

Tutorial_11. Control de motores de continua con puente H

Contenido

1. Motores DC	3
Contenido teórico	3
Motor de corriente continua.....	3
Control de giro y velocidad de un motor DC mediante un puente H.....	10
Proyecto11_01: Giro de un motor DC controlado por un driver L293D	13
Proyecto11_02: Giro de un motor DC controlador por un driver L293D y PWM	15
Ejercicios.....	16

Material necesario

- Placa Arduino
- Cable micro USB
- Placa Protoboard
- Cables Jumper-Wire
- Componentes:
 - Potenciómetro
 - Driver L293D
 - Motor DC-3V

IMPORTANTE:

Comprobaremos que nuestra placa **Arduino** está **desconectada** y sin energía, puesto de no ser así podría dañarse tanto la placa, como el equipo. Una vez hemos realizado esta comprobación, pasaremos a realizar el montaje.

1. Motores DC

Contenido teórico

Motor de corriente continua

El motor de corriente continua (DC) es una máquina que convierte la **energía eléctrica en mecánica**, provocando un **movimiento rotatorio**. Es simple de usar, se le aplica una **tensión continua** y empieza a **girar**. Si se **invierte la polaridad** de la alimentación se consigue que **gire en el sentido opuesto**.

Los motores DC pueden alcanzar muchas revoluciones por minuto (rpm) pero no son muy precisos y además no pueden detenerse de forma inmediata.

Los motores DC demandan una corriente de trabajo alta, normalmente no menos de 300mA.

IMPORTANTE:

1. Las salidas de 5V de Arduino suministran unos 300mA si la placa está alimentada mediante un puerto USB, y si es mediante un alimentador externo será de unos 700mA.
2. Por otra parte, la salida de tensión de 3.3V está regulada y dicho regulador no puede proporcionar más de 150mA, aunque se recomienda que no supere los 50mA.
3. Otra cuestión importante es que cada pin del Microcontrolador **NO** puede proporcionar más de 40 mA. **NO CONECTAR DIRECTAMENTE UN MOTOR A LOS PINES E/S.**

Idealmente la solución para alimentar **motores** pasa por usar una **fuentes de alimentación independiente** de la que alimenta a Arduino.

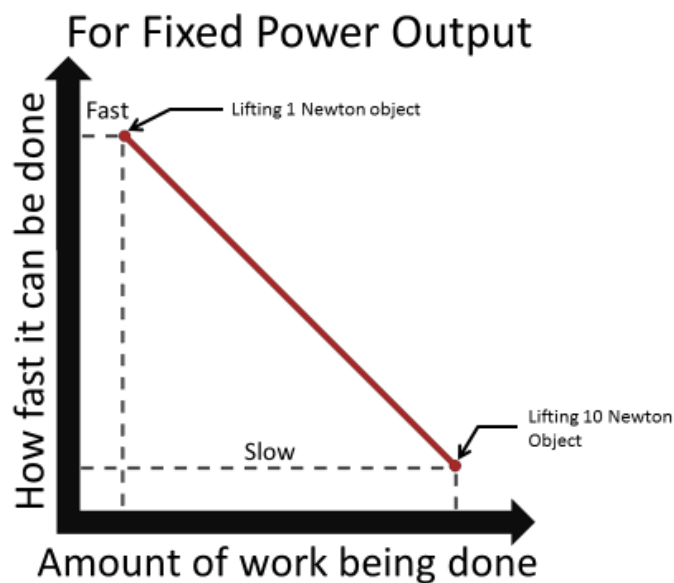
Parámetros básicos del motor DC

Tensión (V)	Tensión de operación	Motor pequeño → 1.5-6V Motor medio → 6-12V Motor grande → 12-24V
Corriente absorbida (I)	Corriente que el motor requiere de la fuente de alimentación.	Motor sin carga → I baja sobre 300mA Motor con carga → I puede ser 200% o 300% la I sin carga. Sobre 2A
Velocidad	Velocidad de giro del eje del motor (rpm). Es proporcional a la tensión aplicada.	Motor sin carga → 18.000 rpm Motor con carga → rpm se reducen
Par (torque)	Fuerza que el motor ejerce sobre la carga (g·cm, oz·inch, N·m, ...). Es proporcional a la corriente suministrada.	Motor pequeño → 10 – 90 g·cm

Fórmulas básicas del motor DC

Par (N·m)= Fuerza x distancia

Potencia (W)= Par x Velocidad



(Imagen: <http://curriculum.vexrobotics.com/>)

A partir de las fórmulas se puede extraer las siguientes conclusiones:

- Si el motor DC gira a una velocidad alta, es posible reducir ésta para conseguir un Par mayor. Para ello se utilizan reductoras, de modo que se consiga un giro del eje a menos rpm.
- Con mayor Par se podrá operar con pesos mayores.
- La primera **limitación vendrá impuesta por la corriente de trabajo** que se quiere mantener. Para esa corriente mediante las gráficas se averigua que par se puede conseguir, si es menor que el necesario, se calcula la relación que sería necesario obtener mediante engranajes.

$$\text{Relación} = \text{Par necesario} / \text{Par disponible}$$

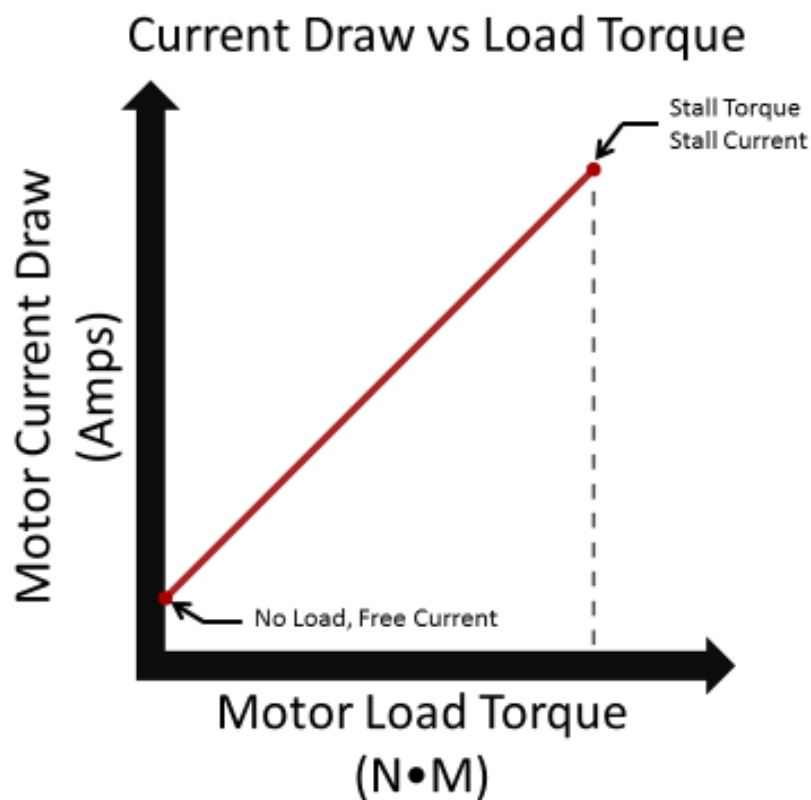
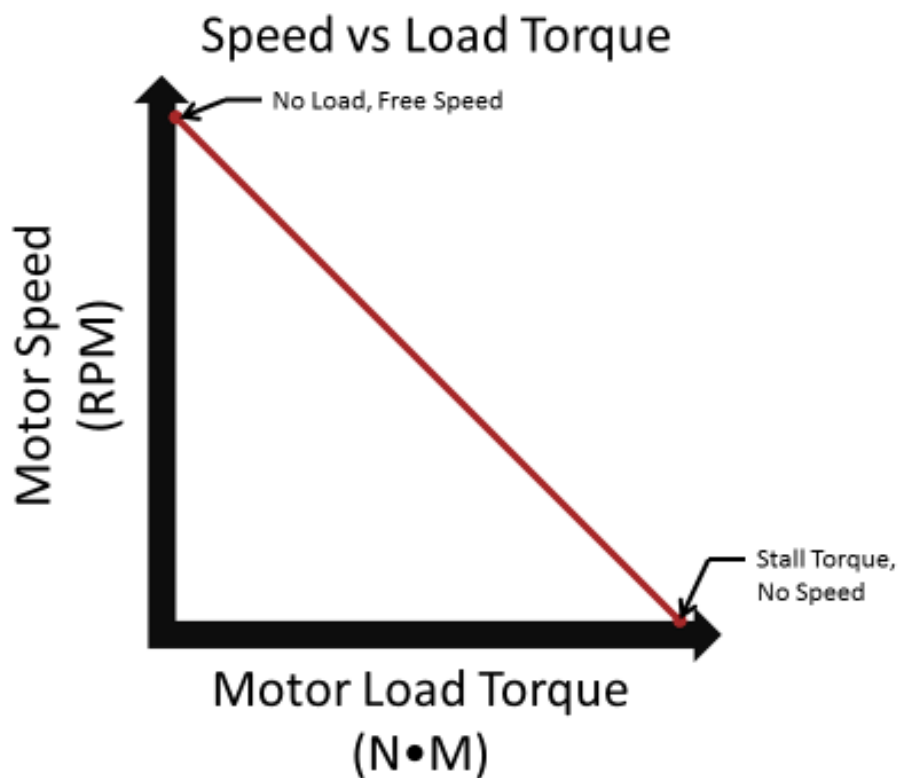
Características que diferencian los distintos tipos de motores Dc

- **Free speed o No Load Speed (rpm):** Velocidad sin carga
- **Stall Torque (N·m):** Carga que consigue detener el motor
- **Stall Current (A):** Corriente que drena el motor en el punto de Stall-Torque
- **Free current o No Load Current (A):** Corriente que drena el motor sin carga

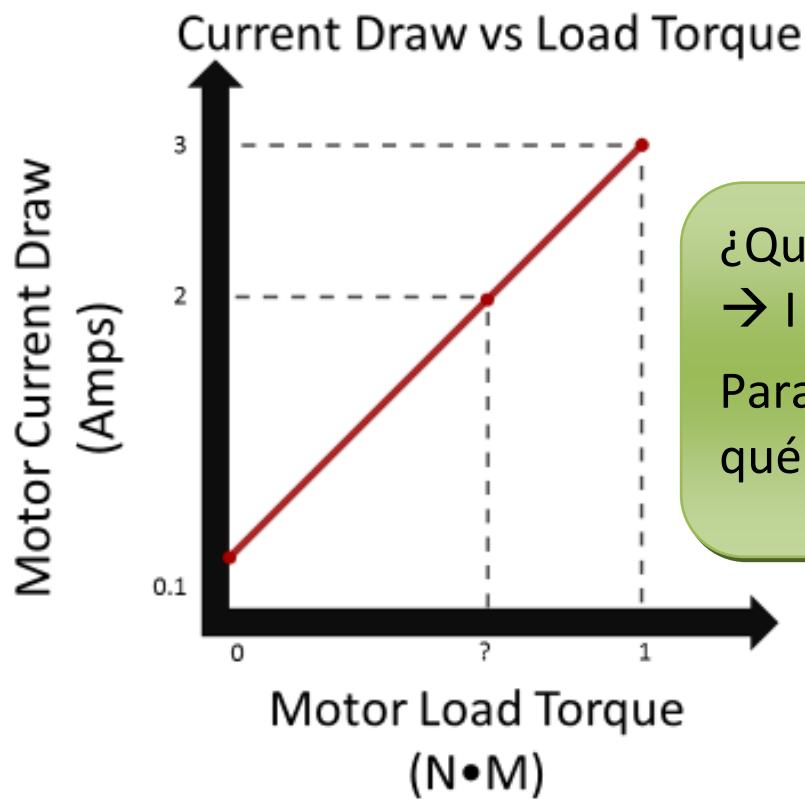
Los fabricantes ofrecen estos datos o bien gráficas que los representan.

Curvas características

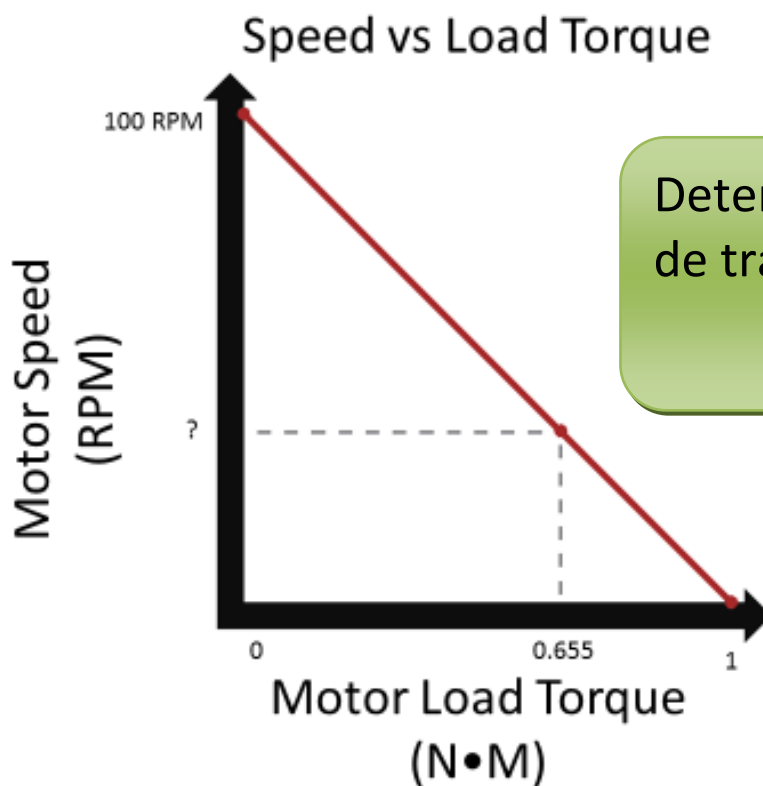
(Imagen: <http://curriculum.vexrobotics.com/>)



Incluso sin carga siempre hay una pequeña demanda de corriente debida a las fricciones a vencer del propio motor.



¿Qué par necesito?
→ I necesaria
Para una I dada →
qué par consigo



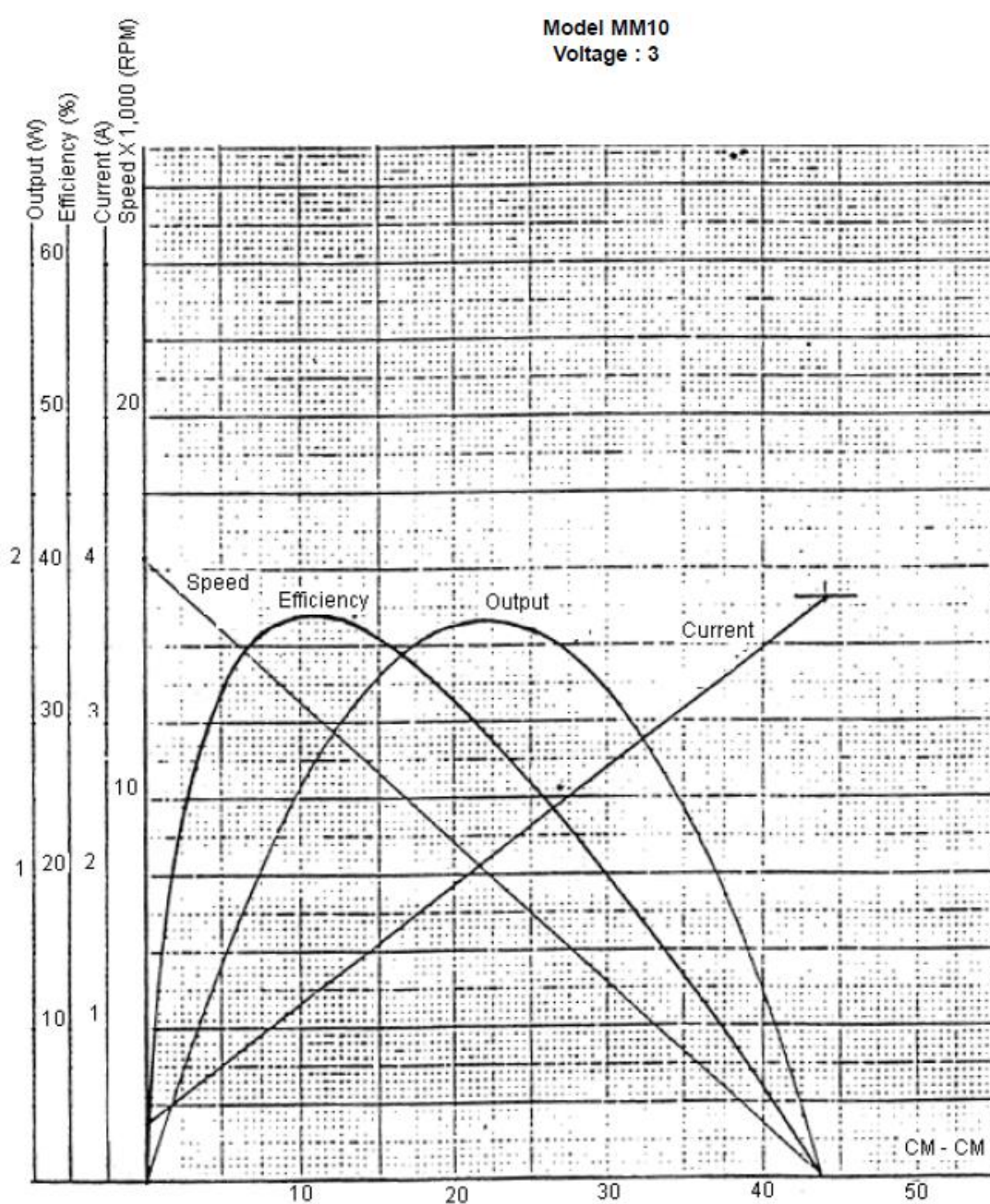
Determinado el par
de trabajo → r.p.m.

Características del motor miniatura de la práctica (Multicomp MM10)

Specification Table

Model	Operating Range	Approximately Weight
MM10	1.5 - 3 V	17 g

Nominal	No Load		At Maximum Efficiency					Stall Torque g-cm
Constant V	Speed rpm	Current A	Speed rpm	Current A	Torque g-cm	Output W	Efficiency %	
1.5 V	8,700	0.32	6,500	0.76	6.2	0.41	33.9	24
3 V	16,300	0.38	12,000	1.1	10	1.23	35.6	44

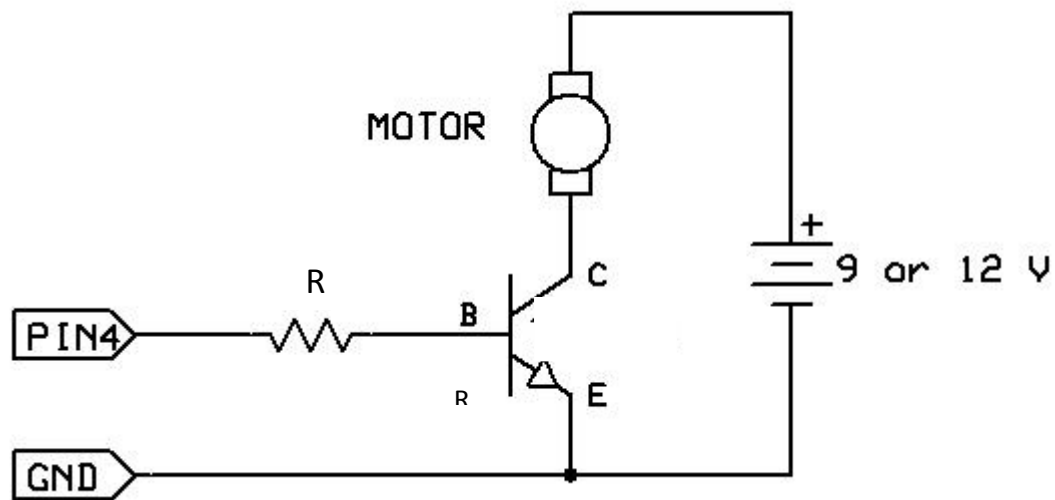


Conexión de un motor DC a un microcontrolador

Como ya se ha comentado anteriormente, cargas como un relé o un motor necesitan para su funcionamiento de una corriente elevada. Los pines de E/S de Arduino no deben utilizarse para suministrar corrientes superiores a los 40mA.

Idealmente, el motor deberá ser alimentado por una fuente externa, y controlaremos su suministro o no de corriente, mediante un “interruptor” gobernado por uno de los pines de E/S del Microcontrolador.

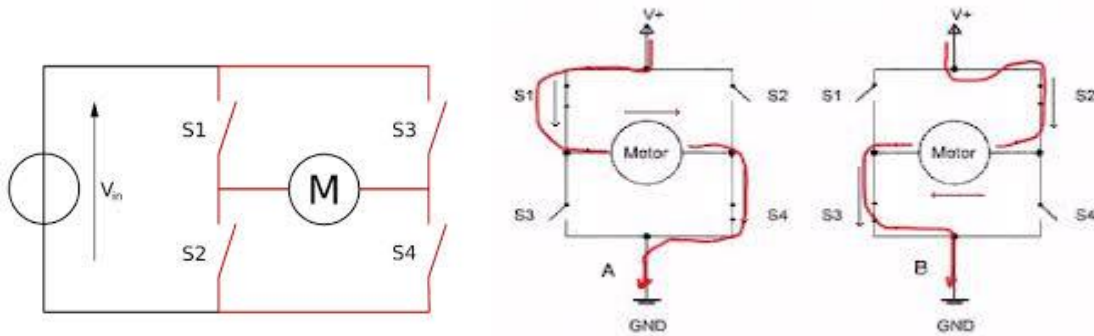
El componente electrónico que usaremos en este proyecto como “interruptor” será un transistor bipolar.



Control de giro y velocidad de un motor DC mediante un puente H

Un **Puente H** es un circuito electrónico que **permite a un motor DC girar en ambos sentidos, avance y retroceso**. Son ampliamente usados en robótica y como convertidores de potencia.

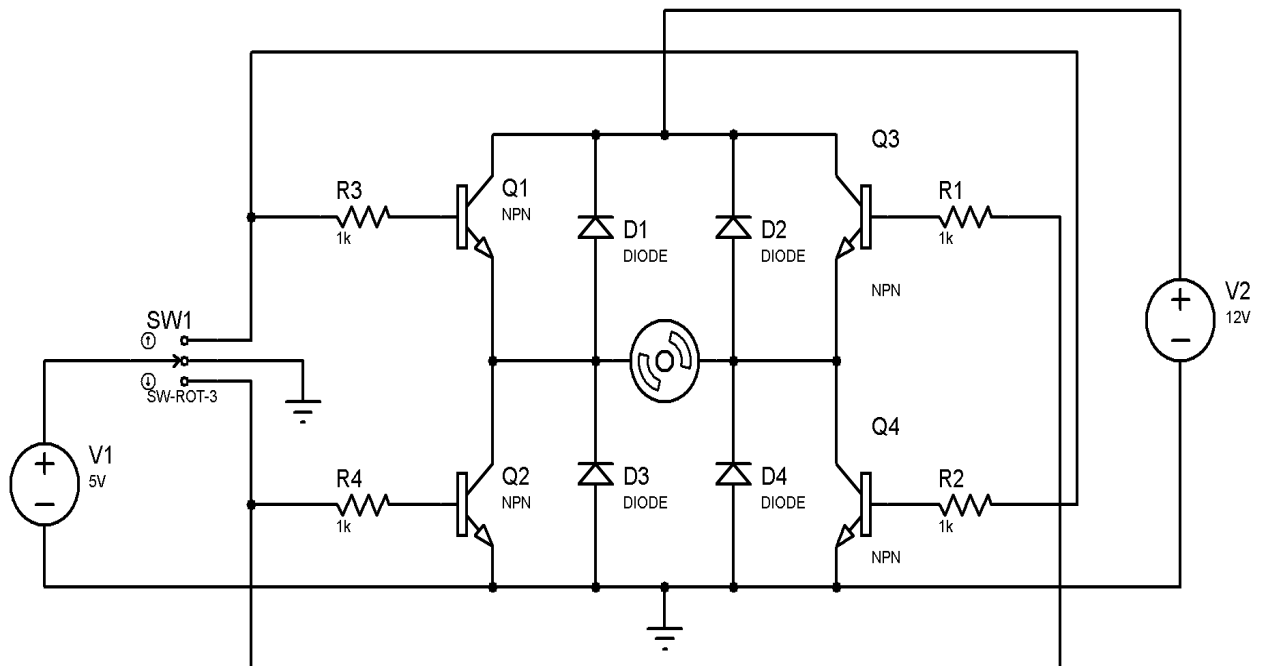
Los puentes H están disponibles como circuitos integrados, pero también pueden construirse a partir de componentes discretos. El término "puente H" proviene de la representación gráfica del circuito. Un puente H se construye con 4 transistores actuando como interruptores (Ver figura).



Cuando los interruptores S1 y S4 están cerrados, S2 y S3 estarán abiertos. La corriente entonces circulará por S1 y S4 haciendo girar el motor en un sentido. Abriendo los interruptores S1 y S4 y cerrando S2 y S3, la corriente circulará por estos últimos permitiendo el giro en el sentido inverso del motor.

Los interruptores S1 y S2 nunca podrán estar cerrados al mismo tiempo, porque ésto cortocircuitaría la fuente de tensión. Lo mismo sucede con S3 y S4.

Este circuito puede montarse usando algunos transistores, resistencias y diodos de protección tal y como puede verse en la siguiente figura:



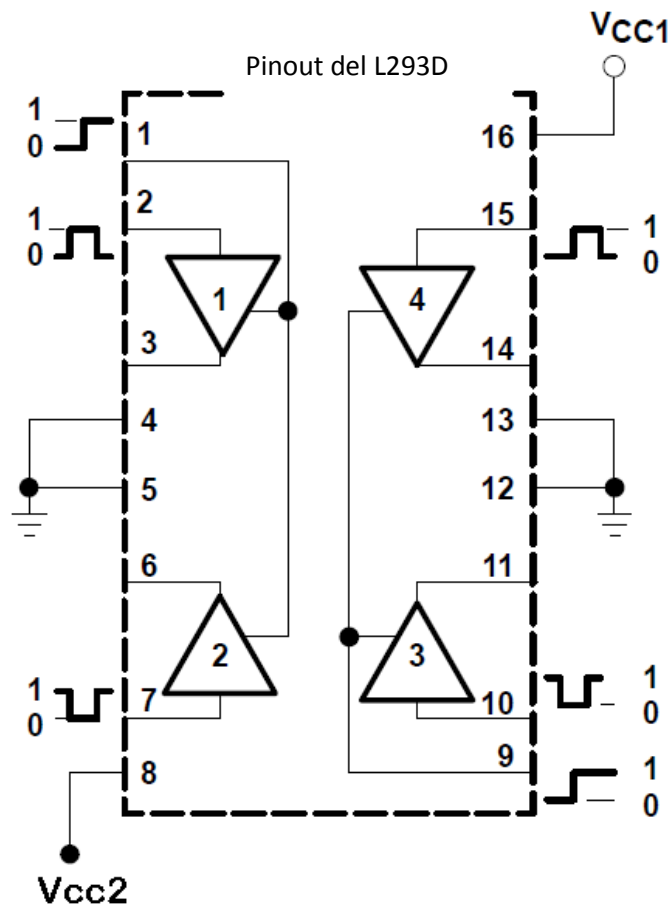
El sentido de giro es controlado por el interruptor Sw1, que aplica la tensión proporcionada por la fuente de tensión V1 a los terminales Base de los transistores Q1 y Q4 en una posición, y a Q2 y Q3 en la otra. Luego, una vez en conducción los transistores correspondientes, V2 proporciona la tensión necesaria para que el motor gire, 12V en el ejemplo de la figura.

Los diodos sirven para proteger a los transistores del mismo modo que ya vimos en el apartado anterior. Por último, las resistencias se utilizan para limitar la corriente de base de los transistores.

El driver L293D

Por suerte, el circuito de puente H podemos encontrarlo como circuito integrado, en concreto el CI **L293D incluye dos puentes H** con sus correspondientes diodos de protección.

Para poder girar en ambos sentidos un motor, será necesario utilizar dos de los cuatro medio-puentes. Así, este circuito integrado nos permitirá gobernar dos motores a la vez.

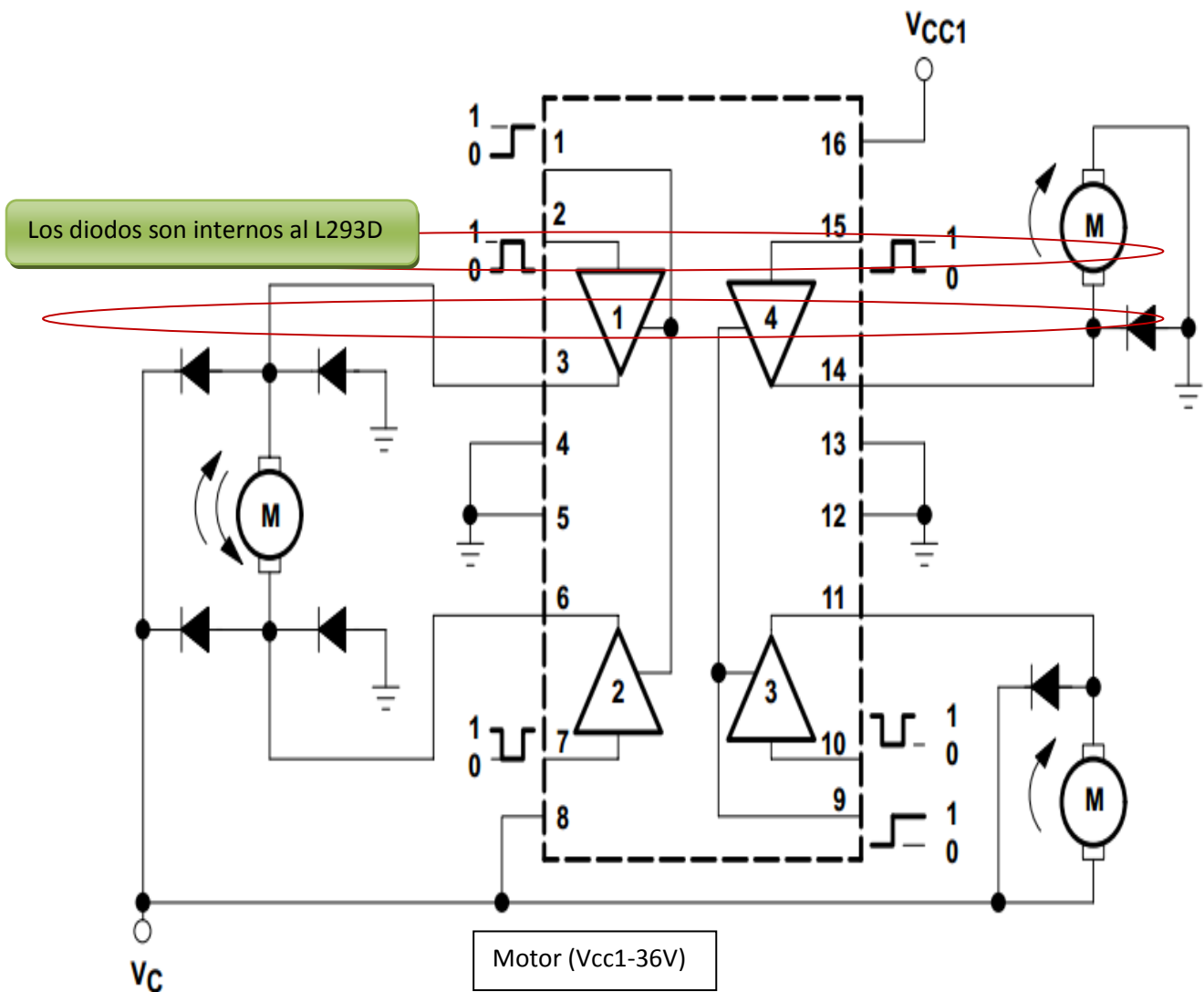


<http://users.ece.utexas.edu/~valvano/Datasheets/L293d.pdf>

Por medio del pin 1 “1,2En” se habilita el puente H del lado izquierdo del chip, y con el pin “3,4EN” el del lado derecho.

Observa que hay dos pines de alimentación Vcc1 (Pin 16) y Vcc2 (Pin 8). El primero alimenta la lógica de control y deberá tener un suministro de de entre 4.5 y 7V. El segundo alimenta el motor y tiene un rango de entre Vcc1 y 36V.

En la siguiente figura vemos como conectar un motor para gobernar su sentido de giro mediante 1A (pin 2) y 2A (pin 6), que pueden conectarse a un microcontrolador. 1Y y 2Y (pines 3 y 6) se conectarán a los terminales del motor.



absolute maximum ratings over operating free-air temperature range (unless otherwise noted)[†]

Supply voltage, V_{CC1} (see Note 1)	36 V
Output supply voltage, V_{CC2}	36 V
Input voltage, V_I	7 V
Output voltage range, V_O	-3 V to $V_{CC2} + 3$ V
Peak output current, I_O (nonrepetitive, $t \leq 5$ ms): L293	± 2 A
Peak output current, I_O (nonrepetitive, $t \leq 100 \mu s$): L293D	± 1.2 A
Continuous output current, I_O : L293	± 1 A
Continuous output current, I_O : L293D	± 600 mA

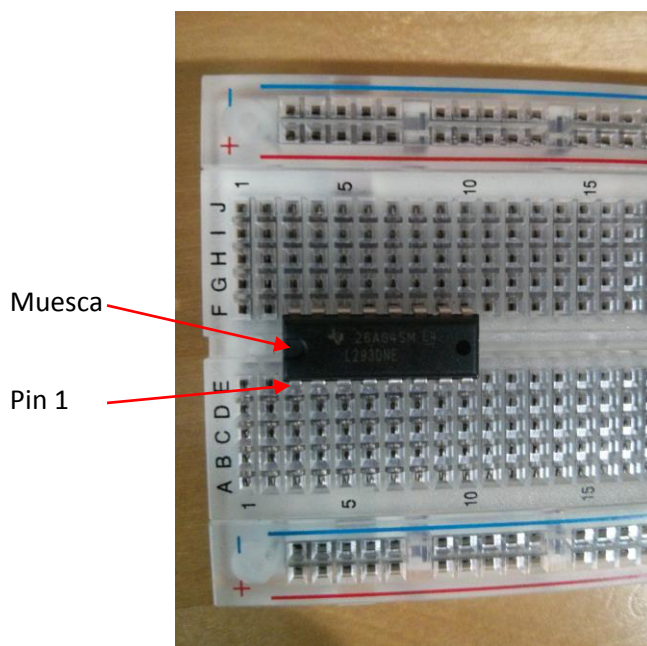
Proyecto11_01: Giro de un motor DC controlado por un driver L293D

Enunciado

Realiza el montaje y código necesarios para controlar la velocidad y el sentido de giro de un motor DC manejado por el driver L293D.

Montaje

Inserta el chip L293 en la placa protoboard, asegúrate de hacerlo disponiendo cada una de las dos hileras de pines a un lado del surco central. Coloca la hilera del pin 1 en la parte de abajo del surco:



Conecta los siguientes pines del L293D a los valores indicados en la tabla:

CI L293D	ARDUINO
Pin 1 (Enable)	5V de la placa Arduino
Pin 16 (Vcc1)	5V de la placa Arduino
Pin 8 (Vcc2-Alimentación del micromotor)	5V de la placa Arduino
Pin 3 (1Y)	Terminal del motor
Pin 6 (2Y)	Terminal del motor
Pin 2 (1A)	Pin de salida de Arduino
Pin 7 (2A)	Pin de salida de Arduino

EN	1A	2A	FUNCTION
H	L	H	Turn right
H	H	L	Turn left
H	L	L	Fast motor stop
H	H	H	Fast motor stop
L	X	X	Fast motor stop

L = low, H = high, X = don't care

Código

```
/* Giro alterno de motor DC con driver L293D
*/
```

```
//Definición de variables
const int in1Pin = 7;
const int in2Pin = 8;
```

```
void setup()
{
  pinMode(in1Pin, OUTPUT);
  pinMode(in2Pin, OUTPUT);
}

void loop()
{
  //Gira Izquierda
  digitalWrite(in1Pin,HIGH);
  digitalWrite(in2Pin,LOW);
  delay(1000);
  //Para
  digitalWrite(in1Pin,LOW);
  digitalWrite(in2Pin,LOW);
  delay(1000);
  //Gira Derecha
  digitalWrite(in1Pin,LOW);
  digitalWrite(in2Pin,HIGH);
  delay(1000);
  //Para
  digitalWrite(in1Pin,LOW);
  digitalWrite(in2Pin,LOW);
  delay(1000);
}
```

Proyecto11_02: Giro de un motor DC controlador por un driver L293D y PWM

Ahora que sabemos controlar el sentido de giro de un motor con el driver L293D, modificaremos el código para controlar la velocidad del motor. Es importante destacar, que en el proyecto anterior hemos conectado la entrada Enable (Pin1) del puente H de la izquierda del chip, a un valor fijo de 5V.

Con este valor fijo y cambiando los valores de las entradas In1 e In2 (pines 2 y 6) conseguíamos el cambio del sentido de giro.

Sin embargo, si queremos realizar un control sobre la velocidad del motor, debemos actuar sobre la entrada Enable (pin 1) mediante PWM.

Enunciado

Realiza el montaje y código necesarios para controlar la velocidad y el sentido de giro de un motor DC manejado por el driver L293D. El sentido de giro debe ser controlado por un pulsador y la velocidad mediante un potenciómetro.

Montaje

Conserva el montaje del proyecto anterior salvo la conexión del pin 1 del chip L293D. Conecta ese pin a la salida 9 de Arduino, u a otro pin con salida PWM. Añade un potenciómetro que nos permitirá regular la velocidad del motor. Recuerda conectarlo a una de las entradas analógicas de Arduino, por ejemplo a A0.

Código

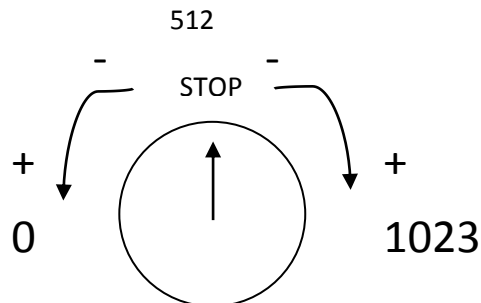
El código debe incluir una función denominada “setMotor” a la que le pasaremos los argumentos “SpeedMotor” y “SpinMotor”.

Ejercicios

1. Crea un nuevo código que incluya dos funciones que hagan girar el motor cada una en un sentido, llámalas **ReverseMotor** y **ForwardMotor**. Añade el código necesario para que use el potenciómetro como un control del sentido de giro y de la velocidad del motor. Es decir, cuando la lectura del potenciómetro esté entre 0 y 500 que vaya adelante y entre 520 y 1023 atrás.

El funcionamiento del motor será el siguiente:

- Si giro desde el centro (500) a izquierda (0), GIRO IZQUIERDA y la velocidad se va incrementando
- Parada del motor entre 500 y 520 (zona central)
- Si giro desde el centro (520) hacia la derecha (1023), GIRO DERECHA y la velocidad se va incrementando



2. Crea una clase de nombre "movROBOT" que incluya funciones para un constructor denominadas "ForwROBOT", "BackROBOT". La primera función hará girar el motor en un sentido, y la otra en el otro sentido. Las funciones deben permitir especificar los pines donde se conectará el motor, y especificar la velocidad de giro de éste.

Importa la librería al IDE de Arduino y comprueba su funcionamiento.