

Motores DC con arduino

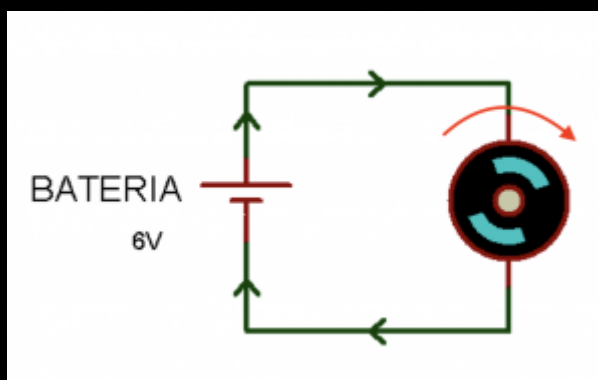
¿Cómo funciona un motor de CD, cómo controlamos su giro?

Un motor de CD básicamente es una carga que, al alimentarlo con el voltaje necesario, realizará giros en una dirección. Todos los motores de CD tienen indicado una polaridad.

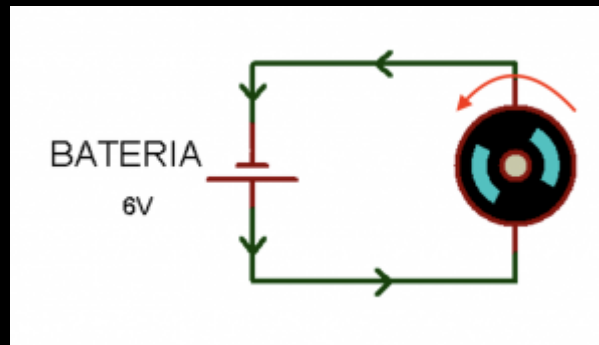
Es decir, una terminal positiva y una negativa. Si a un motor de CD le cambias la polaridad, lo único que pasará es que el giro lo realizará en el otro sentido.



Imagina que tienes un motor pequeño de CD que se alimenta con 6V en la polarización directa (positivo-positivo) y (negativo-negativo) el motor girará en un sentido:



Si ahora invertimos la fuente de voltaje (positivo-negativo y negativo-positivo) el motor simplemente girará en sentido contrario.

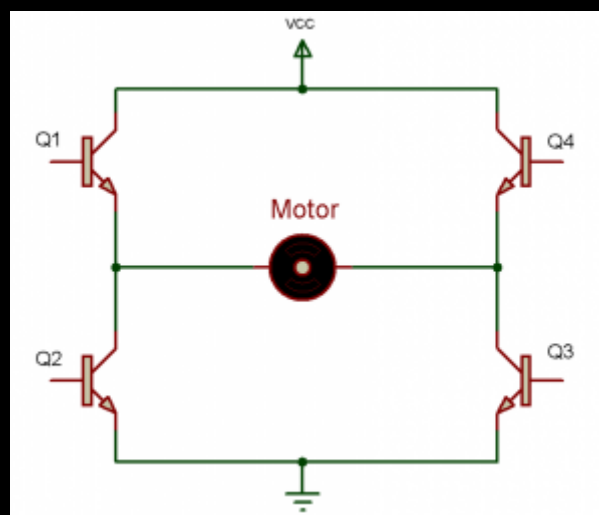


La velocidad del giro y el voltaje de alimentación están dados por el fabricante del motor CD, por lo que siempre debemos revisar las especificaciones técnicas del mismo fabricante para que conozcas sus características técnicas.

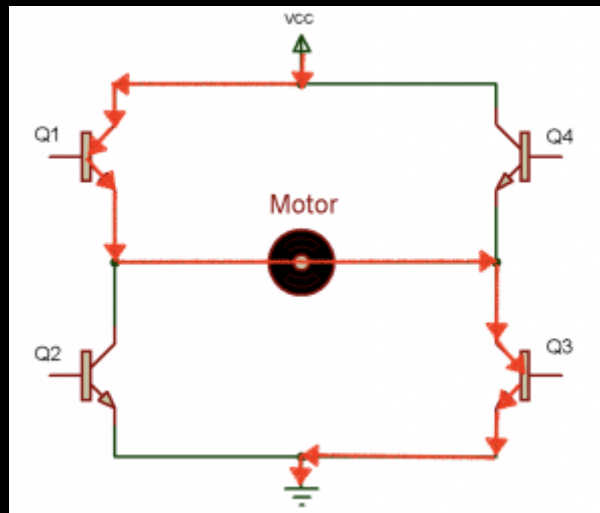
¿Que es un puente H?

Un circuito conocido como puente H, nos permite manipular la dirección de la corriente eléctrica en uno o en otro sentido, esto se logra por medio de cuatro transistores. Están ordenados de manera tal, que cuando necesitamos hacer girar el motor en una dirección, se activan dos; y cuando queremos hacer girar el motor en la otra dirección, activamos otros dos.

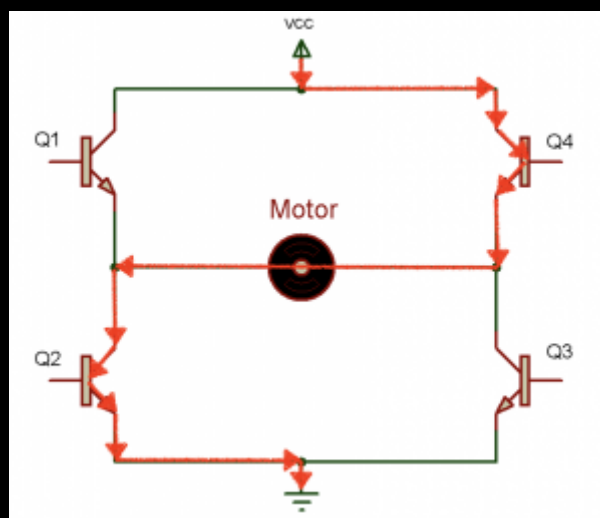
Por la forma en que están distribuidos, es que se conoce como puente H, ya que se parece a esta letra(H)



Si activamos los transistores Q1 y Q3, tendremos la trayectoria mostrada en la siguiente imagen, haciendo que la corriente eléctrica cruce en un sentido al motor



Si activamos los transistores Q2 y Q4, entonces tendremos una trayectoria en el sentido inverso a través del motor.



Entonces el puente H nos ayudará a controlar la dirección de la corriente, por lo tanto la dirección del giro del motor.

Alimentación de motores

Motor de 6V:

Opción 1: 4 pilas AA para los motores y una de 9V para el arduino.

Opción 2: 1 celda de litio (batería de celular) para los motores y una pila de 9V para el Arduino.

Opción 3: Cualquier batería superior a 6V(de suficiente corriente) y un step down para bajar el voltaje.

Motor de 12V:

Opción 1: Pila duracell 9V

Opción 2: Batería lipo o litio de 2 o 3 celdas.

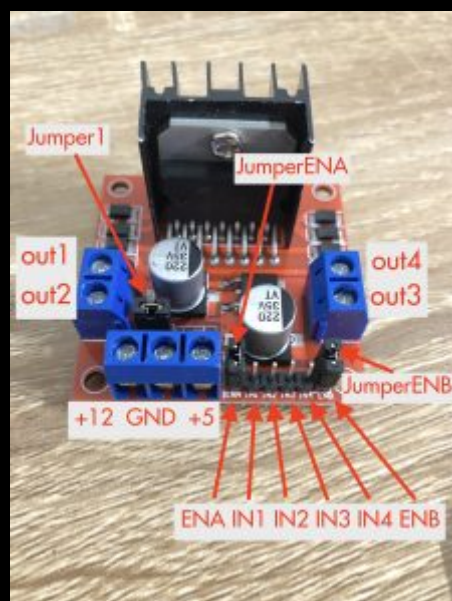
Opción 3: cualquier batería de 12V (de suficiente corriente)

Opción 4: 1 celda de litio (batería de celular) o batería de 3.7V y un step Up para aumentar el voltaje.

Independientemente del valor del voltaje de la batería podemos usar conversores DC-DC para aumentar (step up) o reducir (step down) el voltaje.

Modulos: Modulo L298N

El módulo que contiene el circuito integrado L298N es capaz de controlar el giro de dos motores de CD y contiene además algunas terminales de conexión y algunos Jumpers (puentes de conexión). Es importante que revisemos para qué sirve cada cosa, ya que mediante ellos, controlaremos y alimentamos nuestros motores.



El módulo L298N contiene dos pares de salidas, cada par es para controlar cada motor, por lo que puede controlar dos motores. Para indicarle los sentidos de giro, el módulo contiene cuatro entradas llamadas IN1, hasta IN4. Las entradas IN1 e IN2 son para controlar el primer motor y las entradas IN3 e IN4 son para controlar el segundo motor. También cuenta con entradas de habilitación, estas sirven para indicarle al módulo si queremos o no utilizar algunas de las salidas a motor.

Para habilitar el motor 1 contamos con la entrada ENA y para habilitar el motor 2 utilizaremos la entrada ENB. Si observamos las entradas de habilitación traen un jumper (una conexión físicamente conectado a voltaje, por lo que si el jumper de las entradas de habilitación está colocado, ambas entradas de habilitación estarán encendidas. Si queremos controlar cuándo habilitar o deshabilitar algún motor, debemos remover este jumper y conectarle un cable a alguna salida de nuestro arduino)

También contamos con entradas llamadas +12, GND y +5. Estas terminales tienen distintas funciones dependiendo del estado en el que se encuentre el Jumper 1. Cuando dejamos conectado el Jumper 1, la terminal +12 se utilizará para alimentar motores de entre 6 y 12V, y la terminal +5 será una salida de 5 VCD. Cuando el Jumper 1, la terminal +12 se utilizará para alimentar motores de entre 12 y 35V, la terminal +5 será utilizada como entrada de 5 VCD, como no estamos utilizando el regulador interno, es necesario que insertemos una señal de 5 VCD para alimentar a los transistores internos

Aquí una pequeña tabla de como conectar nuestros motores

Jumper	Terminal +12	Terminal +5
Conectado	Entrada de 6 a 12 VCD	Salida de 5 VDC
Desconectado	Entrada de 12 a 32 VCD	Entrada de 5 VDC

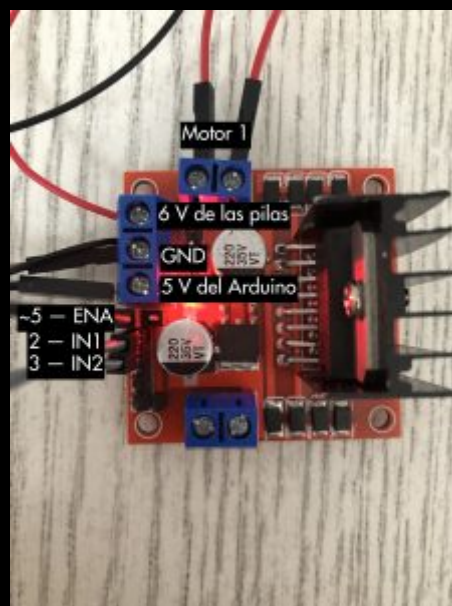
Código de Ejemplo:

```
2  int entrada1 = 2;
3  int entrada2 = 3;
4
5  void setup() {
6      pinMode(entrada1, OUTPUT);
7      pinMode(entrada2, OUTPUT);
8  }
9
10 void loop() {
11     digitalWrite(entrada1, HIGH);
12     digitalWrite(entrada2, LOW);
13     delay(3000);
14     digitalWrite(entrada1, LOW);
15     digitalWrite(entrada2, LOW);
16     delay(3000);
17 }
```

Con este código, le indicaremos al módulo que haga girar al motor en un sentido, haga pausa de 3 segundos y luego lo detenga por otros tres segundos e inmediatamente después gire en el sentido opuesto durante otros 3 segundos. Y esto se estará repitiendo constantemente.

Variar la velocidad de un motor con el módulo L298N

Debemos controlar la entrada de habilitación, removiendo el jumper correspondiente y conectando un cable a alguna salida del tipo PWM del arduino.



Código de ejemplo:

```
1 int entrada1 = 2;
2 int entrada2 = 3;
3 int enableA = 5;
4
5 void setup() {
6   pinMode(entrada1, OUTPUT);
7   pinMode(entrada2, OUTPUT);
8   pinMode(enableA, OUTPUT);
9 }
10
11 void loop() {
12   digitalWrite(entrada1, HIGH);
13   digitalWrite(entrada2, LOW);
14   analogWrite(enableA, 128); // Equivalente a 50 %
15   delay(3000);
16   digitalWrite(entrada1, HIGH);
17   digitalWrite(entrada2, LOW);
18   analogWrite(enableA, 255);
19   delay(3000);
20   digitalWrite(entrada1, LOW);
21   digitalWrite(entrada2, LOW);
22   analogWrite(enableA, 255); // Equivalente al 100 %
23   delay(3000);
24 }
```

Hemos creado una salida llamada enableA para manipular la entrada ENA del módulo. Esta salida tiene que ser de tipo PWM para cambiar el ancho del pulso de la señal que activará el transistor. En este ejercicio, enviamos a escribir la señal PWM mediante analogWrite y le enviamos un valor de 128 para crear un ancho de pulso del 50% del ciclo útil. Recordemos que una señal PWM puede ir de 0 al 100% enviando valores entre 0 y 255.

El código, pondrá al motor a girar a una velocidad del 50% durante 3 segundos, luego a máxima velocidad durante 3 segundos, luego se detendrá durante 3 segundos, hará una pausa de 3 segundos y se repetirá de nuevo.