



**UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO**  
**FACULTAD DE INGENIERIA**  
**LABORATORIO DE MICROCOMPUTADORAS**

**PREVIO N° 6**  
**CONVERTIDOR ANALÓGICO / DIGITAL**

**Alumnos:**

**Alcantar Correa Vianey.**

Grupo Teoria (3)

**Sánchez Rosas Alexis Alejandro.**

Grupo Teoria (1)

**Velazquez Martínez Karla Andrea.**

Grupo Teoria (4)

**Profesor: Moisés Melendez Reyes**

**Grupo: 01**

**FECHA DE ENTREGA: 14/04/2024**

## Objetivo.

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

## Introducción

El convertidor analógico a digital (ADC) es un componente electrónico esencial que permite que los sistemas digitales interactúen con el mundo analógico. Los ADC transforman señales analógicas, como voltajes de corriente continua que varían continuamente, en representaciones digitales que pueden ser procesadas por microcontroladores y otros dispositivos digitales. Esta tecnología es fundamental en numerosas aplicaciones, desde la automatización industrial y dispositivos médicos hasta los sistemas de comunicaciones y electrodomésticos inteligentes, facilitando la integración entre los entornos físicos y los sistemas computacionales.

## Desarrollo de la práctica

### 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 figura 6.1.

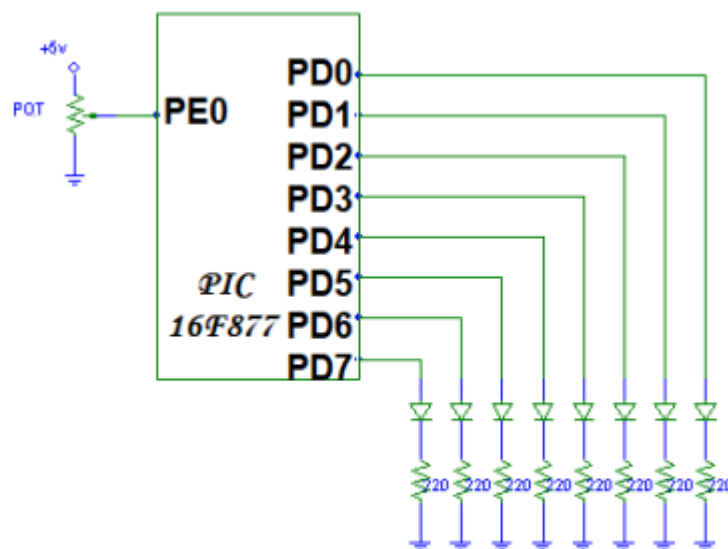


Figura 6.1 Circuito con lectura de una señal analógica



<div>Propuesta de solución</div>	<pre>include &lt;p16f877.inc&gt;  VAL      EQU      0x20      ;Localidad de memoria 0x20 asignada a registro VAL           ORG      0           GOTO     INICIO           ORG      5  INICIO:  CLRF      PORTA      ;Limpia registro PORTA nos permite recibir la señal analógica           BSF      STATUS,RP0 ;Mueve al banco 1           BCF      STATUS,RP1 ;Configura registro PORTB como salida           CLRF      TRISD      ;Limpio ADCON1, defino que el formato de resultado se carga en ADRESH, y puertos A y D como analógicos           CLRF      ADCON1      ;Muevo al banco 0           BCF      STATUS,RP0           MOVLW     B'111101001'           MOVWF     ADCON0      ;ADCON0 &lt;- 11 (frecuencia interna) 001 (QUé canal A1) 0 (Go/Done) 0(No existe) 1(convertidor)           CLRF      PORTD           REPITE: BSF      ADCON0,2 ;GO/Done realiza la conversión                   CALL     RETARDO ;Rutina de retardo que consume el tiempo necesario           ESPERA: BTTFSC    ADCON0,2 ;Espera hasta que la conversión este terminada                   GOTO     ESPERA                   MOVF      ADRESH,W ;ADRESH contiene el resultado de la conversión                   MOVWF     PORTD    ;Transfiero el resultado de la conversión al puertoB                   GOTO     REPITE    ;Para obtener el Voltaje de entrada hay que multiplicar la salida por 19.5 [mV]            RETARDO:                   MOVLW     0x250                   MOVWF     VAL           LOOP:   DECFSZ    VAL                   GOTO     LOOP                   RETURN            END</pre>
<div>Ensamblado correctamente</div>	<div>Release build of project 'C:\Users\Alexis\Desktop\FI\Microcomputadoras\Lab\P6\e1.disposable_mcp' started. Language tool versions: MPASMWIN.exe v5.51, mplink.exe v4.49, mplib.exe v4.49 Sun Apr 14 21:21:26 2024</div> <div>Clean: Deleting intermediary and output files. Clean: Done. Executing: "C:\Program Files (x86)\Microchip\MPASM Suite\MPASMWIN.exe" /q /p16F877A "e1.asm" /I"e1.lst" /e"e1.err" Warning[205] C:\USERS\ALEXIS\DESKTOP\FI\MICROCOMPUTADORAS\LAB\P6\E1.ASM 1 : Found directive in column 1. (include) Message[301] C:\PROGRAM FILES (X86)\MICROCHIP\MPASM SUITE\P16F877.INC 33 : MESSAGE: (Processor-header file mismatch. Verify selected processor.) Message[302] C:\USERS\ALEXIS\DESKTOP\FI\MICROCOMPUTADORAS\LAB\P6\E1.ASM 11 : Register in operand not in bank 0. Ensure that bank bits are correct. Message[302] C:\USERS\ALEXIS\DESKTOP\FI\MICROCOMPUTADORAS\LAB\P6\E1.ASM 12 : Register in operand not in bank 0. Ensure that bank bits are correct. Warning[202] C:\USERS\ALEXIS\DESKTOP\FI\MICROCOMPUTADORAS\LAB\P6\E1.ASM 26 : Argument out of range. Least significant bits used. Message[305] C:\USERS\ALEXIS\DESKTOP\FI\MICROCOMPUTADORAS\LAB\P6\E1.ASM 28 : Using default destination of 1 (file). Executing: "C:\Program Files (x86)\Microchip\MPASM Suite\mplink.exe" /p16F877A "e1.o" /z__MPLAB_BUILD=1 /o"e1.cof" /M"e1.map" /W /x MPLINK 4.49, Linker Device Database Version 1.14 Copyright (c) 1998-2011 Microchip Technology Inc. Errors : 0</div> <div>Loaded C:\Users\Alexis\Desktop\FI\Microcomputadoras\Lab\P6\e1.cof.</div> <div>Release build of project 'C:\Users\Alexis\Desktop\FI\Microcomputadoras\Lab\P6\e1.disposable_mcp' succeeded. Language tool versions: MPASMWIN.exe v5.51, mplink.exe v4.49, mplib.exe v4.49 Sun Apr 14 21:21:27 2024</div> <div>BUILD SUCCEEDED</div>

Resultados:

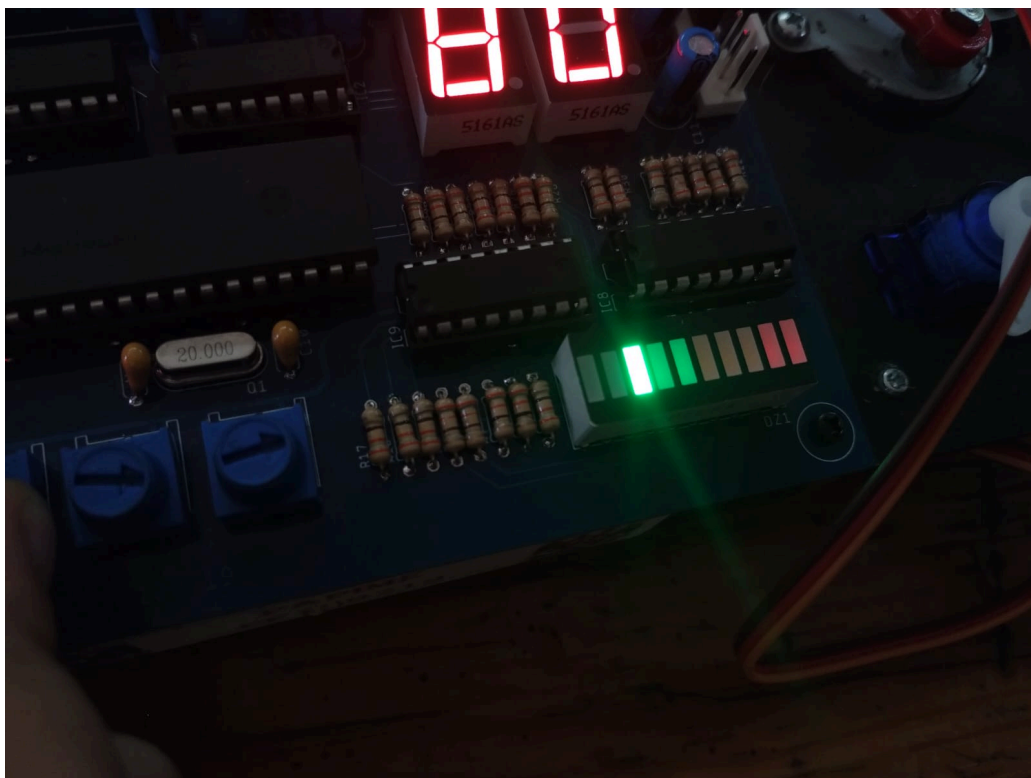
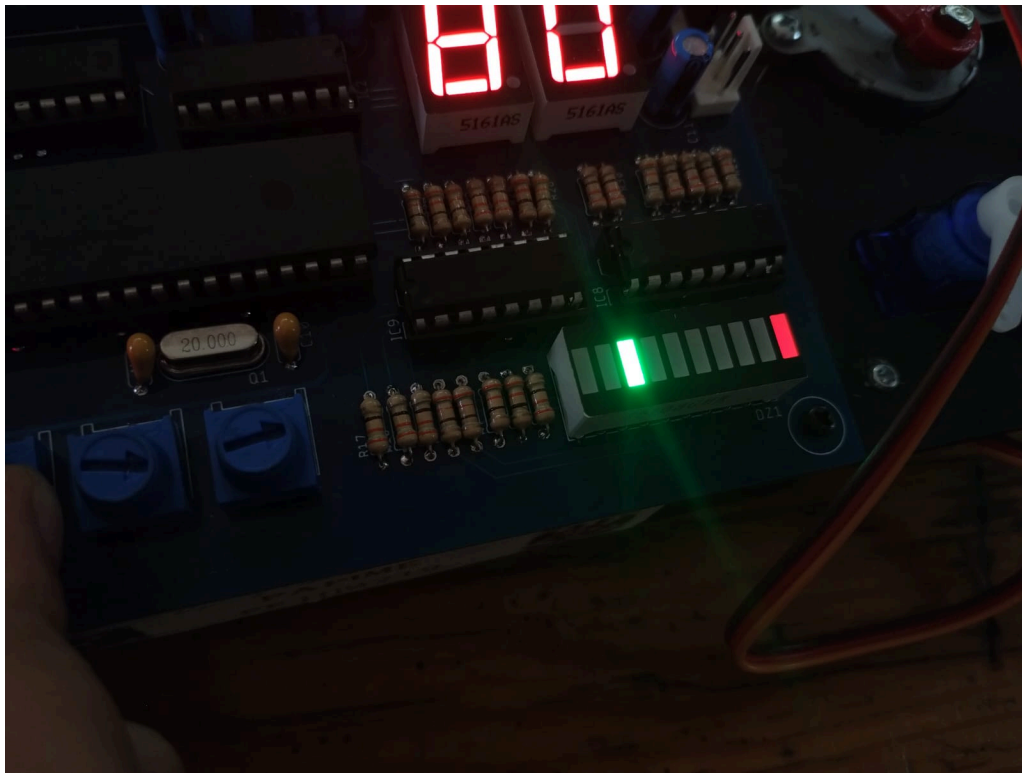
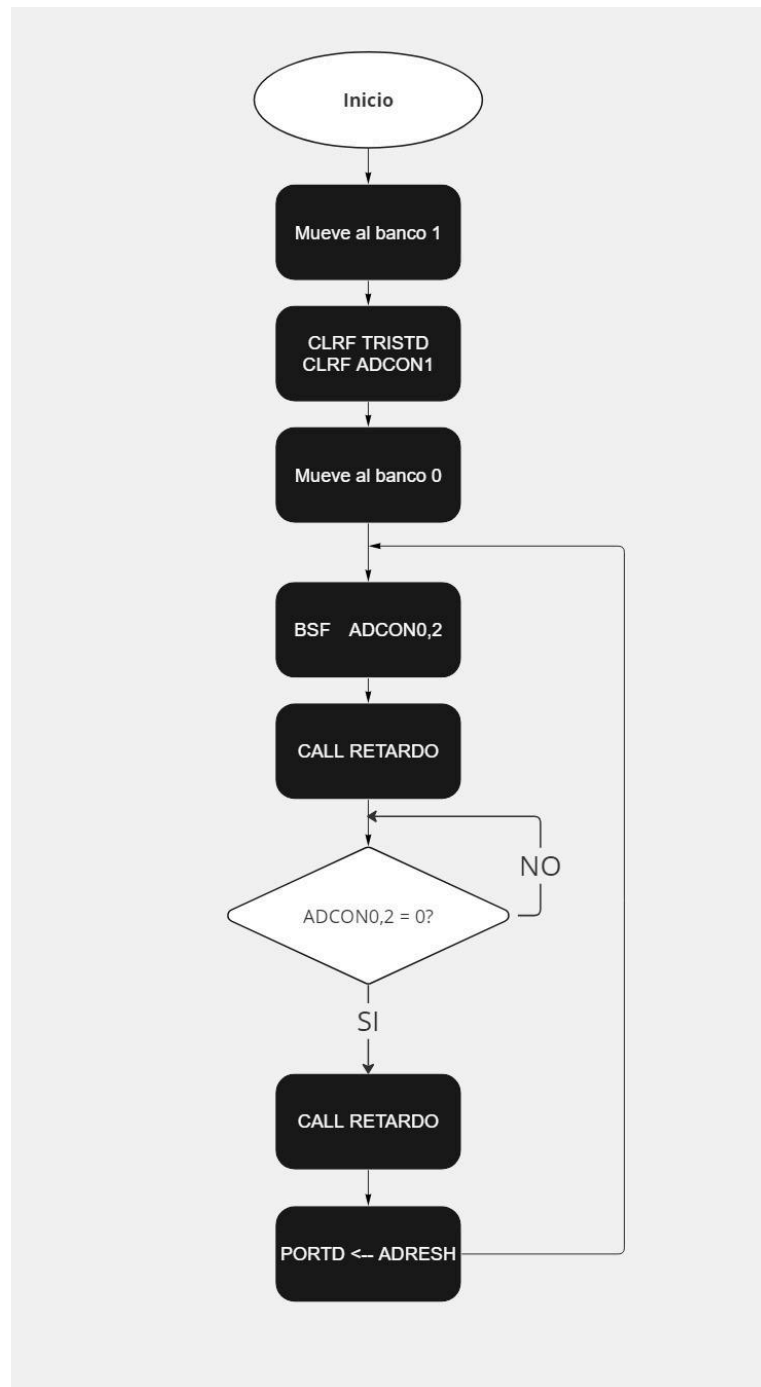


Diagrama de flujo del código:



## Ejercicio 2

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

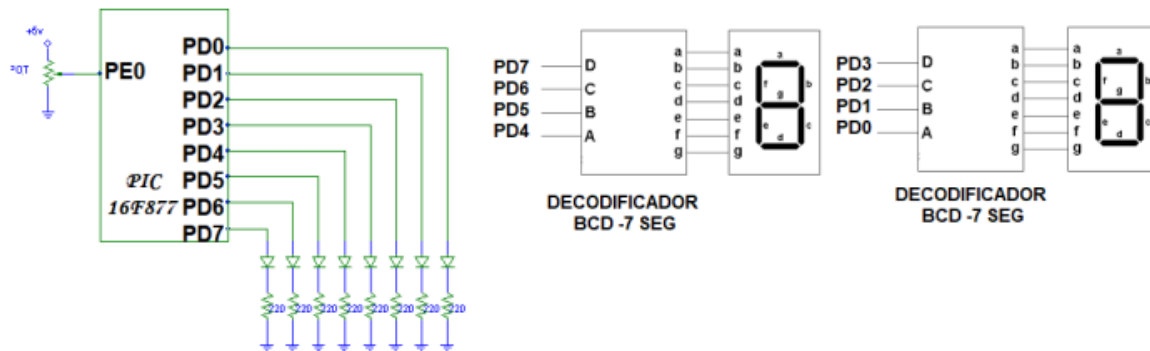


Figura 6.2 Circuito actividad 2

Entrada Analógica $V_e$	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

Tabla 6.1  
Donde  $V_{cc} = 5$  volts



## Propuesta de solución

otor.

```
include <p16f877.inc>

VAL    EQU    0x20    ;Localidad de memoria 0x20 asignada a registro VAL
AUX    EQU    0x21

        ORG    0
        GOTO   INICIO
        ORG    5

INICIO: CLRF    PORTA    ;Limpia registro PORTA nos permite recibir la señal analógica
        BSF    STATUS,RP0
        BCF    STATUS,RP1 ;Mueve al banco 1
        CLRF    TRISD    ;Configura registro PORTB como salida
        CLRF    ADCON1    ;Limpio ADCON1, defino que el formato de resultado se carga en ADRESH,
        BCF    STATUS,RP0 ;Muevo al banco 0
        MOVLW   B'11101001'
        MOVWF   ADCON0    ;ADCON0 <- 11 (frecuencia interna) 001 (QUé canal A1) 0 (Go/Done) 0(No
        CLRF    PORTD
REPITE: BSF    ADCON0,2    ;GO/Done realiza la conversión
        CALL    RETARDO    ;Rutina de retardo que consume el tiempo necesario
ESPERA: BTFSC   ADCON0,2    ;Espera hasta que la conversión este terminada
        GOTO    ESPERA
        MOVFW   ADRESH    ;ADRESH contiene el resultado de la conversión
        MOVWF   AUX
        MOVLW   H'33'
        SUBWF   AUX,W      ;REG_A - 33h -> W
        BTFSS   STATUS,C    ;ADRESH >= 33h ?
        GOTO    MENOR_1
MAYOR_1: ;ADRESH >= 1
        MOVLW   H'66'
        SUBWF   AUX,W
        BTFSS   STATUS,C    ;ADRESH >= 66h ?
        GOTO    MENOR_2
MAYOR_2: ;ADRESH >= 2
        MOVLW   H'99'
        SUBWF   AUX,W
        BTFSS   STATUS,C    ;ADRESH >= 99h ?
        GOTO    MENOR_3
MAYOR_3: ;ADRESH >= 3
        MOVLW   H'CC'
        SUBWF   AUX,W
        BTFSS   STATUS,C    ;ADRESH >= CCh ?
        GOTO    MENOR_4
```



```
MAYOR_4: ;ADRESH >= 4
        MOVLW    H'F5'
        SUBWF    AUX,W
        BTFSS    STATUS,C      ;ADRESH >= F5h ?
        GOTO     MENOR_4_80
        ;ADRESH > 4.80
        MOVLW    0X05
        MOVWF    PORTD
        GOTO     REPITE
MENOR_1: ;ADRESH < 1
        MOVLW    0X00
        MOVWF    PORTD
        GOTO     REPITE
MENOR_2: ;1 <= ADRESH < 2
        MOVLW    0X01
        MOVWF    PORTD
        GOTO     REPITE
MENOR_3: ;2 <= ADRESH < 3
        MOVLW    0X02
        MOVWF    PORTD
        GOTO     REPITE
MENOR_4: ;3 <= ADRESH < 4
        MOVLW    0X03
        MOVWF    PORTD
        GOTO     REPITE
MENOR_4_80: ;4 <= ADRESH < 4.80
        MOVLW    0X04
        MOVWF    PORTD
        GOTO     REPITE      ;Para obtener el Voltaje de entrada hay que multip

RETARDO:
        MOVLW    0X250
        MOVWF    VAL
LOOP:   DECFSZ   VAL
        GOTO     LOOP
        RETURN

END
```

Ensamblado  
correctamente

```
Release build of project 'C:\Users\Alexis\Desktop\FI\Microcomputadoras\Lab\P6\e2.disposable_mcp' started.
Language tool versions: MPASMWIN.exe v5.51, mplink.exe v4.49, mplib.exe v4.49
Sun Apr 14 21:22:09 2024

Clean: Deleting intermediary and output files.
Clean: Done.
Executing: "C:\Program Files (x86)\Microchip\MPASM Suite\MPASMWIN.exe" /q /p16F877A "e2.asm" /"e2.lst" /e"e2.err"
Warning[205] C:\USERS\ALEXIS\DESKTOP\FI\MICROCOMPUTADORAS\LAB\P6\E2.ASM 1 : Found directive in column 1. (include)
Message[301] C:\PROGRAM FILES (X86)\MICROCHIP\MPASM SUITE\P16F877.INC 33 : MESSAGE: (Processor-header file mismatch. Verify selected processor.)
Message[302] C:\USERS\ALEXIS\DESKTOP\FI\MICROCOMPUTADORAS\LAB\P6\E2.ASM 13 : Register in operand not in bank 0. Ensure that bank bits are correct.
Message[302] C:\USERS\ALEXIS\DESKTOP\FI\MICROCOMPUTADORAS\LAB\P6\E2.ASM 14 : Register in operand not in bank 0. Ensure that bank bits are correct.
Warning[202] C:\USERS\ALEXIS\DESKTOP\FI\MICROCOMPUTADORAS\LAB\P6\E2.ASM 75 : Argument out of range. Least significant bits used.
Message[305] C:\USERS\ALEXIS\DESKTOP\FI\MICROCOMPUTADORAS\LAB\P6\E2.ASM 77 : Using default destination of 1 (file).
Executing: "C:\Program Files (x86)\Microchip\MPASM Suite\mplink.exe" /p16F877A "e2.o" /z__MPLAB_BUILD=1 /o"e2.cof" /M"e2.map" /W/x
MPLINK 4.49. Linker
Device Database Version 1.14
Copyright (c) 1998-2011 Microchip Technology Inc.
Errors : 0

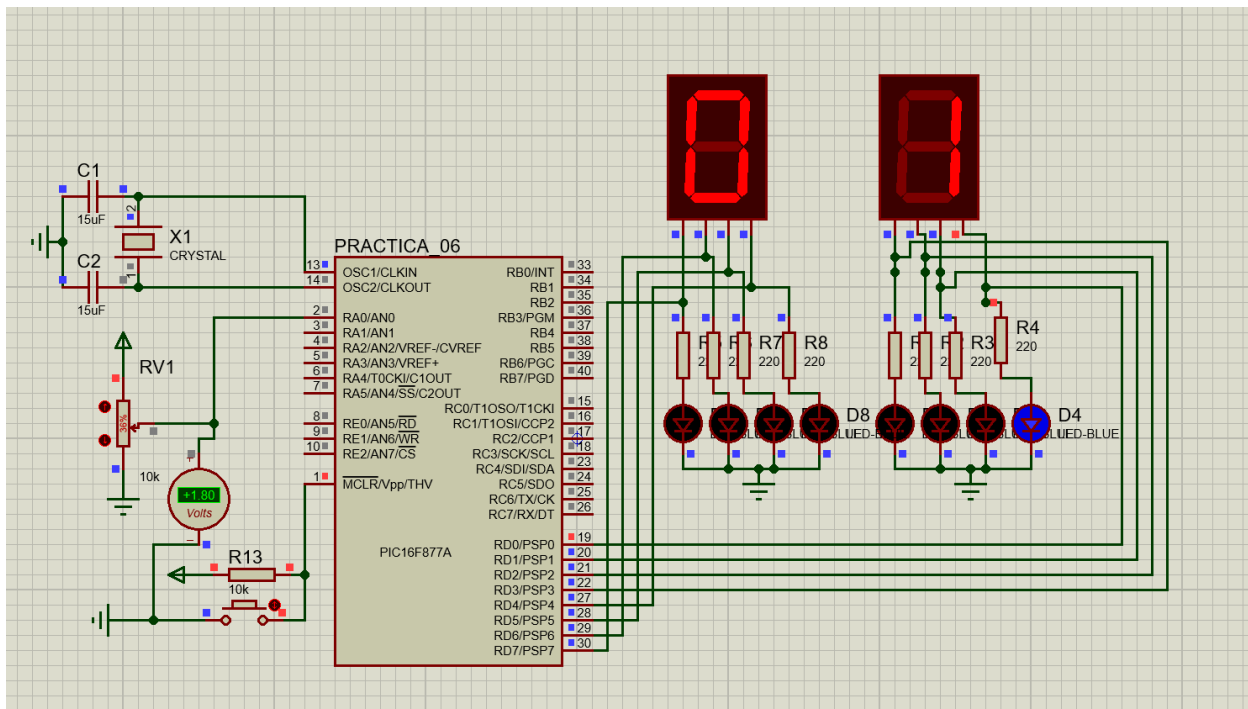
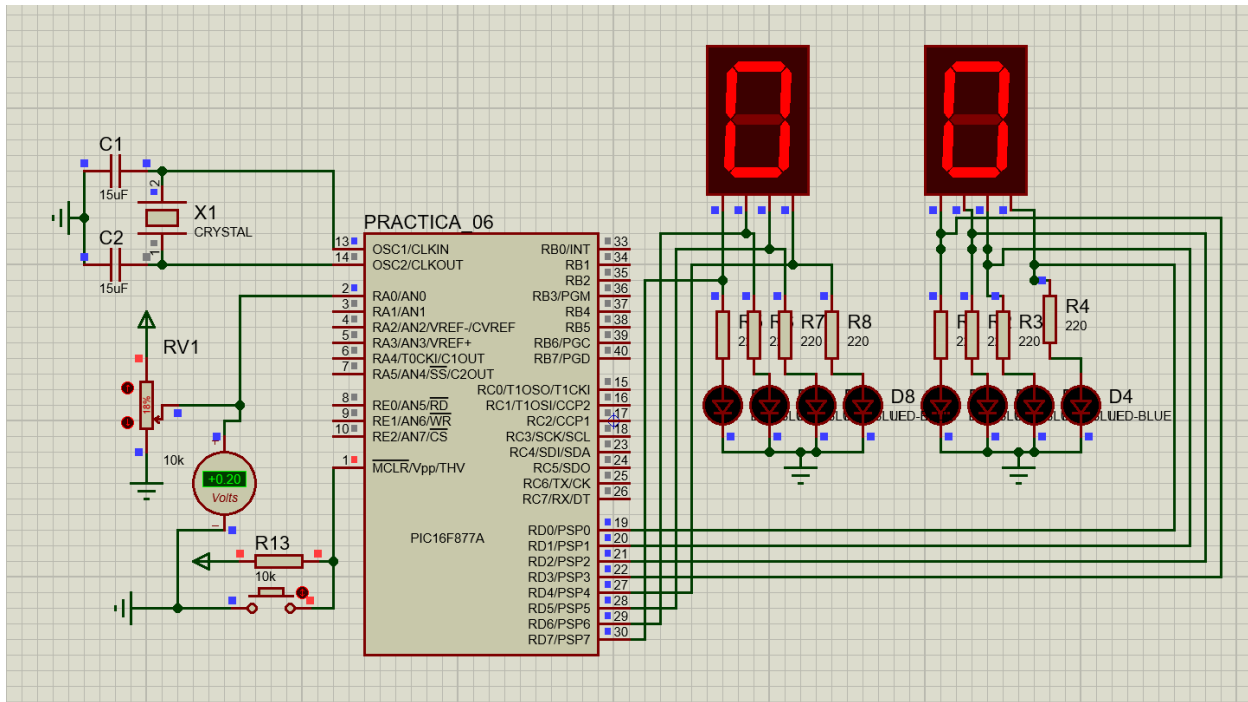
Loaded C:\Users\Alexis\Desktop\FI\Microcomputadoras\Lab\P6\e2.cof.

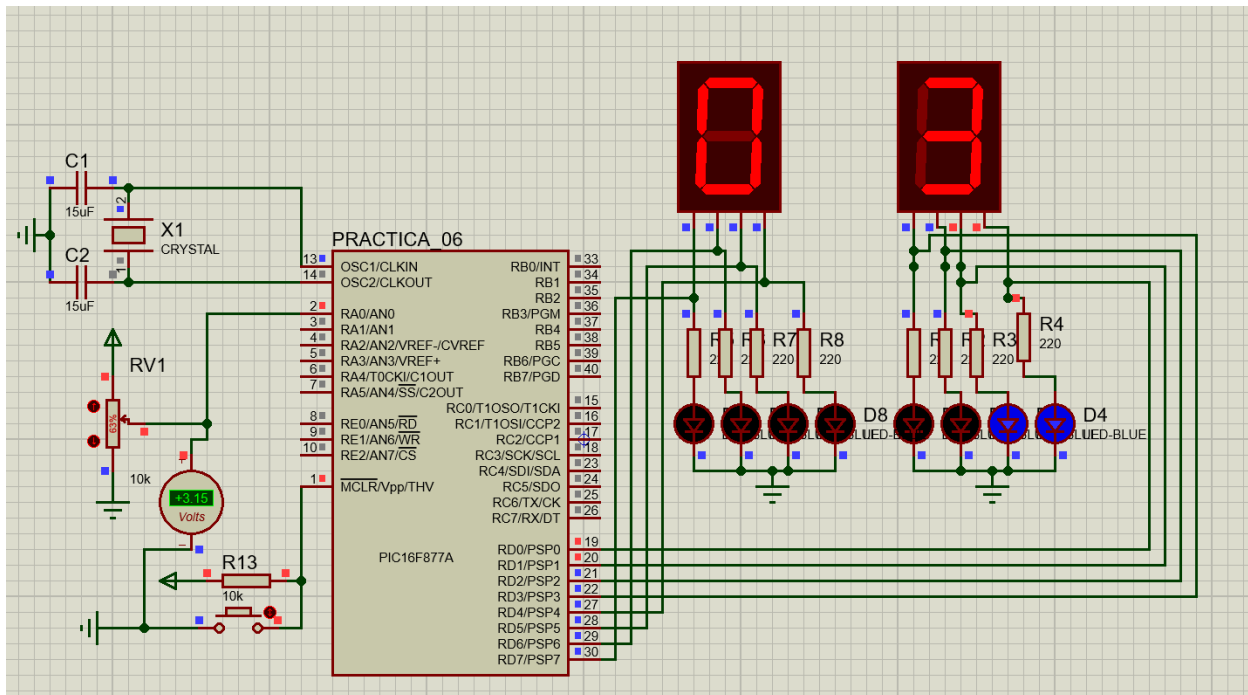
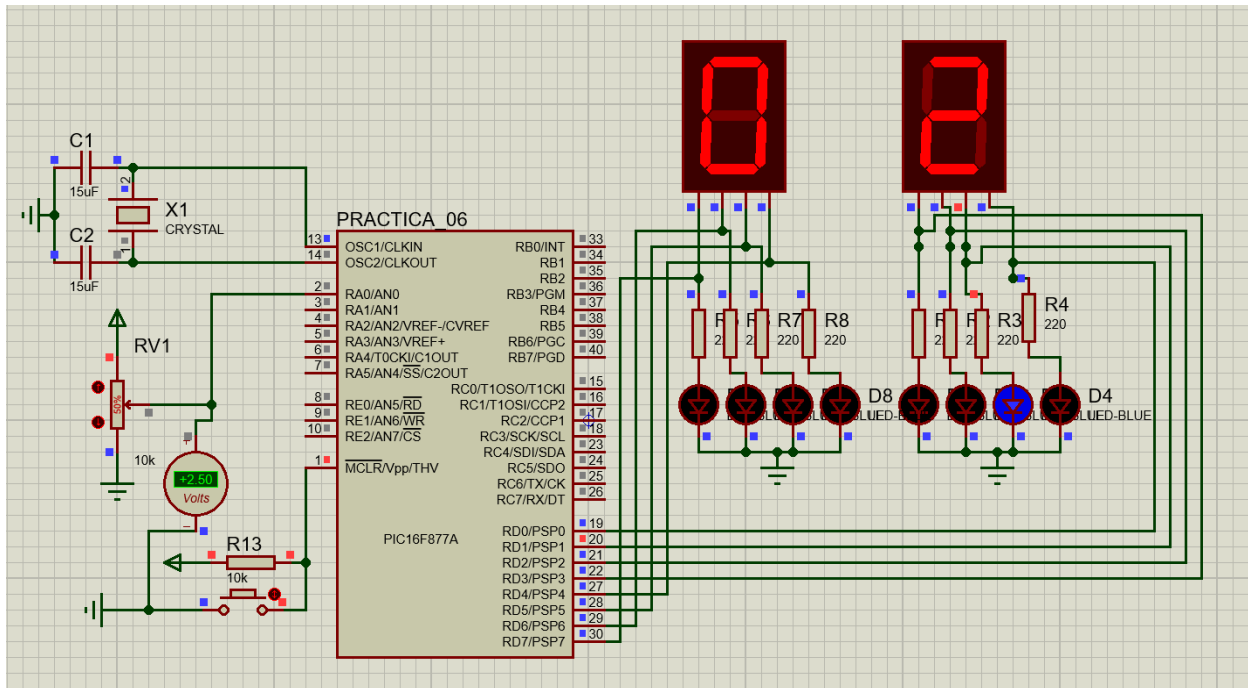
Release build of project 'C:\Users\Alexis\Desktop\FI\Microcomputadoras\Lab\P6\e2.disposable_mcp' succeeded.
Language tool versions: MPASMWIN.exe v5.51, mplink.exe v4.49, mplib.exe v4.49
Sun Apr 14 21:22:10 2024

BUILD SUCCEEDED
```

Resultados:







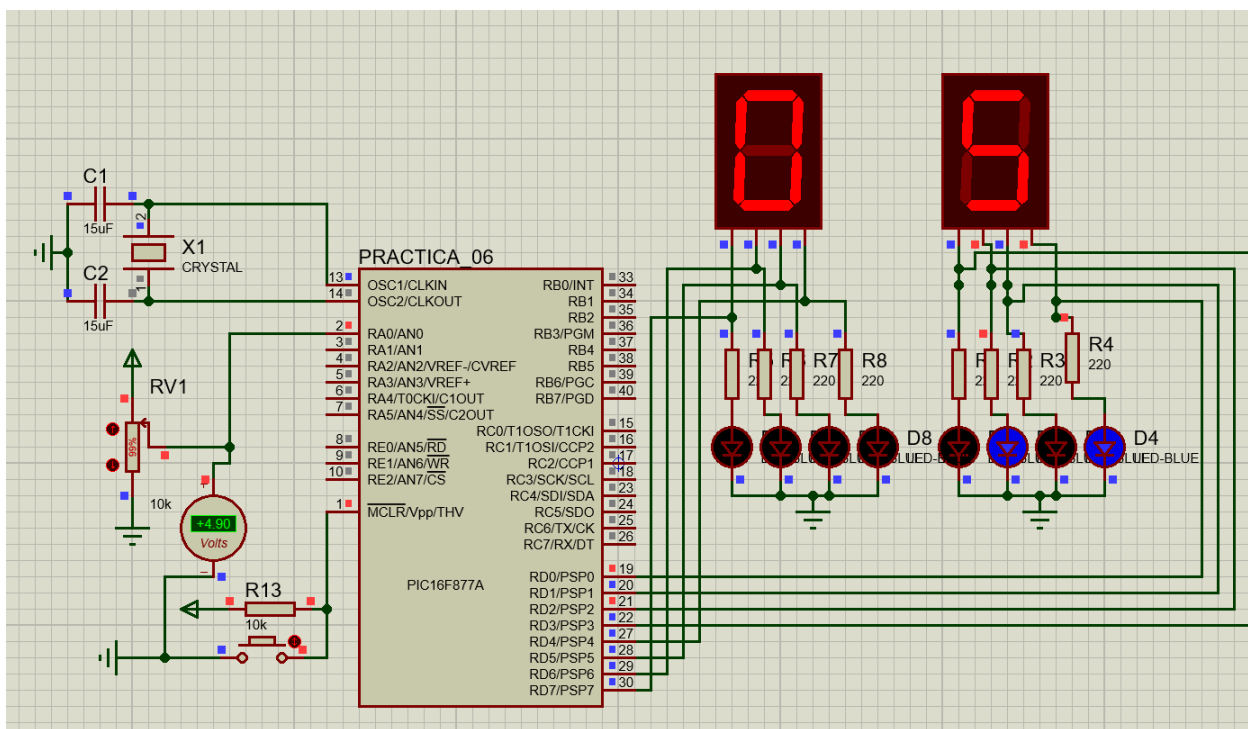
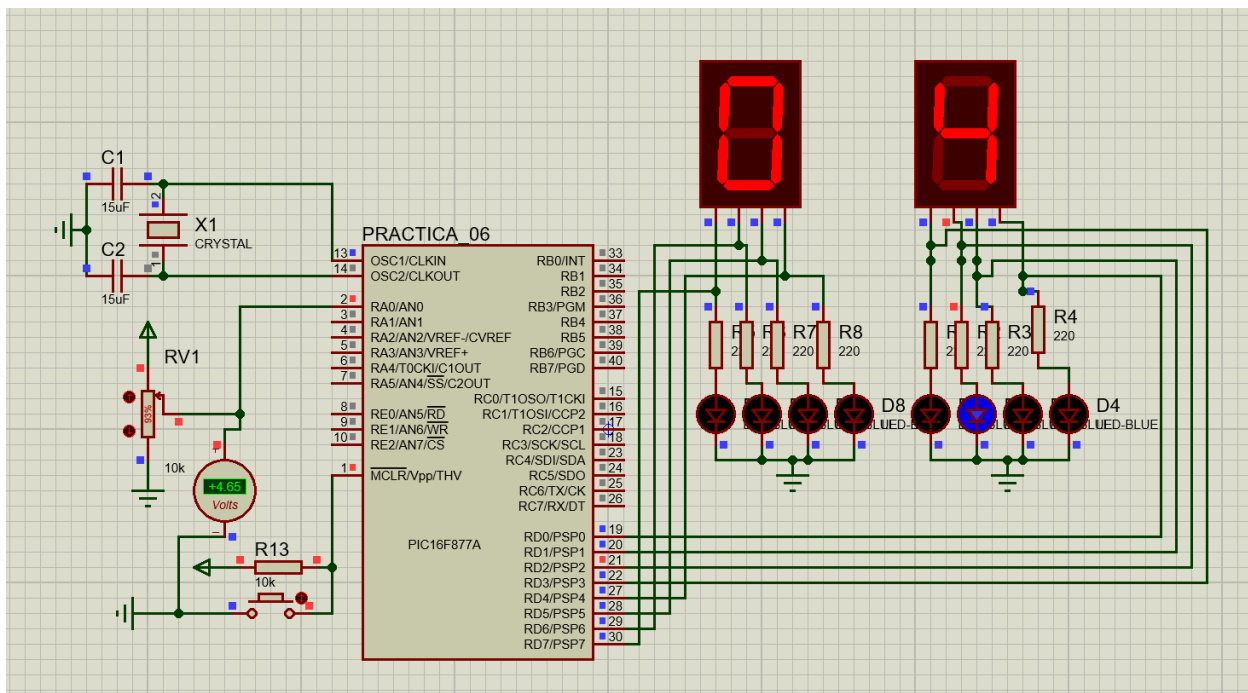
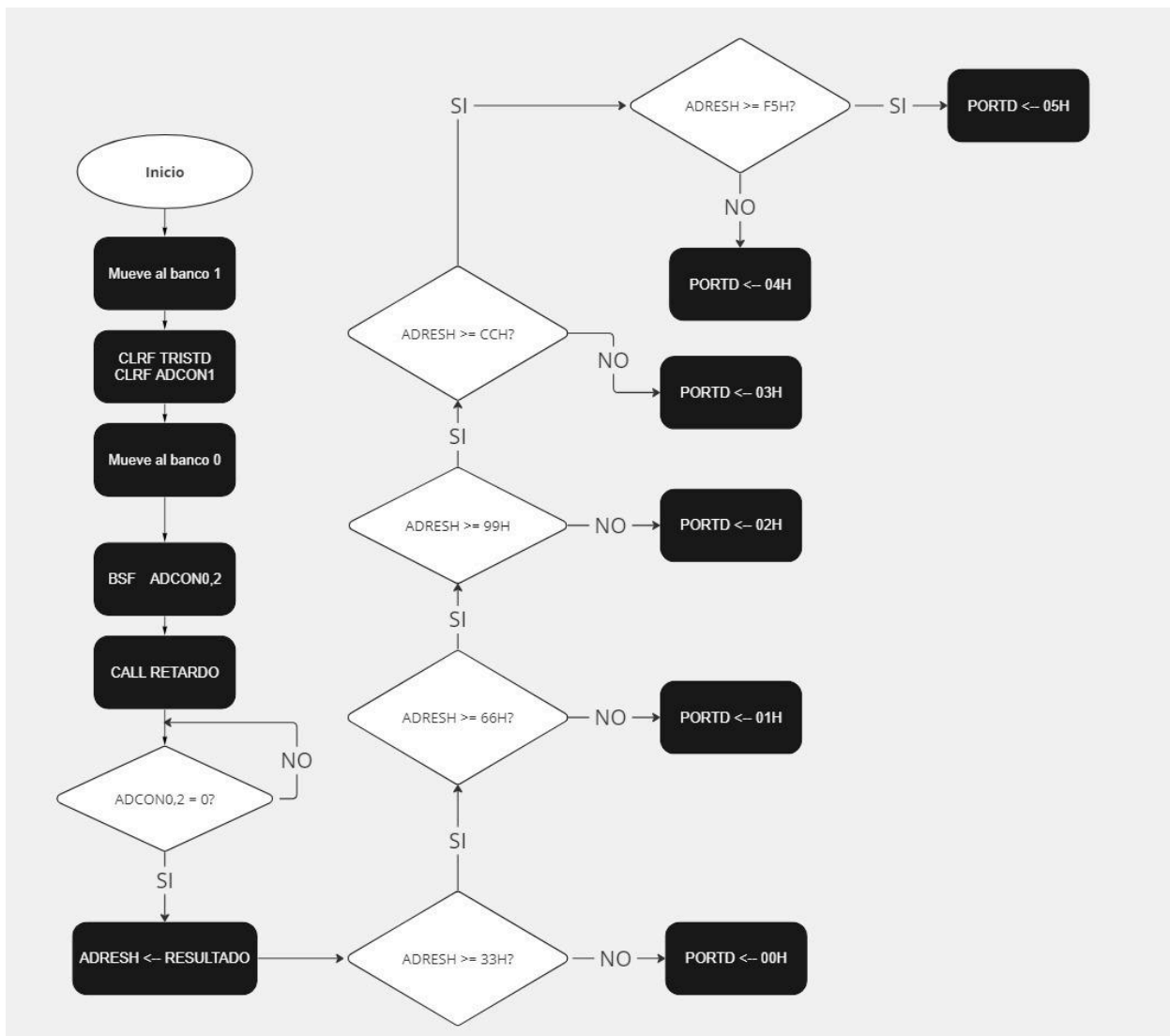


Diagrama de flujo del código:



### 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 al contenido de la tabla 6.2.

Señal	PD2	PD1	PD0
Ve1>Ve2 y Ve3	0	0	1
Ve2>Ve1 y Ve3	0	1	1
Ve3>Ve1 y Ve2	1	1	1

Tabla 6.2

Circuito empleado para este ejercicio.

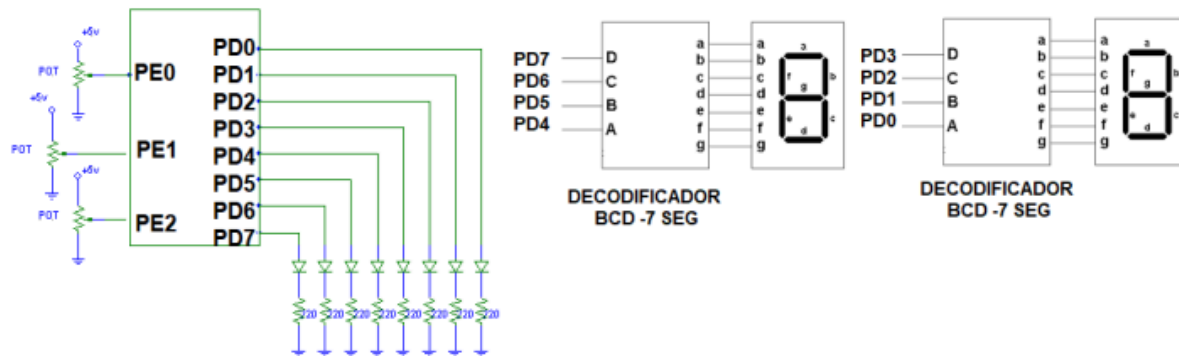


Figura 6.3 Tres señales analógicas

## Propuesta de solución

```

include <p16f877.inc>

VAL EQU 0x20 ;Localidad de memoria 0x20 asignada a registro VAL
VAL2 EQU 0x21
AUX EQU 0x22
AUX2 EQU 0x23
AUX3 EQU 0x24

ORG 0
OOTO INICIO
ORG 5

INICIO: CLRF PORTA ;Limpia registro PORTA nos permite recibir la señal analógica
        BSF STATUS,RP0
        BCF STATUS,RP1 ;Mueve al banco 1
        CLRF TRISD ;Configura registro PORTB como salida
        CLRF ADCON1 ;Limpio ADCON1, defino que el formato de resultado se carga en ADRESH, y puertos A y D como analógicos
        BCF STATUS,RP0 ;Muevo al banco 0
        CLRF PORTD
        ;Para el canal 5
REPITE: BCF STATUS,C ;Limpiamos el carry

        MOVLW B'11101001'
        MOVWF ADCON0 ;ADCON0 <- 11 (frecuencia interna) 001 (QUé canal A1) 0 (Go/Done) 0(No existe) 1(convertidor)
        BSF ADCON0,2 ;GO/DONE INICIA CONVERSION
        CALL RETARDO ;SE GENERA RETARDO
        BCF ADCON0,2 ;La conversion termina resultado en reg ADRESH
        MOVWF ADRESH ;Guarda resultado en registro W
        MOVWF AUX ;Guardamos conversion en var auxiliar
        ;Para el canal 6
        MOVLW B'11110001'
        MOVWF ADCON0 ;ADCON0 <- 11 (frecuencia interna) 001 (QUé canal A1) 0 (Go/Done) 0(No existe) 1(convertidor)
        BSF ADCON0,2 ;GO/DONE INICIA CONVERSION
        CALL RETARDO ;SE GENERA RETARDO
        BCF ADCON0,2 ;La conversion termina resultado en reg ADRESH
        MOVWF ADRESH ;Guarda resultado en registro W
        MOVWF AUX2 ;Guardamos conversion en var auxiliar2
        ;Para el canal 7
        MOVLW B'11111001'
        MOVWF ADCON0 ;ADCON0 <- 11 (frecuencia interna) 001 (QUé canal A1) 0 (Go/Done) 0(No existe) 1(convertidor)
        BSF ADCON0,2 ;GO/DONE INICIA CONVERSION
        CALL RETARDO ;SE GENERA RETARDO
        BCF ADCON0,2 ;La conversion termina resultado en reg ADRESH
        MOVWF ADRESH ;Guarda resultado en registro W
        MOVWF AUX3 ;Guardamos conversion en var auxiliar3
    
```



```
;Consiguiendo el voltaje mayor
SUBWF AUX2,W           ;CANAL 6 - CANAL 7
BTFSC STATUS,C         ;Si carry es 1 canal 6 > 7
GOTO CANAL6_7          ;Si canal 6 es mayor a canal 7
MOVFW AUX3             ;W <- Canal 7
SUBWF AUX,W            ;Canal 5 - Canal 7
BTFSC STATUS,C         ;Si carry es 1 canal 5 > 7
GOTO CANAL5_7          ;Si canal 5 es mayor a canal7
GOTO CANAL7_MAYOR      ;Si canal 7 es el canal mayor

CANAL6_7:
MOVFW AUX2             ;W <- Canal 6
SUBWF AUX,W            ;Canal 5 - Canal 6
BTFSC STATUS,C         ;Si el canal 5 es mayor al canal 6
GOTO CANAL5_MAYOR     ;Carry 0, 5 es mayor
GOTO CANAL6_MAYOR     ;Carry 1, 6 es mayor

CANAL5_7:
MOVFW AUX              ;W <- Canal 5
SUBWF AUX2,W           ;Canal 5 - Canal 6
BTFSC STATUS,C         ;Si el canal 6 es mayor al canal 5
GOTO CANAL6_MAYOR     ;Carry 0, 6 es mayor
GOTO CANAL5_MAYOR     ;Carry 1, 5 es mayor

CANAL5_MAYOR:
MOVLW b'00000001'
MOVWF PORTD
GOTO REPITE

CANAL6_MAYOR:
MOVLW b'00000011'
MOVWF PORTD
GOTO REPITE

CANAL7_MAYOR:
MOVLW b'00000111'
MOVWF PORTD
GOTO REPITE
```

Ensamblado  
correctamente

```
Release build of project 'C:\Users\Alexis\Desktop\FI\Microcomputadoras\Lab\P6\p63.disposable_mcp' started.
Language tool versions: MPASMWIN.exe v5.51, mplink.exe v4.49, mplib.exe v4.49
Sun Apr 14 21:24:21 2024

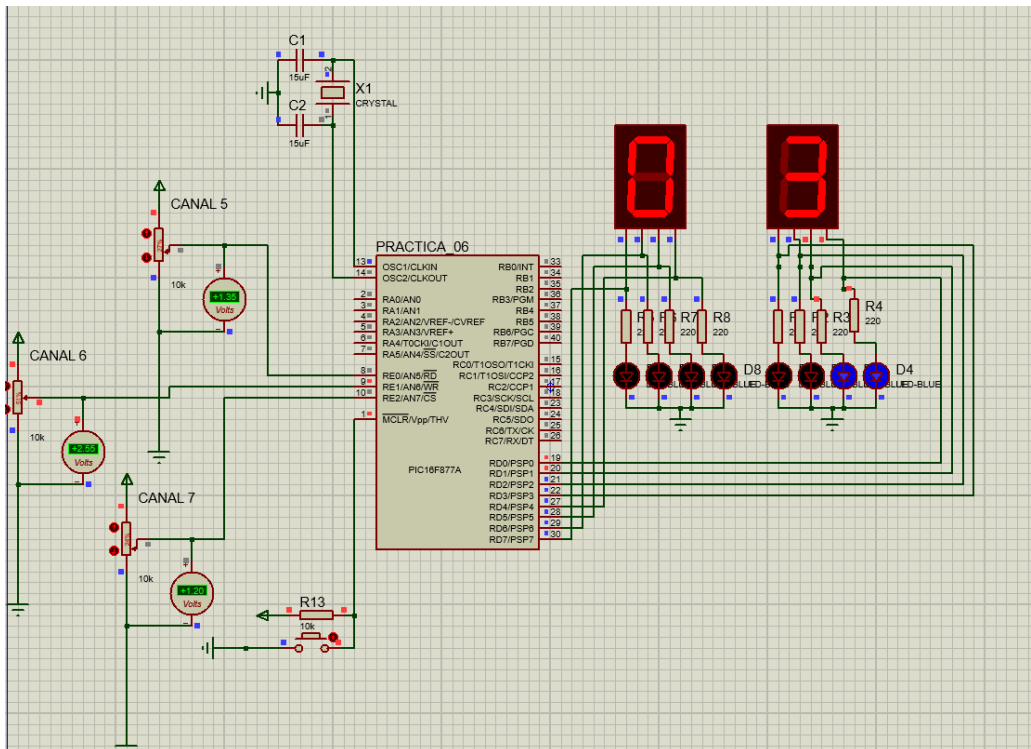
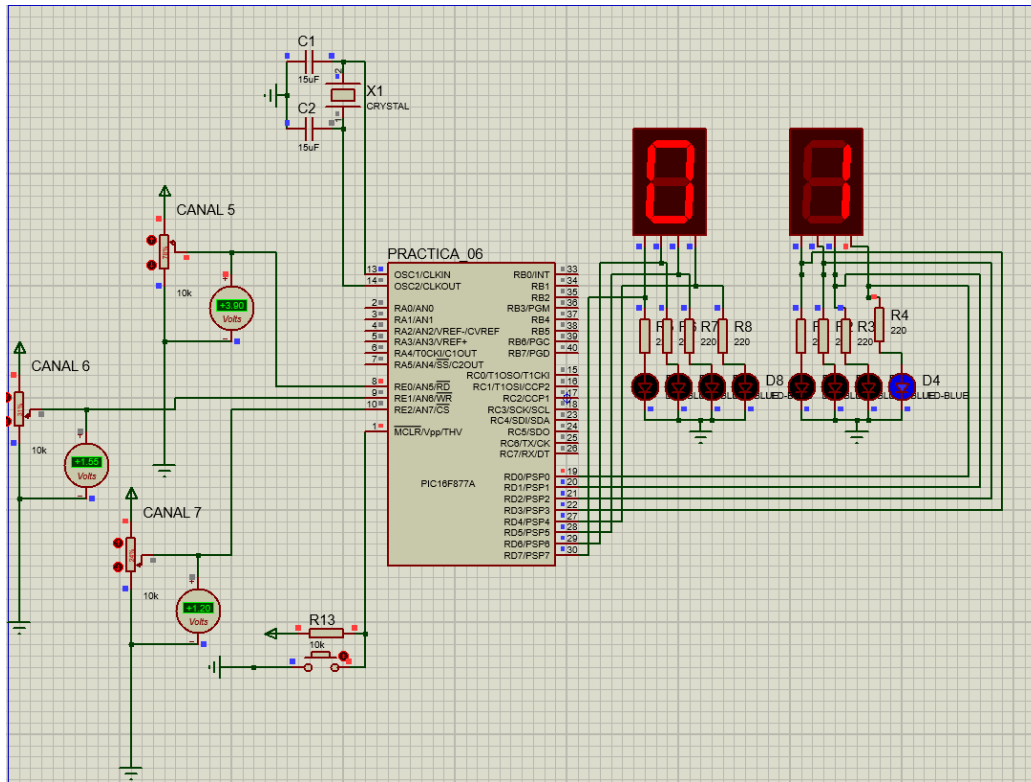
Clean: Deleting intermediary and output files.
Clean: Done.
Executing: "C:\Program Files (x86)\Microchip\MPASM Suite\MPASMWIN.exe" /q /p16F877A "e3.asm" /"e3.lst" /e"e3.err"
Warning[205] C:\USERS\ALEXIS\DESKTOP\FI\MICROCOMPUTADORAS\LAB\P6\E3.ASM 1: Found directive in column 1. (include)
Message[301] C:\PROGRAM FILES (X86)\MICROCHIP\MPASM SUITE\VP16F877.INC 33: MESSAGE: (Processor-header file mismatch. Verify selected processor.)
Message[302] C:\USERS\ALEXIS\DESKTOP\FI\MICROCOMPUTADORAS\LAB\P6\E3.ASM 16: Register in operand not in bank 0. Ensure that bank bits are correct.
Message[302] C:\USERS\ALEXIS\DESKTOP\FI\MICROCOMPUTADORAS\LAB\P6\E3.ASM 17: Register in operand not in bank 0. Ensure that bank bits are correct.
Warning[202] C:\USERS\ALEXIS\DESKTOP\FI\MICROCOMPUTADORAS\LAB\P6\E3.ASM 87: Argument out of range. Least significant bits used.
Message[305] C:\USERS\ALEXIS\DESKTOP\FI\MICROCOMPUTADORAS\LAB\P6\E3.ASM 89: Using default destination of 1 (file).
Message[305] C:\USERS\ALEXIS\DESKTOP\FI\MICROCOMPUTADORAS\LAB\P6\E3.ASM 91: Using default destination of 1 (file).
Executing: "C:\Program Files (x86)\Microchip\MPASM Suite\mplink.exe" /p16F877A "e3.o" /z __MLAB_BUILD=1 /o"e3.cof" /M"e3.map" /W/x
4PLINK 4.49. Linker
Device Database Version 1.14
Copyright (c) 1998-2011 Microchip Technology Inc.
Errors : 0

Loaded C:\Users\Alexis\Desktop\FI\Microcomputadoras\Lab\P6\p63.cof.

Release build of project 'C:\Users\Alexis\Desktop\FI\Microcomputadoras\Lab\P6\p63.disposable_mcp' succeeded.
Language tool versions: MPASMWIN.exe v5.51, mplink.exe v4.49, mplib.exe v4.49
Sun Apr 14 21:24:22 2024

BUILD SUCCEEDED
```

Resultados:



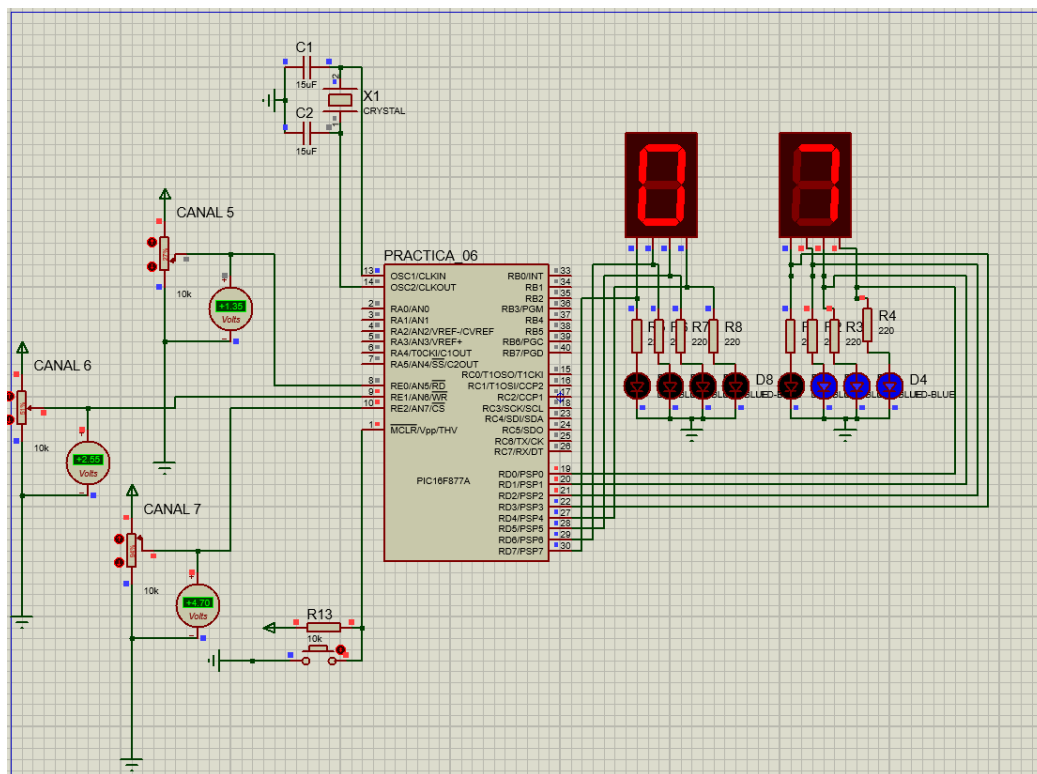
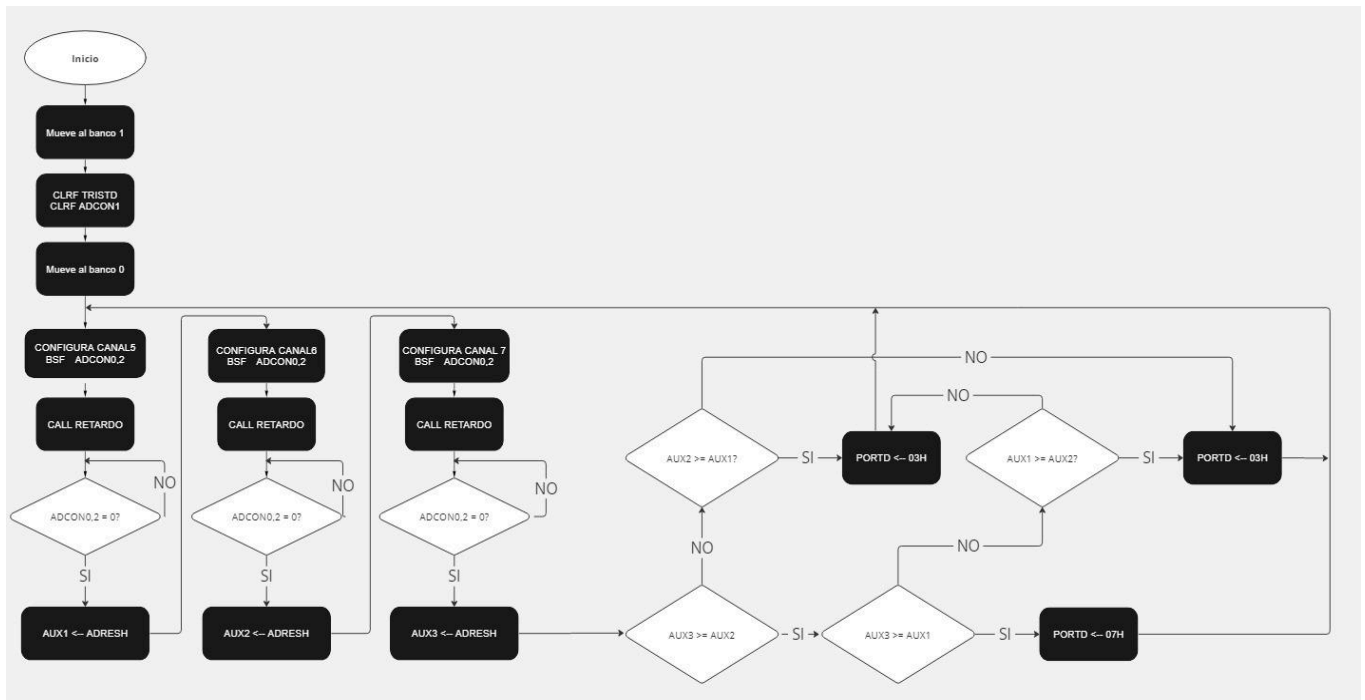


Diagrama de flujo del código:





## Simulación

<https://youtu.be/EYQpo5gfVg8>

## Análisis técnico

### 1. ¿Se puede comprobar que la solución producida funciona?

Gracias a la implementación de circuitos físicos y las simulaciones pudimos observar el correcto funcionamiento de las actividades realizadas durante la clase.

### 2. ¿Se alcanzó el objetivo?

Sí, logramos implementar una solución efectiva para cada actividad. Además, pudimos verificar el correcto desempeño de nuestros códigos gracias a los circuitos físicos como simulados.

## Ejercicio 1

### ¿Cuál es el flujo interno de los datos?

El código en cuestión inicializa y maneja el módulo ADC del microcontrolador PIC16F877, leyendo datos de un canal de entrada analógica específico y enviando el resultado a `PORTD`. Comienza configurando los puertos y el ADC adecuadamente, activa una conversión en el ADC, espera a que esta finalice, y luego transfiere el resultado a `PORTD`, repitiendo este proceso continuamente. Un retardo entre conversiones asegura la estabilidad de la lectura.

### ¿Cuáles fueron los modos de direccionamiento utilizados?

Los modos de direccionamiento utilizados en este programa son principalmente el inmediato, directo y registro indirecto con auto-decremento. `MOVLW` carga valores directamente en el registro de trabajo W, `MOVWF` y `CLRF` manipulan directamente los registros de hardware, y `DECFSZ` se usa en bucles para control de flujo, decrementando un registro y saltando si el resultado es cero, facilitando así operaciones repetitivas eficientes.

## Ejercicio 2

### ¿Cuál es el flujo interno de los datos?

Este programa está diseñado para un microcontrolador PIC16F877A y se encarga de leer datos del ADC, evaluar estos datos contra umbrales predefinidos, y actuar en consecuencia modificando el estado de `PORTD` basado en estos resultados. Inicia configurando los puertos y el ADC para leer entradas analógicas. Después de iniciar y completar una conversión ADC, el resultado se compara secuencialmente contra varios umbrales. Si el resultado es mayor o igual a un umbral, se ejecutan diferentes bloques de código para establecer un valor específico en `PORTD` que representa una categoría de voltaje detectada.

### ¿Cuáles fueron los modos de direccionamiento utilizados?

Los modos de direccionamiento utilizados en este código son mayormente inmediatos y directos. Las instrucciones `MOVLW` cargan literales directamente en el registro de trabajo `W` para comparaciones inmediatas, mientras que `MOVWF` y `CLRF` manipulan directamente los registros para configuración y almacenamiento de datos. `SUBWF` se usa para comparar el resultado del ADC almacenado en `AUX` con valores inmediatos, actualizando las banderas de estado para control de flujo basado en estas comparaciones. Este uso permite una evaluación eficiente y directa de los valores leídos del ADC contra umbrales estáticos.

## Ejercicio 3

### ¿Cuál es el flujo interno de los datos?

El código proporcionado configura y opera un microcontrolador PIC16F877 para leer entradas analógicas de tres diferentes canales (5, 6 y 7) utilizando el módulo ADC. Primero, inicializa y configura los puertos y ajustes del ADC. Luego, realiza conversiones secuenciales para cada uno de los canales, almacenando los resultados en registros auxiliares. Posteriormente, compara estos resultados para determinar cuál canal tiene el valor más alto y transfiere un valor correspondiente a `PORTD` basado en el canal con el valor más alto. El proceso de comparación y salida se repite continuamente.

### ¿Cuáles fueron los modos de direccionamiento utilizados?

Los modos de direccionamiento empleados en este código son principalmente el modo inmediato (como en `MOVLW`), el modo directo (como en `MOVWF` para transferir datos del registro `W` a otro registro) y el modo directo con registros de función especial (como al modificar `ADCON0` y `PORTD`). Se utilizan también instrucciones de comparación (`SUBWF`) para evaluar los valores almacenados en los registros auxiliares, manipulando directamente estos valores para controlar el flujo del programa basado en las condiciones de comparación.

## Conclusiones

### **Alcantar Correa Vianey:**

En la práctica, aprendí el funcionamiento de un convertidor analógico a digital y sus aplicaciones, utilizando el dispositivo para convertir voltajes de corriente directa en señales digitales visualizadas en LEDs. Comprendí la importancia del factor de conversión para interpretar correctamente las señales de entrada. Me familiaricé con los registros ADCON1 y ADCON0 para gestionar las conversiones y verificar rangos de señales con la bandera de acarreo. Utilicé los tres canales del Puerto E para comparar las conversiones y determinar el canal con la señal más alta, logrando entender cómo emplear estos convertidores en diversas aplicaciones prácticas.

### **Sanchez Rosas Alexis Alejandro:**

El desarrollo de las actividades de la práctica fue de gran utilidad para comprender la configuración de las entradas analógicas de nuestro microcontrolador, el trato que tiene el convertidor ADC con las señales nos facilita la lectura y utilización de la misma en aplicaciones prácticas como lo son un medidor de voltaje representando el valor con número hexadecimales, un identificador de voltaje por rangos e inclusive un comparador de voltajes para determinar cuál era el mayor de las entradas. Es importante mencionar que la conversión requiere de un tiempo de espera que debe calcularse cuidadosamente para no generar errores y que los resultados de estas conversiones se almacenan en el registro ADRESH, además la correcta configuración del registro ADCON nos permitirá configurar y utilizar nuestro convertidor de manera adecuada dependiendo de la situación.

### **Velazquez Martinez Karla Andrea:**

Durante el desarrollo de esta práctica pude aprender acerca de los conceptos fundamentales de un convertidor analógico-digital, constituido por una base sólida para ayudarnos a resolver problemas en nuestro entorno, ya que usualmente nos enfrentaremos a este tipo de valores. Así mismo, con ayuda de las actividades realizadas pude observar con mayor facilidad los procesos de las señales implícitas en los display de 7 segmentos, por otra parte, el uso de diferentes canales para ingresar voltajes y determinar la carga de mayor

magnitud nos permitio reconocer la gran importancia de los converrtiores analogicos -digitales.

## Bibliografía

Microchip Technology Inc. (s/f). PIC16F87X Data Sheet. Recuperado de <https://ww1.microchip.com/downloads/en/DeviceDoc/31029a.pdf>  
S.a.. (1997). Section 29. Instruction Set - Microchip Technology. USA:  
S.e..DESCRIPCIÓN DELAS INSTRUCCIONES. (s. f.). Profesores Sanvalero