

**CONTROL DE MOTORES****1. OBJETIVO**

- ✓ Conocer los motores de corriente continua y corriente alterna
- ✓ Comprender la Modulación por Ancho de Pulso PWM y su uso con la librería PWM Library
- ✓ Mostrar en simulación Virtual y también en Hardware su funcionamiento de los ejemplos propuestos.

2. EQUIPOS Y MATERIALES

- ✓ PC equipado con software MiKroC y Proteus
- ✓ Grabador Bootloader
- ✓ Fuente de alimentación variable de 0 a 24 VDC
- ✓ Guía de laboratorio y manual
- ✓ Motores DC y AC, Otros

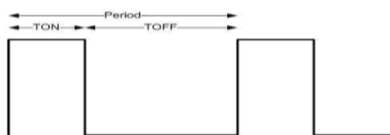
3. MARCO TEÓRICO**3.1. MOTOR DC DE CORRIENTE CONTINUA**

Un motor DC cepillado convencional consiste básicamente de dos partes, el cuerpo estacionario del motor llamado el estator y la parte interior que gira llamada rotor

La velocidad de rotación se determina por el voltaje de DC aplicado y su torque de salida está determinado por la corriente que fluye a través de los devanados del motor. La velocidad de rotación de cualquier motor de corriente continua se puede variar desde unas pocas revoluciones por minuto (rpm) a muchos miles de revoluciones por minuto. Mediante la conexión a cajas de engranajes la velocidad de salida puede ser disminuida y el torque incrementado.

MODULACIÓN POR ANCHO DE PULSO PWM

La modulación de ancho de pulso (PWM) es una técnica utilizada para controlar la energía suplida a dispositivos eléctricos. Esta energía se controla encendiendo y apagando la alimentación de voltaje a una alta frecuencia. PWM se puede utilizar para variar la velocidad de un motor, la brillantez de una lámpara o LED, etc. Elementos básicos de una señal PWM:

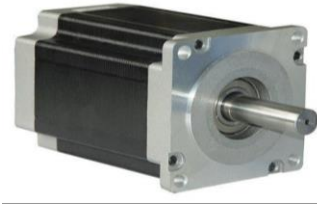


- $\text{Period}_{\text{PWM}} = \text{TON} + \text{TOFF}$
- $\text{Frecuencia}_{\text{PWM}} = 1 / \text{Period}_{\text{PWM}}$
- $\text{Duty cycle (Ciclo de trabajo)} = \text{TON} / \text{Period}_{\text{PWM}} * 100$

3.2. MOTOR PASO A PASO

Los Motores Paso a Paso (Step Motor) se mueven en un determinado ángulo de recorrido (STEP) con cada secuencia de pulso. Por ejemplo, un motor de 1.8 grados de paso requiere de 200 Steps para completar una vuelta completa ($360^\circ/1.8^\circ=200$). Parámetros de un Motor Paso:

- Unipolar (5 o 6 hilos) o Bipolar (4 hilos)
- Paso (Grados de recorrido) (1o, 1.8 o, 4 o, 14 o, etc.)
- Voltaje de Alimentación (+5, +12, +24 volt, etc.)
- Máxima Corriente (0.5 A, 1A, 2A, 4A, etc.)
- Torque (Fuerza del motor) Depende del Modo de Paso
- Velocidad (Depende de la frecuencia de la secuencia de control)

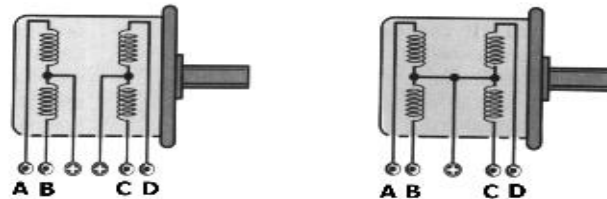


DRIVER DE CORRIENTE:

- Debido a la baja corriente (máximo de 20 mA) que proveen los pines del microcontrolador, para mover el rotor se requiere un driver de corriente.
- Los drivers pueden ser circuitos integrados como el ULN2003, cuatro transistores Darlington o cuatro transistores de potencia como el 2N3055.
- Existen circuitos más elaborados (complejos y costosos) que permiten manejar mayores voltajes y corrientes.

CONEXIONES DE LAS BOBINAS EN UN MOTOR UNIPOLAR (Coils):

Los motores unipolares tienen 5 o 6 hilos dependiendo del modelo. Si el motor tiene 6 hilos, se deben unir los puntos medios de las bobinas y conectarlas al voltaje de alimentación del motor. Los extremos de cada bobina deben ser conectados al driver de corriente como se muestra en el diagrama esquemático.



MODO DE PASO

Existen varios modos de paso que se pueden utilizar para mover el motor:

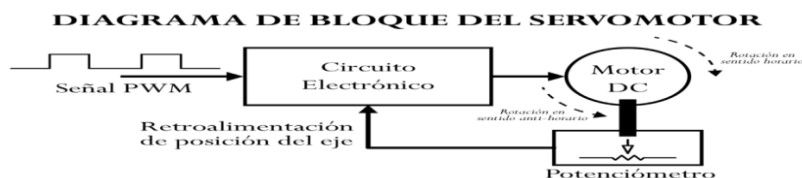
A. PASO SIMPLE (Single Stepping) El modo más simple que activa una bobina a la vez.

- Se debe repetir la siguiente secuencia de 4 pasos.
- Esta secuencia avanza al motor en dirección Horaria (Clockwise) (CW), para girar en dirección anti-horaria (Counter Clockwise) (CCW) la secuencia es inversa

Paso	Coil 1C	Coil 2C	Coil 3C	Coil 4C
1	1			
2		1		
3			1	
4				1

3.3. SERVO MOTOR





- Un servo motor consiste en un motor DC, una caja de cambios de reducción, un dispositivo de retroalimentación de posición y alguna forma de corrección de errores.
- La velocidad o posición es controlada en relación a una señal de entrada o señal de referencia de posición aplicada al dispositivo.

3.4. MOTOR INDUSTRIAL



Los motores eléctricos trifásicos, se fabrican en las más diversas potencias, desde una fracción de caballo hasta varios miles de caballos de fuerza (HP), se los construye para prácticamente, todas las tensiones y frecuencias (50 y 60 Hz) normalizadas y muy a menudo, están equipados para trabajar a dos tensiones nominales distintas. Se emplean para accionar máquinas industriales, herramienta, bombas, montacargas, ventiladores, grúas, maquinaria elevada, sopladores, etc.

3.5. LIBRERÍA PWM LIBRARY DE MIKROC

El microcontrolador PIC18F4550 posee dos salidas PWM1 y PWM2 en los pines del puerto PORTC: RC.2 (CCP1) y RC.1 (CCP2)

- PWM1_Init Inicializa la modulación PWM con la frecuencia deseada. El valor mínimo de la frecuencia cuando usamos un oscilador externo de 4 Mhz es de 245 Hz.
- PWM1_Set_Duty Establece el Ciclo de trabajo. Se mide en términos de porcentaje sobre una escala que varía entre 0 y 255, donde 255 equivale al 100% del ciclo de trabajo
- PWM1_Start Inicia PWM
- PWM1_Stop Detiene PWM

4. PRE INFORME

4.1. Objetivos de laboratorio.

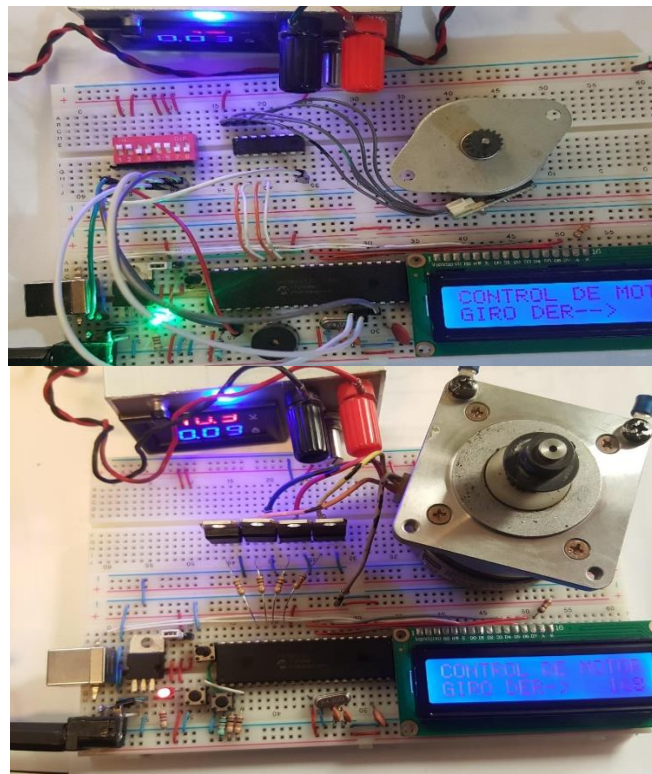
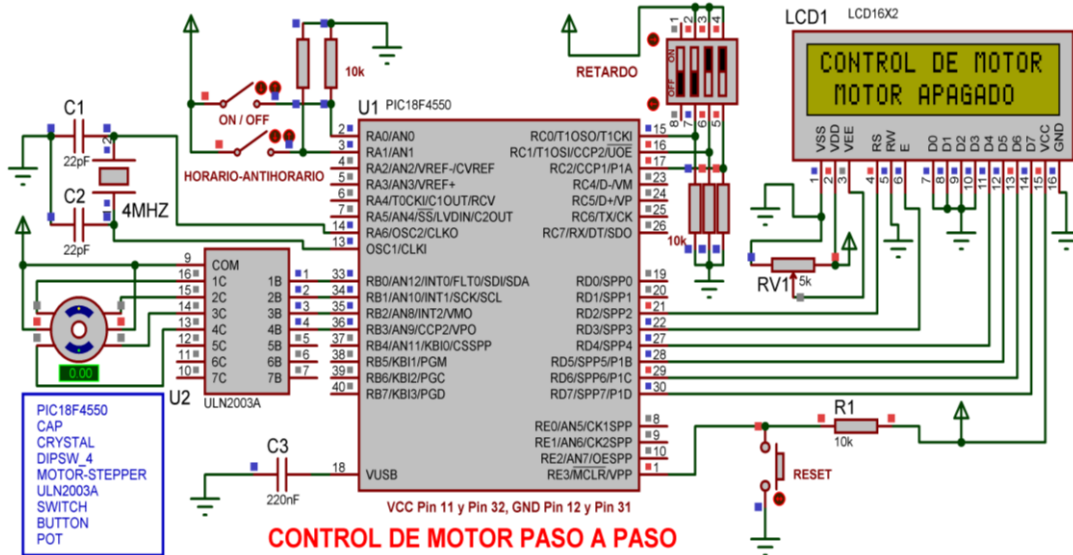
4.2. Características de circuito integrado ULN2003A, L293D y circuito para mayor potencia de los motores

4.3. Características de motor corriente continua y servo motor

5. PROCEDIMIENTO

5.1. Probar el funcionamiento de Control de Motor Paso Unipolar. Demostrar su funcionamiento en simulación virtual y en hardware.

- Mediante el interruptor Activar, se activa (Activar = 0) o desactiva (Activar = 1) el motor paso
- Mediante el interruptor Dirección, el motor gira en sentido horario (Dirección = 1) o en sentido antihorario (Dirección = 0)
- Mediante los 3 interruptores Retardo, se controla la velocidad del motor desde 3 a 250 milisegundos por secuencia de paso
- **Complemente para mostrar en la pantalla LCD las vueltas del motor paso a paso.**



//PROYECTO: CONTROL DE MOTOR PASO A PASO - LCD

1. #define Activar PORTA.F0
2. #define Direccion PORTA.F1

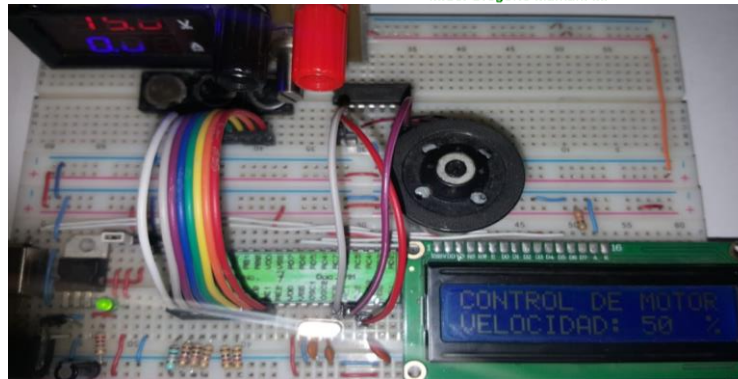
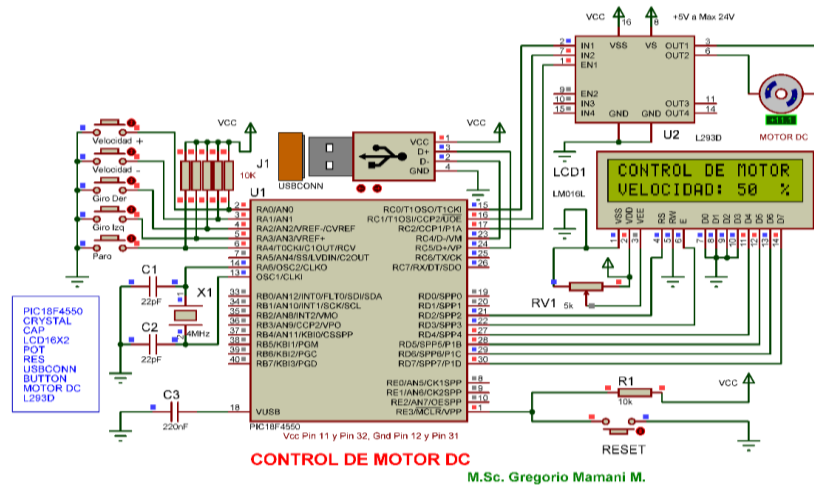
```

3.  #define PMotor LATB
4.  #define PRetardo PORTC
5.  sbit LCD_RS at RD2_bit;
6.  sbit LCD_EN at RD3_bit;
7.  sbit LCD_D4 at RD4_bit;
8.  sbit LCD_D5 at RD5_bit;
9.  sbit LCD_D6 at RD6_bit;
10. sbit LCD_D7 at RD7_bit;
11. sbit LCD_RS_Direction at TRISD2_bit;
12. sbit LCD_EN_Direction at TRISD3_bit;
13. sbit LCD_D4_Direction at TRISD4_bit;
14. sbit LCD_D5_Direction at TRISD5_bit;
15. sbit LCD_D6_Direction at TRISD6_bit;
16. sbit LCD_D7_Direction at TRISD7_bit;
17. unsigned short Retardo;
18. void PasoHorario() {
19.   PMotor=0x01;
20.   Vdelay_ms(Retardo);
21.   PMotor=0x02;
22.   Vdelay_ms(Retardo);
23.   PMotor=0x04;
24.   Vdelay_ms(Retardo);
25.   PMotor=0x08;
26.   Vdelay_ms(Retardo);
27. }
28. void PasoAntiHorario() {
29.   PMotor=0x08;
30.   Vdelay_ms(Retardo);
31.   PMotor=0x04;
32.   Vdelay_ms(Retardo);
33.   PMotor=0x02;
34.   Vdelay_ms(Retardo);
35.   PMotor=0x01;
36.   Vdelay_ms(Retardo);
37. }
38. unsigned short CalcularRetardo() {
39.   unsigned short Valor;
40.   valor=PRetardo & 0x07;
41.   switch (Valor) {
42.     case 0: return 4; break;
43.     case 1: return 8; break;
44.     case 2: return 12; break;
45.     case 3: return 22; break;
46.     case 4: return 42; break;
47.     case 5: return 80; break;
48.     case 6: return 100; break;
49.     case 7: return 150; break;
50.   }
51. }
52. void main() {
53.   ADCON1=15;
54.   CMCON=7;
55.   INTCON=0;
56.   PMotor=0;
57.   TRISB=0b11110000;
58.   TRISC=0b11111111;
59.   TRISA.F0=1;
60.   TRISA.F1=1;
61.   Lcd_Init();
62.   Lcd_Cmd(_LCD_CURSOR_OFF);
63.   Lcd_Out(1,1,"CONTROL DE MOTOR");
64.   Lcd_Out(2,1," PASO A PASO ");
65.   delay_ms(3000);
66.   Ciclo:
67.   Retardo=CalcularRetardo();
68.   if (Activar){
69.     Lcd_Out(2,1,"          ");
70.     if (Direccion){
71.       Lcd_Out(2,1,"GIRO DER-->  ");
72.       PasoHorario();
73.     }
74.     else{
75.       Lcd_Out(2,1,"GIRO IZQ<--  ");
76.       PasoAntiHorario();
77.     }
78.   }else{
79.     Lcd_Out(2,1," MOTOR APAGADO ");
80.   }
81.   goto Ciclo;
82. }

```

5.2. Diseño de programa en MikroC, visualizar los controles en display LCD de CONTROL DE MOTOR DC CON PWM. Demostrar su funcionamiento en simulación virtual y en hardware.

- Complemente para mostrar inversión de giro y además usar librerías PWM1_Start(), PWM1_Stop() en proceso de marcha y paro.



// CONTROL DE MOTOR DC CON PWM

```

1.  sbit LCD_RS at RD2_bit; //LCD module connections
2.  sbit LCD_EN at RD3_bit;
3.  sbit LCD_D4 at RD4_bit;
4.  sbit LCD_D5 at RD5_bit;
5.  sbit LCD_D6 at RD6_bit;
6.  sbit LCD_D7 at RD7_bit;
7.  sbit LCD_RS_Direction at TRISD2_bit;
8.  sbit LCD_EN_Direction at TRISD3_bit;
9.  sbit LCD_D4_Direction at TRISD4_bit;
10. sbit LCD_D5_Direction at TRISD5_bit;
11. sbit LCD_D6_Direction at TRISD6_bit;
12. sbit LCD_D7_Direction at TRISD7_bit;
13. int velocity = 0;
14. int duty = 0;
15. char txt[6] = "0";
16. void main() {
17.     ADCON1 |= 0x0F;
18.     CMCON |= 7;
19.     PORTC = 0; // sets port c to all 0
20.     TRISC = 0; //configures port c as output port
21.     PWM1_Init(5000); // sets pwm frequency to 5000Hz
22.     PWM1_Start(); // starts pwm1 peripheral // enviar a la etapa donde Incremento de PWM, decremento de PWM //
23.     Lcd_Init(); // Inicializa LCD
24.     Lcd_Cmd(_Lcd_CLEAR); // Limpiar display
25.     Lcd_Cmd(_Lcd_CURSOR_OFF); // Desactivar cursor
26.     LATCF1=1;
27.     while(1)
28.     {
29.         wordToStr(velocity, txt);
30.         Lcd_Out(1,1,"CONTROL DE MOTOR"); // Velocidad
31.         Lcd_Out(2,1,"VELOCIDAD:");
32.         Lcd_Out(2,12,Ltrim(txt));
33.         Lcd_Out(2,16,"%");
34.         if (PORTA.F0 == 0){ // Incremento de PWM
35.             Lcd_Out(2,12," ");

```

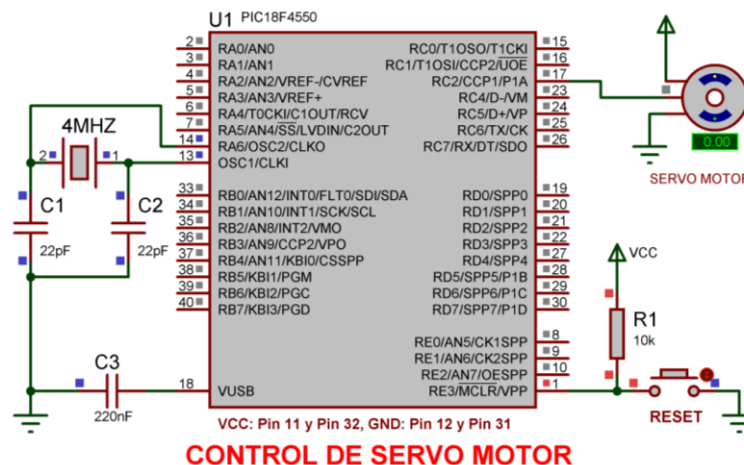


```

36. velocity = velocity + 10;
37. duty = duty + 24;
38. if (duty >= 255) duty = 255;
39. if (velocity >= 100) velocity = 100;
40. Delay_ms(15); // Tiempo 15mseg
41. }
42. if (PORTA.F1 == 0){ // Decremento de PWM
43. Lcd_Out(2,12," ");
44. velocity = velocity - 10;
45. duty = duty - 24;
46. if (duty <= 0) duty = 0;
47. if (velocity <= 0) velocity = 0;
48. Delay_ms(15); // Tiempo 15mseg
49. }
50. if (PORTA.F4 == 0){ // Detener
51. duty = 0;
52. velocity = 0;
53. Lcd_Out(2,12," ");
54. /////// habilitar// PWM1_Stop();
55. }
56. PWM1_Set_Duty(duty); // Establece ciclo de trabajo
57. Delay_ms(15); // Tiempo 15mseg
58. }
59. }

```

5.3. Probar el funcionamiento de Control de Servo Motor y mostrar en la pantalla LCD



//CONTROL DE SERVO MOTOR

```

unsigned int ang=0,i;
void retardo (unsigned int cuenta){
    int j=0;
    Delay_us(1000);
    for(J=0;j<cuenta;j++){
        Delay_us(6);
    }
}
void main(){
    TRISC.F2=0;
    while(1){
        for (ang=0;ang<180;ang+=10){
            for(i=0;i<50; i++){
                LATC.F2=1;
                retardo(ang);
                LATC.F2=0;
                VDelay_ms(18);
            }
        }
    }
}

```

6. CUESTIONARIO

- 6.1. Driver L293B: Permite controlar el sentido de giro y velocidad para 2 motores DC. Investigar el funcionamiento (mostrando el diagrama esquemático) para controladores de motor DC
- 6.2. Controladores de Motor Paso a Paso El chip ULN2003A es apropiado para manejar motores de baja potencia. Investigar y explicar el funcionamiento (mostrando el diagrama esquemático) para controladores de motor paso de mayor potencia.

7. CONCLUSIONES