UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

Práctica 5: Control de Actuadores

INTEGRANTES DE EQUIPO Y GRUPO DE TEORÍA

López Becerra Ricardo - 420053710 - GRUPO 3
Navarrete Zamora Aldo Yael - 317242409 - GRUPO 4

ASIGNATURA

Laboratorio de Microcomputadoras

GRUPO DE LABORATORIO

8

FECHA DE REALIZACIÓN

20 de abril del 2023

FECHA DE ENTREGA

28 de abril de 2023

1. Desarrollo

Utilizando el circuito de potencia de motores de corriente directa y el sistema de desarrollo del microcontrolador PIC, realizar los programas solicitados.

1. Considerando la asignación de terminales asignadas en la figura del manual de prácticas; realizar el programa que ejecute el control indicado en la tabla siguiente.

MOTOR1		
PC1	PB1	PB0
ENABLE_M1	DIR1_M1	DIR2_M1

MOTOR1		
PC1	PB1	PB0
ENABLE_M1	DIR1_M1	DIR2_M1

DATO Puerto Paralelo	ACCION	
	MOTOR M1	MOTOR M2
0x00	PARO	PARO
0x01	PARO	HORARIO
0x02	PARO	ANTI-HORARIO
0x03	HORARIO	PARO
0x04	ANTI-HORARIO	PARO
0x05	HORARIO	HORARIO
0x06	ANTI-HORARIO	ANTI-HORARIO
0x07	HORARIO	ANTI-HORARIO
0x08	ANTI-HORARIO	HORARIO

Algoritmo

Para este problema la solución es sencilla dado que el movimiento de los motores puede ser controlado únicamente con los los bits de los puertos paralelos mostrados en las tablas de la descripción del problema. Además, estos motores no requieren ninguna medición del tiempo para funcionar, por lo que tampoco fue necesario utilizar la subrutina de retardo.

El programa comienza con un menú que dirige el flujo del programa a la sección que hace el movimiento solicitado. En cada una de estas secciones se configuran los bits de los puertos paralelos de tal forma que muevan los motores en las direcciones correspondientes. Ninguno de los flujos requiere de alguna modición especial. El puerto C es utilizado para activar los motores y el puerto B para indicarle la dirección.

Código comentado

```
processor 16f877
include <p16f877.inc>
valor1 equ h'21'
valor2 equ h'22'
valor3 equ h'23'
```

```
cte1 equ h'ff'
cte2 equ 50h
cte3 equ 60h
        ORG 0
GOTO INICIO
        ORG 5
INICIO:
        CLRF PORTA
        BSF STATUS,RP0
        BCF STATUS,RP1
        ;Coinfigura los puertos B y C
        ;como salidas
        MOVLW H'00'
        MOVWF TRISB
        MOVWF TRISC
        ;Configura el puerto A como
        ;digital
        MOVLW 06H
        MOVWF ADCON1
        ;Configura el puerto A como entrada
        MOVLW 3FH
        MOVWF TRISA
        BCF STATUS,RP0
        ;Limpia los puertos B y C
        MOVLW B'00000000'
        MOVWF PORTB
        MOVWF PORTC

loop1:

        ; Loop principal. Dependiendo
        ; de la posicion del switch
        ; se digirge el programa al
        ;flujo correcto
        MOVLW h'00'
        SUBWF PORTA, W
        BTFSC STATUS, Z
        GOTO PARO_PARO

        MOVLW h'01'
        SUBWF PORTA, W
        BTFSC STATUS, Z
        GOTO PARO_H

        MOVLW h'02'
        SUBWF PORTA, W
        BTFSC STATUS, Z
        GOTO PARO_AH

        MOVLW h'03'
```

```
        SUBWF PORTA, W
        BTFSC STATUS, Z
        GOTO H_PARO

        MOVLW h'04'
        SUBWF PORTA, W
        BTFSC STATUS, Z
        GOTO AH_PARO

        MOVLW h'05'
        SUBWF PORTA, W
        BTFSC STATUS, Z
        GOTO H_H

        MOVLW h'06'
        SUBWF PORTA, W
        BTFSC STATUS, Z
        GOTO AH_AH

        MOVLW h'07'
        SUBWF PORTA, W
        BTFSC STATUS, Z
        GOTO H_AH

        MOVLW h'08'
        SUBWF PORTA, W
        BTFSC STATUS, Z
        GOTO AH_H

goto loop1

;Para cada sentido de giro
;de los motores solicitado
;se activa el motor con el
; portc y con el portb se
; determina la direccion de
;giro

PARO_PARO:
        BCF PORTC, 2
        BCF PORTC, 1
        GOTO loop1

PARO_H:
        BCF PORTC, 2
        BSF PORTC, 1
        BSF PORTB, 0
        BCF PORTB, 1
        GOTO loop1
```

```
PARO_AH
        BCF PORTC, 2
        BSF PORTC, 1
        BSF PORTB, 1
        BCF PORTB, 0
        GOTO loop1

H_PARO:
        BCF PORTC, 1
        BSF PORTC, 2
        BSF PORTB, 2
        BCF PORTB, 3
        GOTO loop1

AH_PARO:
        BCF PORTC, 1
        BSF PORTC, 2
        BSF PORTB, 3
        BCF PORTB, 2
        GOTO loop1

H_H:
        BSF PORTC, 1
        BSF PORTC, 2
        BSF PORTB, 2
        BCF PORTB, 3
        BSF PORTB, 0
        BCF PORTB, 1
        GOTO loop1

AH_AH
        BSF PORTC, 1
        BSF PORTC, 2
        BSF PORTB, 3
        BCF PORTB, 2
        BSF PORTB, 1
        BCF PORTB, 0
        GOTO loop1

H_AH
        BSF PORTC, 1
        BSF PORTC, 2
        BSF PORTB, 2
        BCF PORTB, 3
        BSF PORTB, 1
        BCF PORTB, 0
        GOTO loop1

AH_H
        BSF PORTC, 1
```

```
BSF PORTC, 2
BSF PORTB, 3
BCF PORTB, 2
BSF PORTB, 0
BCF PORTB, 1
GOTO loop1
```

2. Realizar un programa que controle la cantidad de pasos que debe dar un motor, así como el sentido de giro.

Dato Puerto Paralelo	**Motor a pasos**
0x00	Motor en paro
0x01	Giro en sentido horario
0x02	Gira en sentido anti horario
0x03	Gira cinco vueltas en sentido horario
0x04	Gira 10 vueltas en sentido anti horario

Algoritmo

Este código es un programa escrito en lenguaje ensamblador para el microcontrolador PIC16F877 que controla un motor en una dirección horaria o antihoraria según la entrada que se le suministre. El programa espera una señal de entrada en el puerto A y según el valor de esa entrada ejecuta uno de los siguientes cinco bloques de código:

a) PARO: Se detiene el motor.

b) H: Gira el motor en dirección horaria en cuatro pasos utilizando el protocolo de control de motores paso a paso unipolar.

c) AH: Gira el motor en dirección antihoraria en cuatro pasos utilizando el protocolo de control de motores paso a paso unipolar.

d) CINCO_H: Gira el motor en dirección horaria dando 5 vueltas completas.

e) DIEZ_AH: Gira el motor en dirección anti-horaria dando 10 vueltas completas.

Calculo para el tiempo de retardo del motor a pasos:

$$Ciclos = 1 + 1 + B(3A + 1) + 1(B - 1) + 2 + 2(B - 1) + 2$$
$$Ciclos = B(3A + 1) + 3B + 1 + 2$$
$$Ciclos = B(3A + 1) + 3B + 3$$

Supongamos que A = 255 y B = 33 entonces tenemos que

$$Ciclos = 255(3(33) + 1) + 3(255) + 3$$
$$Ciclos = 26,268$$

Por lo que el tiempo de retardo para un paso sería

$$T_{retardo} = 26,268(0{,}2\mu s)$$
$$T_{retardo} = 5{,}2ms$$

El código utiliza las funciones GOTO y CALL para saltar a diferentes bloques de código dependiendo del resultado de las comparaciones que se realizan para determinar el estado de la entrada en el puerto A. También se utiliza la función RETARDO para introducir un retardo en la ejecución del programa, ya que el motor necesita cierto tiempo para moverse entre cada paso.

Código comentado

```
                                                        processor
#include <p16f877.inc>
REGB equ h'21'          ; Registro A
REGA equ h'22'          ; Registro B
COUNT1 equ h'35'        ; Contador 1
COUNT2 equ h'36'        ; Contador 2
cte2 equ h'FF'          ; Variable A para calculo del tiempo
cte1 equ .33; B         ; Variable B para calculo del tiempo
        ORG 0
GOTO INICIO
        ORG 5
INICIO:
        CLRF PORTA          ; Limpiar el PortA
        BSF STATUS,RP0      ;
        BCF STATUS,RP1      ; Cambiar al banco 1
        MOVLW H'00'         ;
        MOVWF TRISB         ; Configurar TRISB como salida MOTOR
        MOVWF TRISC         ; Configurar TRISC como salida MOTOR
        MOVWF TRISD         ; Configurar TRISD como salida MOTOR
        MOVLW 06H           ;
        MOVWF ADCON1        ; Configurar entradas digitales
        MOVLW 3FH           ;
        MOVWF TRISA         ; Configurar TRISA como entrada
        BCF STATUS,RP0      ; Cambiamos de banco
        MOVLW B'00000000'   ;
        MOVWF PORTB         ;
        MOVWF PORTC         ;
        MOVWF PORTD         ; Limpieza de los puertos de salida.
loop1:
        MOVLW h'00'             ;
        SUBWF PORTA, W  ; PORTA = 0 ?
        BTFSC STATUS, Z ;
        GOTO PARO       ; VAMOS A PARO

        MOVLW h'01'             ;
        SUBWF PORTA, W  ; PORTA = 1 ?
        BTFSC STATUS, Z ;
```

```
        GOTO H                  ; VAMOS A H = HORARIO

        MOVLW h'02'             ;
        SUBWF PORTA, W  ; PORTA = 2 ?
        BTFSC STATUS, Z ;
        GOTO AH                 ; VAMOS A AH = ANTIHORARIO

        MOVLW h'03'             ;
        SUBWF PORTA, W  ; PORTA = 3 ?
        BTFSC STATUS, Z ;
        GOTO CINCO_H    ; VAMOS A CINCO_H = CINCO HORARIO

        MOVLW h'04'             ;
        SUBWF PORTA, W  ; PORTA = 4 ?
        BTFSC STATUS, Z ;
        GOTO DIEZ_AH    ; VAMOS A DIEZ_AH = DIEZ ANTI HORARIO

goto loop1

PARO:
        CLRF PORTB      ;
        GOTO loop1      ; LIMPIEZA DEL PORTB
H:
        CLRF PORTB      ; Lmpieza del portB
        BSF PORTB, 7    ;
        BSF PORTB, 6    ;
        BCF PORTB, 5    ;
        BCF PORTB, 4    ;PASO 1
        CALL RETARDO
        BCF PORTB, 7    ;
        BSF PORTB, 6    ;
        BSF PORTB, 5    ;
        BCF PORTB, 4    ;PASO 2
        CALL RETARDO
        BCF PORTB, 7    ;
        BCF PORTB, 6    ;
        BSF PORTB, 5    ;
        BSF PORTB, 4    ;PASO 3
        CALL RETARDO
        BSF PORTB, 7    ;
        BCF PORTB, 6    ;
        BCF PORTB, 5    ;
        BSF PORTB, 4    ;PASO 4
        CALL RETARDO
        GOTO loop1
AH:
        CLRF PORTB
        BSF PORTB, 7    ;
        BCF PORTB, 6    ;
        BCF PORTB, 5    ;
```

```
            BSF PORTB, 4      ;PASO 4
            CALL RETARDO
            BCF PORTB, 7      ;
            BCF PORTB, 6      ;
            BSF PORTB, 5      ;
            BSF PORTB, 4      ;PASO 3
            CALL RETARDO
            BCF PORTB, 7      ;
            BSF PORTB, 6      ;
            BSF PORTB, 5      ;
            BCF PORTB, 4      ;PASO 2
            CALL RETARDO
            BSF PORTB, 7      ;
            BSF PORTB, 6      ;
            BCF PORTB, 5      ;
            BCF PORTB, 4      ;PASO 1
            CALL RETARDO
            GOTO loop1
CINCO_H:
                  CLRF COUNT1           ;
                  CLRF COUNT2           ; Limpieza de los contadores
            loop_h_1:
                  MOVLW H'0A'           ; 10 Medias vueltas
                  SUBWF COUNT1, W       ;
                  BTFSC STATUS, Z       ;
                  GOTO end_loop_h_1     ;
            loop_h_2:
                  MOVLW H'FF'           ; 255 pasos
                  SUBWF COUNT2, W       ;
                  BTFSC STATUS, Z       ;
                  GOTO end_loop_h_2     ;
                  CLRF PORTB            ; Limpieza portB
                  BSF PORTB, 7          ;
                  BSF PORTB, 6          ;
                  BCF PORTB, 5          ;
                  BCF PORTB, 4          ;PASO 1
                  CALL RETARDO
                  BCF PORTB, 7          ;
                  BSF PORTB, 6          ;
                  BSF PORTB, 5          ;
                  BCF PORTB, 4          ;PASO 2
                  CALL RETARDO
                  BCF PORTB, 7          ;
                  BCF PORTB, 6          ;
                  BSF PORTB, 5          ;
                  BSF PORTB, 4          ;PASO 3
                  CALL RETARDO
                  BSF PORTB, 7          ;
                  BCF PORTB, 6          ;
                  BCF PORTB, 5          ;
```

```
                BSF PORTB, 4          ;PASO 4
                CALL RETARDO
                INCF COUNT2, 1
                GOTO loop_h_2
end_loop_h_1:
                MOVLW h'03'           ;
                SUBWF PORTA, W        ;
                BTFSC STATUS, Z       ;
                GOTO end_loop_h_1     ;
                GOTO loop1
end_loop_h_2:
                MOVFW COUNT2
                MOVWF PORTD
                INCF COUNT1, 1
                CLRF COUNT2
                GOTO loop_h_1
DIEZ_AH:
                CLRF COUNT1
                CLRF COUNT2
        loop_ah_1:
                MOVLW H'14'           ; 20 Medias vueltas
                SUBWF COUNT1, W       ;
                BTFSC STATUS, Z       ;
                GOTO end_loop_ah_1    ;
        loop_ah_2:
                MOVLW H'FF'           ; 255 pasos
                SUBWF COUNT2, W       ;
                BTFSC STATUS, Z       ;
                GOTO end_loop_ah_2    ;
                CLRF PORTB            ; Limpieza portB
                BSF PORTB, 7          ;
                BCF PORTB, 6          ;
                BCF PORTB, 5          ;
                BSF PORTB, 4          ;PASO 4
                CALL RETARDO
                BCF PORTB, 7          ;
                BCF PORTB, 6          ;
                BSF PORTB, 5          ;
                BSF PORTB, 4          ;PASO 3
                CALL RETARDO
                BCF PORTB, 7          ;
                BSF PORTB, 6          ;
                BSF PORTB, 5          ;
                BCF PORTB, 4          ;PASO 2
                CALL RETARDO
                BSF PORTB, 7          ;
                BSF PORTB, 6          ;
                BCF PORTB, 5          ;
                BCF PORTB, 4          ;PASO 1
                CALL RETARDO
```

```
                INCF COUNT2, 1
                GOTO loop_ah_2
end_loop_ah_1:
                MOVLW h'04'                    ;
                SUBWF PORTA, W          ;
                BTFSC STATUS, Z         ;
                GOTO end_loop_ah_1      ;
                GOTO loop1

end_loop_ah_2:
                MOVFW COUNT2
                MOVWF PORTD
                INCF COUNT1, 1
                CLRF COUNT2
                GOTO loop_ah_1

RETARDO:                ;
        MOVLW cte1      ;
        MOVWF REGB      ;
LOOPB:                  ;
        MOVLW cte2      ;
        MOVWF REGA      ;
LOOPA:                  ;
        DECFSZ REGA     ;
        GOTO LOOPA      ;
        DECFSZ REGB     ;
        GOTO LOOPB      ;
        RETURN          ; CODIGO PARA EL RETARDO
        END
```

3. Utilizando un servo motor realizar el control mostrado en la tabla.

SW2	SW1	SW0	Posición servo	Representación
1	0	0	Izquierda	0
0	1	0	Central	90
0	0	1	Derecha	180

Algoritmo

El algoritmo una vez más es simple. Primero se configuran los puertos de la manera solicitada para poder controlar el servomotor. Después, se determina el ángulo solicitado con el switch haciendo comparaciones. Una vez elegido el ángulo correcto, se manda la señal de encendido por un tiempo entre 1 y 2 ms para después mandar la señal de apagado por un tiempo entre 18 y 19 ms.

Los valores de las constantes A y B necesarias para realizar los retardos de los valores mencionados se obtuvieron de la siguiente manera. Primero se supuso que $A = 255$, por lo que:

$$c = B(3A + 1) + 3B + 3$$
$$c = B(3(255) + 1) + 3B + 3$$
$$c = 769B + 3$$
$$b = \frac{c - 3}{769}$$

Para obtener el número de ciclos requeridos se utilizó la ecuación siguiente:

$$c = \frac{tiempo}{0{,}2x10^{-6}}$$

Por lo que, por ejemplo, para 1.5 ms las ecuaciones dan el siguiente resultado:

$$c = \frac{1{,}5x10^{-3}}{0{,}2x10^{-6}} = 7500$$
$$b = \frac{7500 - 3}{769} = 9{,}74$$

Los resultados se resumen en la siguiente tabla:

Tiempo	Valor B	Valor B código
1 ms	6.49	6
1.5 ms	9.74	10
2 ms	13	12
19 ms	123.5	124
18.5 ms	120.28	120
18 ms	117.03	117

Código comentado

El código para cada uno de los ángulos se muestra aquí.

```
include <p16f877.inc>
REGB     equ h'21'
REGA     equ h'22'
; Constantes de retardo
cteA      equ h'FF';
cteB0     equ .6   ;6.49
cteB01    equ .124 ;123.5
cteB90    equ .10  ;9.74
cteB901   equ .120 ;120.28
cteB180   equ .13  ;13
cteB1801  equ .117   ;117.03

          ORG 0
GOTO INICIO
          ORG 5
```

```
INICIO:
        CLRF PORTA
        BSF STATUS,RP0
        BCF STATUS,RP1
        ;Configura los puertos
        ; B, C y D como salidas
        MOVLW H'00'
        MOVWF TRISB
        MOVWF TRISC
        MOVWF TRISD
        ; El puerto A se configura
        ; como entrada digital
        MOVLW 06H
        MOVWF ADCON1
        ; Configura el puerto A como
        ; entrada
        MOVLW 3FH
        MOVWF TRISA
        BCF STATUS,RP0
        MOVLW B'00000000'
        MOVWF PORTB
        MOVWF PORTC
        MOVWF PORTD
loop1:

        ;Dependiendo de la posicion
        ; del switch se ingresa
        ; a alguno de los flujos
        ; para cada angulo
        MOVLW h'04'             ;
        SUBWF PORTA, W  ;
        BTFSC STATUS, Z ;
        GOTO ang0       ;

        MOVLW h'02'             ;
        SUBWF PORTA, W  ;
        BTFSC STATUS, Z ;
        GOTO ang90              ;

        MOVLW h'01'             ;
        SUBWF PORTA, W  ;
        BTFSC STATUS, Z ;
        GOTO ang180


goto loop1


ang0:
        ;Verifica que se siga en la
```

```
        ; misma opcion para mantener
        ; el angulo
        MOVLW h'04'             ;
        SUBWF PORTA, W  ; PORTA = 4 ?
        BTFSS STATUS, Z ;
        GOTO loop1      ; VAMOS A PARO

        MOVF PORTA
        MOVWF PORTD
; Genera la onda con periodo de 20ms
; y tiermpo encendido de 1ms
        BSF PORTC, 0
        CALL RETARDO0
        CLRF PORTC
        CALL RETARDO01
        GOTO ang0


ang90:
        MOVLW h'02'             ;
        SUBWF PORTA, W  ; PORTA = 2 ?
        BTFSS STATUS, Z ;
        GOTO loop1      ; VAMOS A PARO

        ; Genera la onda con periodo de 20ms
        ; y tiermpo encendido de 1.5ms
        BSF PORTC, 0
        CALL RETARDO90
        CLRF PORTC
        CALL RETARDO901
        GOTO ang90

ang180:
        MOVLW h'01'             ;
        SUBWF PORTA, W  ; PORTA = 1 ?
        BTFSS STATUS, Z ;
        GOTO loop1      ; VAMOS A PARO


        ; Genera la onda con periodo de 20ms
        ; y tiermpo encendido de 2ms
        BSF PORTC, 0
        CALL RETARDO180
        CLRF PORTC
        CALL RETARDO1801
        GOTO ang180
```

El código de los retardo es idéntico entre las diferentes versiones, solo cambiando las constantes, por lo que solo se muestra el retardo de 1.5 ms.

```
RETARDO90:              ;
```

```
        MOVLW cteB90    ;
        MOVWF REGB      ;
LOOPB90:                        ;
        MOVLW cteA      ;
        MOVWF REGA      ;
LOOPA90:                        ;
        DECFSZ REGA     ;
        GOTO LOOPA90    ;
        DECFSZ REGB     ;
        GOTO LOOPB90    ;
        RETURN          ; CODIGO PARA EL RETARDO
```

2. Conclusiones

- López Becerra Ricardo: En esta práctica aprendimos el control de actuadores con el PIC16. Esto es importante ya que el uso de actuadores es muy común en toda clase de proyectos y es una de las principales funciones de un microcontrolador. En esta ocasión experimentamos con 3 de los principales tipos de motores: Corriente directa, a pasos y servomotor. Cada uno se controló de manera característica, siendo el más complicado de controlar en nuestra opinión el motor a pasos, como resultado de que había que controlar la cantidad de ciclos que se ejecutaban.
- Navarrete Zamora Aldo Yael: La práctica de control de actuadores nos muestra la manera de poder controlar los motores a pasos, servomotores y de corriente directa. Pudimos realizar las distintas polarizaciones para los motores así como las configuraciones de los puertos seriales. Hubo algunas complicaciones de tiempo debido a algunos atentados dentro de la institución, sin embargo los ejercicios descritos en el reporte concluyeron de manera exitosa.