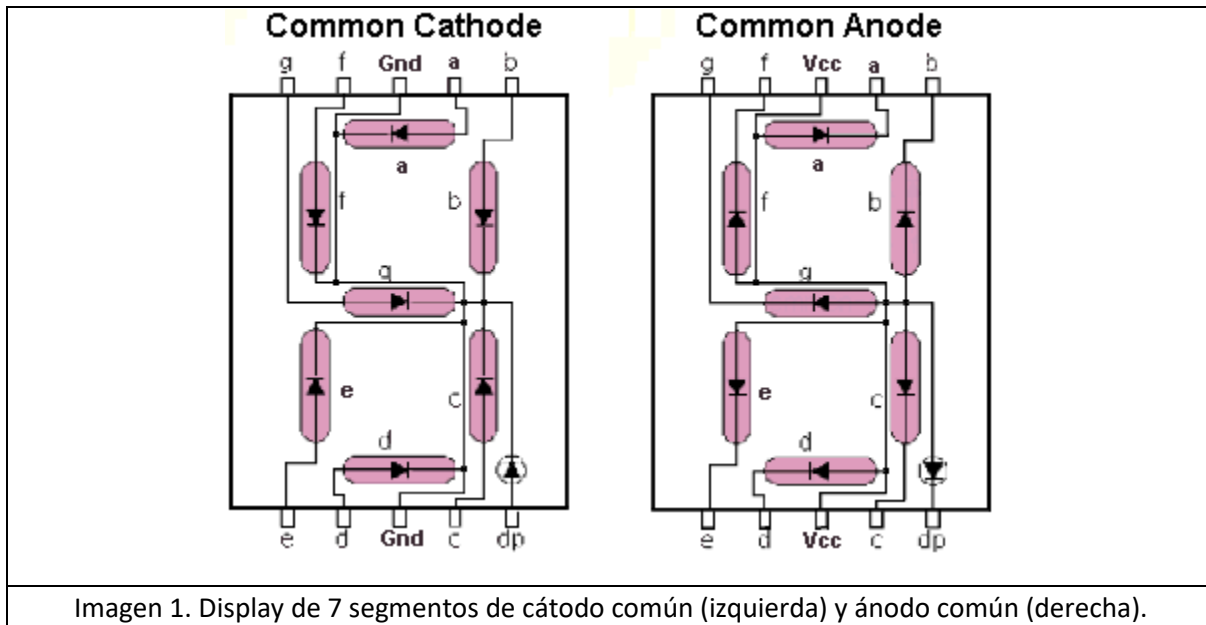
CBLOCK [expr]

Label [:increment] [,label [:increment]]

Endc: Define una lista de símbolos secuenciales nombrados. La lista de nombres termina cuando la directiva **endc** es encontrada.

Para dispositivos **PIC18**, sólo números pares en *expr* son permitidos.

Display de 7 segmentos



El display de 7 segmentos es un dispositivo opto – electrónico que permite visualizar números del 0 al 9. Existen dos tipos de display, de cátodo común y de ánodo común. Especificaciones:

- Voltaje: 3 VCD
- Amperaje: 10 mA
- Número de segmentos: 7
- Cátodo común
- Color del LED: Rojo
- Posiciones de los pines con respecto al punto: Vertical
- Dimensiones: 1.8 cm x 0.9 cm x 0.4 cm

Un display de este tipo está compuesto por siete u ocho leds de diferentes formas especiales y dispuestos sobre una base de manera que puedan representarse todos los símbolos numéricos y algunas letras. Los primeros siete segmentos son los encargados de formar el símbolo y con el octavo podemos encender y apagar el punto decimal. Cada uno de los segmentos que forman la pantalla están marcados con siete primeras letras del alfabeto ('a' – 'g').

En los tipos de ánodo común, todos los ánodos de los segmentos están unidos internamente a una patilla común que debe ser conectada a potencial positivo (nivel '1'). El encendido de cada segmento individual se realiza aplicando potencial negativo (nivel '0') por la patilla correspondiente a través de una resistencia que límite el paso de la corriente.

En los de tipo de cátodo común, todos los cátodos de los segmentos están unidos internamente a una patilla común que debe ser conectada a potencial negativo (nivel '0'). El encendido de cada

El módulo del Timer0 incorpora las siguientes características:

- Operación seleccionable por software como temporizador o contador, ambos en modo de 8 bits o 16 bits.
- Registros para lectura y escritura.
- Prescaler programable por software de 8 bits.
- Fuente seleccionable de reloj (interna o externa).
- Selección de borde para reloj externo.
- Interrupción en desbordamiento.

El registro T0CON controla todos los aspectos de las operaciones del módulo, incluyendo la selección de prescala. En ambos sirve de lectura y escritura.

Un diagrama de bloques simplificado del módulo del timer0 en su modalidad de 8 bits y 16 bits se muestra en las imágenes siguientes.

FIGURE 11-1: TIMER0 BLOCK DIAGRAM (8-BIT MODE)

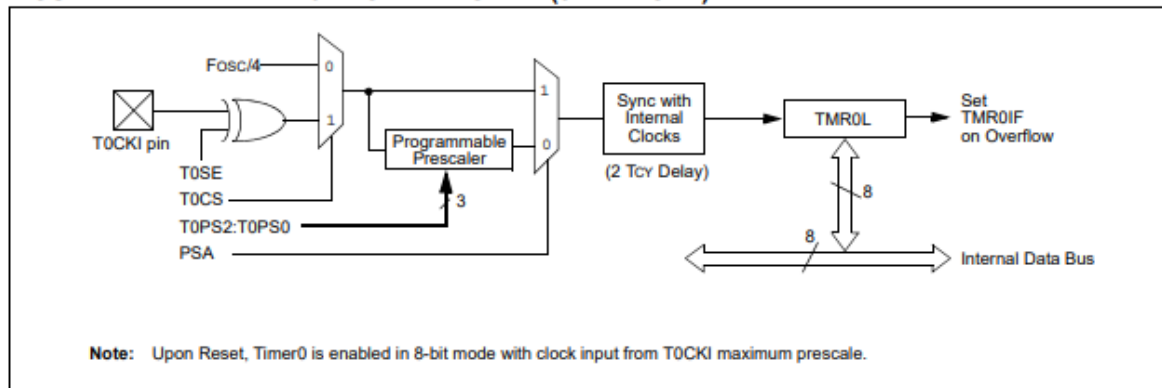


FIGURE 11-2: TIMER0 BLOCK DIAGRAM (16-BIT MODE)

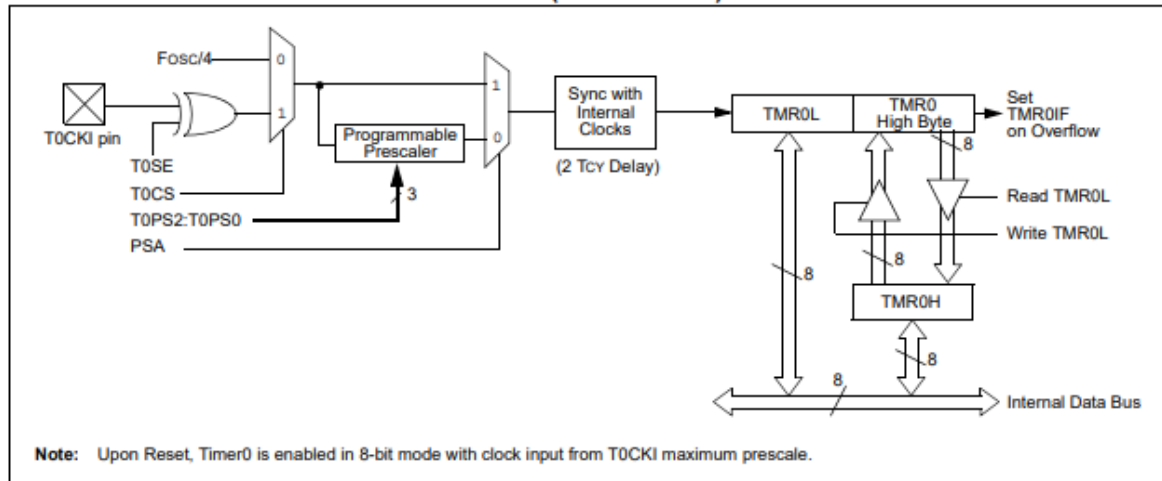


Imagen 3. Timer0 en su modalidad de 8 bits (Figure 11-1) y en su modalidad de 16 bits (Figure 11-2).

T0: Control de registro del Timer0

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
TMR0ON	T08BIT	T0CS	T0SE	PSA	T0PS2	T0PS1	T0PS0
bit 7							bit 0
Legend: R = Readable bit W = Writable bit U = Unimplemented bit, read as '0' -n = Value at POR '1' = Bit is set '0' = Bit is cleared x = Bit is unknown							

Imagen 4. Funcionalidad de los bits del registro T0CON

bit 7	TMR0ON: Timer0 On/Off Control bit 1 = Enables Timer0 0 = Stops Timer0
bit 6	T08BIT: Timer0 8-Bit/16-Bit Control bit 1 = Timer0 is configured as an 8-bit timer/counter 0 = Timer0 is configured as a 16-bit timer/counter
bit 5	T0CS: Timer0 Clock Source Select bit 1 = Transition on T0CKI pin 0 = Internal instruction cycle clock (CLKO)
bit 4	T0SE: Timer0 Source Edge Select bit 1 = Increment on high-to-low transition on T0CKI pin 0 = Increment on low-to-high transition on T0CKI pin
bit 3	PSA: Timer0 Prescaler Assignment bit 1 = Timer0 prescaler is NOT assigned. Timer0 clock input bypasses prescaler. 0 = Timer0 prescaler is assigned. Timer0 clock input comes from prescaler output.
bit 2-0	T0PS2:T0PS0: Timer0 Prescaler Select bits 111 = 1:256 Prescale value 110 = 1:128 Prescale value 101 = 1:64 Prescale value 100 = 1:32 Prescale value 011 = 1:16 Prescale value 010 = 1:8 Prescale value 001 = 1:4 Prescale value 000 = 1:2 Prescale value

Imagen 5. Función de acuerdo con los estados de los bits del registro T0CON

Operación del timer0

El Timer0 puede operar ya sea como temporizador o como contador, para seleccionar alguna de estas opciones se tiene que limpiar el bit T0CS (es decir, el bit 5 del registro T0CON). En el modo temporizador, el modulo incrementa en cada reloj por defecto a no ser que un valor de prescaler diferente sea seleccionado. Si el registro TMR0 está escrito, el incremento es inhibido para los dos siguientes ciclos de instrucción. El usuario puede solucionar esto escribiendo un valor ajustado para el registro del TMR0.

El modo contador es seleccionado al establecer el bit T0CS en 1. En modo contador, el timer0 incrementa ya sea en cada flanco de subida o bajada del pin RA4/T0CKI. El incremento en el flanco es determinado por el bit de selección de flanco de fuente del Timer0, T0SE (es decir, el bit 4 del registro T0CON); limpiando este bit se selecciona el flanco de subida.

Una señal externa de reloj puede ser usada para manejar el Timer0; como sea, se deben conocer ciertos requisitos para asegurarse que la señal externa de reloj esté sincronizada con la fase interna del reloj (Tosc). Existe un retraso entre la sincronización y el comienzo en el incremento el temporizador/contador.

Prescaler

Un contador de 8 bits está disponible como un prescaler para el modulo del Timer0. El prescaler no está directamente establecido para escribirse o leerse; su valor se establece por el PSA y TOPS2: los bits de TOPS0 (TOCON<3:0>) los cuales determinan la asignación del prescaler.

Interrupción del Timer0

La interrupción del TMR0 se genera cuando se desbordan los registros del TMR0 desde FFh hasta 00h en la modalidad de 8 bits, o desde FFFFh hasta 0000h en modalidad de 16 bits. Este desbordamiento establece el bit de bandera del TMR0IF. La interrupción puede ser enmascarada limpiando el bit TMR0IE (el bit 5 del registro INTCON). Antes de rehabilitar la interrupción, el bit TMR0IF debe ser limpiado en el software mediante una rutina de servicio de interrupción.

Desde que el Timer0 se apaga en el modo espera, la interrupción del TMR0 no puede despertar al procesador de ese modo.

Desarrollo práctico

Para la práctica realizada se utilizaron los siguientes materiales a enlistar.

Cantidad	Concepto
1	Pic 18F4550
1	Laptop
1	Display de 7 segmentos cátodo común
8	Resistencias de 330 Ohms
1	Software MPLAB v8.92
1	Datasheet del PIC 18F4550
1	Push button
1	Resistencia de 10k

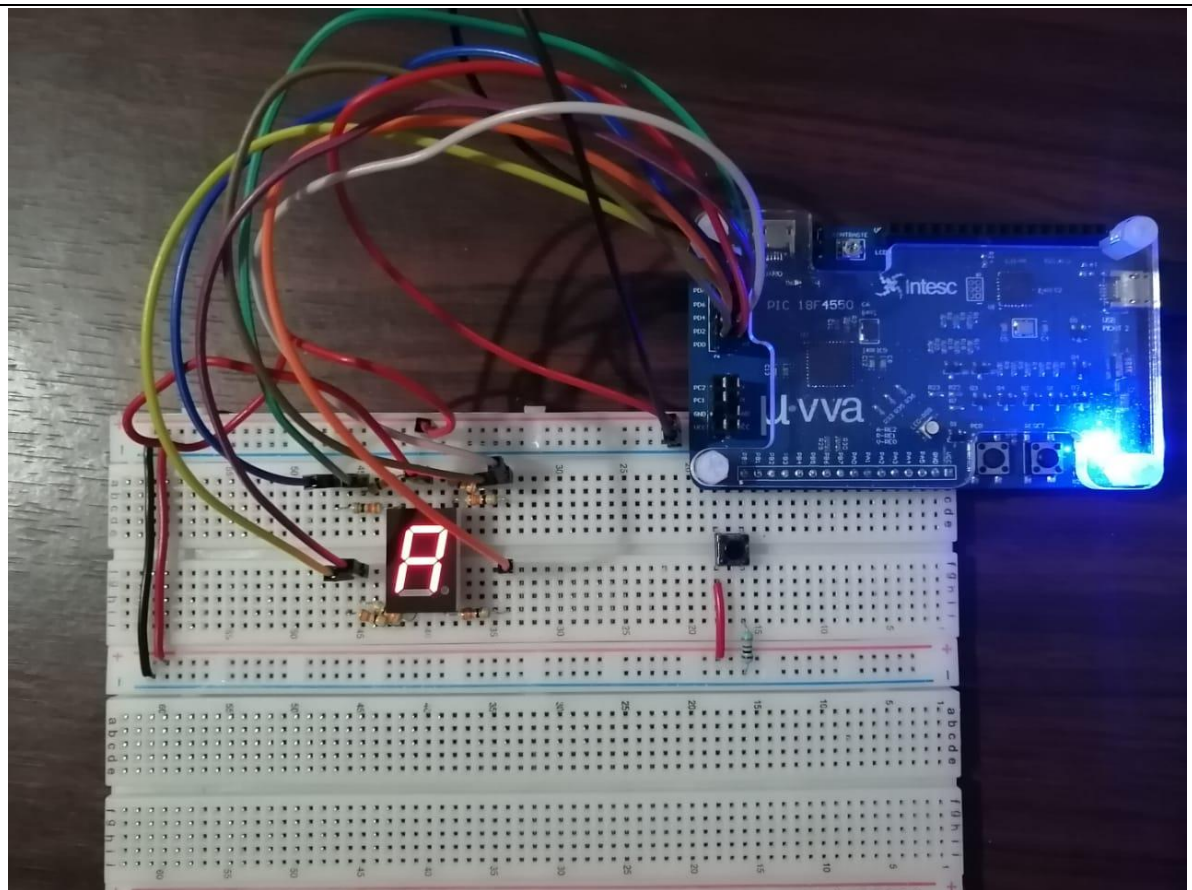


Imagen 6. Circuito montado para la práctica

Se hicieron modificaciones con respecto a la práctica anterior. La primera modificación por mencionar es que las variables definidas disminuyeron a solamente una, la cuál es flags.

```

;Definiciones de variables
cblock      0x0
flags
endc

```

Imagen 7. Variables definidas

En el inicio del programa principal, se establecen algunas funciones como la frecuencia del oscilador interno (4 MHz, cabe recordar que el valor por default es '100', pero para establecer una frecuencia de 4 MHz el valor del registro OSCCON debe ser '110').

Después, se configura mediante el registro ADCON1 los pines de puerto a digitales con el valor 0x0F (1111)₂.

Para configurar el puerto D como salida, es necesario limpiar el registro de dirección de datos TRISD, lo que configurará sus pines como salidas.

Se deben configurar las interrupciones externas INT0 y TMR0 estableciendo el valor 0xB0 en el registro INTCON (1011 0000)₂, cabe recordar que los bits que se habilitan son: GIE/GIEH (Global Interrupt Enable Bit), TMR0IE (TMR0 Overflow Interrupt Enable Bit) e INTOIE (INT0 External Interrupt Enable Bit).

También tendremos que modificar el valor del registro T0CON, el cuál nos servirá para poder trabajar con el temporizador de 16 bits, para esto se establece un valor de 0x95 (1001 0101)₂. Los bits que se habilitan en el registro T0CON son: TMR0ON (Timer0 on/off control bit), T0SE (Timer0 Source Edge Select Bit) y para los bits 2-0 del registro T0CON representa los bits de selección del preescaler, '101' será un valor de preescala de 1:64.

Para establecer un tiempo de 1000 ms (1 segundo), el valor de precarga previamente calculado mediante las ecuaciones dadas en los apuntes deberá ser cargado a los registros TMR0H (Timer0 Register High Byte) y TMR0L (Timer0 Register Low Byte), donde se utilizó en la modalidad de 16 bits (FFFFh -> 0000h), así que el valor que se asignará será 0xC2 al registro TMR0H y 0xF7 al registro TMR0L, todo esto mediante las instrucciones **movlw** y **movwf**.

```

;Inicio del programa principal
inicio  bsf      OSCCON, IRCF2, 0
        bsf      OSCCON, IRCF1, 0
        bcf      OSCCON, IRCF0, 0      ; 110, Oscilador interno a 4 MHz
        movlw    0x0F
        movwf    ADCON1, 0             ; Puertos digitales
        clrf     PORTD, 0
        clrf     TRISD, 0             ; Puerto D configurado como salida
        movlw    0xB0
        movwf    INTCON, 0            ; Configuramos interrupciones externa INT0 y TMR0
        movlw    0x95
        movwf    T0CON, 0             ; Timer 16 bits, preescaler *64
        movlw    0xE1
        movwf    TMR0H, 0
        movlw    0x7C
        movwf    TMR0L, 0            ; Valor de precarga para 500ms a 4 MHz, preescaler 64

```

Imagen 8. Instrucciones para configurar el PIC18

Para la secuencia de los números mostrados en el display, se debe cargar previamente los valores para mostrar la numeración desde el 0 hasta F en sistema hexadecimal. Para las subrutinas esp0, esp1, ..., espF se utiliza un código similar en estas. Lo primero que realiza es verificar que el bit de interrupción este encendido mediante la instrucción **btfs** (bit test F skip if set), si está encendido entonces se saltará a la instrucción donde se apaga el bit de interrupción con la instrucción **bcf** (bit clear f). Hay que recordar que el bit 0 del registro *flags* se pone en 1 para cuando el timer haya contado un segundo, por lo que de acuerdo con la subrutina esp0, esp1, ..., espF si aún no ha pasado ese segundo no podrá saltarse el retorno a la misma subrutina y no podrá avanzar a la siguiente instrucción, la cual apaga el bit de interrupción y después verifica el bit de 1 del registro *flags*, que se encarga de monitorear la interrupción mediante la interrupción de alta prioridad que se activa cuando se presiona el switch, la cuál si es 1 saltará la instrucción "**bra f**" y empezará el conteo ascendente, de lo contrario será un conteo descendente.

cero	movlw	0xC0	; Código del cero
	movwf	PORTD, 0	
esp0	btfs	flags, 0	; ¿pasaron 500 ms?
	bra	esp0	
	bcf	flags, 0	
	btfs	flags, 1, 0	; verifica dirección de conteo
	bra	f	
uno	movlw	0xF9	; Código del uno
	movwf	PORTD, 0	
esp1	btfs	flags, 0	; ¿pasaron 500 ms?
	bra	esp1	
	bcf	flags, 0	
	btfs	flags, 1, 0	; verifica dirección de conteo
	bra	cero	

Imagen 9. Código para la secuencia de conteo ascendente/descendente

Hay que aclarar que la subrutina para la secuencia del display que muestra F16 se tiene que hacer el llamado a dos subrutinas, la de "E" y la de "cero", esto debido a que si el bit de interrupción está encendido entonces continuará con el conteo ascendente y se irá directamente a "cero", de lo contrario retrocederá hacia "E".

f	movlw	0x8E	; Código del F
	movwf	PORTD, 0	
espF	btfs	flags, 0	; ¿pasaron 500 ms?
	bra	espF	
	bcf	flags, 0	
	btfs	flags, 1, 0	; verifica dirección de conteo
	bra	e	
	bra	cero	

Imagen 10. Código de la subrutina para F16

Para la subrutina de servicio de interrupción *RSI* en la dirección 0x08 lo primero que se hace es verificar si el bit *TMROIF* del registro *INTCON* está encendido (hay que recordar que este bit cuando está encendido indica que se ha desbordado, si está apagado es lo contrario), esto quiere decir que

una vez que el timer0 haya registrado un segundo dicho bit estará encendido. Si el bit *TMROIF* no está encendido, se verificará si el bit *INTOIF* (el bit de bandera de interrupción externa) está encendido, si lo está los bits del registro *flags* serán cambiados de valor mediante la instrucción **btg** (bit toggle), de lo contrario se regresará al programa principal después de la interrupción.

Si el bit *TMROIF* está encendido, se apagará mediante la instrucción **bcf** (bit clear f) y después se le asignará a los registros *TMROH* y *TMROL* el valor de precarga previamente calculado (en este caso 0xC2 y 0xF7) para después apagar el bit 0 del registro *flags* y regresar de la interrupción.

```

;*****RUTINA DE SERVICIO DE INTERRUPTCIÓN*****
RSI      btfss    INTCON, TMROIF
          bra      SINT0
          bcf      INTCON, TMROIF
          movlw    0xC2
          movwf    TMR0H, 0
          movlw    0xF7
          movwf    TMR0L, 0           ; valor de precarga para 1000 ms a 4 MHz preescaler 64
          bsf      flags, 0           ; monitor interrupción del timer 0
          retfie
SINT0    bcf      INTCON, INTOIF      ; apaga bit de bandera
          btg      flags, 1           ; bit monitor de interrupción
          retfie
          END

```

Imagen 11. Rutina de servicio de interrupción

Cálculos

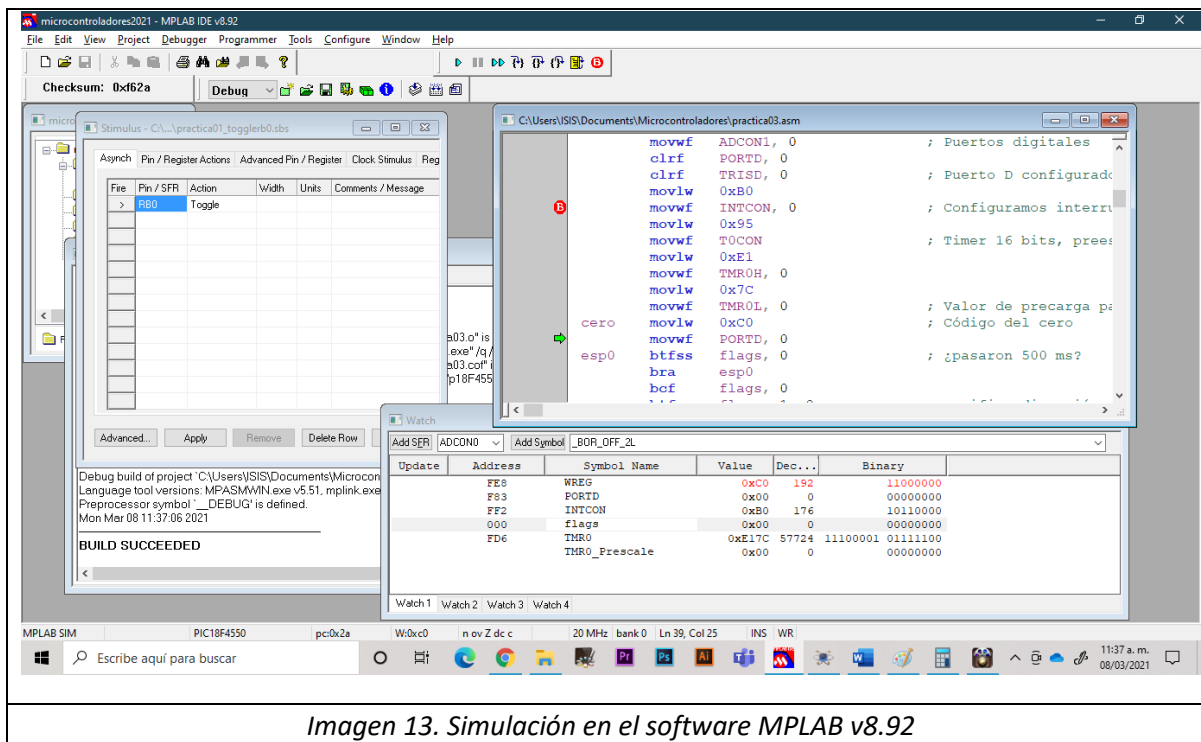
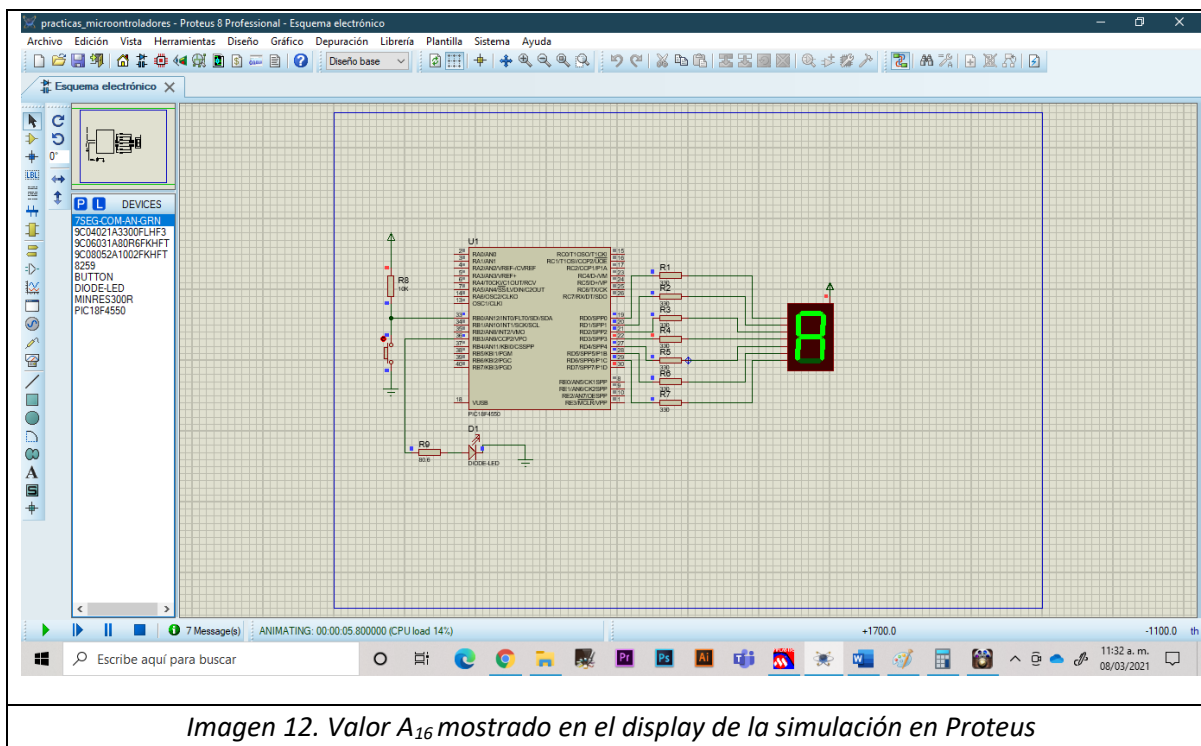
Los cálculos realizados en esta práctica fueron hechos para calcular el valor del timer y el valor de precarga.

$$\text{Valor del timer} = \frac{\text{Temporización}}{4 \times T_{osc} \times \text{Valor_preescaler}} = \frac{1000 \text{ ms}}{4 \times 0.25 \mu\text{seg} \times 64} = 15625$$

$$\text{Valor de precarga} = 65536 - 15625 = 49911$$

Este valor decimal lo tendremos que cargar a los registros del timer de 16 bits, por lo que será necesario convertirlo a hexadecimal, lo que es equivalente a 0xC2F7.

Simulación



Conclusiones

Con el desarrollo de la práctica utilizando el Timer0 del PIC 18F4550 podemos observar que, a diferencia de las herramientas utilizadas en prácticas anteriores, el uso de los timers dentro del PIC son de gran beneficio ya que el tiempo es más exacto, así que puede usarse para algún proceso que requiera usar el valor del tiempo como parámetro. Siempre es necesario revisar la hoja de datos de nuestro PIC para consultar el manejo adecuado de los bits y nuestra implementación sea correcta, de esa forma obtendremos los resultados esperados.

Link del vídeo: <https://youtu.be/CzSeiOWViuU>

Bibliografía

Microchip PIC18F Instruction Set. (2021). Retrieved 9 February 2021, from http://technology.niagarac.on.ca/staff/mboldin/18F_Instruction_Set/

(2021). Retrieved 23 February 2021, from <http://ww1.microchip.com/downloads/en/devicedoc/33014j.pdf>