



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO



FACULTAD DE INGENIERÍA

MATERIA

- Laboratorio de Microcomputadoras
 - Grupo:04

PRÁCTICA 06

Puertos paralelos E/S

PROFESOR

- M.I. Rubén Anaya García

ALUMNOS

- Carreón Guzmán Mariana Ivette
 - Núm. Cta.: 312103914
 - Gpo. Teoría: 04
 - Rojas Méndez Gabriel
 - Gpo. Teoría: 01
 - Núm. Cta.: 314141712

SEMESTRE 2022-1

Objetivo.

Familiarizar al alumno con el uso y aplicación del Convertidor Analógico/Digital de un microcontrolador.

Introducción.

El microcontrolador PIC16F877 tiene 8 posibles canales de entrada por los cuales se pueden procesar señales analógicas de 10 bits de resolución.

Los registros involucrados para este periférico son los mostrados a continuación, la dirección y banco donde están ubicados se pueden consultar en la información dada en la práctica uno.

- Registro de configuración ADCON0 0x1F
- Registro de configuración ADCON1 0x9F
- Registro de resultados parte alta ADRESH 0x1E
- Registro de resultados parte baja ADRESL 0x9E

El algoritmo para emplear para el uso del convertidor A/D, con resolución de 8 bits:

1. Ubicado en el banco cero, limpiar el puerto A, usando CLRF PORTA.
2. Cambiar al banco uno.
3. Configurar el puerto A como entradas analógicas, escribir 00H al registro ADCON1.
4. Regresar al banco 0.
5. Realizar la configuración de la fuente de reloj, el canal de entrada y prender al convertidor A/D, en el registro ADCON0.
6. #.
7. Generar un tiempo de retardo de 20 microsegundos.
8. Esperar a que GO/DONE# sea igual a cero, lo que indica que ha concluido el proceso de conversión.
9. Lee el resultado de la conversión del registro ADRESH.

Desarrollo

Ejercicio 1

Empleando el canal de su elección del convertido A/D, realizar un programa en el cuál, de acuerdo a una entrada analógica que se ingrese por este canal, se represente el resultado de la conversión en un puerto paralelo utilizar el arreglo de leds para ver la salida, como se muestra en la tabla.

```

PROCESSOR 16F877
INCLUDE<P16F877.INC>

VAL EQU H'20' ;VARIABLE PARA EL DELAY

ORG 0H
GOTO INICIO

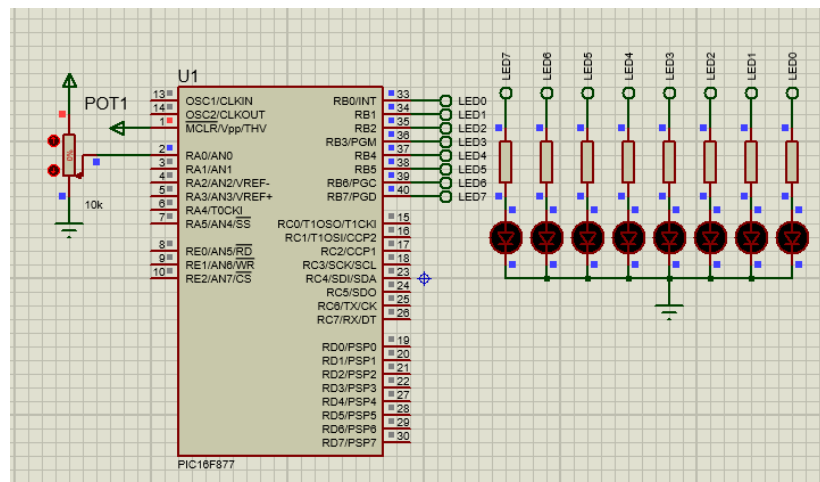
ORG 05H
INICIO: CLRF PORTA ;LIMPIA EL PUERTO A
        CLRF PORTE ;LIMPIA EL PUERTO A
        BSF STATUS,RP0
        BCF STATUS,RP1 ;CAMBIO LA BANCO 1
        MOVLW 00H
        MOVWF ADCON1 ;CONFIGURA PUERTO A Y E COMO ANALÓGICOS
        MOVLW 3FH
        MOVWF TRISA ;CONFIGURA EL PUERTO A COMO ENTRADA
        MOVLW H'0'
        MOVWF TRISB ;CONFIGURA PUERTO B COMO SALIDA
        BCF STATUS,RP0 ;REGRESA AL BANCO 0
        MOVLW B'11000001' ;CONFIGURACIÓN ADCON0
        MOVWF ADCON0 ;ADCS1=1 ADCS0=1 CHS2=0 CHS1=0 CHS0=0
                        ; GO/DONE=0 - ADON=1
CICLO:  BSF ADCON0,2 ;CONVERSIÓN EN PROGRESO GO=1
        CALL RETARDO ;ESPERA QUE TERMINE LA CONVERSIÓN

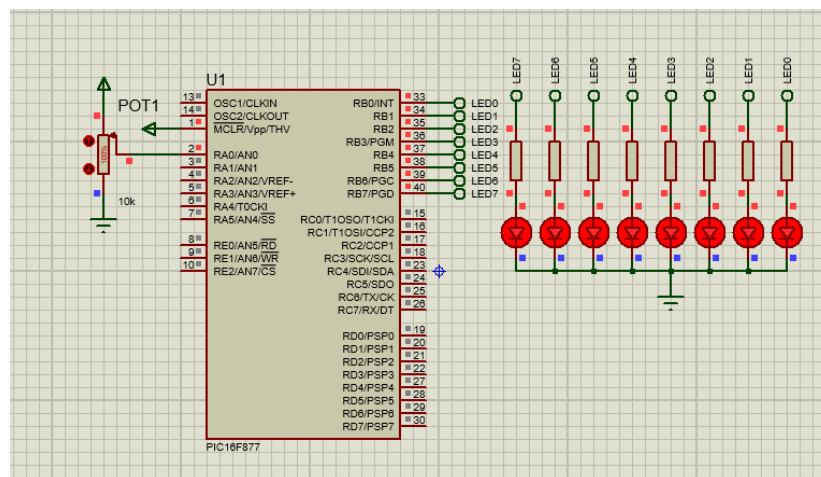
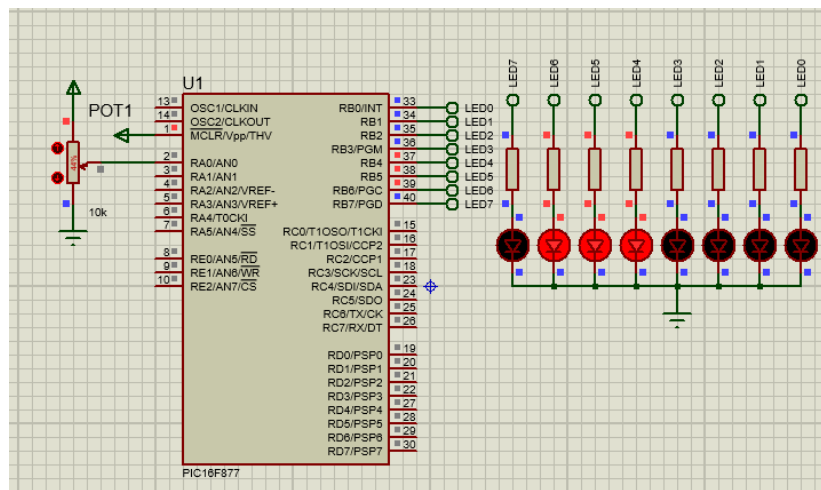
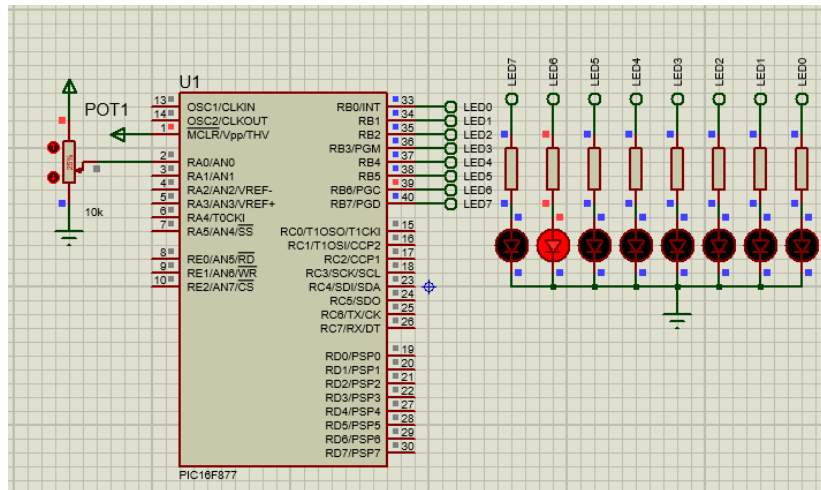
ESPERA: BTFSF ADCON0,2 ;PREGUNTA POR DONE=0?(TERMINÓ CONVERSIÓN?)
        GOTO ESPERA ;NO, VUELVE A PREGUNTAR
        MOVF ADRESH,0 ;SI
        MOVWF PORTE ;MUESTRA EL RESULTADO EN PORTE
        GOTO CICLO

RETARDO: MOVLW H'30' ;DELAY DE 20 MICROSEG APROX.
        MOVWF VAL

LOOP:   DECFSZ VAL
        GOTO LOOP
        RETURN

```





Ejercicio 2

Utilizando el circuito anterior, realizar un programa que indique el rango en el cual se encuentra el voltaje a la entrada del convertidor canal seleccionado. Mostrar el valor en un display de 7 segmentos.

Entrada Analógica Ve	Salida
0 – 0.99 V	0
1.0 – 1.99 V	1
2.0 – 2.99 V	2
3.0 – 3.99 V	3
4.00 – 4.80 V	4
4.80 – 5.00 V	5

```

PROCESSOR 16F877
INCLUDE<P16F877.INC>

VAL EQU H'20'      ; VARIABLES PARA EL DELAY

ORG 0H
GOTC INICIO

ORG 05H
INICIO: CLRF PORTA
        CLRF PORTE
        BSF STATUS,RP0 ;CAMBIA LA BANCO 1
        BCF STATUS,RP1
        MOVWL 00H      ;CONFIGURA PUERTOS A Y E COMO ANALÓGICOS
        MOVWF ADCON1
        MOVWL 3FH      ;CONFIGURA EL PUERTO A COMO ENTRADA
        MOVWF TRISA
        MOVWL H'0'
        MOVWF TRISE
        BCF STATUS,RP0 ;CONFIGURA PUERTO B COMO SALIDA
        MOVWL B'11000001' ;REGRESA AL BANCO 0
        MOVWF ADCON0    ;CONFIGURACIÓN ADCON0
        ;ADCS1=1 ADCS0=1 CHS2=0 CHS1=0 CHS0=0
        ;GO/DONE=0 - ADON=1

CICLO:  BSF ADCON0,2    ;CONVERSIÓN EN PROGRESO GO=1
        CALL RETARDO   ;ESPERA QUE TERMINE LA CONVERSIÓN

ESPERA: BTFSC ADCON0,2 ;PREGUNTA POR DONE(TERMINO CONVERSIÓN?)
        GOTC ESPERA    ;NO, VUELVE A PREGUNTAR
        MOVF ADRESH,0  ;MUEVE VE A W (VE=VOLTAJE DE ENTRADA)
        SUBLW 130H     ;REALIZA 0CAH - W (1/5VCC=0CCH)
        BTFSC STATUS,0 ;VERIFICA SI W ES MENOR (CARRY=0?)
        GOTC SALIDA0   ;SI, VE A SALIDA1 (VE < 1/3VCC)
        MOVF ADRESH,0  ;NO, MUEVE VE A W
        SUBLW 265H     ;194H - W (2/3VCC=155H)
        BTFSC STATUS,0 ;VERIFICA SI W ES MENOR (CARRY=0)
        GOTC SALIDA1   ;SI, SE CUMPLE QUE 1/3VCC<VE< 2/3VCC, NO, VERIFICA QUE W MENOR QUE VCC
        MOVF ADRESH,0  ;MUEVE VE A W
        SUBLW 398H     ;3FFH - W (VCC=3FFH)
        BTFSC STATUS,0 ;VERIFICA SI W ES MENOR A VCC (CARRY=0?)
        GOTC SALIDA2   ;SI, SE CUMPLE QUE 2/3VCC<VE<VCC
        MOVF ADRESH,0  ;MUEVE VE A W
        SUBLW 3CAH     ;3FFH - W (VCC=3FFH)
        BTFSC STATUS,0 ;VERIFICA SI W ES MENOR A VCC (CARRY=0?)
        GOTC SALIDA3   ;MUEVE VE A W
        MOVF ADRESH,0  ;3FFH - W (VCC=3FFH)
        SUBLW H'3F7'   ;VERIFICA SI W ES MENOR A VCC (CARRY=0?)
        BTFSC STATUS,0 ;MUEVE VE A W
        GOTC SALIDA4   ;3FFH - W (VCC=3FFH)
        MOVF ADRESH,0  ;VERIFICA SI W ES MENOR A VCC (CARRY=0?)
        SUBLW H'3FF'   ;MUEVE VE A W
        BTFSC STATUS,0 ;3FFH - W (VCC=3FFH)
        GOTC SALIDA5   ;VERIFICA SI W ES MENOR A VCC (CARRY=0?)

SALIDA0: MOVWL H'0'
        MOVWF PORTE
        GOTC CICLO

SALIDA1: MOVWL H'1'
        MOVWF PORTE
        GOTC CICLO

SALIDA2: MOVWL H'2'
        MOVWF PORTE
        GOTC CICLO

SALIDA3: MOVWL H'3'
        MOVWF PORTE
        GOTC CICLO

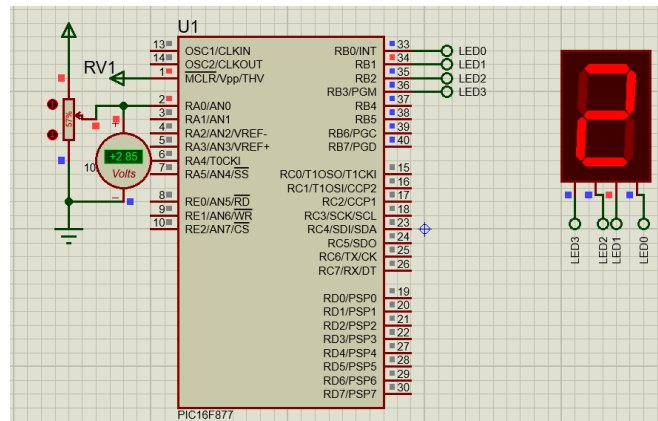
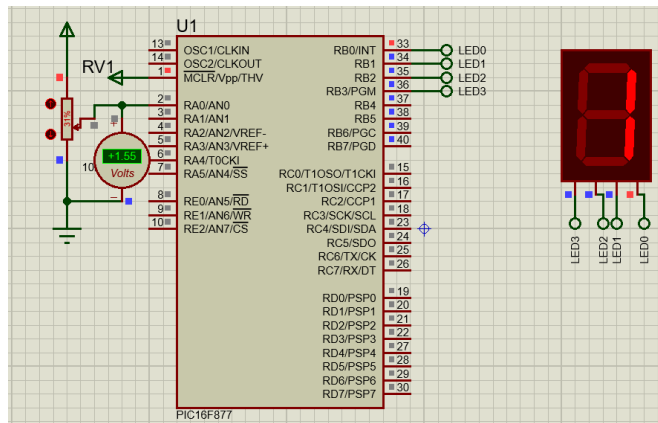
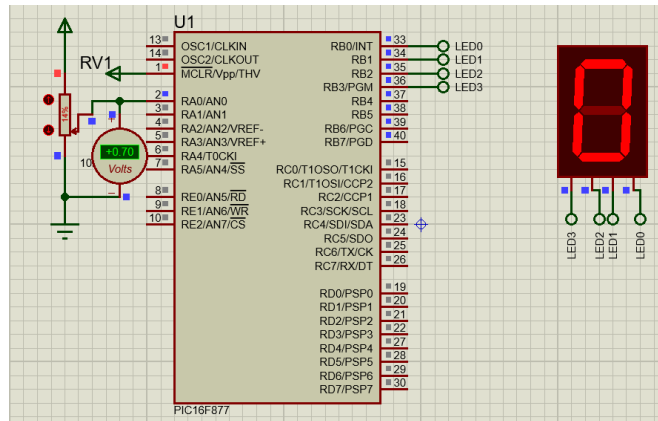
SALIDA4: MOVWL H'4'
        MOVWF PORTE
        GOTC CICLO

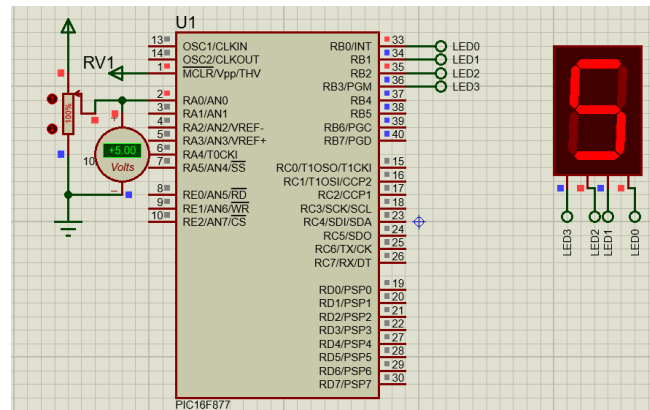
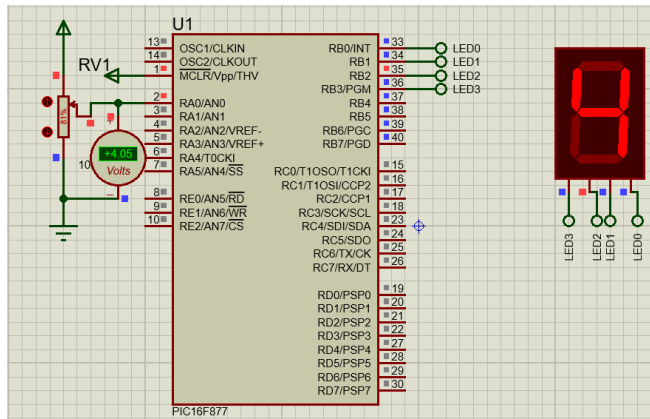
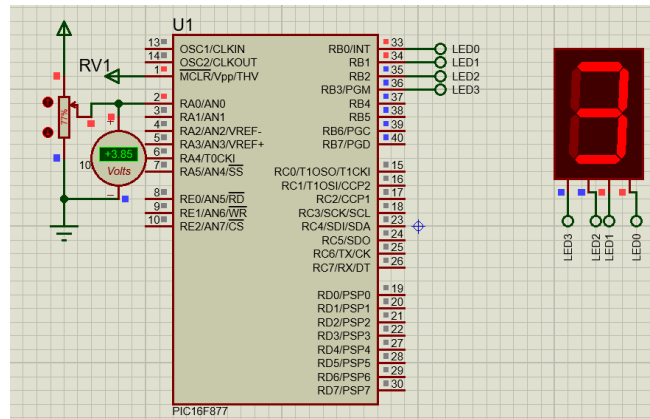
SALIDA5: MOVWL H'5'
        MOVWF PORTE
        GOTC CICLO

RETARDO: MOVWL H'30' ;DELAY DE 20 MICROSEG APROX.
        MOVWF VAL

LOOP:   DECFSZ VAL
        GOTC LOOP
        RETURN
        END

```





Ejercicio 3

Realizar un programa, de manera que identifique cuál de tres señales analógicas que ingresan al convertidor A/D es mayor que las otras dos; representar el resultado de acuerdo con el contenido de la tabla.

Señal	PB2	PB1	PB0
Ve1>Ve2 y Ve3	0	0	1
Ve2>Ve1 y Ve3	0	1	1
Ve3>Ve1 y Ve2	1	1	1

```

PROCESSOR 16F877
INCLUDE<P16F877.INC>

VAL EQU H'20'    ;VARIABLES PARA EL DELAY
CAL1 EQU H'21'   ;CANALES
CAL2 EQU H'22'
CAL3 EQU H'23'

ORG 0H
GOTO INICIO

ORG 05H
INICIO: CLRF PORTA
        CLRF PORTB
        BSF STATUS,RP0    ;CAMBIA LA BANCO 1
        BCF STATUS,RP1
        MOVLW 00H         ;CONFIGURA PUERTOS A Y E COMO ANALÓGICOS
        MOVWF ADCON1
        MOVLW 3FH         ;CONFIGURA EL PUERTO A COMO ENTRADA
        MOVWF TRISA
        MOVLW H'0'
        MOVWF TRISE       ;CONFIGURA PUERTO B COMO SALIDA
        BCF STATUS,RP0    ;REGRESA AL BANCO 0

CICLO:  MOVLW B'11000001'  ;CONFIGURACIÓN ADCON0 CANAL 0
        MOVWF ADCON0
        BSF ADCON0,2       ;CONVERSION EN PROGRESO GO=1
        CALL RETARDO       ;ESPERA QUE TERMINE LA CONVERSIÓN

ESPERA1: BTFSF ADCON0,2    ;PREGUNTA POR DONE(TERMINO CONVERSIÓN?)
        GOTO ESPERA1
        MOVF ADRESH,0      ;MUEVE A W
        MOVWF CAL1         ;GUARDAMOS EL VALOR EN CAL1
        MOVLW B'11001001' ;CONFIGURACIÓN ADCON0 CANAL 1
        MOVWF ADCON0
        BSF ADCON0,2       ;CONVERSIÓN EN PROGRESO GO=1
        CALL RETARDO       ;ESPERA QUE TERMINE LA CONVERSIÓN

```



```

ESPERA2:    BTFSF ADCON0,2    ;PREGUNTA POR DONE(TERMINO CONVERSIÓN?)
            GOTO ESPERA1
            MOVF ADRESH,0    ;MUEVE A W
            MOVWF CAL2       ;GUARDAMOS EL VALOR EN CAL2
            MOVLW B'11010001' ;CONFIGURACIÓN ADCON0 CANAL 1
            MOVWF ADCON0
            BSF ADCON0,2     ;CONVERSIÓN EN PROGRESO GO=1
            CALL RETARDO     ;ESPERA QUE TERMINE LA CONVERSIÓN

ESPERA3:    BTFSF ADCON0,2    ;PREGUNTA POR DONE(TERMINO CONVERSIÓN?)
            GOTO ESPERA3
            MOVF ADRESH,0    ;MUEVE A W
            MOVWF CAL3       ;GUARDAMOS EL VALOR EN CAL3
            MOVF CAL1, W     ;MUEVE CAL1 A W
            SUBWF CAL2, W     ;RESTA CAL 2 - CAL1
            BTFSF STATUS, C   ;VERIFICA SI W ES MAYOR
            GOTO CAL1CAL2    ;SI CAL1 > CAL2
            GOTO CAL2CAL1    ;SI CAL2 < CAL1

CAL1CAL2:   MOVF CAL1, W
            SUBWF CAL3, W
            BTFSF STATUS, C
            GOTO SALIDA1
            GOTO SALIDA3

CAL2CAL1:   MOVF CAL2, W
            SUBWF CAL3, W
            BTFSF STATUS, C
            GOTO SALIDA2
            GOTO SALIDA3

SALIDA1:    MOVLW H'1'       ;SALIDA 0001
            MOVWF PORTB
            GOTO CICLO

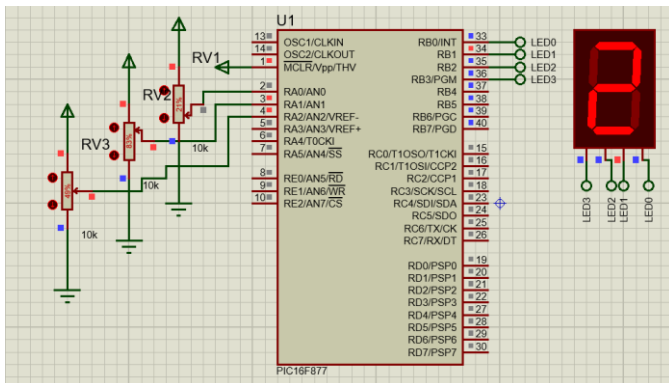
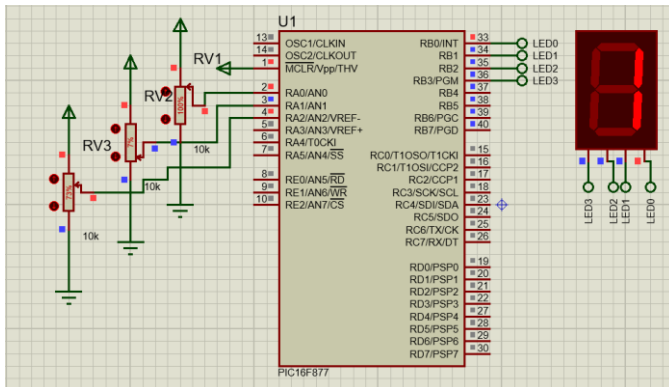
SALIDA2:    MOVLW H'2'       ;SALIDA 0010
            MOVWF PORTB
            GOTO CICLO

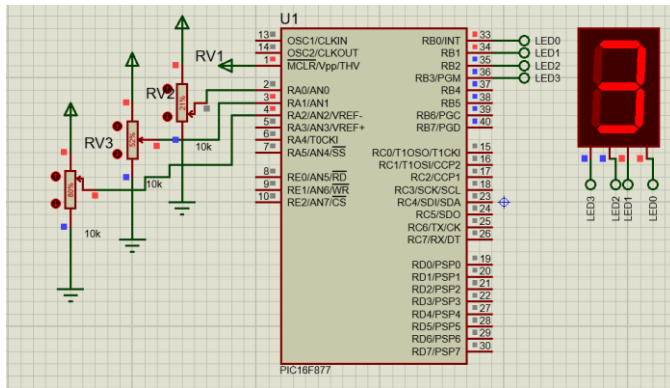
SALIDA3:    MOVLW H'3'       ;SALIDA 0011
            MOVWF PORTB
            GOTO CICLO

RETARDO:    MOVLW H'30'     ;DELAY DE 2MICROSEG APROX.
            MOVWF VAL

LOOP:       DECFSZ VAL
            GOTO LOOP
            RETURN
            END

```





Conclusiones

Mariana Carreón Guzmán

En esta práctica aprendí a utilizar un convertidor analógico, en el primer caso fue sencillo, terminé de comprender el uso de los puertos entrada y salida porque aún tenía algunas dudas respecto a ello y pude ir desarrollando un poco más los convertidores analógicos. En cuanto al código pude aplicar lo visto en otras prácticas para el diseño y desarrollo de esta práctica.

Rojas Méndez Gabriel

En este caso pudimos seguir desarrollando nuestras habilidades con el lenguaje ensamblador y emplear convertidores analógicos para poder controlar un display de 7 segmentos, seguimos ocupando Proteus y cada vez me doy cuenta que comprendo mejor cómo ir desarrollando un controlador PIC.