

## Laboratorio 5: Control de Motores DC y Servomotores

### 1. Datos de la Práctica

Carrera	INGENIERÍA ELECTRÓNICA		
Semestre		Grupo	
Tipo de Práctica	<input type="checkbox"/> Laboratorio <input type="checkbox"/> Simulación	Fecha	
Asignatura	Electrónica Digital II		
Unidad Temática			
Nº Alumnos por práctica	2	Nº Alumnos por reporte	2
Nombre del Profesor			
Nombre(s) de Alumno(s)	1. 2.		
Tiempo estimado		Vo. Bo. Profesor	
Comentarios			

### 2. Objetivo

- Conocer el funcionamiento de los Motores DC, Servo Motores y Motores Paso a Paso.
- Familiarizarse con el módulo driver L298N para el control de Motores DC y Servo Motores.
- Controlar Motores DC y Servo Motores mediante la implementación de PWM en Arduino.

### 3. Componentes a Utilizar

Por cada práctica y por cada puesto de laboratorio, los materiales a utilizar son:

Cantidad	Descripción
1	Computadora
1	Arduino MEGA
2	Motores DC
1	MicroServoMotor
1	Módulo L298N

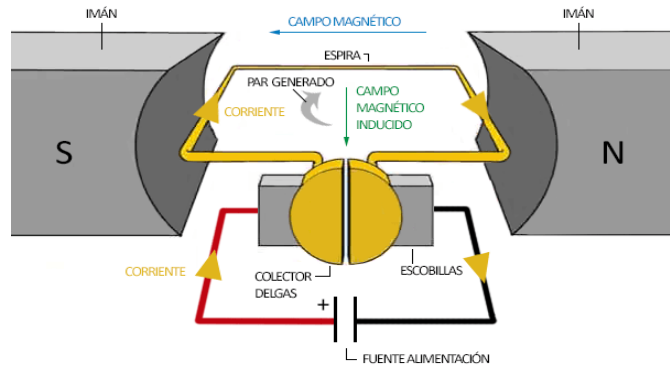
### 4. Introducción

Un motor es la parte sistemática de una máquina capaz de hacer funcionar el sistema, transformando algún tipo de energía (eléctrica, de combustibles fósiles, etc.), en energía mecánica capaz de realizar un trabajo. Existen diversos tipos, siendo los más comunes: los Motores térmicos (el trabajo se obtiene a partir de energía calórica) y los Motores eléctricos (el trabajo se obtiene a partir de una corriente eléctrica). Los motores eléctricos utilizan la inducción electromagnética que produce la electricidad para producir movimiento, según sea la constitución del motor.

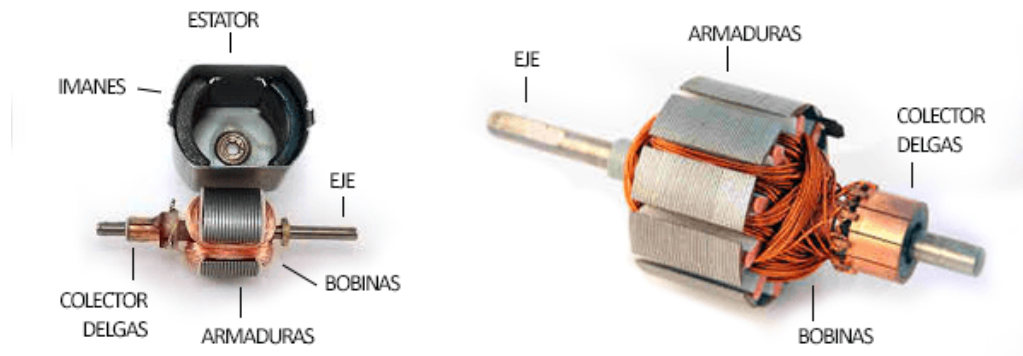
## Tipos de Motores

### Motores de Corriente Continua

Los motores de corriente continua (DC) son unos de los actuadores más comunes. Su funcionamiento se basa en el alineamiento de dos campos magnéticos (Ver **Fig. 1**). El estator, la parte fija del motor, dispone de un imán permanente que genera un campo magnético en el interior del motor a como se muestra en la **Fig. 2**.



**Fig. 1 Alineación de Campo Magnético de un Motor DC.**



**Fig. 2 Partes Internas de un Motor DC.**

En su interior introducimos una espira y hacemos circular una corriente eléctrica, con lo que se genera un campo magnético. El desfase angular entre ambos campos magnéticos genera un par de giro, que hace que el rotor gire hasta que los dos campos magnéticos se alineen. Al pasar un cierto ángulo las escobillas pasaran de una delga a la siguiente. Esto provoca la inversión de la corriente en la espira. De esta forma el conjunto escobillas – colector de delgas actúa como un inversor mecánico y permite al motor girar de forma continua.

La ventaja de este sistema es que la sincronización es siempre perfecta independientemente de la velocidad y par ejercidos, ya que es el propio ángulo de giro del motor el que marca la inversión de la corriente. El lado negativo es que el rozamiento supone una pérdida de eficiencia y reducen la vida útil del motor. Los motores de corriente continua se encuentran disponibles en distintas tensiones nominales, siendo habituales 6V, 12V y 24V. En cuanto a potencia, encontramos motores de todo tipo de tamaños.

### Servo Motores

Los servos son otro actuador muy común en proyectos de robótica (Ver **Fig. 3**). Un servo recibe una señal pulsada generada por un procesador, que transmite la posición que deseamos y el servo autónomamente se posiciona en esa posición, no puede dar una vuelta completa, siendo su rango habitual de 180°. A cambio, proporcionan un control total (fácil de usar) en posición y giro y de una alta precisión.



Fig. 3 Servo Motores

Internamente un servo está constituido por un motor DC acoplado a un reductor, junto con un controlador que se encarga de posicionar el eje en el ángulo indicado (Ver **Fig. 4**). Al estar acoplados a un reductor la velocidad de un servo es relativamente baja y proporcionan un alto par. Son ampliamente utilizados en proyectos de robótica, como brazos robóticos, hexápodos, o robots bípedos.



Fig. 4 Partes Internas de un Servo Motor.

## Motores Paso a Paso

Los motores paso a paso (también llamados stepper) son otro tipo de motor muy empleado en robótica (Ver **Fig. 5**). En este tipo de motores el eje gira un ángulo fijo llamado «paso» cuando es indicado por un procesador. El paso varía del modelo de motor, siendo valores habituales 1.  $8^\circ$  (200 pasos por vuelta) y 3.  $75^\circ$  (96 pasos por vuelta). A rasgos generales, un motor paso a paso está formado por un estator con dos bobinas desfasadas a  $90^\circ$  y un rotor formado por un imán permanente instalado solidariamente al eje.

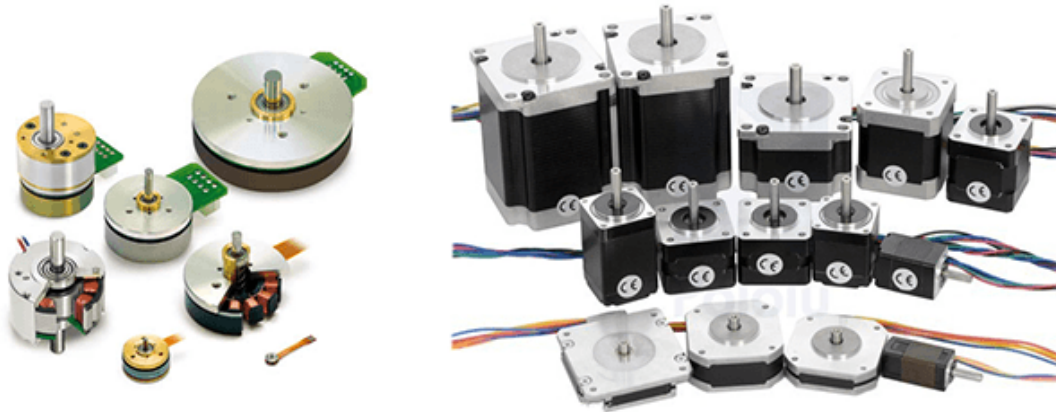


Fig. 5 Motores Paso a Paso.

Aplicando una secuencia de encendido adecuada a las bobinas podemos hacer que el imán se oriente progresivamente, hasta dar un giro completo (Ver **Fig. 6**). Si la secuencia es incorrecta el motor no se moverá. Por este motivo, un motor paso a paso necesita obligatoriamente un procesador para funcionar no siendo posible activarlos simplemente conectándolos a corriente.

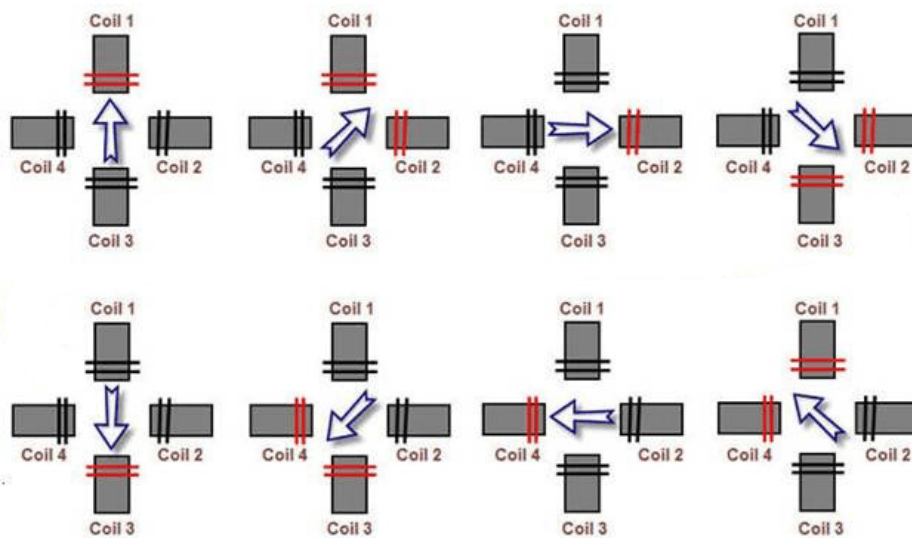


Fig. 6 Secuencia de Activación de Bobinas del Motor.

En el mundo real, un motor que sólo es capaz a pasos de  $90^\circ$  no sería muy útil. Para conseguir un paso inferior podríamos pensar en añadir múltiples bobinas, pero añadiríamos peso sin incrementar la potencia. Para conseguir el efecto de tener «múltiples bobinas» se emplea la variación de la reluctancia magnética. Se dispone de una rueda dentada acoplada al eje (Ver **Fig. 6**), con un número de dientes tal que el siguiente diente esté desfasado respecto a las restantes bobinas. Al activar una bobina, la rueda dentada es atraída por el campo magnético generado de forma que el rotor gira para minimizar la distancia del circuito magnético.

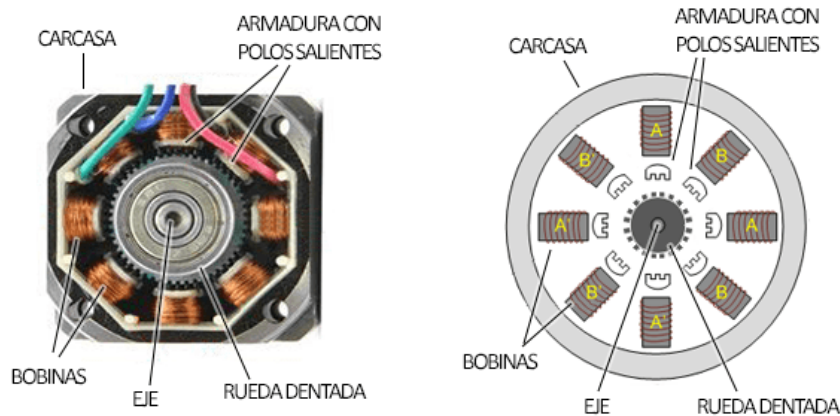


Fig. 7 Partes Internas de un Motor Paso a Paso.

En la imagen puede parecer que hay «8 bobinas» pero en realidad solo hay 2 bobinas A y B, cada una con 4 polos (Ver Fig. 8). Al activar una bobina todos los polos funcionan de forma conjunta. Existen motores unipolares o bipolares. La diferencia es que los motores unipolares tienen ambas bobinas divididas en dos, para lo cual disponen de un terminal adicional en cada fase.

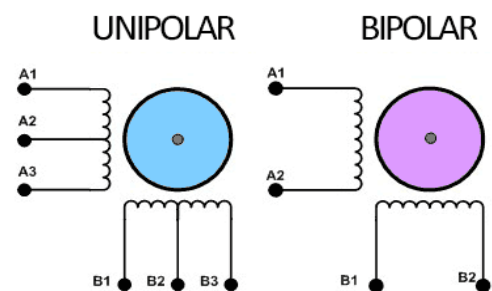


Fig. 8 Tipos de Motores Paso a Paso.

## Conexión de los Motores

Algunos motores DC y Servo Motores pequeños de bajo consumo pueden conectarse directamente al Arduino, pero cuando se necesita controlar más de uno ya es necesario hacer uso de un driver para motor como lo es el L293D o el L298N que pueden controlar motores de mayor consumo e incluso hacer inversión de giro. En esta práctica de laboratorio se hará uso del driver L298N que puede controlar: 4 motores DC para un solo giro, 2 Motores DC con inversión de giro, un motor Paso a Paso y un Servo Motor.

El L298N es un driver de motor con doble canal de Puente H capaz de manejar dos motores DC con inversión de giro y un Servo Motor, capaz de alimentarlos con un voltaje de **3V a 46V** y una corriente de hasta **2A**. El diagrama de pines se muestra en la Fig. 9 y la función de cada uno de ellos se muestra en la **Tabla #1**.

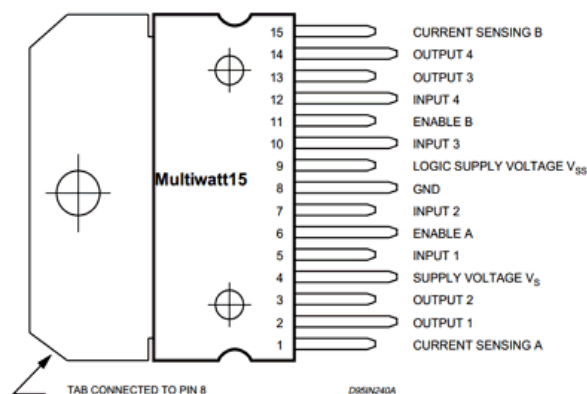


Fig. 9 Diagrama de Pines L298N

Pines		Descripción
Nombre	Numero	
<b>ENABLE</b>	6, 11	Habilitación Canal 1 y 2 activos en Alto
<b>INPUT</b>	5, 7, 10, 12	Entradas del Driver
<b>OUTPUT</b>	2, 3, 13, 14	Salidas del driver
<b>3, 4 EN</b>	9	Habilitación Canal 3 y 4 activos en Alto
<b>GROUND</b>	8	Conexión a tierra y disipador
<b>Vss</b>	9	Alimentación de 5V para IC
<b>Vs</b>	4	Alimentación Motores de 3V hasta 46V
<b>SENSE A, B</b>	1, 15	Detección de Corriente para la carga

Tabla 1 Funciones de los Pines

Para el control de estos motores se habilitan o deshabilitan los pines que controlan los motores siguiendo una secuencia. Esta secuencia se establece en la **Tabla #2**.

IN1	IN2	IN3	IN4	Dirección de los Motores
LOW	LOW	LOW	LOW	Motores OFF
HIGH	LOW	HIGH	LOW	Adelante
LOW	HIGH	LOW	HIGH	Retroceso
HIGH	HIGH	HIGH	HIGH	Motores OFF

Tabla 2 Modos de Activación de Motores.



## 5. Prácticas de Laboratorio

### Practica #1 Controlando un Motor DC con el Driver L298N

#### Código Arduino

El siguiente sketch nos ayudará a comprender como controlar la velocidad y la dirección de un Motor DC usando el driver L298N. Este código puede servir para el control de las llantas de un carrito de juguete, haciendo uso de funciones como **moveForward()**, **moveBackward()** y **fullStop()**.

```
int pinIN1 = 17, pinIN2 = 18, pinIN3 = 19, pinIN4 = 20;
int pinENA = 11, pinENB = 12;
const int waitTime = 2000; //espera entre fases
int sensorPin = A0; //Pin del Potenciometro
int sensorValue, outputValue; //Valor del Potenciometro
const int pinMotorA[3] = { pinENA, pinIN1, pinIN2 };
const int pinMotorB[3] = { pinENB, pinIN3, pinIN4 };

void setup()
{
  pinMode(pinIN1, OUTPUT);
  pinMode(pinIN2, OUTPUT);
  pinMode(pinENA, OUTPUT);
  pinMode(pinIN3, OUTPUT);
  pinMode(pinIN4, OUTPUT);
  pinMode(pinENB, OUTPUT);
}

void loop()
{
  sensorValue = analogRead(sensorPin);
  outputValue = map(sensorValue, 0, 1023, 0, 255);
  moveForward(pinMotorA, outputValue);
  moveForward(pinMotorB, outputValue);
  delay(waitTime);

  moveBackward(pinMotorA, outputValue);
  moveBackward(pinMotorB, outputValue);
  delay(waitTime);

  fullStop(pinMotorA);
  fullStop(pinMotorB);
  delay(waitTime);
}
```

```
void moveForward(const int pinMotor[3], int speed)
{
  digitalWrite(pinMotor[1], HIGH);
  digitalWrite(pinMotor[2], LOW);
  analogWrite(pinMotor[0], speed);
}

void moveBackward(const int pinMotor[3], int speed)
{
  digitalWrite(pinMotor[1], LOW);
  digitalWrite(pinMotor[2], HIGH);
  analogWrite(pinMotor[0], speed);
}

void fullStop(const int pinMotor[3])
{
  digitalWrite(pinMotor[1], LOW);
  digitalWrite(pinMotor[2], LOW);
  analogWrite(pinMotor[0], 0);
}
```

## Simulación en Proteus

1. Para la simulación de esta práctica se hará uso de varios componentes electrónicos que se detallaran en la siguiente tabla para su respectiva búsqueda en el Proteus.

Nombre de Componente	Código o Nombre del en Proteus	Características	Cantidades
<b>Arduino</b>	Arduino MEGA 2560	Librería de Arduino	1
<b>DUAL FULL-BRIDGE DRIVER</b>	L298	Driver para Motores	1
<b>Potenciómetro</b>	POT-HG	1K	1
<b>Motor DC</b>	MOTOR	Animado	2
<b>Osciloscopio</b>	Osciloscopio	4 Canales	1

2. Monte los componentes del circuito según la **Fig. 10**.



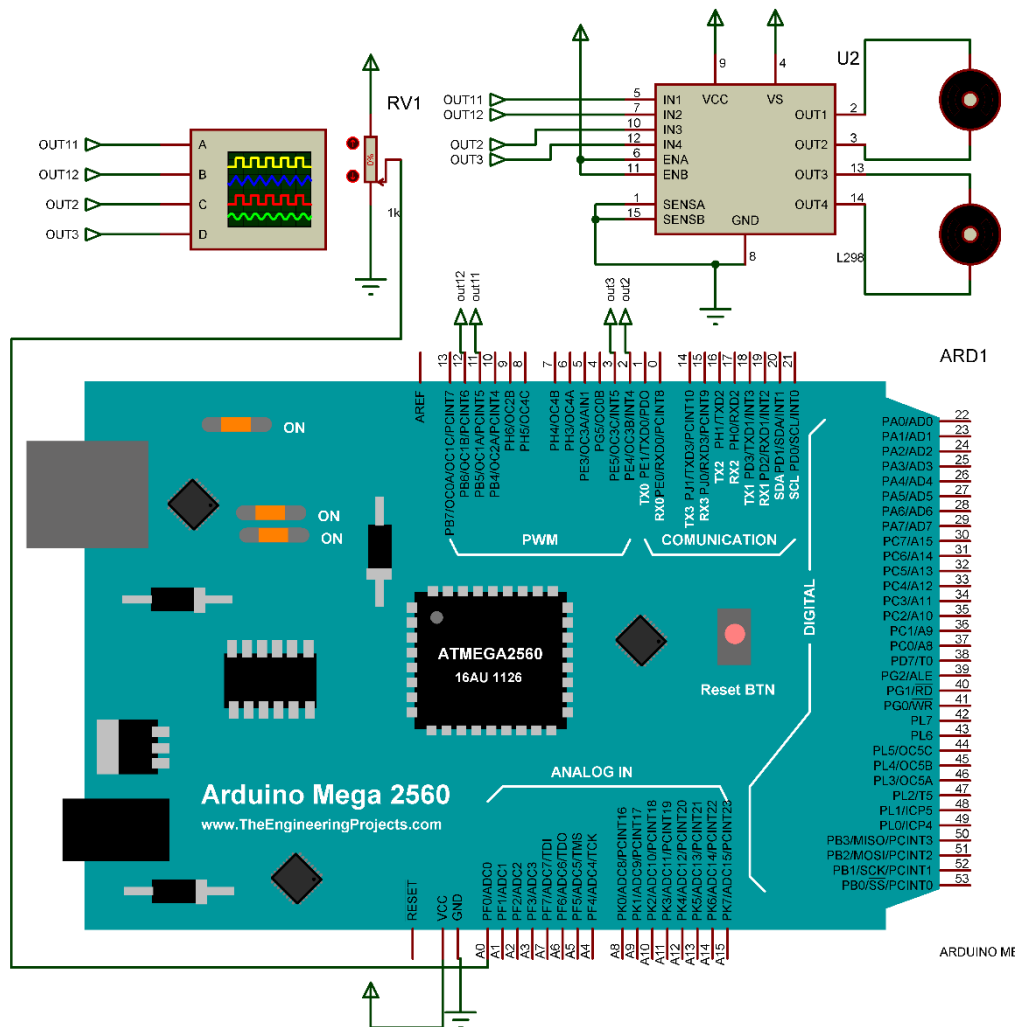


Fig. 10 Simulación de Control de Motores DC.

- Observe la Simulación, cambie los valores de los **logicstate** según la **Tabla #2** y vea el comportamiento de los motores. Según sus cambios, habrá 3 estados distintos: un estado de paro, un estado de mover los motores hacia adelante y un momento de los motores en retroceso.
- Por cada cambio de estado observe en el osciloscopio las señales PWM para cada motor. ¿Han cambiado las salidas del driver L298N?
- Varíe el valor del potenciómetro y observe los nuevos cambios en la señal PWM. ¿Qué pasa cuando el potenciómetro está en su mínimo valor y en su máximo valor?

## Montaje del Circuito

Para montar la práctica, es necesario tener el modulo driver L298N que se muestra en la **Fig. 11**, el cual posee un jumper que habilita o deshabilita un regulador de voltaje de 5V para la circuitería interna del IC (Pin llamado **Vlógico**). Por tanto:

- Si el regulador está activado (jumper cerrado) **Vlógico** es una salida de 5V que podemos emplear para alimentar otros dispositivos.
- Si el regulador está desactivado (jumper abierto), **Vlógico** es una entrada a la que tendremos que proporcionar un voltaje de 4.5 a 5.5V.

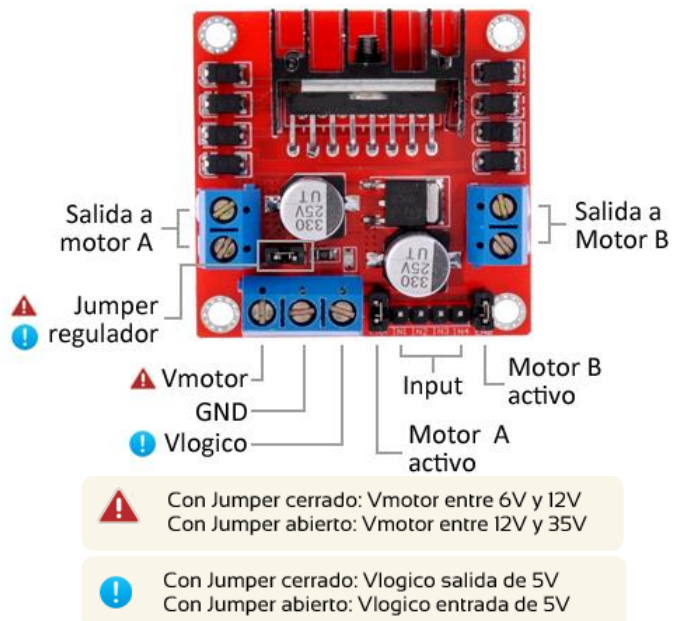


Fig. 11 Modulo L298N

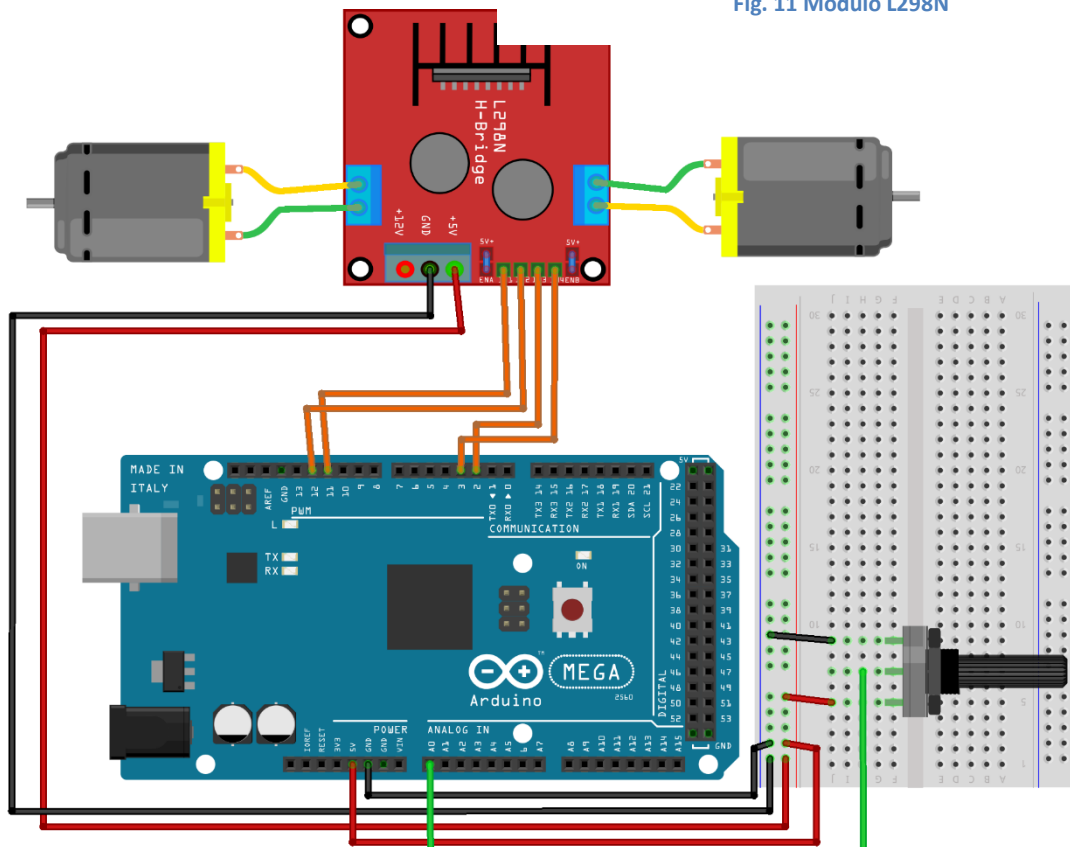


Fig. 12 Diagrama de Montaje de Motores DC.

## Actividad

Agregue al circuito dos funciones más, de tal forma que ahora tenga dos modos extra para el motor, un modo que gire a la izquierda y un modo que gire a la derecha. Recuerde que estos modos se

hacen deteniendo un motor y haciendo que el otro trabaje en dependencia del sentido del giro que se quiera hacer.

## Practica #2 Control de un Servo Motor

### Código Arduino

Así a cómo podemos controlar Motores DC con los pines digitales, también podemos hacerlo con los Servo Motores, haciendo uso del mismo driver L298N para el caso de motores que soliciten mayor consumo. Para este control se agregará en el código la librería **Servo.h** que usa la técnica PWM para hacer el movimiento por ángulo de los motores.

```
#include <Servo.h>
Servo myServo;
int angulo = 0;

void setup()
{ myServo.attach(9);
  pinMode(0, OUTPUT);}

void loop()
{
  digitalWrite(0, HIGH);
  for (angulo = 0; angulo < 180; angulo += 10)
  {
    myServo.write(angulo);
    delay(500);
  }
  digitalWrite(0, LOW);
  for (angulo = 180; angulo > 1; angulo -= 10)
  {
    myServo.write(angulo);
    delay(500);
  }
}
```

## Simulación en Proteus

6. La simulación de esta práctica se hará uso de varios componentes electrónicos que se detallarán en la siguiente tabla para su respectiva búsqueda en el Proteus.

Nombre de Componente	Código o Nombre del en Proteus	Características	Cantidades
Arduino	Arduino MEGA 2560	Librería de Arduino	1
DUAL FULL-BRIDGE DRIVER	L298	Driver para Motores	1
Servo Motor	MOTOR-PWMSERVO	Animado	1
Osciloscopio	Osciloscopio	4 Canales	1

7. Monte los componentes del circuito según la Fig. 10.

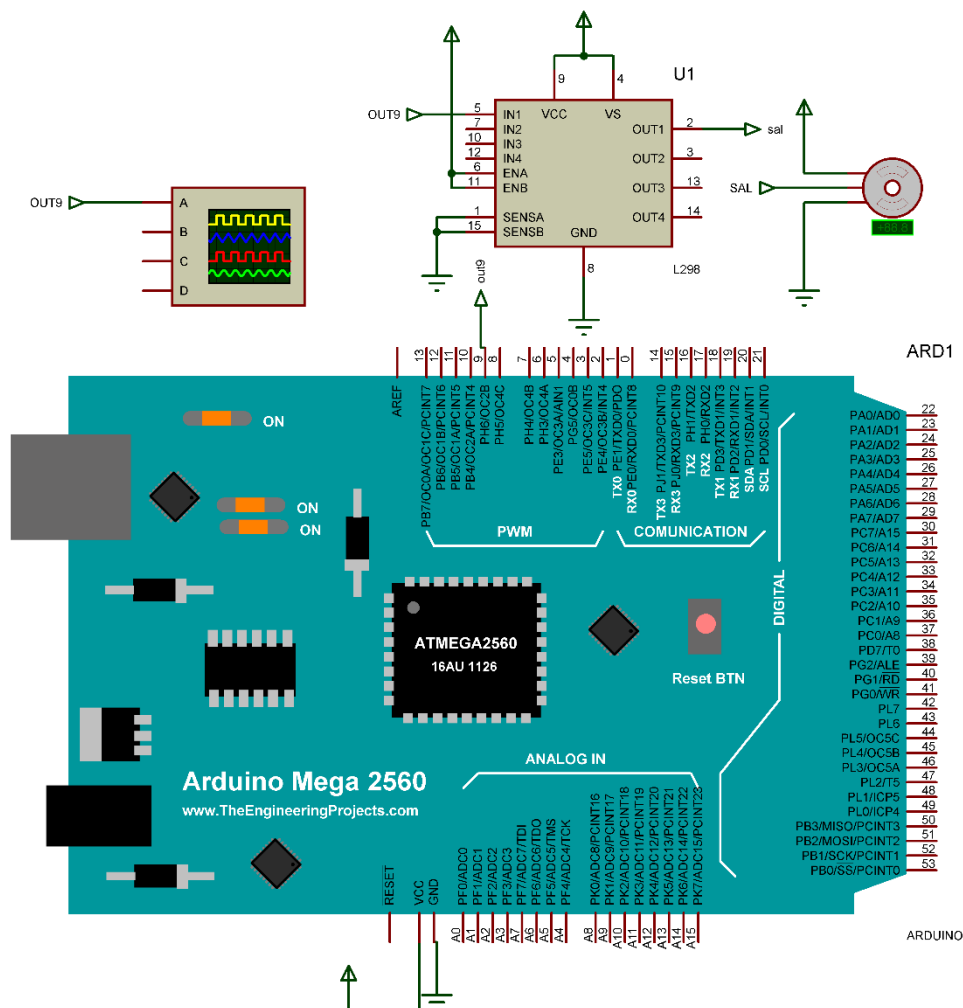


Fig. 13 Simulación de Servo Motor.

8. Observe la Simulación, mire los cambios del Servo Motor, cada vez que el numero incrementa, el ángulo también lo hace, en caso contrario cambia de giro el motor.

## 6. Actividades Propuestas

Realizado las dos prácticas de laboratorio, es necesario combinar los códigos para que se realice la secuencia de los motores DC a través de unas direccionales y que con un pulsador el Servo Motor cambie el giro de los ángulos.

