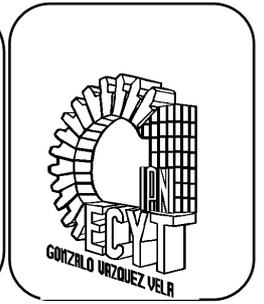




INSTITUTO POLITÉCNICO NACIONAL
Centro de Estudios Científicos y Tecnológicos N° 1
“Gonzalo Vázquez Vela”
Academia de Sistemas Digitales
Prácticas de Micro Electrónica Programable



NOMBRE DEL ALUMNO: _____
Apellido Paterno

_____ Apellido Paterno Nombre

N° DE BOLETA: _____ GRUPO: _____

ASIGNATURA: **Micro Electrónica Programable**

HOJA	DE	FECHA			EVALUACION
		DIA	MES	AÑO	

PROFESOR: _____

Práctica 5

Manejo de Motores

Competencias de La Unidad:

- Emplea el microcontrolador en la comunicación de datos y control de periféricos de forma multiplexada.

Resultado de Aprendizaje Propuesto (RAP):

- Usa diferentes elementos periféricos utilizando los puertos del microcontrolador de forma multiplexada.
- Utiliza los puertos del microcontrolador de forma bidireccional en la solución de un problema.

Objetivos de la Práctica:

1. Realizar la simulación de un programa para comprobar su funcionamiento utilizando herramientas computacionales
2. Realizar un control de actuadores utilizando diversas técnicas de programación.
3. Implementar técnicas programación con el fin de manipular distintos tipos de motores para aplicarlos en diversos ámbitos.
4. Implementar programas en un circuito basado en microcontrolador, para comprobar su funcionamiento

Equipo Necesario	Material Necesario
<p>Computadora (con el Software MPLAB IDE, IC-PROG o similar, compilador C, Simulador de circuitos electrónicos "Proteus")</p> <p>Programador tipo JDM o similar.</p>	<p>Instrucciones del PIC 16F887 u otro de gama media o alta</p> <p>Microcontrolador PIC16F887 u otro de gama media o alta</p> <p>Capacitores (C1=C2=39p)</p> <p>LED's</p> <p>Resistencias</p> <p>Push Button</p> <p>Motor de CD de 5V</p> <p>Circuito L293D o similar</p> <p>CD4093</p> <p>(Para los valores de estos elementos ver figuras)</p>

Introducción Teórica

Motor de corriente continúa

Un motor eléctrico de Corriente Continua es esencialmente una máquina que convierte energía eléctrica en movimiento o trabajo mecánico, a través de medios electromagnéticos (Ver Fig. 5a).

Motor a pasos

Los motores paso a paso pueden ser utilizando cuando se requieren mecanismos donde los movimientos deben ser muy precisos. La característica principal de estos motores es poder moverlos un paso a la vez por cada pulso que se le aplique. Este paso puede variar desde 90° hasta pequeños movimientos de tan solo 1.8°, es decir, que se necesitan 4 pasos en el primer caso (90°) y 200 para el segundo caso (1.8°), para completar un giro completo o de 360°. (Ver Fig. 5b).

Servomotor

Un Servo es un dispositivo pequeño que tiene un eje de rendimiento controlado y puede ser llevado a posiciones angulares específicas al enviar una señal codificada. Al presentarse la señal codificada en la línea de entrada, el servo mantendrá la posición angular. Este dispositivo consta de tres terminales dos para alimentación y una para la señal de control codificada. (Ver Fig. 5c).



Fig. 5a Motor de CD



Fig. 5b Motor a pasos



Fig. 5c Servomotor

ACTIVIDADES TEÓRICAS PREVIAS

Investigar los siguientes conceptos:

- **Puente H utilizando transistores bipolares**
- **Puente H utilizando transistores MOSFET**
- **Menciona como funciona el Circuito L293D**
- **Menciona como funciona el Circuito SN754410**
- **Configuración de transistor como interruptor.**
- **Menciona como funciona el Circuito ULN2803**
- **Modulación por ancho de pulso**
- **Ciclo de trabajo de una señal**
- **Características y tipos de motores a paso (Unipolar y bipolar)**
- **Características de los Servomotores**

ACTIVIDADES PREVIAS

- **Crear un proyecto de nombre pra5 en la carpeta c:\MEPIC\practica5 en MPLAB o PIC C Compiler. Los programas de cada ejercicio deben ser guardados con el nombre practica5X.c con X= 1, 2, 3...,A.**
- **En el caso de utilizar MPLAB, realizar los siguientes pasos:**
 - Utilizar Project wizard y seleccionar el compilador de c**
 - Agregar al proyecto los archivos adecuados con extensión c y h.**
 - Habilitar Simulador MPLAB SIM y modificar la frecuencia del simulador a 4 Mhz.**
 - Utilizaremos la herramienta de stopwatch, para obtener la elija Debugger >> Stopwatch.**
 - Obtener la herramienta de watch, de la siguiente manera View>> watch.**
 - Y seleccione los registros PORTA, PORTB, PORTC, PORTD, PORTE, TRISA, TRISB, TRISC, TRISD, TRISE y W**
- **Si usa PIC C compiler crear el proyecto únicamente e incluir un archivo nuevo.**

ACTIVIDADES PRÁCTICAS

Parte 1

A. Armar el siguiente circuito

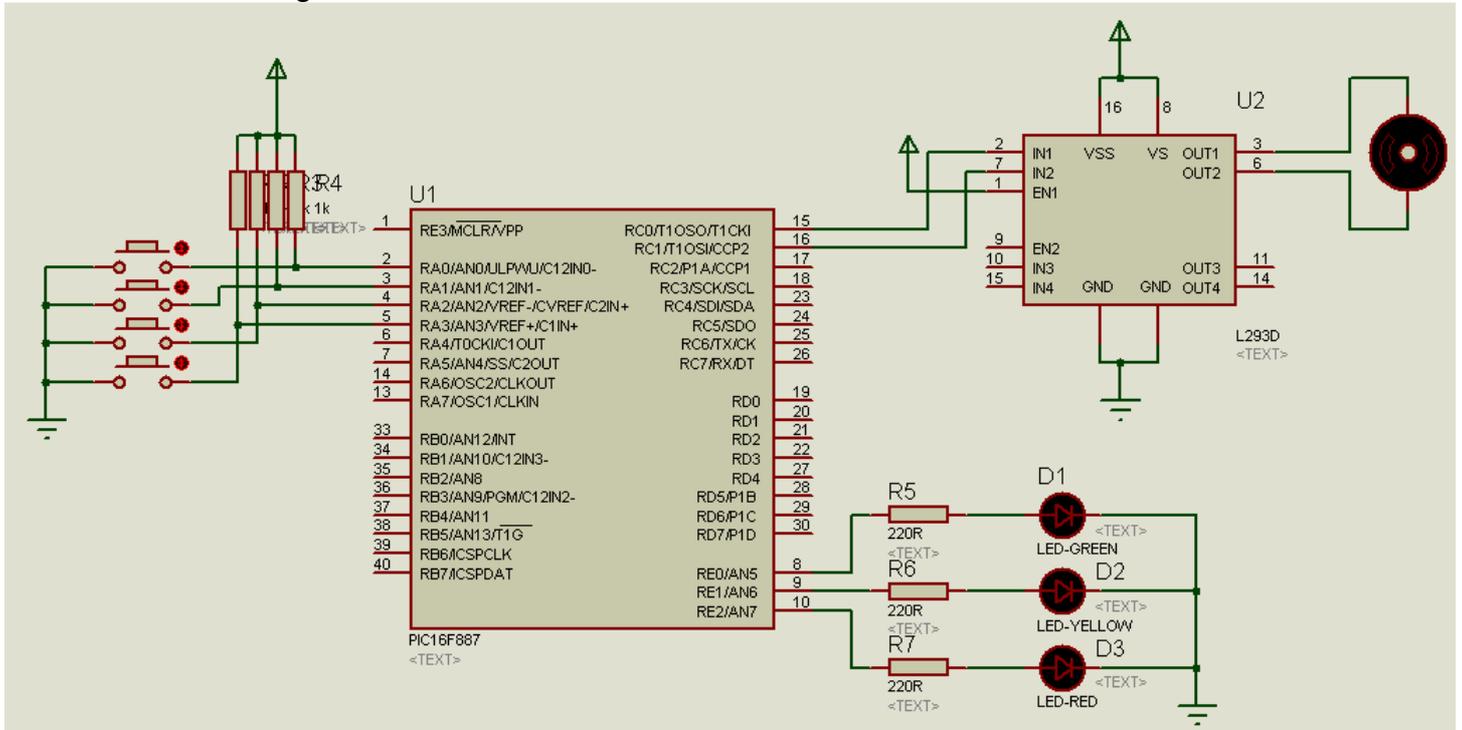


Figura 5.1

Nota: La terminal 12 o 31 del PIC16F887 se conectan a tierra. Es caso de implementarlo

A grabar el microcontrolador deshabilitar WDT y LVP y habilitar PWRTE y BODEN en la palabra de configuración, además recuerde seleccionar el tipo de oscilador a XT

Ejemplo 1

Realizar el siguiente programa y observar su funcionamiento utilizando la opción watch y stopwatch del simulador MPLAB (en caso de usarlo) y posteriormente grabarlo en el circuito de la figura 5.1, así como simularlo en PROTEUS

El siguiente código siguiente permite el manejo de un motor de CD a través de un interruptor conectado al pin RA0.

- **Cuando no es presionado el interruptor, el led amarillo debe de mantenerse encendido**
- **Cuando es presionado el interruptor debe encenderse el led verde y girar el motor durante 3 segundos a la derecha y al término debe de volver a encender el led Amarillo**

```
#include <16f887.h>
#use delay(clock=4000000)
#fuses XT,NOWDT,NOPUT,NOMCLR,NOPROTECT,NOCPD,NOBROWNOUT,NOIESO,NOFCMEN,NOLVP
#use standard_io(c)
#use standard_io(e)
```

```

void main()
{
output_e(2); // Enciende led amarillo
while(true)
{
output_c(0);
if(input(PIN_A0)==0)
{ delay_ms(100); // rebote de interruptor
output_c(1); // Enciende led verde y motor a la derecha
output_e(1);
delay_ms(3000);
output_e(2);
}
else
output_e(2); // Enciende led amarillo
}
}

```

Ejemplo 2

Realizar el siguiente programa, armar el circuito de la figura 5.2, grabarlo en el microcontrolador, así como simularlo en PROTEUS

El siguiente programa ejemplifica una forma de obtener una variación de velocidad de un motor utilizando la técnica PWM.

- **Cuando no es presionado ningún interruptor , el led amarillo debe de mantenerse encendido y el motor apagado**
- **Cuando es presionado el interruptor RA0 debe encenderse el led verde y girar el motor a la derecha con velocidad1 y se detiene al oprimir RA3**
- **Cuando es presionado el interruptor RA1 debe encenderse el led verde y girar el motor a la izquierda y se detiene al oprimir RA3**
- **Cuando es presionado el interruptor RA2 debe encenderse el led verde y girar el motor a la derecha con velocidad2 y se detiene al oprimir RA3**

//Código de para manejar un motor de CD utilizando PWM

```

#include <16f887.h>
#use delay(clock=4000000)
#fuses XT,NOWDT,NOPUT,NOMCLR,NOPROTECT,NOCPD,NOBROWNOUT,NOIESO,NOFCMEN,NOLVP
#use standard_io(c)
#use standard_io(e)
#use standard_io(a)

int i;
void main()

{
output_e(2); // Enciende led amarillo
output_c(1); // Enciende led amarillo y motor detenido
delay_ms(20);
output_c(0);
delay_ms(20);

```

```

while(true)
{
if(input(PIN_A0)==0)
{
delay_ms(10); // rebote de interruptor
while(input(PIN_A3)==1)
{
delay_ms(10);
for(i=0;i<=100;i++)
{
output_e(1); // Enciende led verde y motor a la derecha
output_c(1);
delay_ms(35);
output_c(0);
delay_ms(5);
}
}
}
else if(input(PIN_A1)==0)
{
delay_ms(10); // rebote de interruptor
while(input(PIN_A3)==1)
{ delay_ms(10);
for(i=0;i<=100;i++)
{
output_e(4); // Enciende led rojo y motor a la derecha
output_c(0);
delay_ms(10);
output_c(1);
delay_ms(30);
}
}
}
else if(input(PIN_A2)==0)
{ delay_ms(10); // rebote de interruptor
while(input(PIN_A3)==1)
{
delay_ms(10);

for(i=0;i<=100;i++)
{
output_e(1); // Enciende led verde y motor a la derecha
output_c(1);
delay_ms(28);
output_c(0);
delay_ms(12);
}
}
}
else
output_e(2); // Enciende led amarillo
output_c(1); // Enciende led amarillo y motor a la detenido
delay_ms(10);
output_c(0);
delay_ms(10);
}
}
}

```

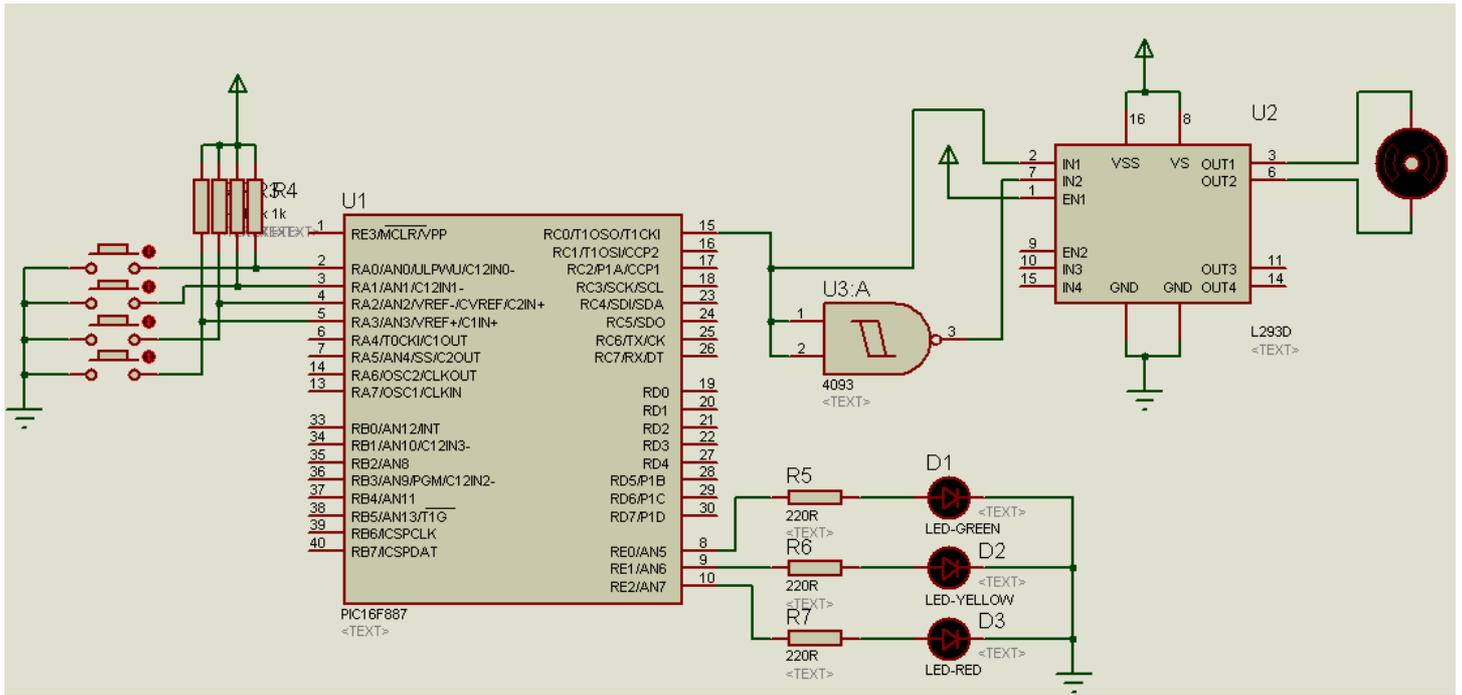


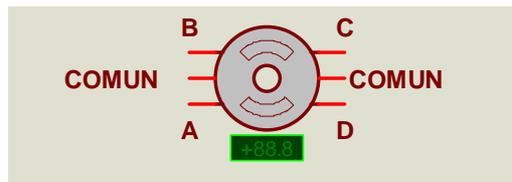
Figura 5.2

Nota: Tome en cuenta en los ejemplos 1 y 2 que la alimentación del motor de CD en manejador (Driver) L293D se determina en la terminal Vs (8) en algunos casos es diferente a la alimentación del circuito y depende del motor que se desee utilizar.

Ejemplo 3

Realizar el siguiente programa en el circuito de la figura 5.3, y simularlo en PROTEUS

La configuración de un motor a pasos en PROTEUS es:



El microcontrolador está configurado de la siguiente manera

A= RD4; B=RD5; C=RD6; D=RD7

Una opción para moverlo es la siguiente tabla de verdad

D	C	B	A	
0	0	0	1	PASO 1
0	0	1	0	PASO 2
0	1	0	0	PASO 3
1	0	0	0	PASO 4

El siguiente programa maneja un motor a pasos:

```
#include <16f887.h>
#use delay(clock=4000000)
#fuses XT,NOWDT,NOPUT,NOMCLR,NOPROTECT,NOCPD,NOBROWNOUT,NOIESO,NOFCMEN,NOLVP
#use standard_io(c)
#use standard_io(e)
#use standard_io(a)
int i;
void main()
{
output_e(2); // Enciende led amarillo
while(true)
{
output_e(2);
if(input(PIN_A0)==0)
{ delay_ms(20); // rebote de interruptor
for(i=0;i<=10;i++){
output_e(1);
output_c(0x01); // Enciende led verde y motor a la derecha
delay_ms(500);
output_c(0x02);
delay_ms(500);
output_c(0x04); // Enciende led verde y motor a la derecha
delay_ms(500);
output_c(0x08);
delay_ms(500);
}
output_e(2); //Enciende led amarillo
}
else
output_e(2); // Enciende led amarillo
}
}
```

¿Cuántos grados giro el motor?

(Nota: En este caso el simulador proporciona 8° por paso:)

Modifique el programa anterior con la siguiente secuencia

D	C	B	A	
0	0	1	1	PASO 1
0	1	1	0	PASO 2
1	1	0	0	PASO 3
1	0	0	1	PASO 4

¿Qué modificación se tendrían que realizar al programa para cambiar de sentido del motor?

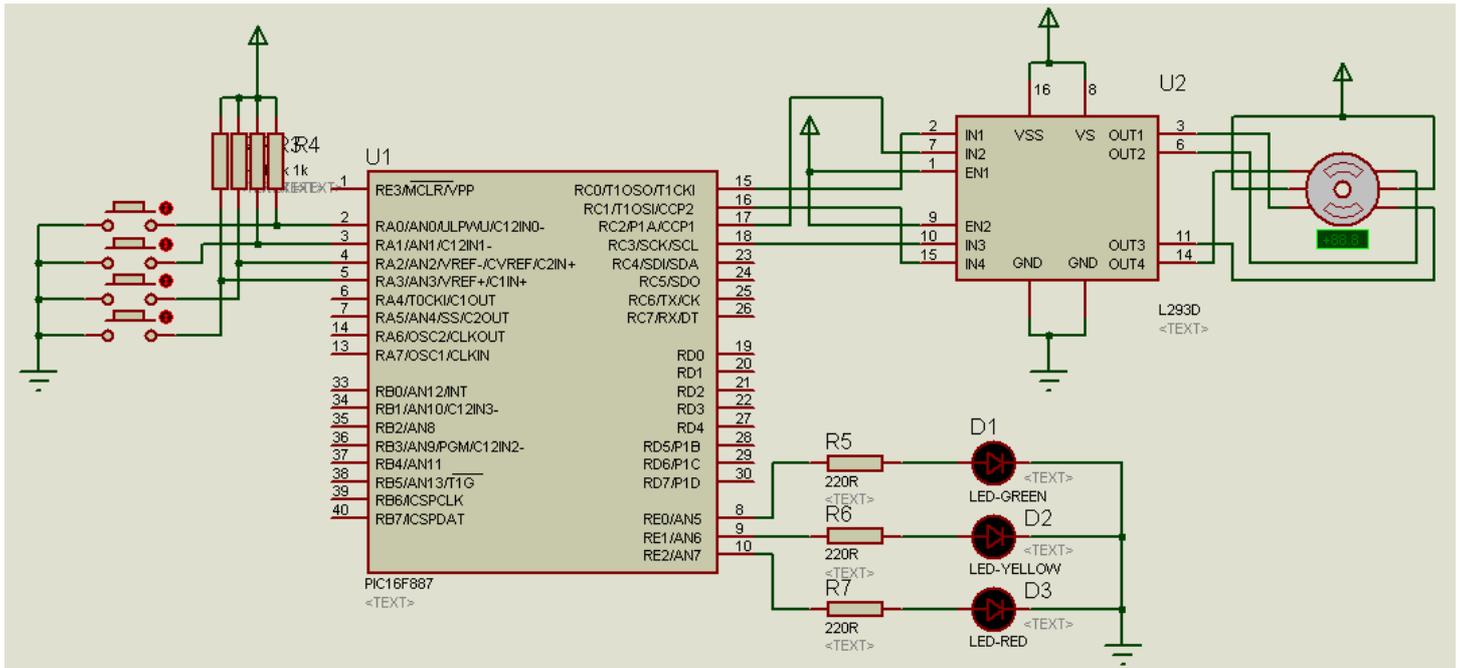


Figura 5.3

Ejemplo 4

Realizar el siguiente programa y simularlo en PROTEUS, utilizando el circuito de la figura 5.4.

```
#include <16f887.h>
#include <delay.h>
#define delay_time 4000000
#define XT oscillator_type
#define NOPDT
#define NOPUT
#define NOMCLR
#define NOPROTECT
#define NOCPD
#define NOBROWNOUT
#define NOIESO
#define NOFCMEN
#define NOLVP
#define standard_io_e
#define standard_io_c
#define standard_io_a
int i,j;
void main()
{
while(true)
{
output_e(7);
j=~input_A()&15;
delay_ms (2);
switch(j)
{
case 1: //giro izquierda 90 grados
for (i = 0; i < 50; i++)
{
output_e(1);
output_c(1);
delay_ms (1);
output_c(0);
delay_ms (19);}
break;}
}
```

```

case 2: //giro derecha 90 grados
{for (i = 0; i < 50; i++)
{
output_e(4);
output_c(1);
delay_ms (3);
output_c(0);
delay_ms (17);
}
break;}
case 4: // giro izquierda
{
for (i = 0; i < 50; i++)
{
output_e(2);
output_c(1);
delay_ms (1);
delay_us (500);
output_c(0);
delay_ms (18);
delay_us (500);}
break;}
case 8: //giro derecha 2
{
for (i = 0; i < 50; i++)
{
output_e(2);
output_c(1);
delay_ms (2);
delay_us(500);
output_c(0);
delay_ms (17);
delay_us(500); //Retardo en microsegundos
}
break;}
default: //detenido en el centro
{
output_b(7);
output_c(1);
delay_ms (2);
output_b(0);
delay_ms (18);
break;}
}
}
}

```

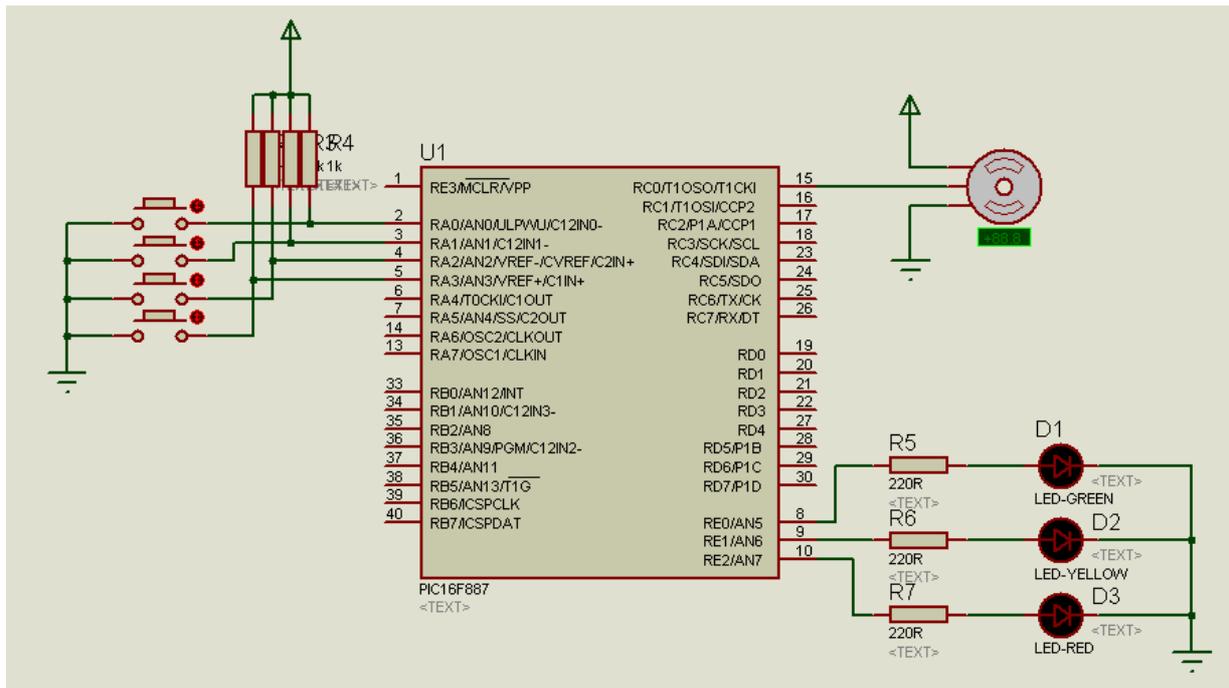


Figura 5.4

Parte 2

Realizar los siguientes programas y sus respectivas simulaciones, así como la implementación en el circuito, en caso de que el profesor lo indique.

A. Realice un programa de control de giro de un motor de CD, para efectuar las siguientes acciones:

- **3 led's conectados al puerto E y el motor conectado al puerto C, el motor debe conectar por medio del circuito manejador (Driver) correspondiente.**
- **Además contara con tres interruptores conectados al puerto A, que habilitaran las siguientes funciones:**
 - Si se activa PIN RA0,
 - Se enciende un led rojo y el motor gira hacia la derecha 5 segundos; para posteriormente detenerse y apagarse motor y led respectivamente; y encenderse un led amarillo parpadeando durante 2s y al terminar dejarlo encendido.
 - Si se activa PIN RA1,
 - Se enciende un led verde y el motor gira hacia la izquierda 8 segundos; para posteriormente detenerse y apagarse motor y led respectivamente; y encenderse un led amarillo parpadeando durante 2s y al terminar dejarlo encendido.
 - Si se activa PIN RA2,
 - Se enciende un led verde y el motor gira hacia la izquierda 10 segundos,
 - Después, se detiene por 5 segundos y enciende un led amarillo parpadeando.
 - Posteriormente un led rojo y el motor gira hacia la derecha 15 segundos; para posteriormente detenerse y apagarse motor y led respectivamente
 - y encenderse un led amarillo.

Si no se activa ningún PIN debe estar encendido un led amarillo que indica el motor estas detenido. Dicho led se apaga al activar alguna de las funciones.

Implemente un archivo c, MPLAB o PIC Compiler y posteriormente grabarlo en el circuito de la figura 5.1 y simularlo en PROTEUS, para observar su funcionamiento

B. Realice un programa de control de velocidad un motor de CD, para efectuar las siguientes acciones:

- **3 led's conectados al puerto E y el motor conectado al puerto C, el motor debe conectar por medio del circuito manejador (Driver) correspondiente.**
- **Considere que el periodo de la señal PWM es de 25hz**
- **Además contara con cuatro interruptores conectados al puerto A, que habilitaran las siguientes funciones:**
 - Si se activa PIN RA0,
 - Se enciende un led rojo y el motor gira al 75% de su velocidad hacia la derecha 5 segundos; para posteriormente detenerse y apagarse motor y led respectivamente; y encenderse un led amarillo parpadeando durante 2s y al terminar dejarlo encendido.
 - Si se activa PIN RA1,
 - Se enciende un led verde y el motor gira al 75% de su velocidad hacia la izquierda 8 segundos; para posteriormente detenerse y apagarse motor y led respectivamente; y encenderse un led amarillo parpadeando durante 2s y al terminar dejarlo encendido.
 - Si se activa PIN RA2,
 - Se enciende un led rojo y el motor gira al 35% de su velocidad hacia la derecha 3segundos; para posteriormente detenerse y apagarse motor y led respectivamente; y encenderse un led amarillo y verde parpadeando durante 3s y al terminar dejarlo encendido.
 - Si se activa PIN RA3,
 - Se enciende un led verde y el motor gira al 35% de su velocidad hacia la izquierda 4 segundos; para posteriormente detenerse y apagarse motor y led respectivamente; y encenderse un led amarillo y verde parpadeando durante 3s y al terminar dejarlo encendido.

Si no se activa ningún PIN debe estar encendido un led amarillo que indica el motor estas detenido. Dicho led se apaga al activar alguna de las funciones.

Implemente un archivo c, compílelo con ayuda de MPLAB o PIC Compiler y posteriormente simularlo en PROTEUS en el circuito de la figura 5.2 y, para observar su funcionamiento.

C. Realice un programa de control de giro de un motor de paso a paso unipolar, para efectuar las siguientes acciones:

- **3 led's conectados al puerto E y el motor conectado al puerto C, el motor debe conectar por medio del circuito manejador (Driver) correspondiente.**
- **Además contara con tres interruptores conectados al puerto A, que habilitaran las siguientes funciones:**
 - Si se activa PIN RA0,
 - Se enciende un led rojo y el motor gira hacia la derecha 5 segundos; para posteriormente detenerse y apagarse motor y led respectivamente; y encenderse un led amarillo parpadeando durante 2s y al terminar dejarlo encendido.
 - Si se activa PIN RA1,
 - Se enciende un led verde y el motor gira hacia la izquierda 4 segundos; para posteriormente detenerse y apagarse motor y led respectivamente; y encenderse un led amarillo parpadeando durante 3s y al terminar dejarlo encendido.
 - Si se activa PIN RA2,
 - Se enciende un led verde y el motor gira hacia la izquierda 8 segundos,
 - Después, se detiene por 5 segundos y enciende un led amarillo parpadeando.
 - Posteriormente un led rojo y el motor gira hacia la derecha 16 segundos; para posteriormente detenerse y apagarse motor y led respectivamente
 - y encenderse un led amarillo.

Si no se activa ningún PIN debe estar encendido un led amarillo que indica el motor estas detenido. Dicho led se apaga al activar alguna de las funciones.

Implemente un archivo c, compílelo con ayuda de MPLAB o PIC Compiler y posteriormente simular el circuito de la figura 5.3 en PROTEUS , para observar su funcionamiento

D. Realice un programa de control de un servomotor, para efectuar las siguientes acciones:

- **3 led's conectados al puerto E y el motor conectado al puerto C, el motor debe conectar como en la figura 5.4**
- **Además contara con tres interruptores conectados al puerto A, que habilitaran las siguientes funciones:**
 - Si se activa PIN RA0,
 - Se enciende un led rojo y el motor se posiciona en 60° y regresa al centro durante 5 veces y a terminar regresa al centro encendiendo un led amarillo parpadeando.
 - Si se activa PIN RA1,
 - Se enciende un led rojo y el motor se posiciona en 30° y regresa al centro durante 4 veces y a terminar regresa al centro encendiendo un led amarillo parpadeando.

- Si se activa PIN RA3,
 - Se enciende un led verde y el motor se posiciona en -30° y regresa al centro durante 4 veces y a terminar regresa al centro encendiendo un led amarillo parpadeando.
- Si se activa PIN RA4,
 - Se enciende un led verde y el motor se posiciona en -45° y regresa al centro durante 4 veces y a terminar regresa al centro encendiendo un led amarillo parpadeando.

Si no se activa ningún PIN debe estar encendido un led amarillo de de parpapadear indicando que el motor estas detenido y en la posición central. Dicho led se apaga al activar alguna de las funciones.

NOTA : Es obligatorio utilizar sentencias if-else

Implemente un archivo c, compílelo con ayuda de MPLAB o PIC Compiler y posteriormente simular el circuito de la figura 5.4 en PROTEUS , para observar su funcionamiento

1. Conclusiones

A. Realizar conclusiones de manera individual.

2. Cuestionario

- a) ¿Qué función realiza la compuerta inversora en el control de velocidad del motor con PWM de la figura 5.2?
- b) ¿Cuál es la frecuencia de la señal PWM implementada en el programa de ejemplo 2?
- c) ¿Cómo conectaría un motor a pasos unipolar transistores bipolares?
- d) ¿Cómo conectaría un motor a pasos bipolar a un L293D?
- e) ¿Que considera que sucede al modificar el tiempo de activación de cada bobina en un motor a pasos unipolar?
- f) ¿Cuál es la ventaja de utilizar servomotores?
- g) ¿Qué diferencia existe al usar sentencias if-else y switch-case?

Comentarios Finales

- El alumno entrega un reporte de la práctica, como el profesor lo indique.
- El reporte debe contener el diagrama de flujo o algoritmo (Seudo código) de cada uno de los programas.
- Además, en el reporte deben anexarse las conclusiones y cuestionario contestado.