



Laboratorio de Microcomputadoras

Profesor(a): Dra. Lourdes Angelica Quiñonez Juárez

Asignatura: Laboratorio de Microcomputadoras

Grupo Laboratorio: 5

Grupo Teoría: 3

No. de Práctica(s): 6

Integrante(s): Hernández Diaz Sebastián

Semestre: 2025-2

Fecha de entrega: 31 de marzo del 2025



Laboratorio de Microcomputadoras

Práctica No. 6

Convertidor Analógico/Digital

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

Desarrollo. Realizar los programa solicitados y comprobar su funcionamiento.

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.

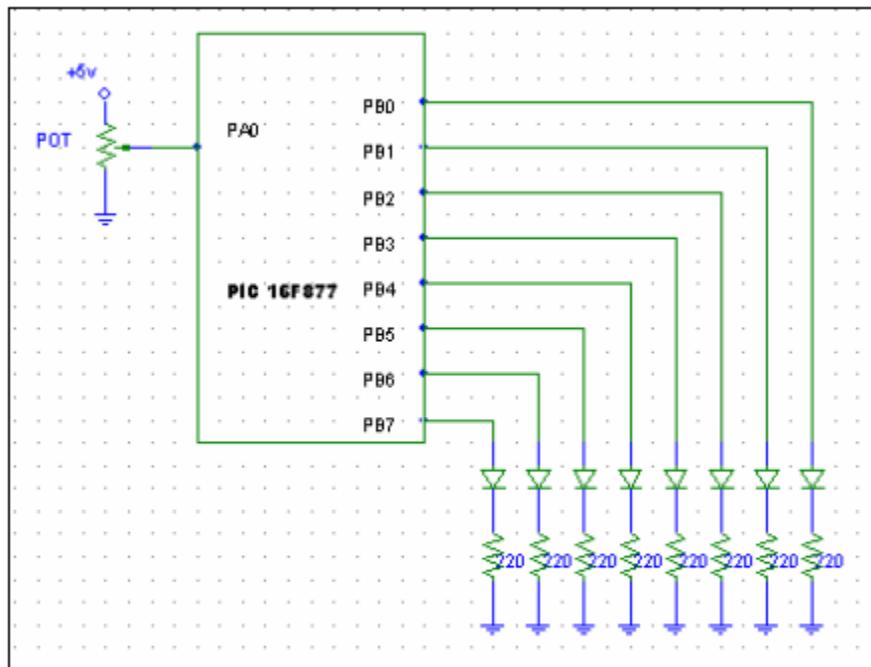


Figura 6.1 Circuito con lectura de una señal analógica

El código de este ejercicio es el siguiente:

```
include <p16f877.inc>           ;Incluye la librería de la versión del procesador

J      EQU 0X20                 ;LOCALIDADES PARA SUBRUTINA DE RETRASO
K      EQU 0X21                 ;Carga al vector de RESET la dirección de inicio

goto inicio

ORG 05H                         ;Dirección de inicio del programa del usuario
inicio: CLR PORTA
       BSF STATUS,RPO          ;Cambia a banco 1
       BCF STATUS,RP1
       MOVLW 00H
       MOVWF ADCON1            ;Configura puertos A y E como analógico
       MOVLW 00H
       MOVWF TRISD
       BCF STATUS,RPO          ;Regresa al banco cero
       MOVLW b'11101001'        ;Establecemos frecuencia de reloj, canal, el convertidor a/d
       MOVWF ADCONO
       CLRF PORTD

INICIO
       BSF ADCONO,2            ;Iniciamos la conversión a/d
       CALL retardo             ;Damos tiempo para que realice la conversión
       BCF ADCONO,2             ;Damos fin a la conversión
       MOVF ADRESH,W           ;Leemos el resultado de la conversión
       MOVWF PORTD              ;Lo cargamos en el puerto B
       GOTC INICIO              ;Volvemos al inicio

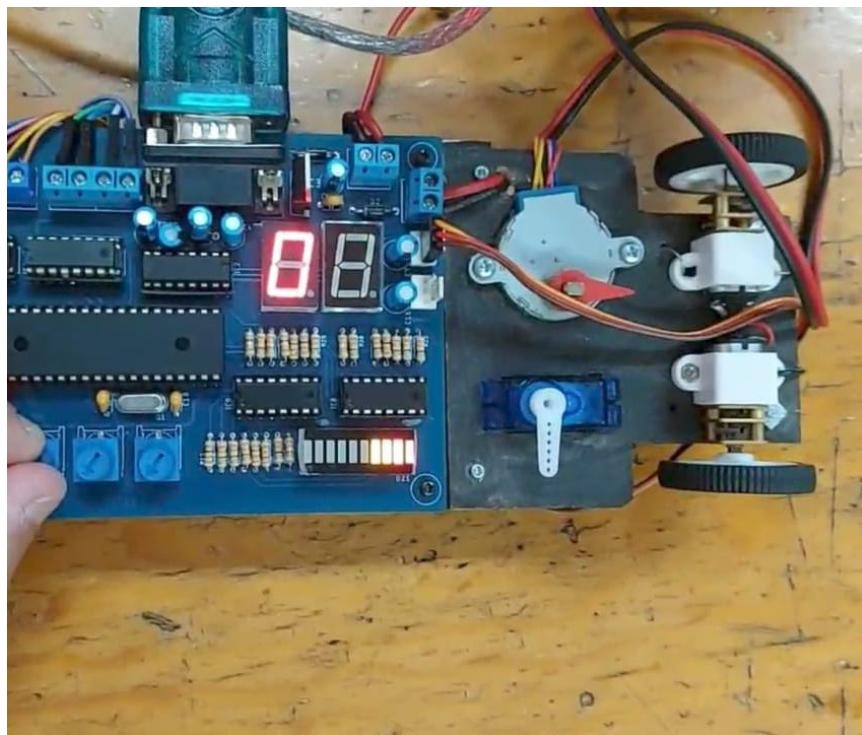
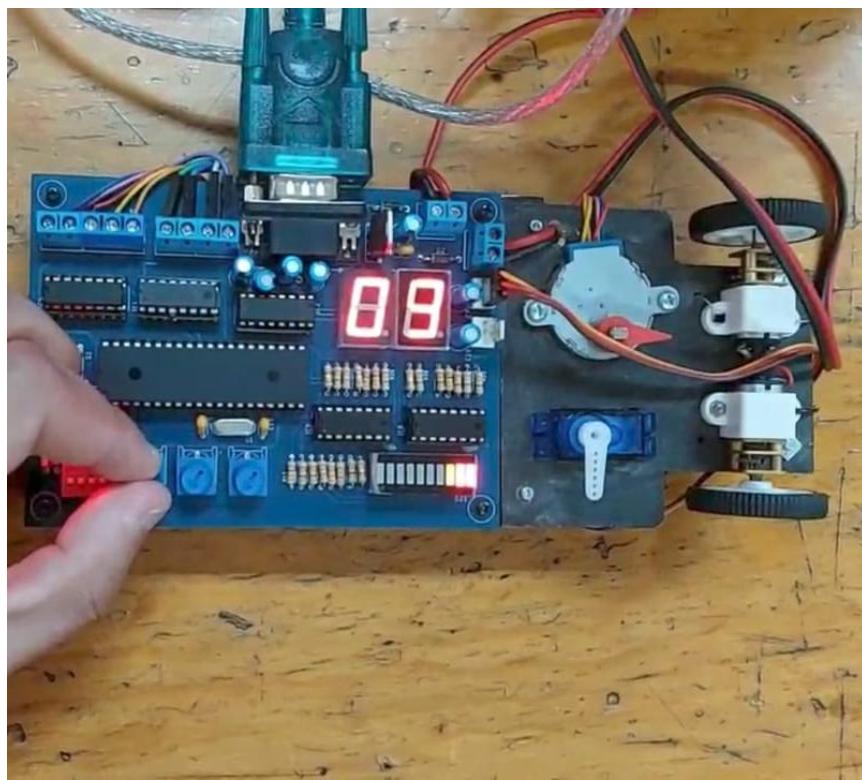
;Rutina de retardo de 20 milisegundos
retardo: MOVLW D'25'            ; W = 25 decimal
         MOVWF J                ; J = W
jloop:  MOVWF K                ; K = W
kloop:  DECFSZ K,f            ; K = K-1, omite si es cero
         DECFSZ J,f            ; J = J-1, omite si es cero
         GOTC jloop
         RETURN

END
```

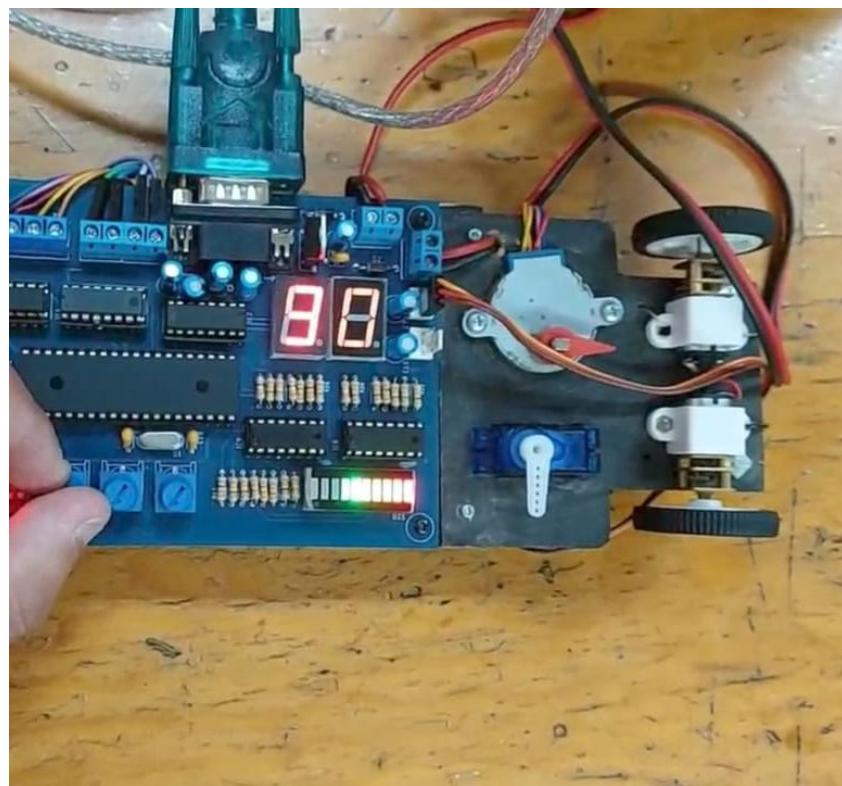
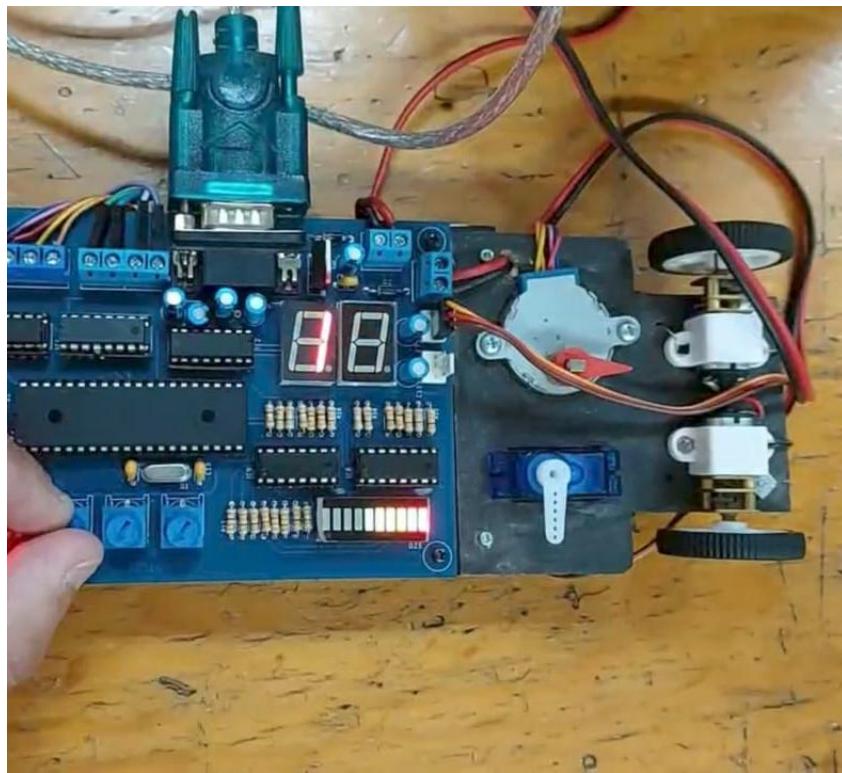
Este código realiza la conversión de una señal analógica a digital y muestra el resultado en el Puerto D. Primero, configura los puertos y el módulo ADC para que el microcontrolador pueda leer señales analógicas. Luego, en un ciclo repetitivo, inicia la conversión, espera un tiempo con una rutina de retardo, obtiene el valor digital resultante y lo muestra en PORTD. Esto permite visualizar en tiempo real la conversión de una señal analógica en una representación digital.

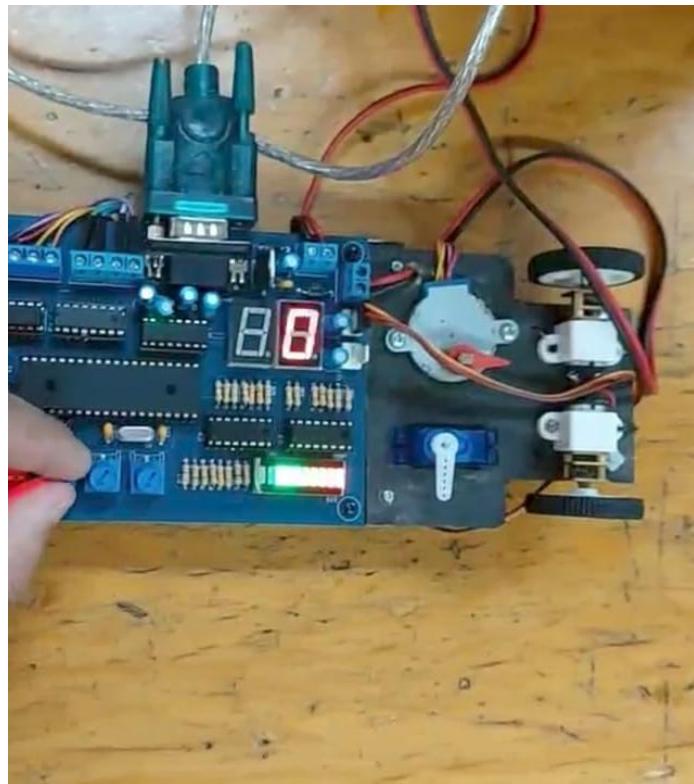
Hernández Diaz Sebastian

El programa en ejecución es el siguiente:



Hernández Diaz Sebastian





Como se puede ver los leds de la tarjeta se encienden de la manera esperada mostrando que se ejecutó de manera correcta.

2.- Utilizando el circuito anterior, realizar un programa que indique si el valor del voltaje a la entrada del convertidor A/D, se encuentra entre los siguientes rangos de voltaje.

ENTRADAS	SALIDAS		
	PX2	PX1	PX0
$V_e < 1/3 V_{cc}$	0	0	1
$1/3V_{cc} < V_e < 2/3V_{cc}$	0	1	1
$2/3 < V_e < V_{cc}$	1	1	1

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

El código de este ejercicio es el siguiente:

```
include <p16f877.inc> ;Incluye la librería de la versión del procesador

J      EQU 0X20          ;LOCALIDADES PARA SUBRUTINA DE RETRASO
K      EQU 0X21          ;Carga al vector de RESET la dirección de inicio

goto inicio

ORG 05H                      ;Dirección de inicio del programa del usuario
inicio: CLRF PORTA
        BSF STATUS,RPO    ;Cambia la banco 1
        BCF STATUS,RP1
        MOVLW 00H
        MOVWF ADCON1      ;Configura puertos A y E como analógico
        MOVLW 00H
        MOVWF TRISD
        BCF STATUS,RPO    ;Regresa al banco cero
        MOVLW b'11101001'  ;Establecemos frecuencia de reloj, canal, el convertidor a/d
        MOVWF ADCONO
        CLRF PORTD

INICIO
        BCF STATUS,C       ;Limpiamos el CARRY
        BSF ADCONO,2        ;Iniciamos la conversión a/d
        CALL retardo        ;Damos tiempo para que realice la conversión
        BCF ADCONO,2        ;Damos fin a la conversión
        MOVEF ADRESH,W     ;Leemos el resultado de la conversión
        SUBLW H'55'          ;Le restamos el primer tercio
        BTFSC STATUS,C     ;Si el resultado es negativo se apaga el CARRY
        GOTC OPI            ;Si es positivo va a op1
        MOVEF ADRESH,W     ;Leemos el resultado de la conversión
        SUBLW H'AC'          ;Restamos el segundo tercio
        BTFSC STATUS,C     ;Si el resultado es negativo se apaga el CARRY
        GOTC OP2            ;Si es positivo salta a opción 2
        MOVLW H'07'          ;Si no es ninguna de las anteriores está en el tercer tercio
        MOVWF PORTD          ;Carga un 7 en el puerto B
        GOTC INICIO         ;Vuelve al inicio

OP1   MOVLW H'01'          ;Carga un 1 en el puerto B
        MOVWF PORTD
        GOTC INICIO

OP2   MOVLW H'03'          ;Carga un 3 en el puerto B
        MOVWF PORTD
        GOTC INICIO

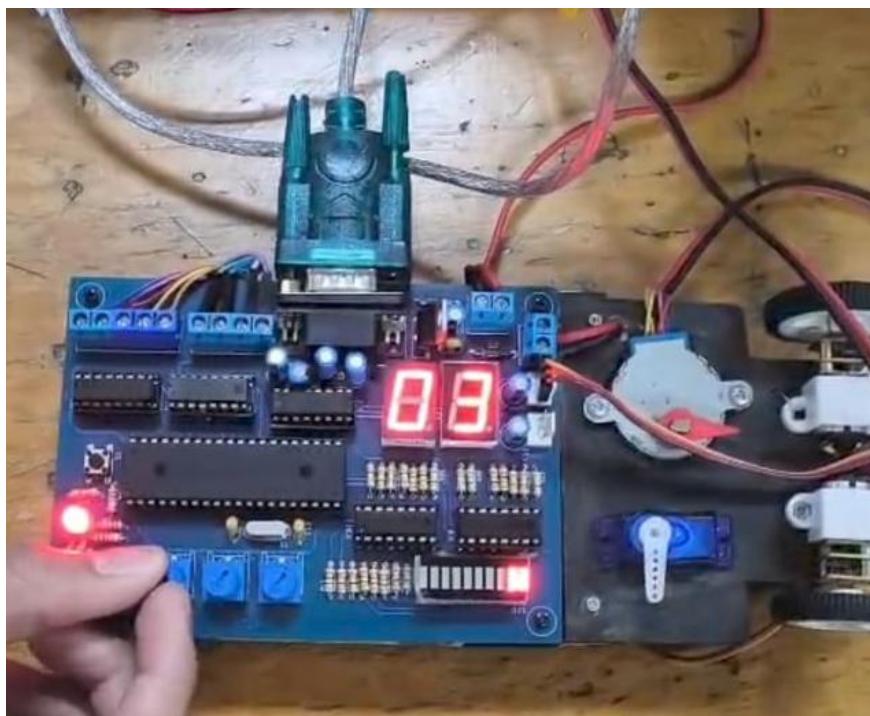
        ;Rutina de retardo de 20 milisegundos
        retardo: MOVLW D'25'    ; W = 25 decimal
                  MOVWF J          ; J = W
        jloop:  MOVF K           ; K = W
        kloop:  DECFSZ K,f      ; K = K-1, omite si es cero
                  GOTC kloop
                  DECFSZ J,f      ; J = J-1, omite si es cero
                  GOTC jloop
        RETURN

END
```

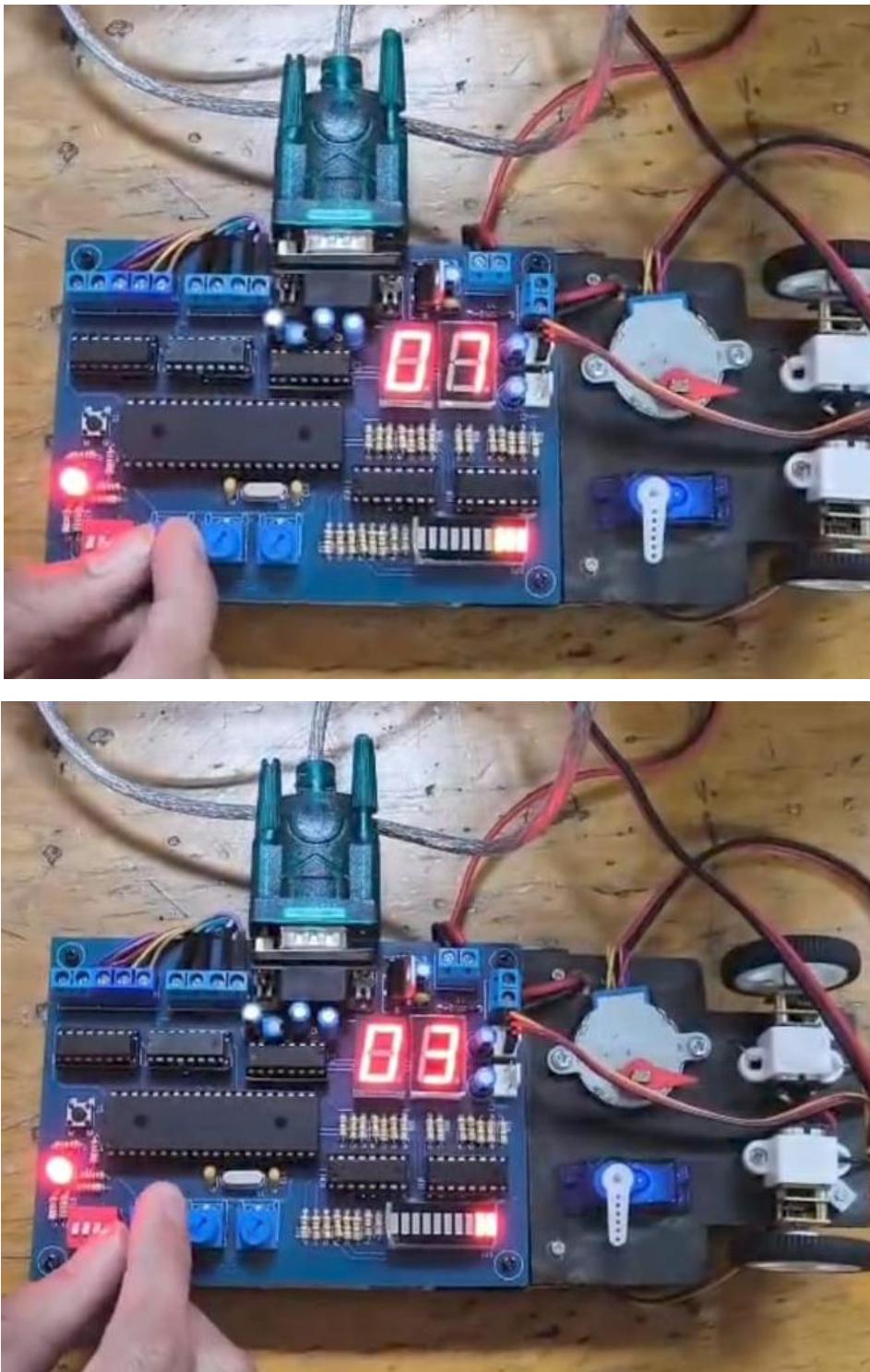
Este código realiza la conversión de una señal analógica a digital y clasifica el resultado en tres rangos distintos. Primero, configura los puertos y el módulo ADC

para habilitar la lectura de señales analógicas. Luego, en un ciclo continuo, inicia la conversión y espera un tiempo con una rutina de retardo. Después de la conversión, compara el valor obtenido con dos umbrales (0x55 y 0xAC), asignando un número distinto al Puerto D según el rango en el que se encuentre el valor digital: 1 si es menor que 0x55, 3 si está entre 0x55 y 0xAC, y 7 si es mayor que 0xAC. Esto permite clasificar la señal analógica en tres niveles diferentes y representarlos en el puerto de salida.

El programa en ejecución es el siguiente:



Hernández Diaz Sebastian



Como se puede ver los leds de la tarjeta se encienden de la manera esperada mostrando que se ejecutó de manera correcta.

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	PB2	PB1	PB0
$V_e1 > V_e2 \text{ y } V_e3$	0	0	1
$V_e2 > V_e1 \text{ y } V_e3$	0	1	1
$V_e3 > V_e1 \text{ y } V_e2$	1	1	1

Tabla 6.2

Circuito empleado para este ejercicio.

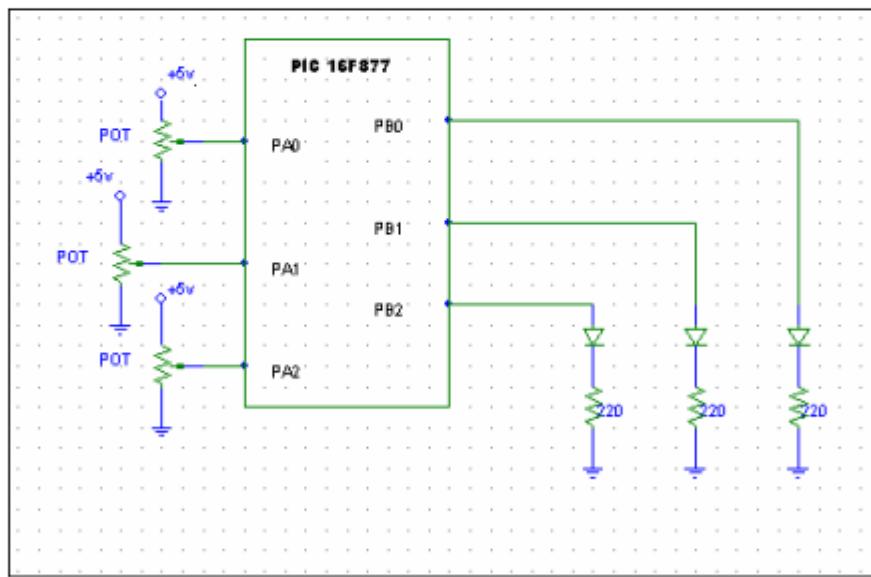


Figura 6.2 Tres señales analógicas

El código de este ejercicio es el siguiente:

```
include <p16f877.inc> ; Incluir la librería del microcontrolador

; Definición de variables en memoria
Vel EQU 0x20      ; Variable para almacenar la primera lectura ADC
Ve2 EQU 0x21      ; Variable para almacenar la segunda lectura ADC
Ve3 EQU 0x22      ; Variable para almacenar la tercera lectura ADC

GOTO inicio        ; Ir al inicio del programa
ORG 05H            ; Vector de reset

ORG 0x05           ; Inicio del programa del usuario
inicio:
    CLRF PORTB      ; Limpiar el puerto de salida
    BSF STATUS, RP0  ; Cambiar a banco 1
    BCF STATUS, RP1
    MOVLW 0x00
    MOVWF ADCON1     ; Configurar puertos A y E como analógicos
    MOVLW 0x00
    MOVWF TRISB      ; Configurar puerto B como salida
    BCF STATUS, RP0  ; Volver a banco 0
    MOVLW b'11000001' ; Seleccionar canal 0, configurar ADC y encenderlo
    MOVWF ADCON0
    CLRF PORTB      ; Limpiar puerto de salida

; Lectura de la primera señal
lee_Vel:
    BCF STATUS, C
    BSF ADCON0, 2    ; Iniciar conversión
    CALL retardo      ; Esperar tiempo suficiente
    MOVF ADRESH, W   ; Guardar el resultado
    MOVWF Vel

; Lectura de la segunda señal
    MOVLW b'11001001' ; Configurar ADC para canal 1
    MOVWF ADCON0
    BSF ADCON0, 2
    CALL retardo
    MOVF ADRESH, W
    MOVWF Ve2
```

```
; Lectura de la tercera señal
    MOVLW b'11010001' ; Configurar ADC para canal 2
    MOVWF ADCONO
    BSF ADCONO, 2
    CALL retardo
    MOVEF ADRESH, W
    MOVWF Ve3

; Comparación de señales
compara:
    MOVEF Vel, W
    SUBWF Ve2, W
    BTFSC STATUS, C ; Si Vel < Ve2, salta a comparar con Ve3
    GOTC check_Ve2
    MOVEF Vel, W
    SUBWF Ve3, W
    BTFSC STATUS, C
    GOTC check_Ve3
    MOVLW b'00000001' ; Vel es la mayor
    GOTC mostrar

check_Ve2:
    MOVEF Ve2, W
    SUBWF Ve3, W
    BTFSC STATUS, C
    GOTC check_Ve3
    MOVLW b'00000011' ; Ve2 es la mayor
    GOTC mostrar

check_Ve3:
    MOVLW b'00000111' ; Ve3 es la mayor

mostrar:
    MOVWF PORTB ; Mostrar el resultado en los LEDs
    GOTC lee_Vel ; Repetir el proceso

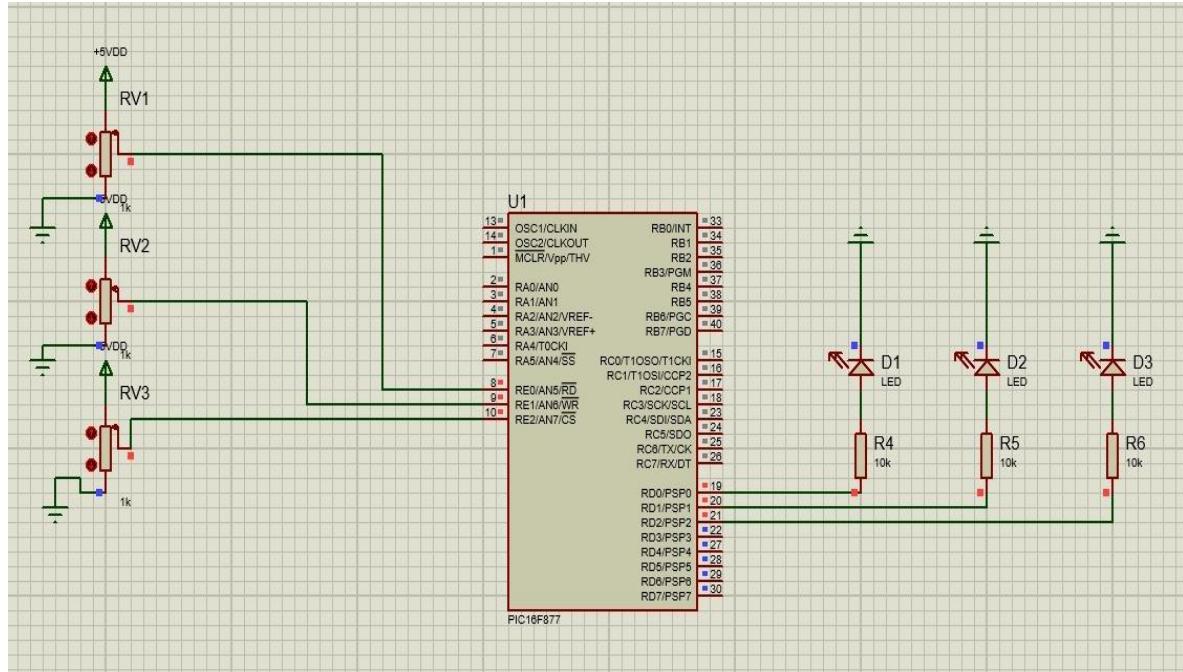
    ; Rutina de retardo
retardo:
    MOVLW D'25'
    MOVWF 0x23
loop1:
    MOVWF 0x24
loop2:
    DECFSZ 0x24, f
    GOTC loop2
    DECFSZ 0x23, f
    GOTC loop1
    RETURN

END
```

Este código lee tres señales analógicas mediante el convertidor A/D, las convierte a valores digitales y determina cuál es la mayor. Luego, enciende los LEDs en el

puerto B según la señal de mayor voltaje: si V_{e1} es la mayor, enciende un LED, si V_{e2} es la mayor, enciende dos, y si V_{e3} es la mayor, enciende tres. El proceso se repite continuamente para actualizar la salida en tiempo real.

La simulación fue la siguiente:



Conclusiones

El desarrollo de estos programas en ensamblador para el PIC16F877 permitió implementar y comprender el funcionamiento del convertidor analógico-digital (ADC) en un microcontrolador. En la primera práctica, se observó cómo el valor digitalizado de una señal analógica puede ser representado directamente en un puerto de salida y visualizado mediante LEDs. Esto es útil para la adquisición de datos en sistemas embebidos donde se requiera monitoreo en tiempo real.

En la segunda práctica, se implementó una clasificación del voltaje de entrada en tres niveles, comparando el resultado de la conversión A/D con valores umbrales predefinidos.

Además, se podría complementar con técnicas de filtrado digital para mejorar la precisión de la medición. Estas prácticas son fundamentales en aplicaciones como instrumentación, monitoreo de sensores y sistemas de control embebidos, donde la conversión y procesamiento de señales analógicas es un requisito esencial.

Por último, se implementó un sistema de comparación de señales analógicas utilizando el convertidor A/D del microcontrolador PIC16F877. Se logró identificar cuál de las tres señales de entrada era la de mayor amplitud y representarla en un puerto paralelo utilizando LEDs. Se verificó el funcionamiento del código mediante pruebas con diferentes niveles de voltaje en las entradas del ADC.

Se concluye que el PIC16F877 ofrece una capacidad eficiente para la conversión de señales analógicas a digitales y permite realizar comparaciones básicas sin necesidad de hardware adicional. Sin embargo, el tiempo de conversión y el retardo pueden afectar el rendimiento si se requiere un procesamiento en tiempo real.

Bibliografía

Díaz, J. (2015). *Microcontroladores PIC: Programación en ensamblador y C.* Alfaomega.

Predko, M. (2008). *Programming and Customizing the PIC Microcontroller.* McGraw-Hill.

Peatman, J. B. (1998). *Design with PIC Microcontrollers.* Pearson Education.

Texas Instruments. (2021). *Understanding ADC Parameters and Specifications.* Recuperado de www.ti.com

Microchip Technology. (2020). *PIC16F877A Datasheet.* Recuperado de www.microchip.com