**Universidad Nacional Autónoma de México**

**Facultad de Ingeniería**

**Laboratorio de Microcomputadoras**

# Práctica No.5: Control de actuadores

Profesor: Rubén Anaya García

Alumnos:

- Murrieta Villegas Alfonso
- Reza Chavarría, Sergio Gabriel
- Valdespino Mendieta Joaquín

Grupo: 4

Semestre: 2021-2

**Práctica 5: Control de actuadores**

**Objetivos**

Emplear los puertos paralelos que contiene un microcontrolador, para controlar la operación de dos motores de corriente directa, motores a pasos y servomotores.

**Desarrollo (Práctica 5)**

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 5.1, descritas en la tabla 5.1; realizar un programa que ejecute el control indicado en la tabla 5.1.**

Nota: Las tierras de los ambos circuitos están conectados entre sí

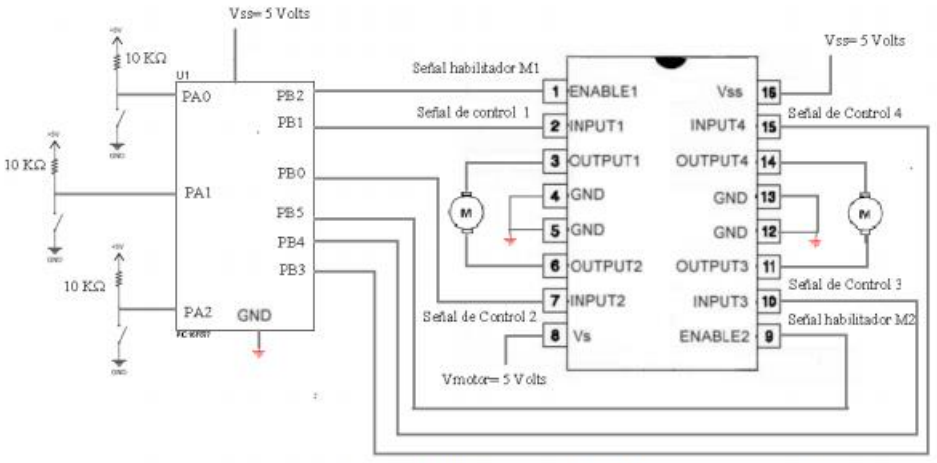



Figura 5.10 Circuito control de motores de CD

La asignación de las terminales queda de la siguiente manera:

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Función | - | - | Eable_M2 | Dir1_M2 | Dir2_M2 | 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 |

*Código del ejercicio:*

```
        processor 16f877
        include<p16f877.inc>

        org 0
        goto INICIO
        org 5

INICIO:
       clrf PORTA
       bsf STATUS,RP0  ;Cambio al Banco 1
       bcf STATUS,RP1
       movlw h'0'
       movwf TRISB     ;Configura Puerto B como salida
       clrf PORTB      ;Limpia los bits de Puerto 1

       movlw 06h       ;Configura puertos A y E como digitales
       movwf ADCON1
       movlw 3fh       ;Configura el Puerto A como entrada
       movwf TRISA
       bcf STATUS,RP0  ;Regresa al Banco 0


LOOP:
    MOVF PORTA,W    ;w<--PORTA
    ANDLW 0X0F  ;W<--PORTA & 00000111
    ADDWF PCL,F         ;PCL<--PORTA & 00000111
    GOTO CONF0          ;PC+0
    GOTO CONF1          ;PC+1
    GOTO CONF2          ;PC+2
    GOTO CONF3          ;PC+3
    GOTO CONF4          ;PC+4
    GOTO CONF5          ;PC+5
    GOTO CONF6          ;PC+6
    GOTO CONF7          ;PC+7
    GOTO CONF8          ;PC+8
```

```
CONF0:              ;PARO PARO
    MOVLW 0X00
    MOVWF PORTB
    GOTO LOOP
CONF1:              ;PARO horario
    MOVLW 0X06
    MOVWF PORTB
    GOTO LOOP
CONF2:              ;PARO ANTIHORIO
    MOVLW 0X05
    MOVWF PORTB
    GOTO LOOP
CONF3:              ;HORARIO PARO
    MOVLW 0X30
    MOVWF PORTB
    GOTO LOOP
CONF4:              ;ANTIHORARIO PARO
    MOVLW 0X28
    MOVWF PORTB
    GOTO LOOP
CONF5:              ;HORARIO HORARIO
    MOVLW 0X36
    MOVWF PORTB
    GOTO LOOP
CONF6:              ;antihorario antihorario
    MOVLW 0X2D
    MOVWF PORTB
    GOTO LOOP
CONF7:              ;horario antihorario
    MOVLW 0X35
    MOVWF PORTB
    GOTO LOOP
CONF8:              ;ANTIHORARIO HORARIO
    MOVLW 0x2E
    MOVWF PORTB
    GOTO LOOP


    END
```

*Código 1: Ejercicio 1, motores de corriente directa*

Para la asignación de los puertos se realizó la configuración al puerto A y E como digitales, al puerto A como entrada y al puerto B como salida.

Para el manejo del ejercicio se utilizó el valor de PC (PCL) y el valor obtenido de la entrada A. Al obtener el valor obtenido de la entrada se obtendrá el valor a sumar a PCL. La suma se realizará para llegar a cada estado a partir de GOTO.

A continuación, se mencionarán los valores obtenidos a partir de la asignación de terminales. Para que el motor se active se necesita activar el bit correspondiente al habilitador, si esta abajo se apagará. Una vez activado el habilitador se necesita tener un bit activo y un bit abajo para generar el movimiento, dependiendo del orden se moverá en sentido horario y anti horario. Los valores de las terminales se le asignarán al puerto B.

- Estado 0 (Paro, Paro): Valor 00 o 0000 0000
- Estado 1 (Paro, Horario): Valor 06 o 0000 0110
- Estado 2 (Paro, Anti horario): Valor 05 o 0000 0101
- Estado 3 (Horario, Paro): Valor 30 o 0011 0000
- Estado 4 (Anti horario, Paro): Valor 28 o 0010 1000
- Estado 5 (Horario, Horario): Valor 36 o 0011 0110
- Estado 6 (Anti horario, Anti horario): Valor 2D o 0010 1101
- Estado 7 (Horario, Anti horario): Valor 35 o 0011 0101
- Estado 8 (Anti horario, Horario): Valor 2E o 0010 1110

*Diagrama de flujo*

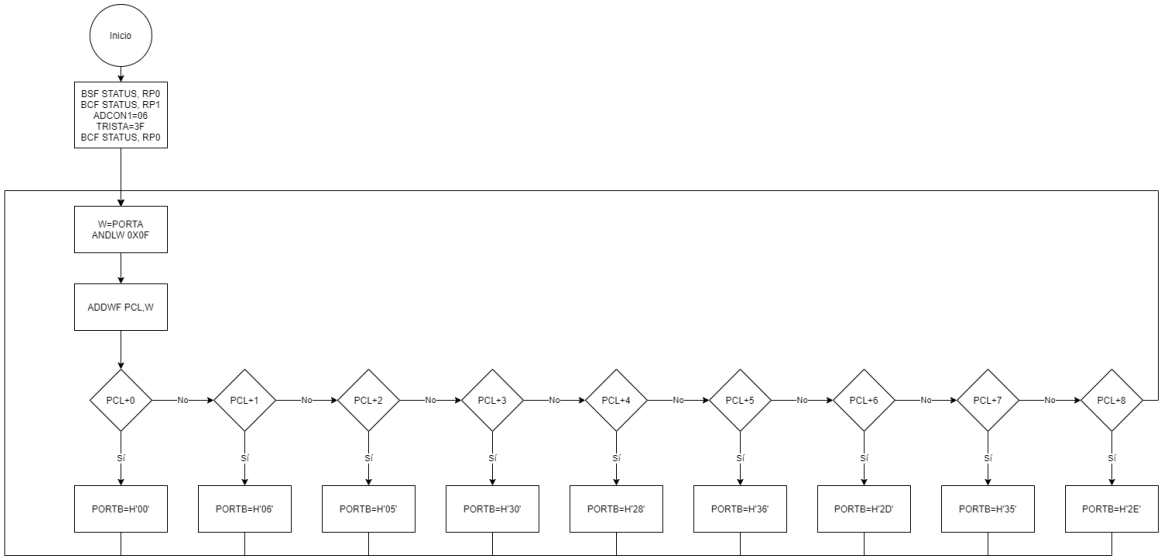



*Diagrama 1: Ejercicio 1 Motores de corriente directa*

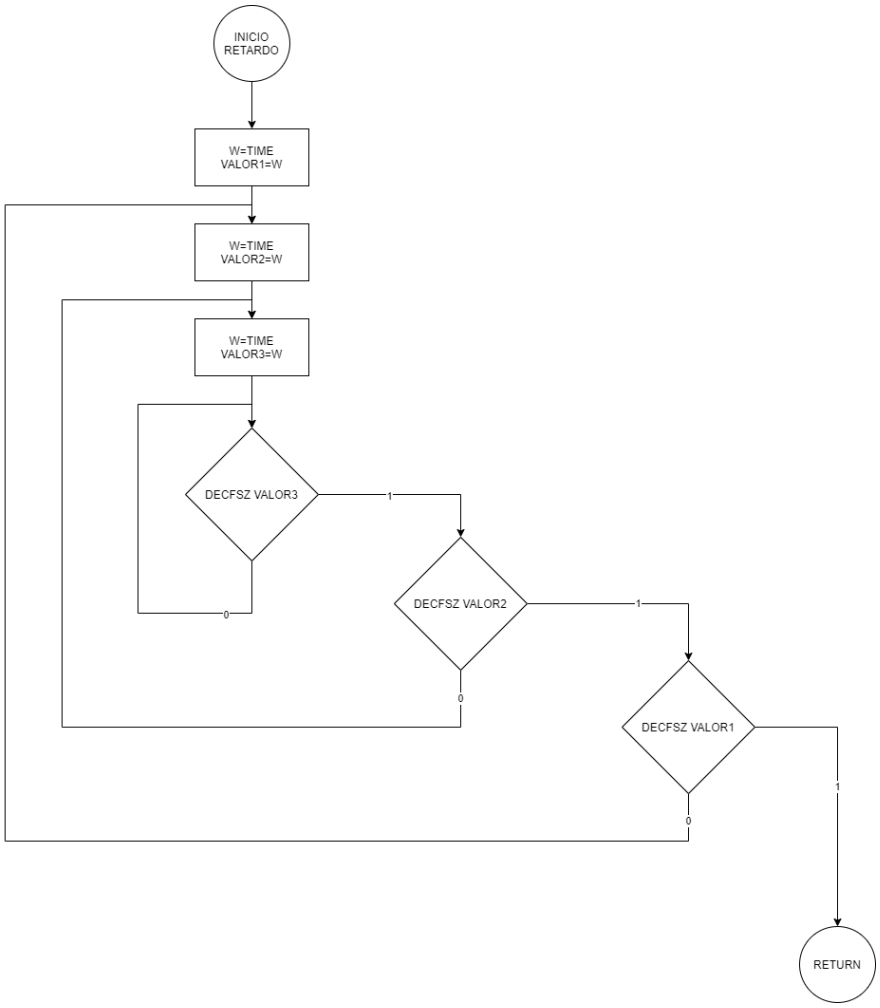
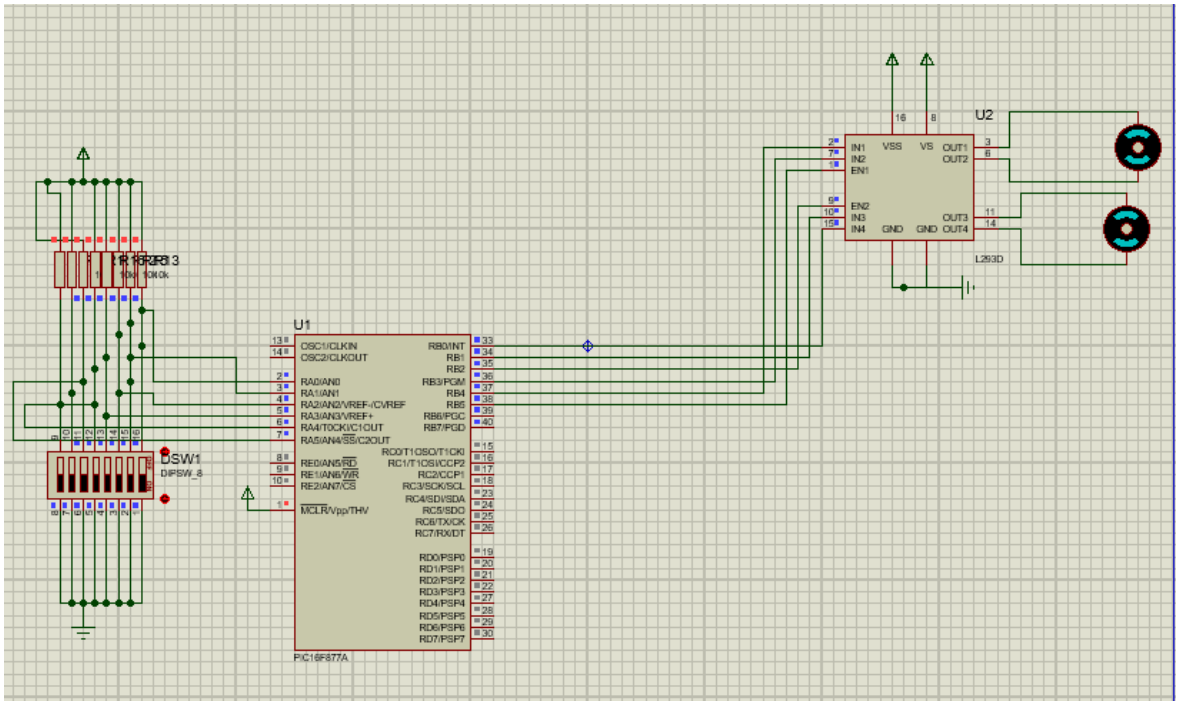
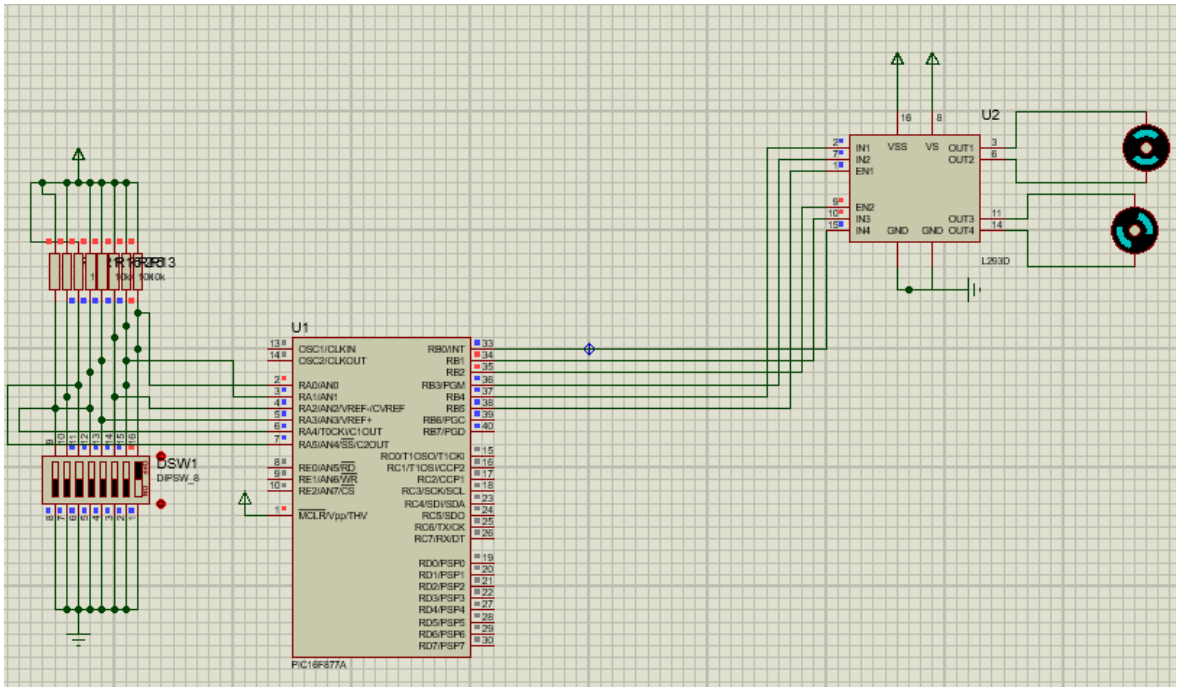



*Diagrama 2: Subrutina general de retardo*

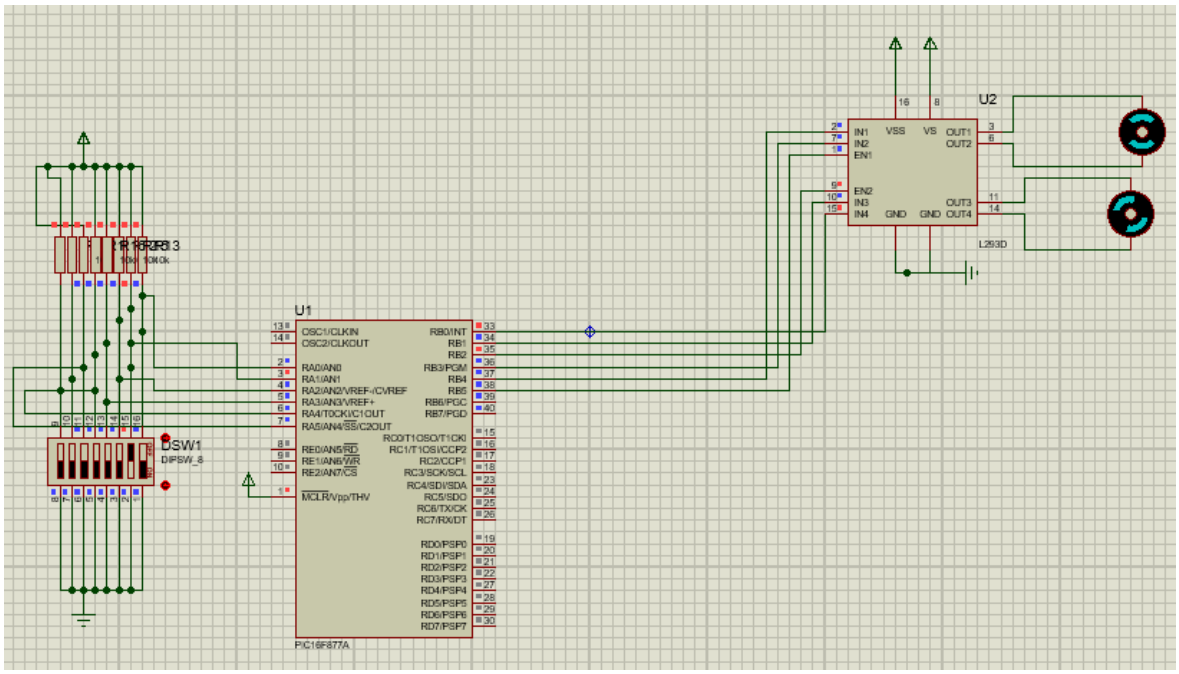
*Simulación*



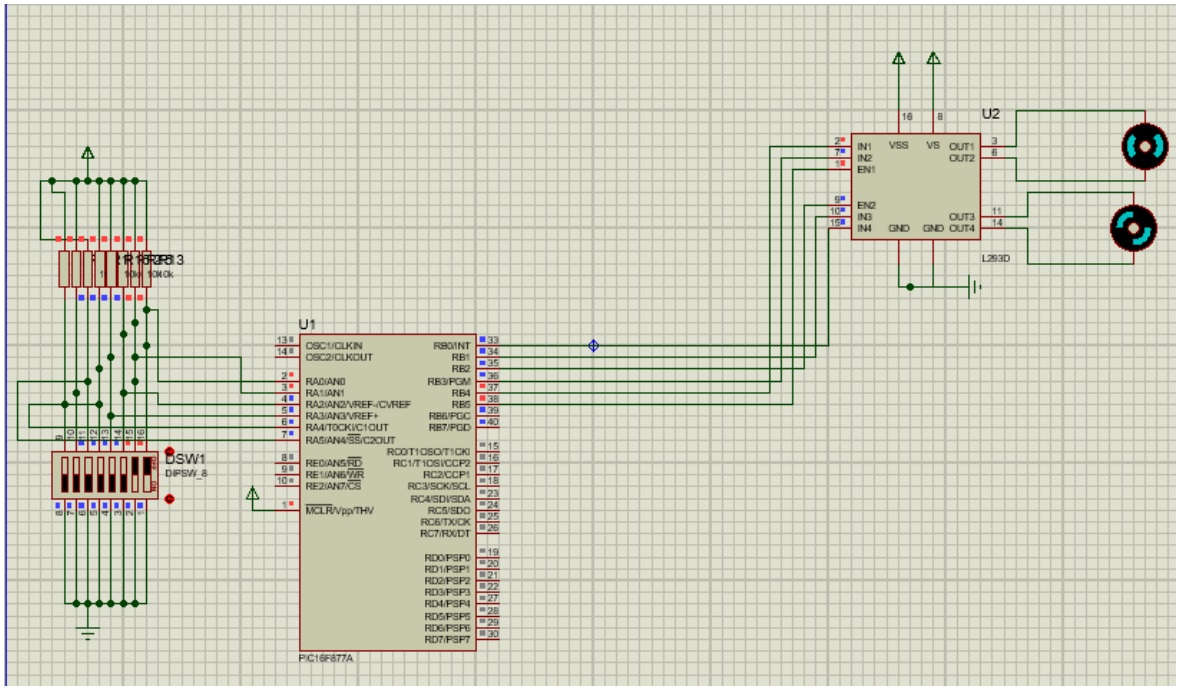

*Simulación 1: Estado 0 (Paro, Paro)*



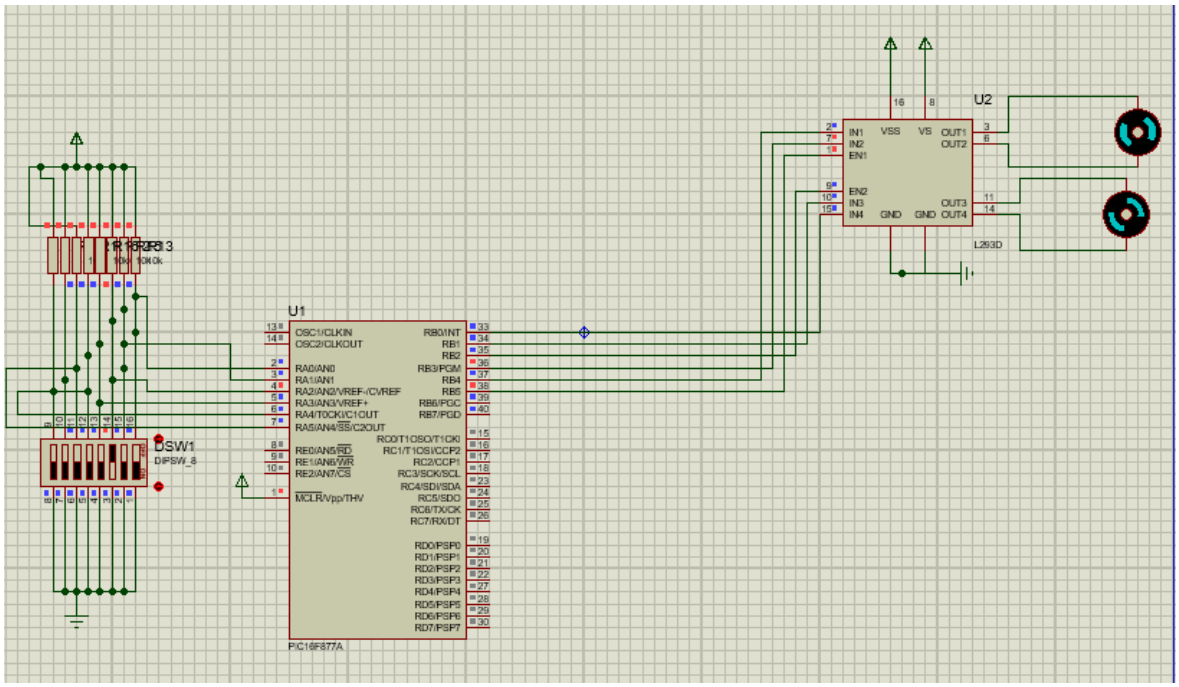

*Simulación 2: Estado 01 (Paro, Horario)*



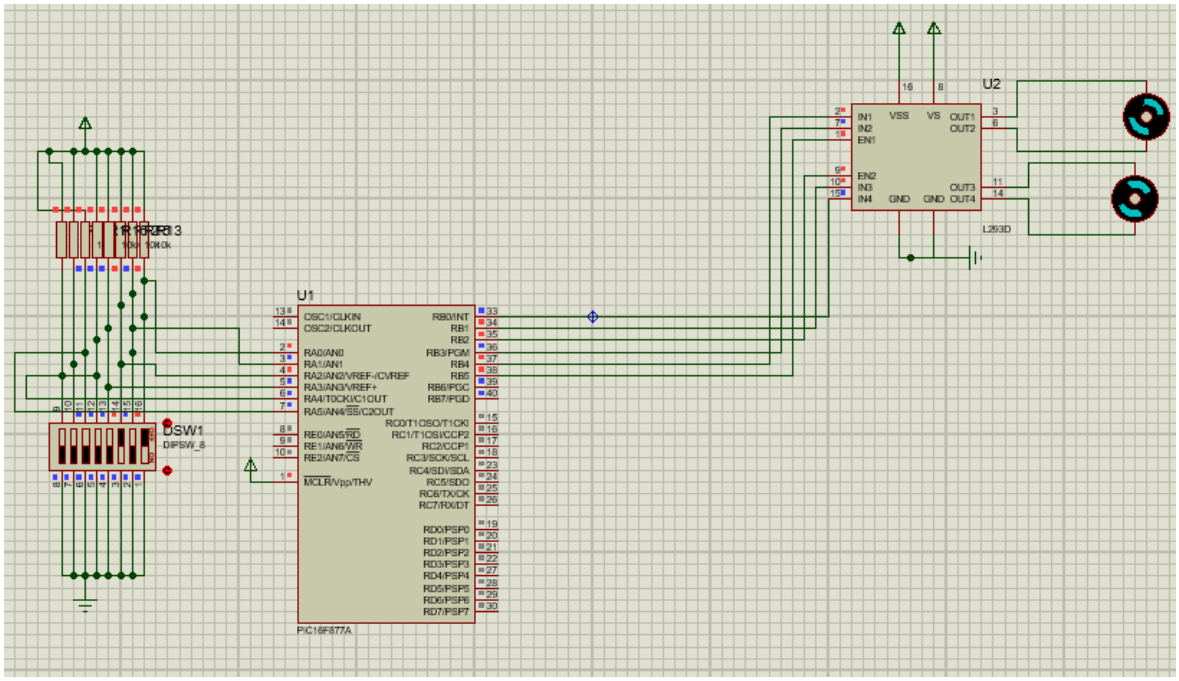

*Simulación 3: Estado 02( Paro, Anti horario)*



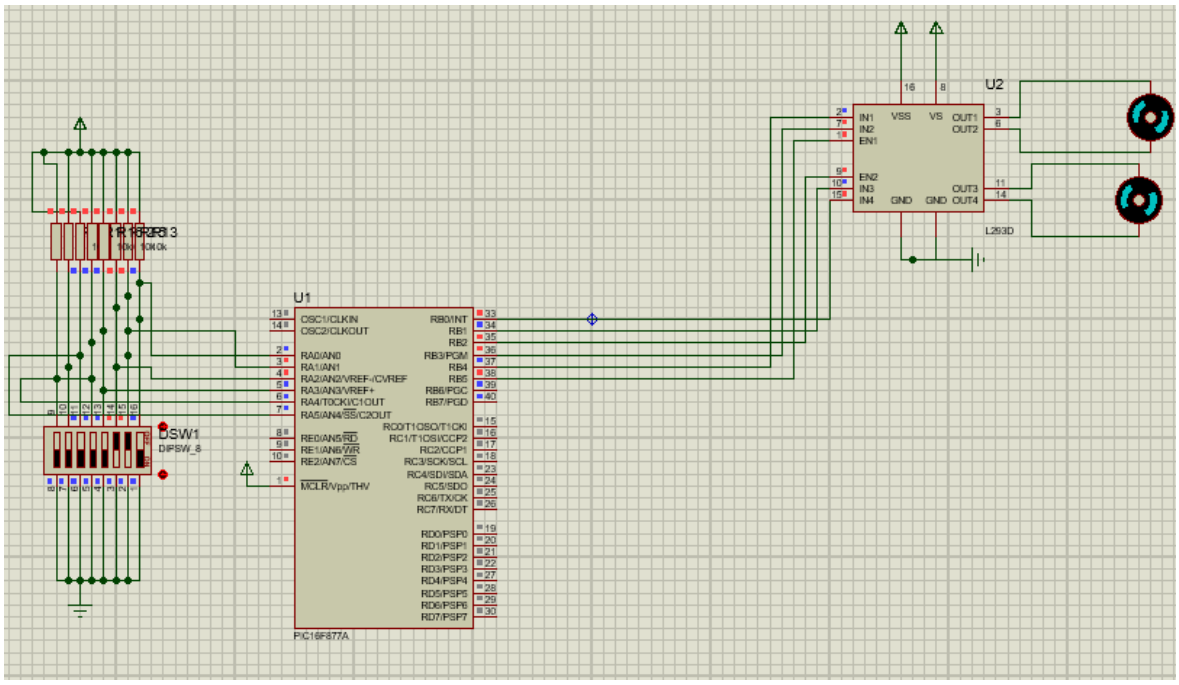

*Simulación 4: Estado 03 (Horario, Paro)*



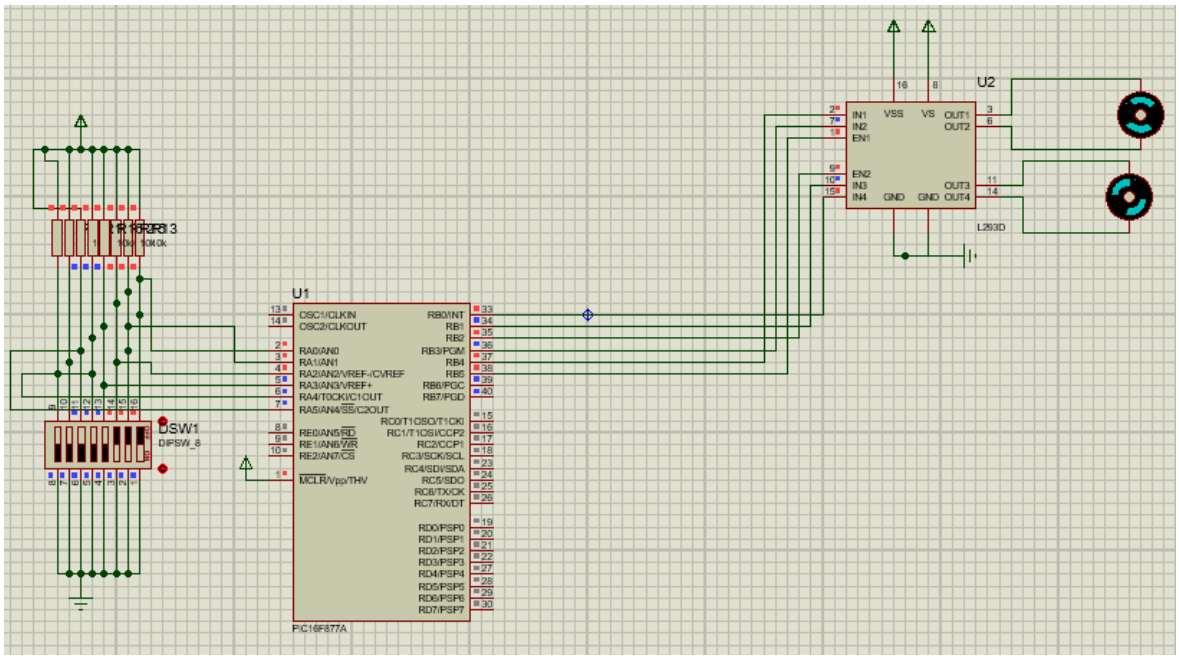

*Simulación 5: Estado 04 (Anti horario, Paro)*



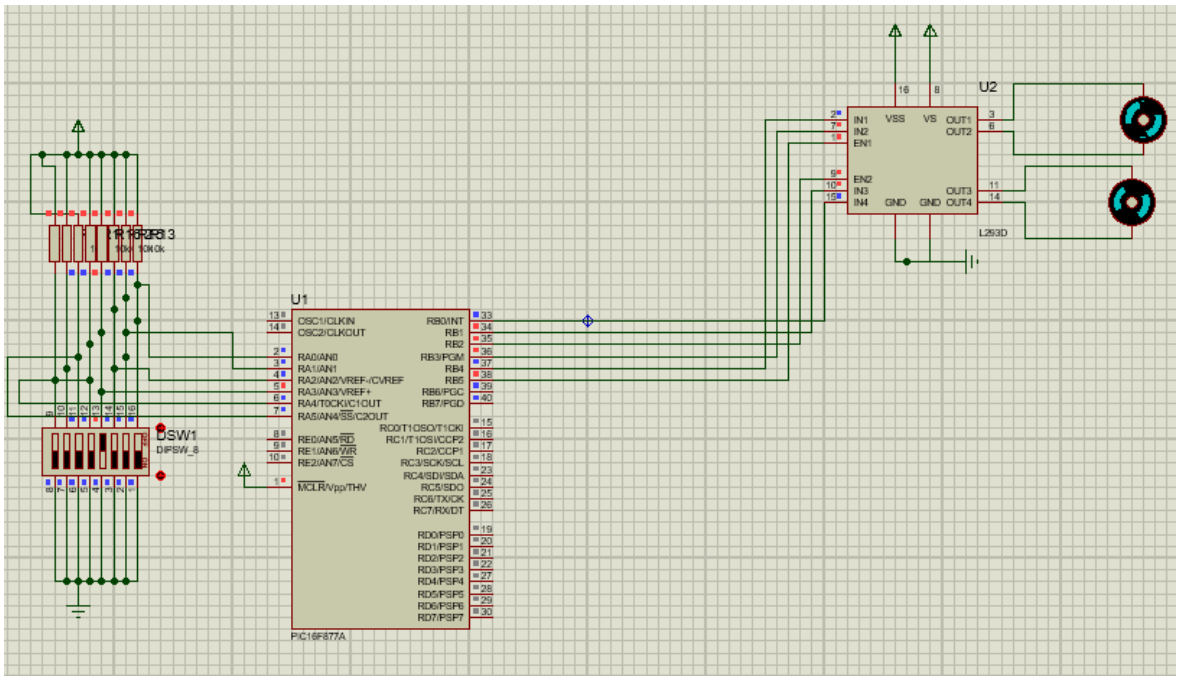

*Simulación 6: Estado 05 ( Horario, Horario)*




*Simulación 7: Estado 06 (Anti horario, Anti horario)*




*Simulación 8: Estado 07 (Horario, Anti horario)*




*Simulación 9: Estado 08 (Anti horario, Horario)*

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

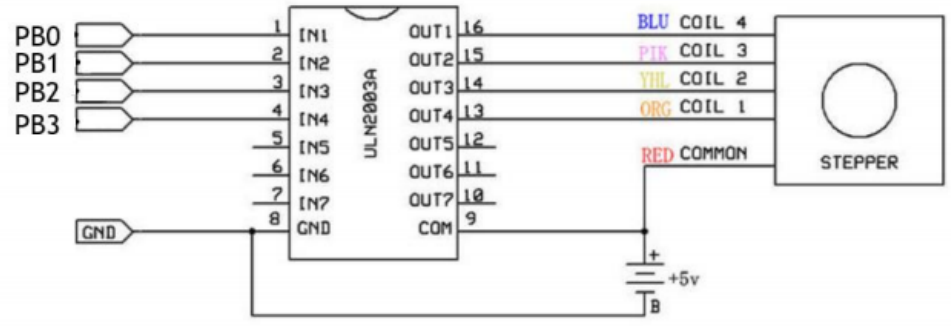



Figura 5.11 Conexión de motor a pasos

| Dato Puerto Paralelo | Motor a pasos |
|---|---|
| 0x00 | Motor en paro |
| 0x01 | Gira en sentido horario durante 5 segundos |
| 0x02 | Gira en sentido anti horario por 10 segundos |
| 0x03 | Gira cinco vueltas en sentido horario |
| 0x04 | Gira 10 vueltas en sentido anti horario |

Tabla 5.2 Control del motor a pasos

*Código del ejercicio:*

Al igual que en los ejercicios anteriores se configuró el puerto A como puerto de entrada y el puerto b como salida.

Se utilizaron valores para realizar los retardos correspondientes para generar esperas de 5 y 10 segundos. Además, se utilizaron secciones de memoria para el conteo de vueltas que se han generado, con respecto a los nuevos casos.

Se realizó la operación de obtener el valor de la entrada y esta sumarla al valor de PC, para que el programa se dirija al estado correspondiente.

Para el estado 0, se manda la salida de 0x00 para generar un paro hasta que se seleccione otra opción.

Para el estado 1, se realizó la salida correspondiente para el movimiento horario del motor. Estos valores generan giro de 90, los cuales son 0x0C, 0x06, 0x03 y 0x09. En este caso por cada realización del giro se realizará un retardo equivalente a 5 segundos.

$$A = B = FF = 255$$

$$C = 7F = 127$$

$$C[B(3A + 1) + 3B + 1] + 3C + 3 = 24904576\mu s$$

$$(24903576) * 0.2 = 4.9809152\ seg$$

Para el estado 2, se realizó la salida correspondiente al movimiento anti horario del motor. Se invirtieron la asignación de la salida B, 0x09, 0x03, 0x06 y 0x0C. La subrutina de retardo genera una pausa aproximada de 10 segundos.

$$A = B = C = FF = 255$$

$$C[B(3A + 1) + 3B + 1] + 3C + 3 = 50005248\ \mu s$$

$$(50005248) * 0.2 = 10.0010496\ seg$$

Para el estado 3 y 4 se siguió la asignación de valores al puerto de salida para el giro horario y anti horario, respectivamente. Se utilizó una comparación entre el valor de vueltas asignados y el valor de vueltas completadas (5 vueltas para el estado 3 y 10 para el estado 4). La comprobación de la realización de los giros correspondientes se realizó a partir de la operación XORWF, para saber si ambos valores eran iguales y revisar si al generar el resultado de 0.

```
    processor 16f877
    include<p16f877.inc>

valor1 equ h'21'    ;equivalencia
valor2 equ h'22'
valor3 equ h'23'

giroder equ h'24'   ;Contadores de giros
giroizq equ h'25'

giroderTop equ h'26'    ;Topes de giros
giroizqTop equ h'27'

ctel equ 70h        ;Retardo normal
cte2 equ 70h
cte3 equ 70h

ctel_5 equ 0xff     ;Retardo de 5 segundos
cte2_5 equ 0xff
cte3_5 equ 0x7f
                    ;Retardo de 10 segundos
ctel_10 equ 0xff
cte2_10 equ 0xff
cte3_10 equ 0xff


        org 0
        goto INICIO
        org 5


INICIO:
       clrf PORTA
       bsf STATUS,RP0  ;Cambio al Banco 1
       bcf STATUS,RP1
       movlw h'0'
       movwf TRISB     ;Configura Puerto B como salida
       clrf PORTB      ;Limpia los bits de Puerto 1

       movlw 06h       ;Configura puertos A y E como digitales
       movwf ADCON1
       movlw 0x3f      ;Configura el Puerto A como entrada
       movwf TRISA
       bcf STATUS,RP0  ;Regresa al Banco 0


LOOP:
    CLRF  giroder
    CLRF  giroizq
    MOVF PORTA,W    ;w<--PORTA
    ANDLW 0X0F  ;W<--PORTA & 00000111
    ADDWF PCL,F         ;PCL<--PORTA & 00000111
    GOTO CONF0          ;PC+0
    GOTO CONF1          ;PC+1
    GOTO CONF2          ;PC+2
    GOTO CONF3          ;PC+3
    GOTO CONF4          ;PC+4

CONF0:
    MOVLW 0X00
    MOVWF PORTB
    GOTO LOOP
```

```
CONF1:              ;derecho
    MOVLW 0X0C
    MOVWF PORTB
    CALL RETARDO_1
    MOVLW 0X06
    MOVWF PORTB
    CALL RETARDO_1
    MOVLW 0X03
    MOVWF PORTB
    CALL RETARDO_1
    MOVLW 0X09
    MOVWF PORTB
    CALL RETARDO_1
    GOTO LOOP
CONF2:              ;izquierdo
    MOVLW 0X09
    MOVWF PORTB
    CALL RETARDO_2
    MOVLW 0X03
    MOVWF PORTB
    CALL RETARDO_2
    MOVLW 0X06
    MOVWF PORTB
    CALL RETARDO_2
    MOVLW 0X0C
    MOVWF PORTB
    CALL RETARDO_2
    GOTO LOOP


  CONF3:              ;derecho

      MOVLW 0X0C
      MOVWF PORTB
      CALL RETARDO
      MOVLW 0X06
      MOVWF PORTB
      CALL RETARDO
      MOVLW 0X03
      MOVWF PORTB
      CALL RETARDO
      MOVLW 0X09
      MOVWF PORTB
      CALL RETARDO

      INCF giroder
      MOVLW 0X05
      MOVWF giroderTop
      MOVF giroder,W
      XORWF giroderTop
      BTFSC STATUS,2
      GOTO RETARDO_SPACE
      GOTO CONF3
```

```
CONF4:              ;izquierdo
    MOVLW 0X09
    MOVWF PORTB
    CALL RETARDO
    MOVLW 0X03
    MOVWF PORTB
    CALL RETARDO
    MOVLW 0X06
    MOVWF PORTB
    CALL RETARDO
    MOVLW 0X0C
    MOVWF PORTB
    CALL RETARDO

    INCF giroizq
    MOVLW 0X0A
    MOVWF giroizqTop
    MOVF giroizq,W
    XORWF giroizqTop
    BTFSC STATUS,2
    GOTO RETARDO_SPACE
    GOTO CONF4


RETARDO ;retardo de tres niveles
    MOVLW cte1
    MOVWF valor1
tres MOVLW cte2
    MOVWF valor2
dos MOVLW cte3
    MOVWF valor3
uno DECFSZ valor3 ;desincrementa valor 3
    GOTO uno
    DECFSZ valor2 ;desincrementa valor 3
    GOTO dos
    DECFSZ valor1 ;desincrementa valor 3
    GOTO tres
    RETURN

RETARDO_SPACE ;retardo de tres niveles
    MOVLW cte1_5
    MOVWF valor1
tres_s
    MOVLW cte2_5
    MOVWF valor2
dos_s
    MOVLW cte3_5
    MOVWF valor3
uno_s
    DECFSZ valor3 ;desincrementa valor 3
    GOTO uno_s
    DECFSZ valor2 ;desincrementa valor 3
    GOTO dos_s
    DECFSZ valor1 ;desincrementa valor 3
    GOTO tres_s
    GOTO LOOP
```

```
RETARDO_1 ;retardo de tres niveles
    MOVLW ctel_5
    MOVWF valor1
tres_1
    MOVLW cte2_5
    MOVWF valor2
dos_1
    MOVLW cte3_5
    MOVWF valor3
uno_1
    DECFSZ valor3 ;desincrementa valor 3
    GOTO uno_1
    DECFSZ valor2 ;desincrementa valor 3
    GOTO dos_1
    DECFSZ valor1 ;desincrementa valor 3
    GOTO tres_1
    RETURN


RETARDO_2   ;retardo de tres niveles
    MOVLW ctel_10
    MOVWF valor1
tres_2
    MOVLW cte2_10
    MOVWF valor2
dos_2
    MOVLW cte3_10
    MOVWF valor3
uno_2
    DECFSZ valor3 ;desincrementa valor 3
    GOTO uno_2
    DECFSZ valor2 ;desincrementa valor 3
    GOTO dos_2
    DECFSZ valor1 ;desincrementa valor 3
    GOTO tres_2
    RETURN

    END
```

*Código 2: Ejercicio con el motor a pasos*

*Diagrama de flujo*

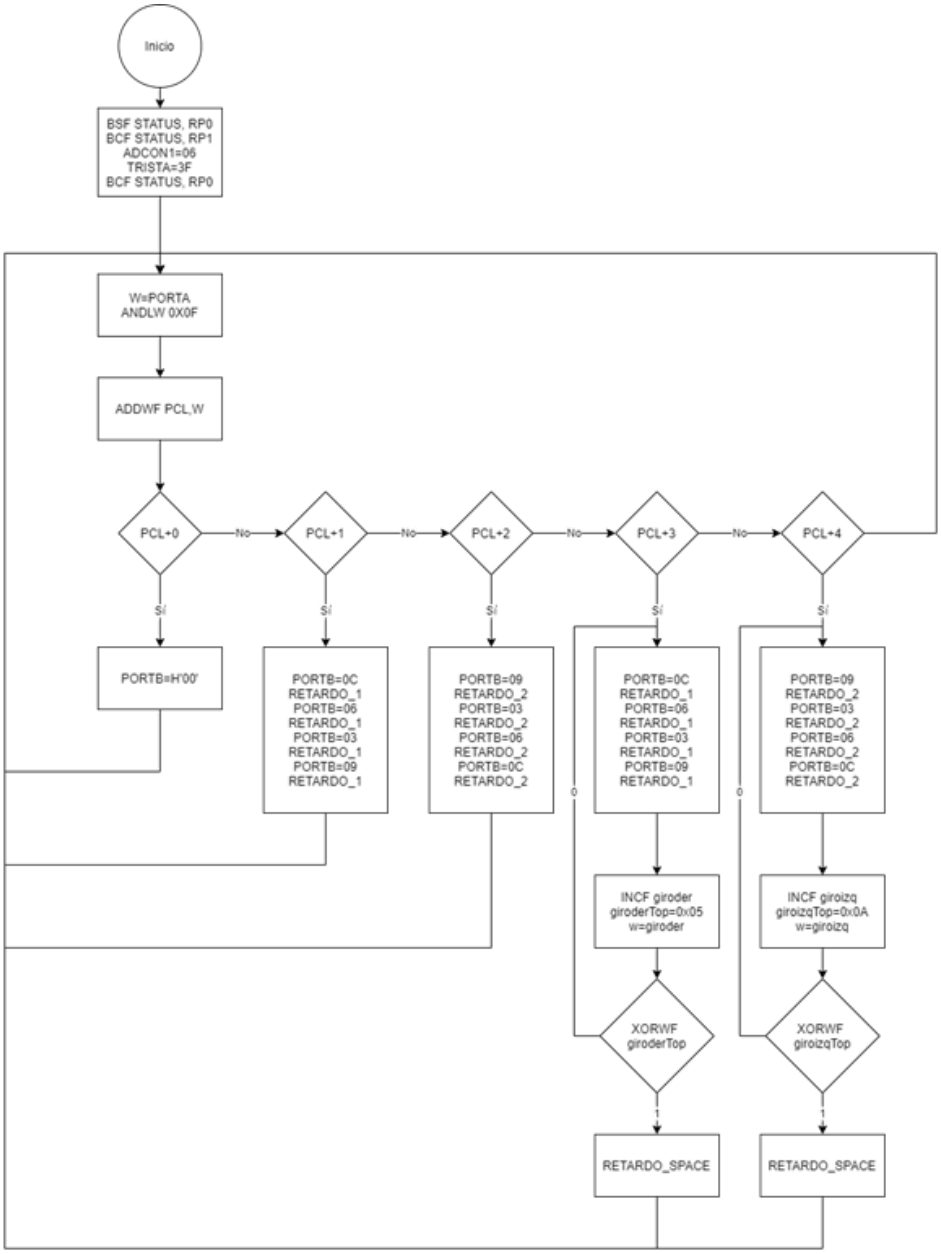
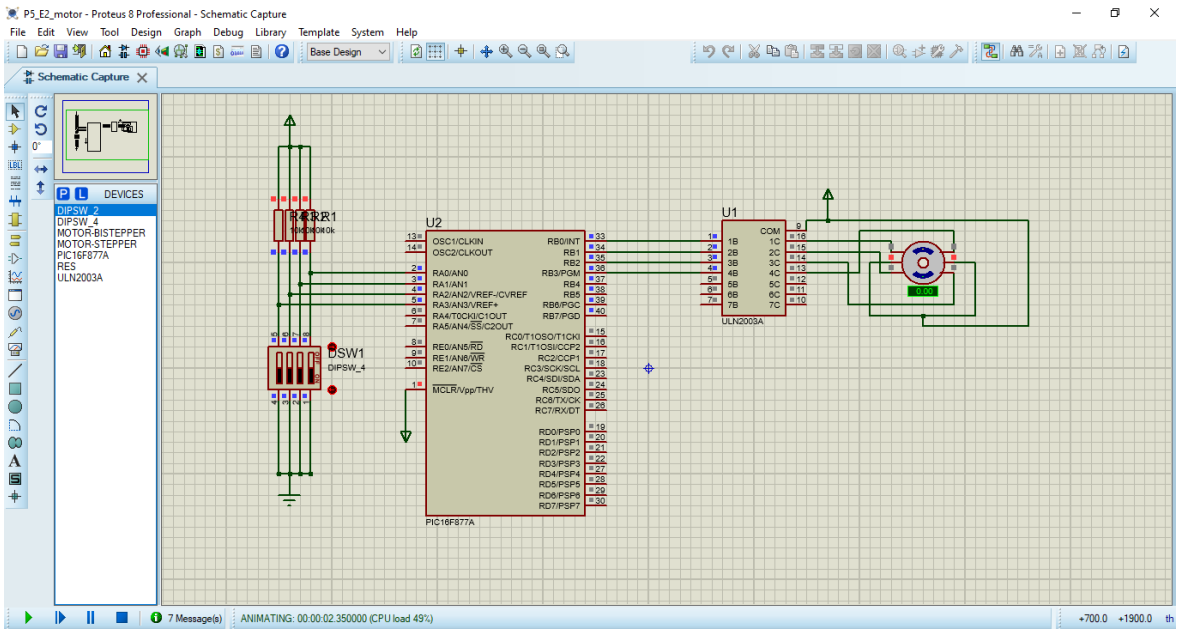


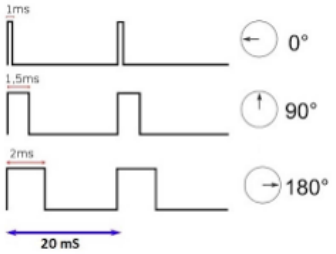

*Diagrama 3: Motores a paso*

*Simulación*




*Simulación 10: Simulación ejercicio 2*

## 3.- Utilizando un servo motor realizar el control mostrado en la tabla No. 5.3




| SW2 | SW1 | SW0 | Posición Servo | Representación |
|---|---|---|---|---|
| 1 | 0 | 0 | Izquierda | 0° |
| 0 | 1 | 0 | Central | 90° |
| 0 | 0 | 1 | Derecha | 180° |

*Código del ejercicio:*

Se realizó la asignación del puerto A como entrada y el puerto B como salida.

Se realizaron los 3 estados correspondientes para la entrada. Para revisar la entrada se tomó en cuenta la resta del valor obtenido de la entrada y el valor correspondiente para comprobar el estado.

Para el primer giro descrito se asignó el valor a la salida correspondiente al giro. Una vez asignado se le dan valores diferentes a las variables que utilizarán la subrutina de retardo, esto para generar las diferentes señales, tanto en alto y en bajo, en el puerto de salida.

```
#INCLUDE<P16F877A.inc>

VALOR1 EQU H'51'
VALOR2 EQU H'52'
VALOR3 EQU H'53'

CTE2 EQU .2
CTE3 EQU .82

TIME EQU 0x20       ;Dirección que determinará el tiempo en milisegundos

org 0
goto INICIO
org 5

INICIO:
    CLRF PORTA      ;Limpiamos PORTA
    BSF STATUS,RP0
    BCF STATUS,RP1  ;Cambiamos al banco 1
    MOVLW h'0'
    MOVWF TRISB     ;Establecemos al PORTB como salida
    CLRF PORTB      ;Limpiamos el puerto B
    MOVLW 0x06
    MOVWF ADCON1    ;Puerto Ay E como digitales
    MOVLW 0x3f
    MOVWF TRISA     ;Puerto A como entrada
    BCF STATUS,RP0  ;Regresamos al banco 0

CICLO:
    MOVLW 0x04          ;W <- PORTA
    SUBWF PORTA,W       ;W <- 0x04 - PORTA
    BTFSS STATUS,Z      ;¿STATUS[Z]==1?, PORTA == 0x04
    GOTO GIRO_90_0
```

```
GIRO_0:
    MOVLW 0x01
    MOVWF PORTB      ;(PORTB) <- 00000001
    MOVLW .5         ;1ms
    MOVWF TIME       ;(TIME) <- 0x01
    CALL RETARDO
    CLRF PORTB       ;(PORTB) <- 0
    MOVLW .190       ;19 ms
    MOVWF TIME       ;Tiempo en bajo
    CALL RETARDO
    GOTO CICLO
GIRO_90_0:
    MOVLW 0x02
    SUBWF PORTA,W    ;W <- PORTA - 0x02
    BTFSS STATUS,Z   ;STATUS[Z] == 1,
    GOTO COMP_180
GIRO_90:
    MOVLW 0x01
    MOVWF PORTB      ;(PORTB) <- 00000001
    MOVLW .15        ;1,5ms
    MOVWF TIME       ;(TIME) <- 0x01
    CALL RETARDO
    CLRF PORTB       ;(PORTB) <- 0
    MOVLW .185       ;18,5ms
    MOVWF TIME       ;Tiempo en bajo
    CALL RETARDO
    GOTO CICLO

COMP_180:
    MOVLW 0x01
    SUBWF PORTA,W    ;W <- PORTA - 0x01
    BTFSS STATUS,Z   ;¿PORTA == 0x01?, ¿Z==1?
    GOTO CICLO


GIRO_180:
    MOVLW 0x01
    MOVWF PORTB      ;(PORTB) <- 00000001
    MOVLW .20        ;2 ms
    MOVWF TIME       ;(TIME) <- 0x01
    CALL RETARDO
    CLRF PORTB       ;(PORTB) <- 0
    MOVLW .180       ;18 5ms
    MOVWF TIME       ;Tiempo en bajo
    CALL RETARDO
    GOTO CICLO


RETARDO:
    MOVF TIME,W      ;Configuración del tiempo
    MOVWF VALOR1
TRES:
    MOVLW CTE2
    MOVWF VALOR2
DOS:
    MOVLW CTE3
    MOVWF VALOR3
UNO:
    DECFSZ VALOR3
    GOTO UNO
    DECFSZ VALOR2
    GOTO DOS
    DECFSZ VALOR1
    GOTO TRES
    RETURN

    END
```

*Diagrama de flujo*

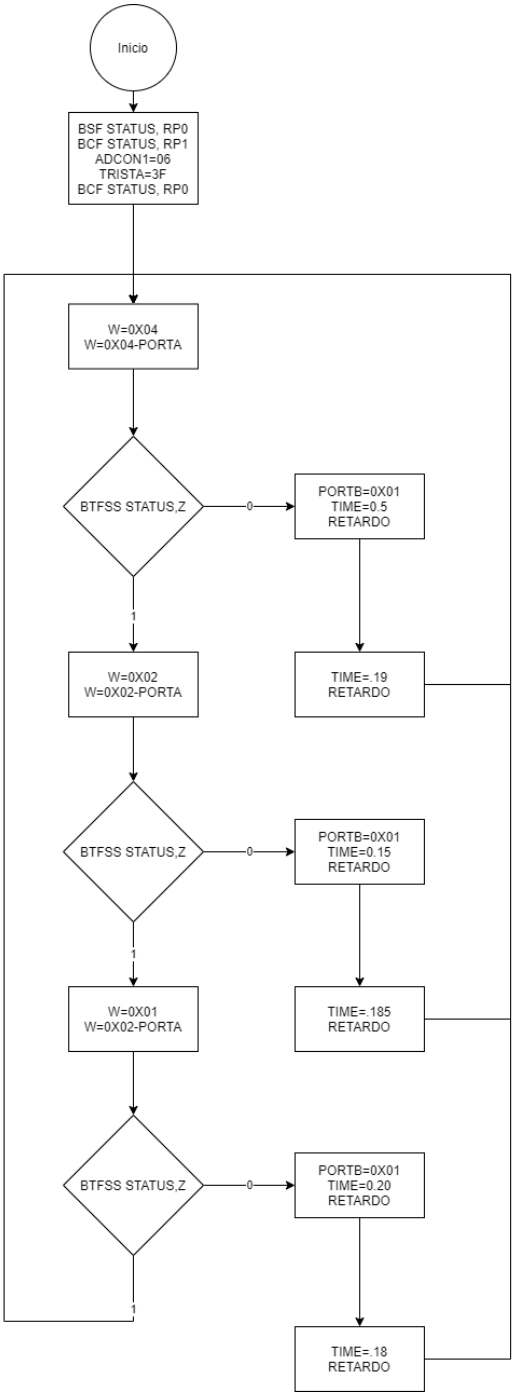
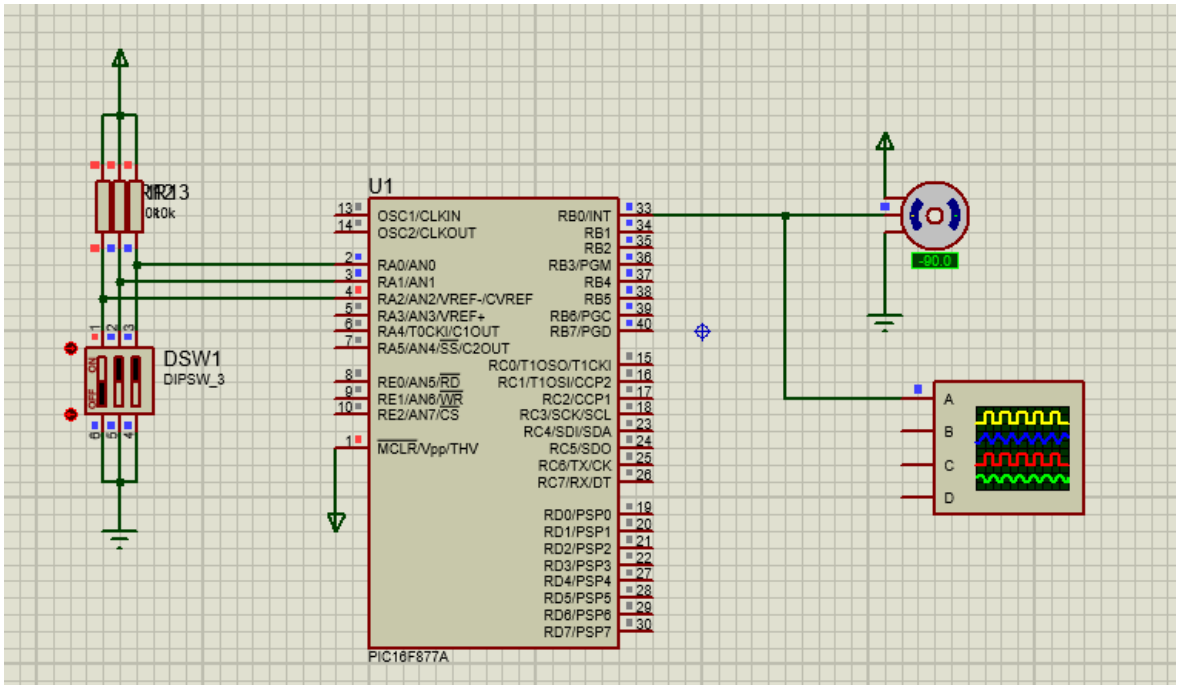
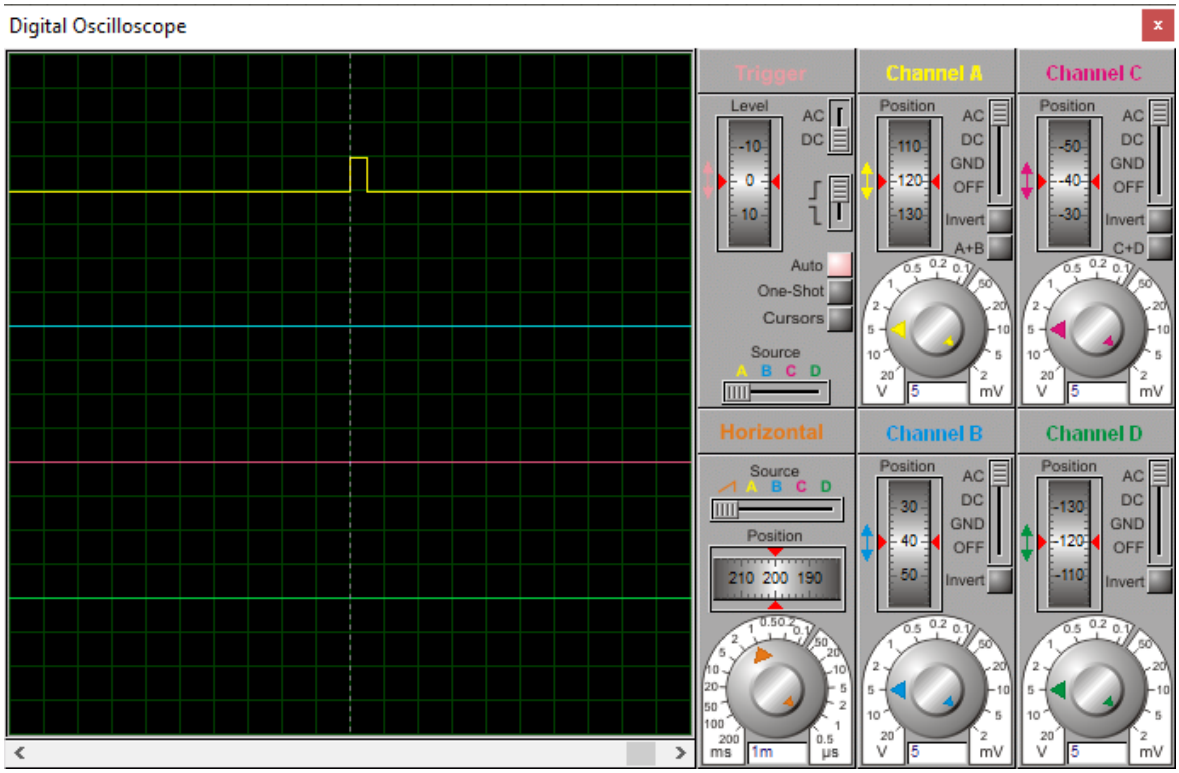



*Diagrama 4: Servomotores*

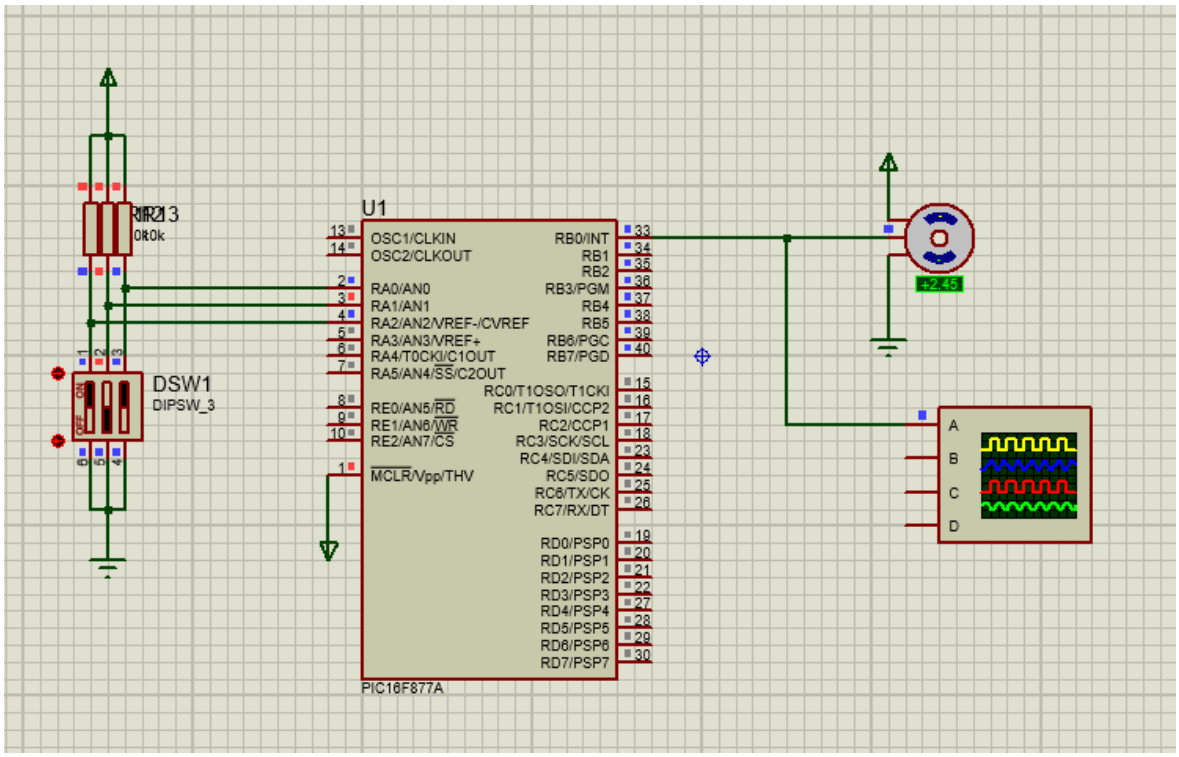
*Simulación*



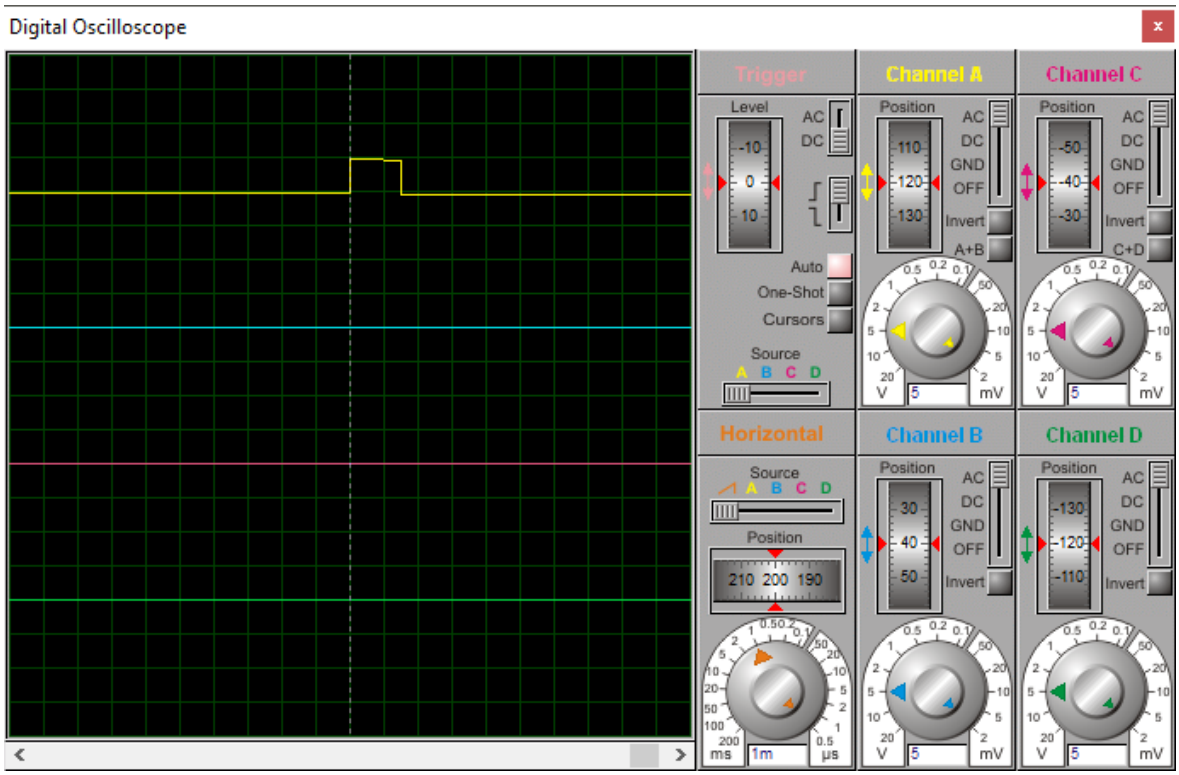

*Simulación 11: Servo 0 grados Izquierda*



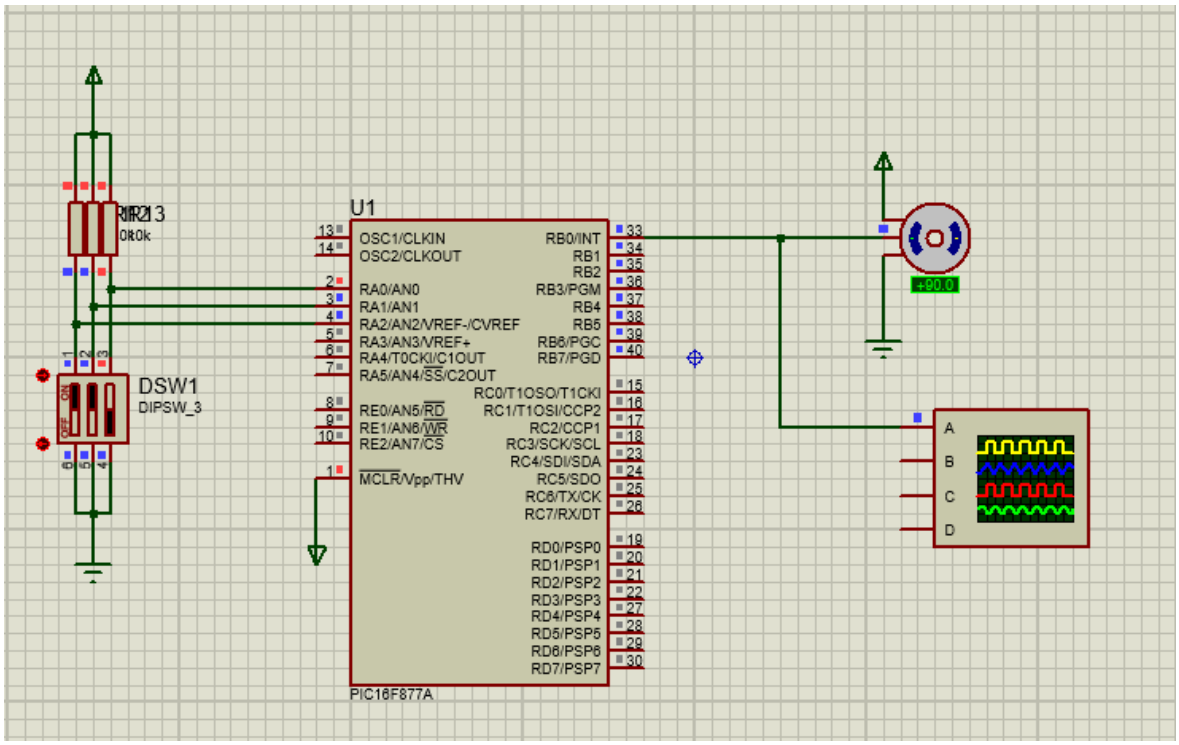

*Oscilograma 1: Señal 1 obtenida de 1 ms*



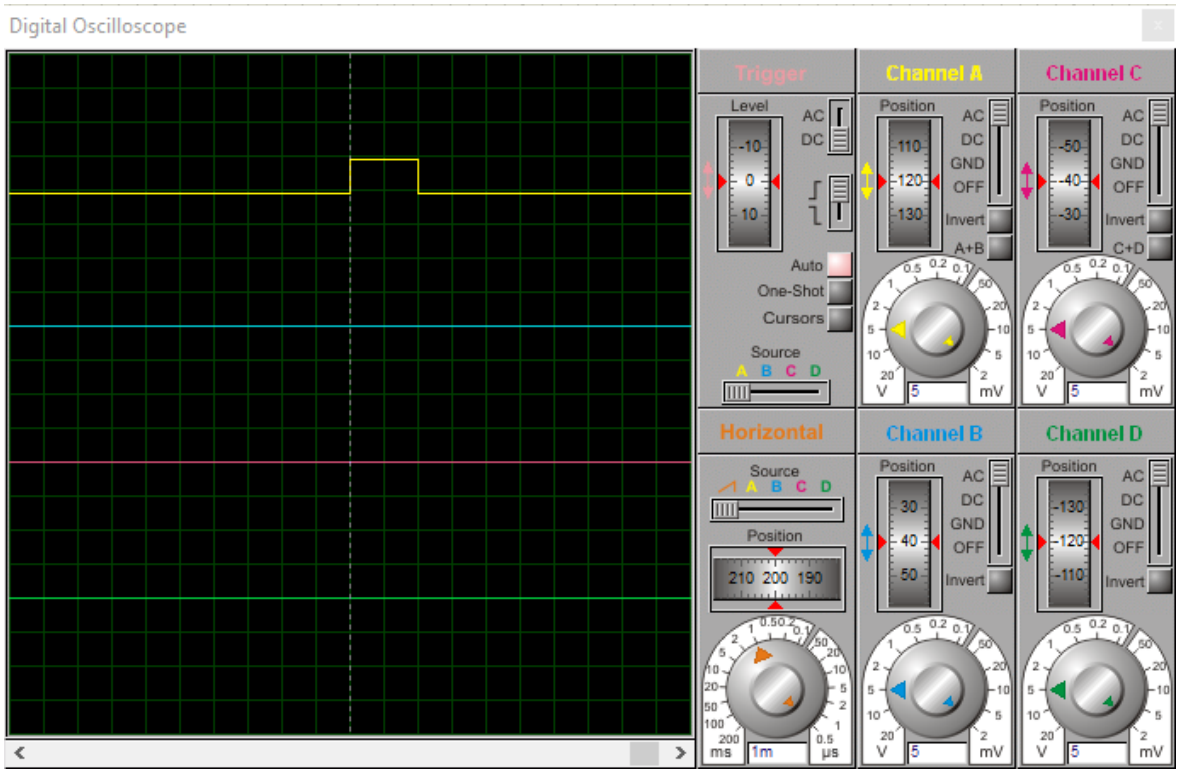

*Simulación 12: Servo 90 grados Central*




*Oscilograma 2: Señal 2 obtenida de 1.5 ms*




*Simulación 13: Servo 180 grados Derecha*




*Oscilograma 3: Señal 3 obtenida de 2 ms*

## Conclusiones:

*Alfonso Murrieta Villegas*

En la presente práctica aprendimos a emplear los puertos de nuestro microcontrolador para poder controlar hardware externo.

A su vez y como recopilación de materias previas, empleamos conceptos de electrónica digital para el uso de puentes H para poder determinar la dirección o estado en el que se quería tener al servomotor.

*Sergio Gabriel Reza Chavarría*

Para la presente práctica se dio el manejo práctico, en el apartado del uso de entradas y salidas, la introducción a los diferentes tipos y los diferentes métodos de uso de motores, como lo fueron el de corriente directa, a pasos y los servomotores. Esto permitirá la implementación de estos elementos en proyectos futuros.

*Joaquín Valdespino Mendieta*

En la realización de la práctica pudimos implementar y comprender las funciones para el control del flujo de datos en diversas aplicaciones, como observamos con la manipulación del motor o servomotor, con técnicas abstraídas de un análisis previo, además se emplearon los puertos paralelos para operar los dichos motores, haciendo énfasis en los de corriente directa, a pasos y servomotores.