



Laboratorio de Microcomputadoras

Profesor(a): Dra. Lourdes Angelica Quiñonez Juárez

Asignatura: Laboratorio de Microcomputadoras

Grupo Laboratorio: 5

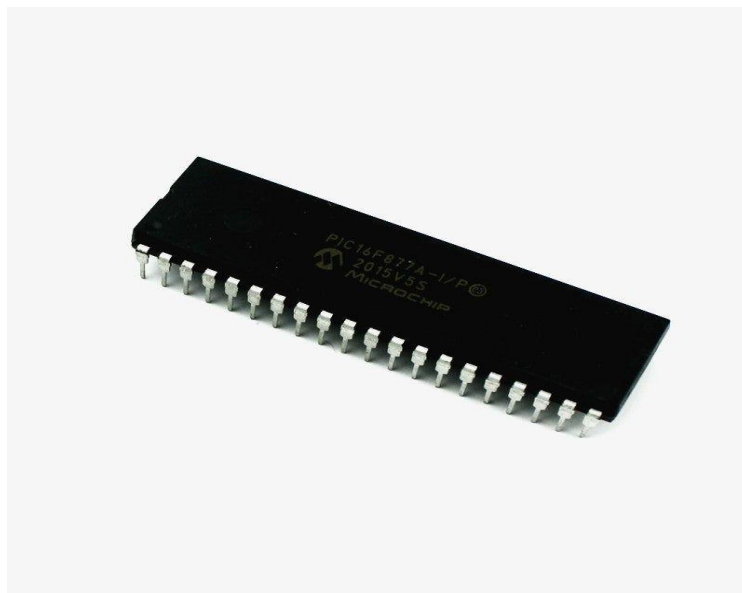
Grupo Teoría: 3

No. de Práctica(s): 5

Integrante(s): Hernández Diaz Sebastián

Semestre: 2025-2

Fecha de entrega: 24 de marzo del 2025



Laboratorio de Microcomputadoras

Práctica No. 5

Puertos Paralelos IV (Lectura de sensores ópticos)

Objetivo: Emplear los puertos paralelos, para hacer lecturas de señales externas (sensores reflectivos) con el microcontrolador y realizar operaciones de acuerdo a los valores recibidos.

Desarrollo: Realizar los apartados siguientes.

Realizar un programa que, de acuerdo a la entrada generada por los switch, se controle la operación de los motores, tal como se muestra en la siguiente tabla.

ENTRADAS			ACCION	
Sensor Izquierdo	Sensor Central	Sensor Derecho	MOTOR IZQUIERDO	MOTOR DERECHO
B	N	N	ATRÁS	ADELANTE
N	B	N	ADELANTE	ADELANTE
N	N	B	ADELANTE	ATRÁS
N	N	N	PARO	PARO

El código para esta práctica es el siguiente:

```
processor 16F877
include <pl6f877.inc>

ORG 0
GOTO inicio ; Salta a la configuración inicial.

ORG 5
inicio:
    bsf STATUS, RP0 ; Cambia al Banco 1.
    BCF STATUS, RP1 ; Asegura que estamos en el Banco 1.
    MOVLW H'07' ; Configura ADCON1 para que todos los pines sean digitales.
    MOVWF ADCON1 ; Deshabilita el ADC.
    MOVLW H'FF' ; Configura el Puerto A como entrada.
    MOVWF TRISA
    CLRF TRISE ; Configura el Puerto B como salida.
    CLRF TRISC ; Configura el Puerto C como salida.
    BCF STATUS, RP0 ; Vuelve al Banco 0.

bucle:
    MOVF PORTA, 0 ; Lee el estado de los switches en PORTA.

    XORLW H'00' ; Compara con 00H (Motores apagados).
    BTFSC STATUS, Z
    CALL motores_apagados

    XORLW H'01' ; Compara con 01H (Motor derecho en sentido horario).
    BTFSC STATUS, Z
    CALL motor_derecho_horario

    XORLW H'02' ; Compara con 02H (Motor derecho en sentido antihorario).
    BTFSC STATUS, Z
    CALL motor_derecho_antihorario

    XORLW H'04' ; Compara con 04H (Motor izquierdo en sentido horario).
    BTFSC STATUS, Z
    CALL motor_izquierdo_horario

    XORLW H'08' ; Compara con 08H (Motor izquierdo en sentido antihorario).
    BTFSC STATUS, Z
    CALL motor_izquierdo_antihorario

    XORLW H'03' ; Compara con 03H (Ambos motores en sentido horario).
    BTFSC STATUS, Z
    CALL motores_horario
```

```
XORLW H'06'          ; Compara con 06H (Ambos motores en sentido antihorario).
BTFSC STATUS, Z
CALL motores_antihorario

XORLW H'07'          ; Compara con 07H (Motor izquierdo horario y derecho antihorario).
BTFSC STATUS, Z
CALL motores_opuestos1

XORLW H'05'          ; Compara con 05H (Motor derecho horario y izquierdo antihorario).
BTFSC STATUS, Z
CALL motores_opuestos2

GOTO bucle           ; Regresa a verificar la entrada.

motores_apagados:
    CLRF PORTC        ; Apaga los motores.
    CLRF PORTB
    RETURN

motor_derecho_horario:
    MOVLW H'02'        ; Activa el motor derecho en sentido horario.
    MOVWF PORTC
    MOVLW H'02'
    MOVWF PORTE
    RETURN

motor_derecho_antihorario:
    MOVLW H'02'        ; Activa el motor derecho en sentido antihorario.
    MOVWF PORTC
    MOVLW H'01'
    MOVWF PORTE
    RETURN
```

```
motor_izquierdo_horario:
    MOVLW H'04'          ; Activa el motor izquierdo en sentido horario.
    MOVWF PORTC
    MOVLW H'08'
    MOVWF PORTB
    RETURN

motor_izquierdo_antihorario:
    MOVLW H'04'          ; Activa el motor izquierdo en sentido antihorario.
    MOVWF PORTC
    MOVLW H'04'
    MOVWF PORTB
    RETURN

motores_horario:
    MOVLW H'06'          ; Ambos motores giran en sentido horario.
    MOVWF PORTC
    MOVLW H'0A'
    MOVWF PORTB
    RETURN

motores_antihorario:
    MOVLW H'06'          ; Ambos motores giran en sentido antihorario.
    MOVWF PORTC
    MOVLW H'05'
    MOVWF PORTB
    RETURN

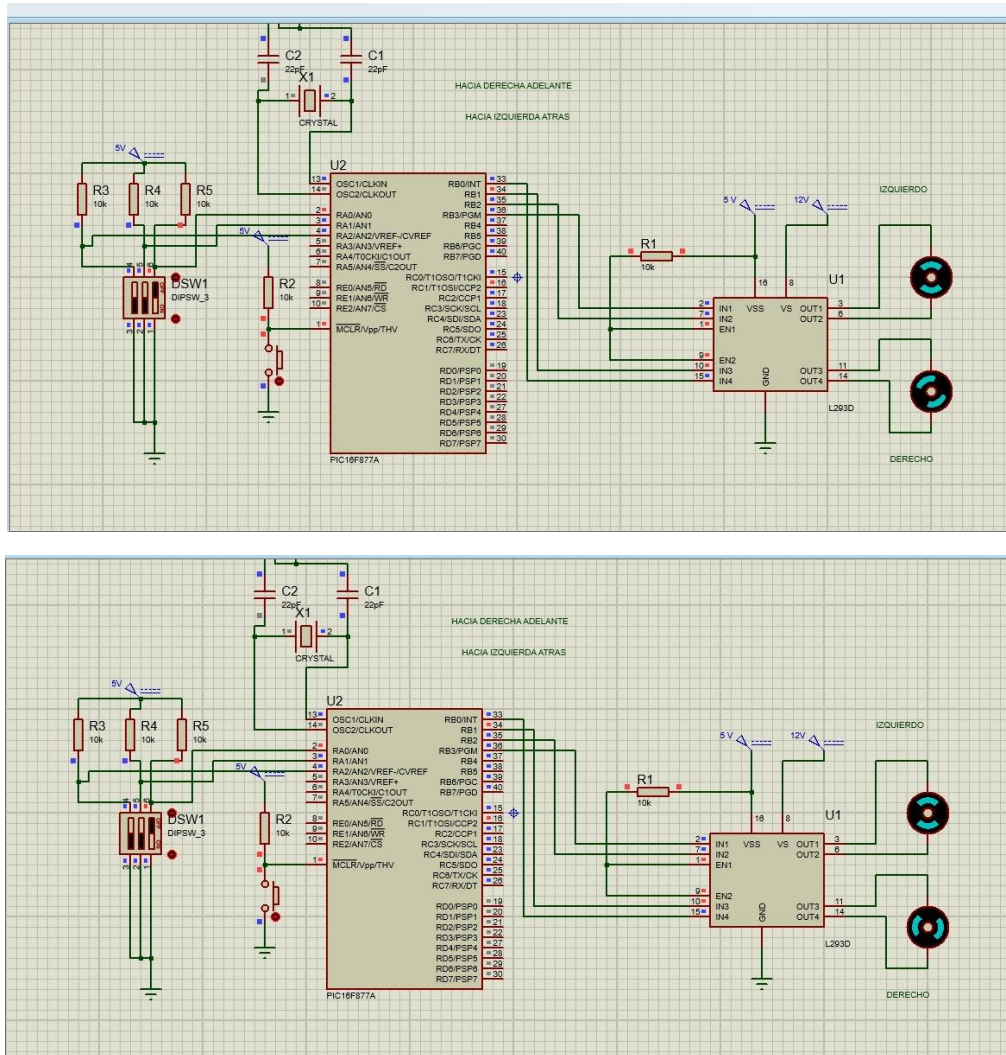
motores_opuestos1:
    MOVLW H'06'          ; Motor izquierdo horario, motor derecho antihorario.
    MOVWF PORTC
    MOVLW H'09'
    MOVWF PORTB
    RETURN

motores_opuestos2:
    MOVLW H'06'          ; Motor derecho horario, motor izquierdo antihorario.
    MOVWF PORTC
    MOVLW H'06'
    MOVWF PORTB
    RETURN

END
```

Este código controla los switches y los compara con xor para saber cuál es que esta activo y de esta manera poder girar los motores hacia un lado u otro dependiendo de lo que se solicite.

A continuación, se presentan unas imágenes de la simulación:



En este caso no se puede apreciar de manera correcta ya que son solo imágenes, pero se pueden ver las diferencias en el movimiento que se le provoco a los motores y como es que estos giraron de la manera esperada.

Conclusiones

Esta práctica permitió consolidar conocimientos sobre el control de motores mediante un microcontrolador PIC16F877A, utilizando Proteus como entorno de simulación. Se trabajó con la configuración de puertos digitales, estableciendo PORTA como entrada para la lectura de switches y PORTB y PORTC como salidas para activar el puente H, encargado de manejar la dirección de los motores. Se comprobó la importancia de la configuración del ADCON1, ya que, al deshabilitar el conversor analógico-digital, los pines del Puerto A pueden utilizarse como entradas digitales sin interferencias.

Se aplicaron instrucciones de comparación y control de flujo como XORLW, BTFSC y GOTO, permitiendo que el código detectara combinaciones de switches y ejecutara la función correspondiente para mover los motores en diferentes direcciones. La modularización del código en funciones facilitó la estructura y mantenimiento del programa, haciendo que cada acción del motor estuviera claramente definida y separada.

Entre los resultados obtenidos, se logró visualizar el movimiento de los motores en distintas direcciones según la combinación de switches activados. Esto incluyó el control individual de cada motor en sentido horario y antihorario, así como el movimiento sincronizado en la misma dirección o en direcciones opuestas. Se validó el correcto funcionamiento del código en la simulación, asegurando que las salidas digitales correspondieran con los comandos de control previstos.

En conclusión, esta práctica reforzó el conocimiento sobre la configuración de puertos de entrada y salida digital., permitiendo simular y validar el control de motores sin necesidad de hardware físico.

Bibliografía

Microchip Technology. (2024). *PIC16F877 Datasheet*. Recuperado de <https://www.microchip.com>

Microchip Technology. (2018). *Mid-Range MCU Family Reference Manual*. Recuperado de <https://ww1.microchip.com/downloads/en/DeviceDoc/33023a.pdf>

All About Circuits. (2023). "Introduction to Microcontrollers". Recuperado de <https://www.allaboutcircuits.com>

Gooligum Electronics. (2023). "PIC Tutorial – Mid-Range Microcontrollers". Recuperado de <http://www.gooligum.com.au/tutorials>