Benemérita Universidad Autónoma de Puebla

Facultad de Ciencias de la Electrónica

Microcontroladores

Materia

Práctica No. 6

Camarillo Bautista Alfredo, González Escalona Miguel Ángel, Lázaro Bonilla Ramiro, López Arce Roberto

Integrantes del equipo

Ingeniería en mecatrónica

Carrera

Ricardo Álvarez González

Docente

20 de mayo de 2021, Primavera 2021

Fecha de entrega

Contenido

Objetivo

Entender el funcionamiento del PWM del PIC18F4550 y poder manipularlo para que funcione con distinto porcentaje de su capacidad utilizando sus registros correspondientes y otras herramientas previamente vistas.

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.

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

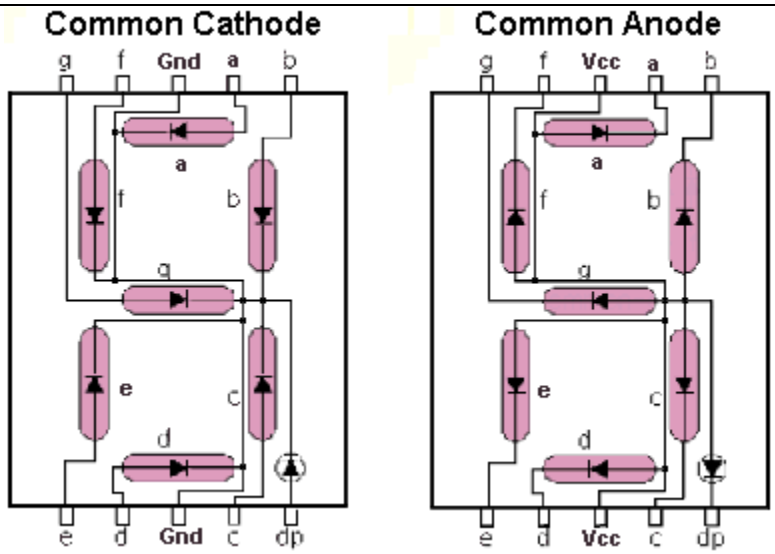


Imagen 1. Display de 7 segmentos de cátodo común (izquierda) y ánodo común (derecha).

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 segmento individual se realiza aplicando potencial positivo (nivel ‘1’) por la patilla correspondiente a través de una resistencia que limite el paso de la corriente.

Interrupciones externas

Las interrupciones externas en los pins RB0/AN12/INT0/FLT0/SDI/SDA, RB1/AN10/INT1/SCK/SCL y RB2/AN8/INT2/VMO son activados por flanco. Si el bit correspondiente INTEDGx en el registro INTCON2 es puesto (=1), la interrupción es activa en un flanco; si el bit está limpio, entonces la interrupción se activa en un flanco de bajada.

Cuando un flanco válido aparece en el pin RBx/INTx, el bit de bandera correspondiente, INTxIF, es puesto. Esta interrupción puede deshabilitar limpiando el bit de activación correspondiente, INTxIE. El bit de bandera, INTxIF, debe ser limpiado en software en la rutina de servicio de interrupción antes de habilitar de nuevo la interrupción.

Todas las interrupciones externas (INT0, INT1 e INT2) pueden “despertar” al procesador del modo de manejo de energía si el bit INTxIE fue establecido para configurar el manejo de energía. Si el bit de habilitación de interrupción global, GIE, este puesto, el procesador se ramificará al vector de interrupción después de “despertarse”.

La prioridad de interrupción de INT1 e INT2 está determinada por el valor contenido en los bits de interrupción de prioridad INT1IP (INTCON3<6>) e INT2IP (INTCON3<7>). No hay algún bit de prioridad asociado a INT0, siempre es una fuente de interrupción de alta prioridad.

Bits de control para la configuración del puerto A/D

PCFG3:PCFG0: A/D Port Configuration Control bits:

PCFG3: PCFG0	AN12	AN11	AN10	AN9	AN8	AN7(2)	AN6(2)	AN5(2)	AN4	AN3	AN2	AN1	AN0
0000(1)	A	A	A	A	A	A	A	A	A	A	A	A	A
0001	A	A	A	A	A	A	A	A	A	A	A	A	A
0010	A	A	A	A	A	A	A	A	A	A	A	A	A
0011	D	A	A	A	A	A	A	A	A	A	A	A	A
0100	D	D	A	A	A	A	A	A	A	A	A	A	A
0101	D	D	D	A	A	A	A	A	A	A	A	A	A
0110	D	D	D	D	A	A	A	A	A	A	A	A	A
0111(1)	D	D	D	D	D	A	A	A	A	A	A	A	A
1000	D	D	D	D	D	D	A	A	A	A	A	A	A
1001	D	D	D	D	D	D	D	A	A	A	A	A	A
1010	D	D	D	D	D	D	D	D	A	A	A	A	A
1011	D	D	D	D	D	D	D	D	D	A	A	A	A
1100	D	D	D	D	D	D	D	D	D	D	A	A	A
1101	D	D	D	D	D	D	D	D	D	D	D	A	A
1110	D	D	D	D	D	D	D	D	D	D	D	D	A
1111	D	D	D	D	D	D	D	D	D	D	D	D	D

A = Analog input D = Digital I/O

Imagen 2. Bits de control para la configuración del puerto A/D.

Configuración del puerto D como salida

	7	6	5	4	3	2	1	0
TRISD	0	0	0	0	0	0	0	0
PORTD	Salida	Salida	Salida	Salida	Salida	Salida	Salida	Salida

Timer 0

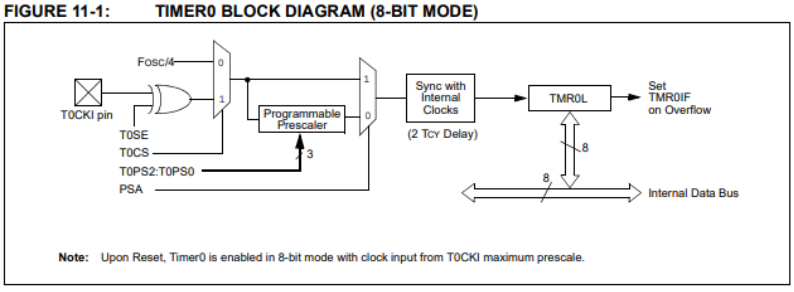
El módulo del Timer0 incorpora las siguientes características:

- 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.

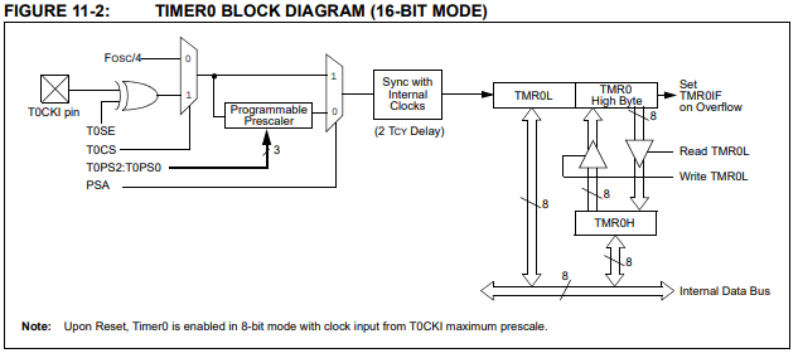
Un diagrama de bloques simplificado del módulo del timer0 en su modalidad de 8 bits y 16 bits se muestra en las imágenes siguientes.

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



Note: Upon Reset, Timer0 is enabled in 8-bit mode with clock input from T0CKI maximum prescale.

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



Note: Upon Reset, Timer0 is enabled in 8-bit mode with clock input from T0CKI maximum prescale.

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

Timer 2

El módulo del Timer2 incorpora las siguientes características:

- Temporizador de 8 bits y registros de periodo (TMR2 y PR2, respectivamente).
- Registros para lectura y escritura
- Prescaler programable por software (1:1, 1:4 y 1:16)
- Postscaler programable por software (1:1 hasta 1:16)
- Interrupción en TMR2 hasta PR2
- Uso opcional como reloj de cambio para el módulo MSSP

El módulo es controlado a través del registro T2CON el cuál habilita y deshabilita el temporizador y configura el prescaler y postscaler. El timer2 puede ser apagado limpiando el bit de control, TMR2ON (T2CON<2>), para minimizar el consumo de poder.

Un diagrama de bloques simplificado del módulo del timer2 se muestra en la figura 13-1.

FIGURE 13-1: TIMER2 BLOCK DIAGRAM

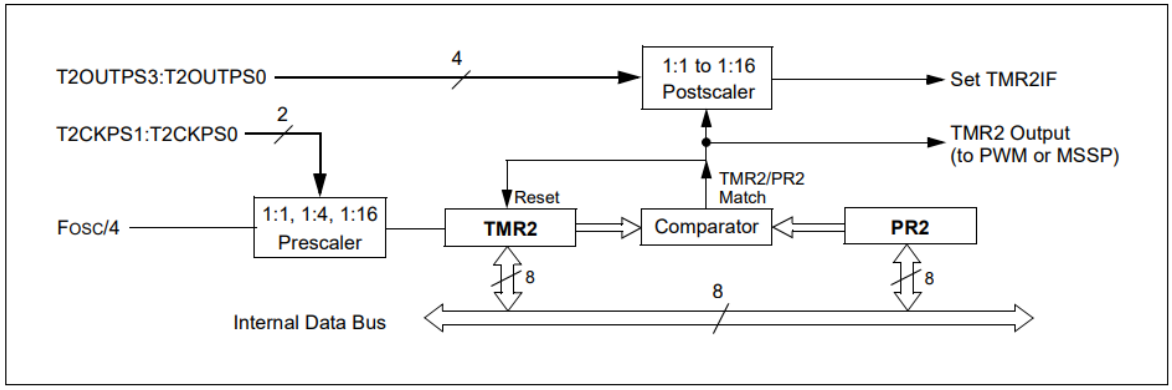


Imagen 6. Diagrama de bloques del Timer2

REGISTER 13-1: T2CON: TIMER2 CONTROL REGISTER

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2OUTPS0	TMR2ON	T2CKPS1	T2CKPS0
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

bit 7 **Unimplemented:** Read as '0'

bit 6-3 **T2OUTPS3:T2OUTPS0**: Timer2 Output Postscale Select bits
0000 = 1:1 Postscale
0001 = 1:2 Postscale
•
•
•
1111 = 1:16 Postscale

bit 2 **TMR2ON**: Timer2 On bit
1 = Timer2 is on
0 = Timer2 is off

bit 1-0 **T2CKPS1:T2CKPS0**: Timer2 Clock Prescale Select bits
00 = Prescaler is 1
01 = Prescaler is 4
1x = Prescaler is 16

Imagen 7. Registros de control del timer2

TABLE POINTER REGISTER (TBLPTR) y TABLE LATCH REGISTER (TABLAT)

El Table Latch (TABLAT) es un registro de 8 bits mapeado en el espacio de los SFR (Special Function Registers). El registro TABLAT es usado para guardar datos de 8 bits durante transferencia de datos entre la memoria de programa y los datos de la RAM.

El registro Table Pointer (TBLPTR) direcciona un byte dentro de la memoria de programa. El TBLPTR está compuesto de tres SFR: Table Pointer Upper Byte, Table Pointer High Byte y Table Pointer Low Byte (TBLPTRU:TBLPTRH:TBLPTRL). Estos tres registros se unen para formar un apuntador de 22 bits. El orden bajo de 21 bits permite al dispositivo direccionar hasta 2 MB de espacio de memoria de programa. El bit 22 permite el acceso al ID del dispositivo, del usuario y la configuración de bits. El puntero de tabla, TBLPTR, es usado por las instrucciones TBLRD y TBLWT. Estas instrucciones pueden actualizar el TBLPTR en una de cuatro maneras de acuerdo con la tabla de operaciones. Estas operaciones se muestran en la tabla 6 – 1. Estas operaciones en el registro TBLPTR solamente afectan el orden bajo de 21 bits.

TABLE 6-1: TABLE POINTER OPERATIONS WITH TBLRD AND TBLWT INSTRUCTIONS

Example	Operation on Table Pointer
TBLRD* TBLWT*	TBLPTR is not modified
TBLRD*+ TBLWT*+	TBLPTR is incremented after the read/write
TBLRD*- TBLWT*-	TBLPTR is decremented after the read/write
TBLRD+* TBLWT+*	TBLPTR is incremented before the read/write

FIGURE 6-3: TABLE POINTER BOUNDARIES BASED ON OPERATION

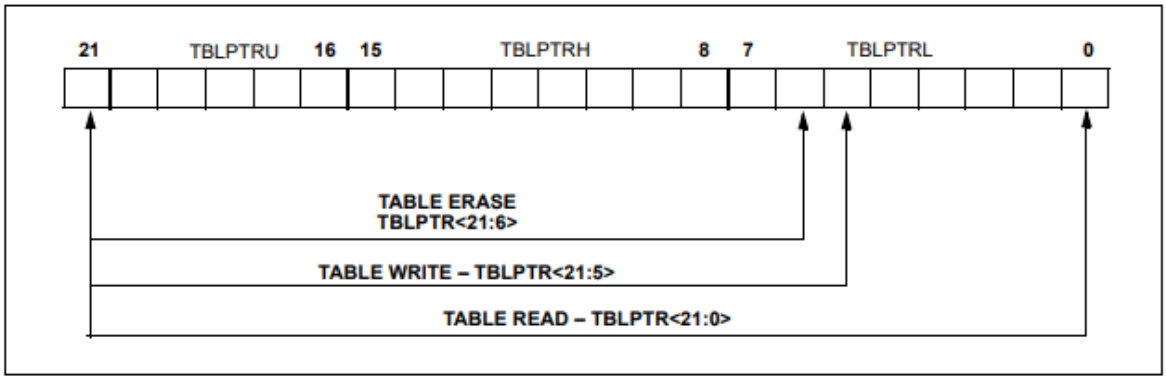


Imagen 8. Operaciones de apuntador de tabla (tabla 6 – 1) y límites de puntero de tabla basados en la operación (figura 6 – 3).

El registro TBLPTR es usado en lectura, escritura y borrado de la memoria flash del programa.

Cuando un TBLRD es ejecutado, todos los 22 bits del TBLPTR determinan cual byte es leído de la memoria del programa dentro de TABLAT.

Los cálculos utilizados para modificar el trabajo del PWM son los siguientes:

$$CC^{\cdot}PR1L:CCP1CON<5:4>=\frac{10\ ms}{8\mu seg\ \times 16}=78.125\approx 78\rightarrow(1001110)_2\rightarrow(0x13)_{16}$$

$$CC^{\cdot}PR1L:CCP1CON<5:4>=\frac{20\ ms}{8\mu seg\ \times 16}=156.25\approx 156\rightarrow(10011100)_2\rightarrow(0x27)_{16}$$

$$CC^{\cdot}PR1L:CCP1CON<5:4>=\frac{30\ ms}{8\mu seg\ \times 16}=234.375\approx 234\rightarrow(11101010)_2\rightarrow(0x3A)_{16}$$

$$CC^{\cdot}PR1L:CCP1CON<5:4>=\frac{40\ ms}{8\mu seg\ \times 16}=312.5\approx 313\rightarrow(100111001)_2\rightarrow(0x4E)_{16}$$

$$CC^{\cdot}PR1L:CCP1CON<5:4>=\frac{50\ ms}{8\mu seg\ \times 16}=390.625\approx 391\rightarrow(110000111)_2\rightarrow(0x61)_{16}$$

$$CC^{\cdot}PR1L:CCP1CON<5:4>=\frac{60\ ms}{8\mu seg\ \times 16}=468.75\approx 469\rightarrow(111010101)_2\rightarrow(0x75)_{16}$$

$$CC^{\cdot}PR1L:CCP1CON<5:4>=\frac{70\ ms}{8\mu seg\ \times 16}=546.875\approx 547\rightarrow(10001000111)_2\rightarrow(0x88)_{16}$$

$$CC^{\cdot}PR1L:CCP1CON<5:4>=\frac{80\ ms}{8\mu seg\ \times 16}=625\rightarrow(1001110001)_2\rightarrow(0x9C)_{16}$$

$$CC^{\cdot}PR1L:CCP1CON<5:4>=\frac{90\ ms}{8\mu seg\ \times 16}=703.125\approx 703\rightarrow(1010111111)_2\rightarrow(0xAF)_{16}$$

Desarrollo práctico

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

Cantidad	Concepto
1	Tarjeta de desarrollo Miuva
1	Laptop
4	Display de 7 segmentos ánodo común
1	Software MPLAB v8.92
1	Datasheet del PIC 18F4550
7	Resistencia de 220
1	Resistencia de 330
3	Resistencia de 10k
4	Resistencia de 1k
4	Transistor BC547
2	Protoboard
1	Diodo LED
3	Pushbutton
20	Jumpers
1	Mt de alambre calibre 22

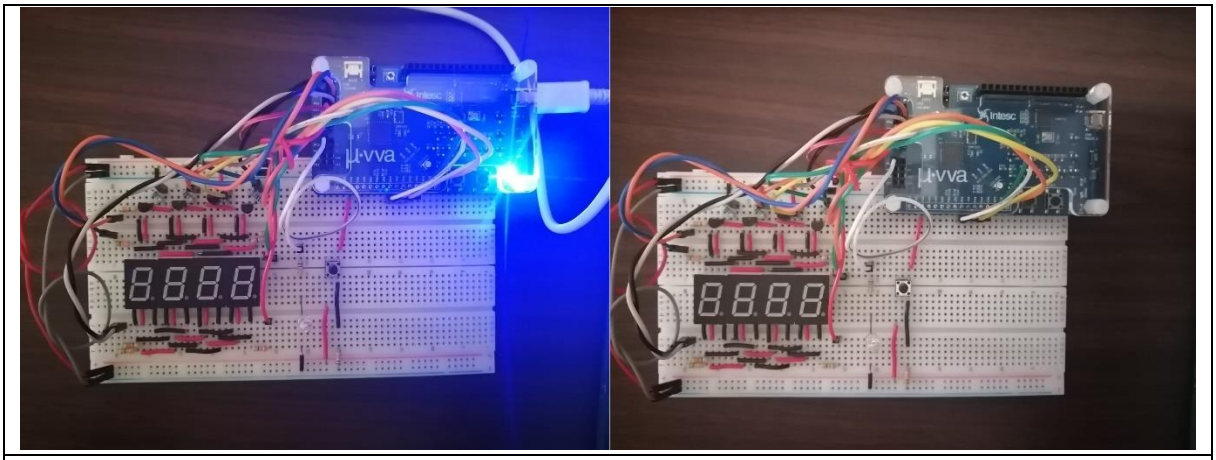


Imagen 9. Implementación del circuito y funcionamiento de este en la tarjeta de desarrollo Miuva

Explicación del código

```
;Definición de variables
        cblock  0x0
        cont
        codigo
        pwm
        displayT
        indices
        endc
```

Imagen 10. Definición de variables

En esta parte del código definimos las variables que utilizaremos más adelante.

```
; Inicio del programa principal
inicio  bcf     OSCCON, IRCF2, 0
        bsf     OSCCON, IRCF0, 0    ; Oscilador interno a 125 kHz
        movlw   0x0F
        movwf   ADCON1, 0           ; Puertos digitales
        clrf    PORTD, 0
        clrf    TRISD, 0            ; Puerto D configurado como salida
        movlw   0xF0
        movwf   TRISA               ; RA3:RA0 salidas para multiplexar displays

;CONFIGURANDO INTERRUPCIÓN 2
        bcf     TRISB, 3            ; RB3 salida
        bcf     INTCON3, INT2IF     ; INT1IF = 0
        bsf     INTCON3, INT2IE     ; Enable INT1
        bsf     INTCON, GIE
        movlw   0x0F                ; Configuramos INT0-INT2 como entradas digitales
        movwf   ADCON1
;INTERRUPCIÓN
```

Imagen 11. Configuración de puertos y de la interrupción 2.

En la imagen 11, podemos observar que se configuran los puertos de salida, así como la interrupción 2.

```
movlw   0x1C                ; 0001 1100
movwf   CCP1CON             ; modo PWM
movlw   0x07
movwf   T2CON, 0            ; Configuración preescaler timer2 x 16
movlw   d'194'
movwf   PR2, 0              ; Periodo PWM 100 ms
bcf     TRISC, 2, 0         ; PIN ccp1 SALIDA (hacia el osciloscopio)

movlw   0x03
movwf   TBLPTRH             ; TBLPTR = 0x000300
movlw   0x13
movwf   CCPR1L, 0           ; PWM a 10%

movlw   0x16
movwf   displayT
movlw   0x06
movwf   pwm
```

Imagen 12. Configuración de otros registros

En la imagen 12 podemos observar que se configuran los registros del PWM, el Timer2 y el pin 2 del puerto C que es el bit CCP1, donde observaremos el comportamiento del PWM mediante un osciloscopio.

```
start   movff   indices, TBLPTRL
        tblrd   *
        movff   TABLAT, PORTA
        movff   codigo, TBLPTRL
        tblrd   *
        movff   TABLAT, PORTD

        call    retardo
        incf    indices
        incf    codigo
        movff   codigo, WREG
        xorlw   0x03
        btfsc   STATUS, Z, 0
        bra     thrddsply
        movff   indices, WREG
        xorlw   0x13
        btfsc   STATUS, Z, 0
        bra     thrddsply
        bra     start
```

Imagen 13. Subrutina "start"

La subrutina “start” es la encargada de que los displays muestren el carácter que les corresponde de acuerdo con la posición del apuntador, además de que es donde se establece el límite para encender determinados pines del puerto A y así poder encender todos los displays.

```
thrddsply   movlw   0x00
            movwf   codigo
            movlw   0x10
            movwf   indices
            movlw   0x04
            movwf   PORTA
            movff   displayT, TBLPTRL
            tblrd   *
            movff   TABLAT, PORTD
            movff   pwm, TBLPTRL
            tblrd   *
            movff   TABLAT, CCPR1L
            call    retardo
            bra     start
```

Imagen 14. Subrutina “thrddsply”

La subrutina “thrddsply” es la encargada de que se encienda el tercer display (de ahí los nemónicos del nombre de la subrutina, *third display*). Una vez que se encendió el último display en la subrutina anterior, se ramifica a la subrutina thrddsply y reinicia los valores de las variables *código* e *índices*, de tal manera que después se encarga de mostrar el número del valor de la variable *pwm* y *displayT*, las cuales se encargan de mostrar la unidad de decena del porcentaje de trabajo del PWM.

```
retardo movlw   0x7f                ; esta rutina tarda 64 ms en ejecutarse
        movwf   cont, 0
nada    nop
        decfsz  cont, F, 0
        bra     nada
        return

RSI     bcf     INTCON3, INT2IF
        movff   displayT, WREG
        xorlw   0x1F
        btfsc   STATUS, Z, 0
        bra     clrPWM
        incf    displayT
        incf    pwm
        movff   pwm, TBLPTRL
        tblrd   *
        movff   TABLAT, CCPR1L
        movff   displayT, WREG
        xorlw   0x1F
        btfss   STATUS, Z, 0
        retfie
clrPWM  movlw   0x06
        movff   WREG, pwm
        movlw   0x16
        movff   WREG, displayT
        retfie
;*****************************************************************************
       org     0x300                        ; dirección para el código de los displays
       DB      0xa1, 0xa7, 0xc0        ; dc_0
       org     0x306                        ; dirección para el pwm
       DB      0x13, 0x27, 0x3A, 0x4e, 0x61, 0x75, 0x88, 0x9c, 0xaf
       org     0x310                        ; indica que display se encenderá
       DB      0x01, 0x02, 0x08
       org     0x316                        ; códigos del 1-9 para el tercer display del pwm
       DB      0xf9, 0xa4, 0xb0, 0x99, 0x92, 0x82, 0xb8, 0x80, 0x98
       END
```

Imagen 15. Última parte del código

Por último, se encuentra la subrtuina retardo la cuál se encarga de realizar un pequeño lapso de tiempo muerto y la subrutina de interrupción *RSI*, la cuál se encargará de aumentar el valor de las unidades de decena del tercer display y la carga de trabajo del PWM, verificando si está en 9 para saltar a 1 de nuevo.

Código completo escrito en ASM

```
        LIST P=18F4550          ;directiva para definir el procesador
                #include <P18F4550.INC>         ;definiciones de variables especificas del
procesador

;*****************************************************************************
;Bits de configuración
                        CONFIG FOSC = INTOSC_XT ;Oscilador interno para el uC, XT para USB
                        CONFIG BOR = OFF                ;Brownout reset deshabilitado
                        CONFIG PWRT = ON                ;Pwr up timer habilitado
                        CONFIG WDT = OFF                ;Temporizador vigia apagado
                        CONFIG MCLRE = OFF              ;Reset apagado
                        CONFIG PBADEN = OFF
                        CONFIG LVP = OFF
;*****************************************************************************
;Definición de variables
                cblock  0x0
                cont
                codigo
                pwm
                displayT
                indices
                endc
;*****************************************************************************
; Reset vector
                ORG             0X0000
                bra             inicio
                ORG             0x08
                bra             RSI
; Inicio del programa principal
inicio  bcf             OSCCON, IRCF2, 0
                bsf             OSCCON, IRCF0, 0        ; Oscilador interno a 125 kHz
                movlw   0x0F
                movwf   ADCON1, 0                       ; Puertos digitales
                clrf    PORTD, 0
                clrf    TRISD, 0                        ; Puerto D configurado como salida
                movlw   0xF0
                movwf   TRISA                           ; RA3:RA0 salidas para multiplexar displays

;CONFIGURANDO INTERRUPCIÓN 2
                bcf     TRISB, 3                        ; RB3 salida
                bcf             INTCON3, INT2IF                 ; INT1IF = 0
                bsf             INTCON3, INT2IE                 ; Enable INT1
                bsf             INTCON, GIE
                movlw   0x0F                            ; Configuramos INT0-INT2 como entradas
digitales
```

```
                movwf  ADCON1
;INTERRUPCIÓN
                movlw  0x1C                     ; 0001 1100
                movwf  CCP1CON                          ; modo PWM
                movlw  0x07
                movwf  T2CON, 0                 ; Configuración preescaler timer2 x 16
                movlw  d'194'
                movwf  PR2, 0                   ; Periodo PWM 100 ms
                bcf             TRISC, 2, 0             ; PIN ccp1 SALIDA (hacia el
osciloscopio)

                movlw  0x03
                movwf  TBLPTRH                          ; TBLPTR = 0x000300
                movlw  0x13
                movwf  CCPR1L, 0                ; PWM a 10%

                movlw  0x16
                movwf  displayT
                movlw  0x06
                movwf  pwm

start    movff  indices, TBLPTRL
                tblrd   *
                movff   TABLAT, PORTA
                movff   codigo, TBLPTRL
                tblrd   *
                movff   TABLAT, PORTD

                call    retardo
                incf    indices
                incf    codigo
                movff   codigo, WREG
                xorlw   0x03
                btfsc   STATUS, Z, 0
                bra             thrddsply
                movff   indices, WREG
                xorlw   0x13
                btfsc   STATUS, Z, 0
                bra             thrddsply
                bra             start

thrddsply       movlw  0x00
                        movwf  codigo
                        movlw  0x10
                        movwf  indices
                        movlw  0x04
                        movwf  PORTA
                        movff   displayT, TBLPTRL
```

```
                        tblrd   *
                        movff   TABLAT, PORTD
                        movff   pwm, TBLPTRL
                        tblrd   *
                        movff   TABLAT, CCPR1L
                        call    retardo
                        bra             start
;*************************************************************************
*****
retardo movlw  0x7f                             ; esta rutina tarda 64 ms en ejecutarse
                movwf  cont, 0
nada    nop
                decfsz  cont, F, 0
                bra             nada
                return

RSI             bcf             INTCON3, INT2IF
                movff   displayT, WREG
                xorlw   0x1F
                btfsc   STATUS, Z, 0
                bra             clrPWM
                incf    displayT
                incf    pwm
                movff   pwm, TBLPTRL
                tblrd   *
                movff   TABLAT, CCPR1L
                movff   displayT, WREG
                xorlw   0x1F
                btfss   STATUS, Z, 0
                retfie
clrPWM          movlw  0x06
                movff   WREG, pwm
                movlw  0x16
                movff   WREG, displayT
                retfie
;*************************************************************************
*****
                org             0x300                                   ; dirección para el
código de los displays
                DB              0xa1, 0xa7, 0xc0                ; dc_0
                org             0x306                                   ; dirección para el
pwm
                DB              0x13, 0x27, 0x3A, 0x4e, 0x61, 0x75, 0x88, 0x9c, 0xaf
                org             0x310                                   ; indica que display
se encenderá
                DB              0x01, 0x02, 0x08
                org             0x316                                   ; códigos del 1-9
para el tercer display del pwm
```

```
DB          0xf9, 0xa4, 0xb0, 0x99, 0x92, 0x82, 0xb8, 0x80, 0x98
ENDConclusiones
```

Conclusión

Con la realización de esta práctica podemos ahora controlar el porcentaje de trabajo de nuestro PIC18F4550, además de reforzar nuestros conocimientos en apuntadores.

Link del vídeo: https://youtu.be/FZx2BRsWy_Q

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

Rafiquzzaman, M., 2007. *Microprocessors theory and applications*. 2nd ed. New Delhi, II: Prentice-Hall of India.