

Auxiliares: Isao Parra, Rodrigo Asenjo, Rodrigo Galilea  
Rodrigo Muñoz, Sebastian Chamas

# Control de motores de corriente continua

## 1. Introducción

Los motores de corriente continua (DC) son empleados en muchas aplicaciones, en nuestro contexto de se usan para generar movimientos constantes en sistemas que, en general, no requieran una alta precisión. Los sistemas de tracción de robots móviles constituyen un ejemplo clásico del uso de estos motores, pues solo se requiere, usualmente, de movimientos constantes, cambios de rapidez y de sentido de giro. Por otro lado, si necesitamos un control adecuado sobre la posición angular es recomendable usar un servomotor o dotar al motor DC con un *encoder*.

El control de rapidez y sentido de giro de un motor DC se realiza mediante un dispositivo electrónico llamado **punte H**. El puente H nos permitirá controlar un motor con un voltaje y corriente muy superiores a las que puede controlar un microcontrolador, como Arduino, de forma directa.

## 2. Funcionamiento del puente H

Para entender el funcionamiento del puente H es útil verlo como un conjunto de interruptores conectados entre la fuente de voltaje  $V_{cc}$  y el motor (figura 1).

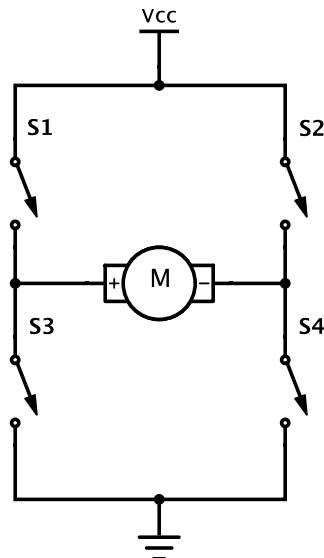


Figura 1: Esquema de funcionamiento del puente H.

El sentido de giro del motor DC depende del sentido en que circula la corriente a través de él. Así si activamos los interruptores  $S_1$  y  $S_4$  el motor girará en avance, por otro lado, si activamos solo los interruptores  $S_2$  y  $S_3$  la corriente entrará por en el otro terminal del motor, provocando un cambio en el sentido de giro. Si habilitamos  $S_1$  y  $S_2$  el motor responderá ofreciendo resistencia al movimiento, como un freno. Finalmente, si encendemos  $S_3$  y  $S_4$  el motor quedara libre.

El control de rapidez de giro se realiza preniendo y apagando los interruptores rápidamente, estos impulsos serán tan rápidos que se percibirá un movimiento constante a menor velocidad.

En la practica los interruptores del puente H se implementan con transistores. El encendido y apagado de los interruptores se realiza mediante una señal PWM. Es importante destacar que no se pueden habilitar  $S_1$  y  $S_3$  de forma simultanea, pues se generará un corto circuito, usualmente este problema se soluciona usando un circuito lógico en el puente H.

### 3. Placa Puente H

Debido a la gran utilidad que presenta el puente H, este ya se encuentra implementado en un PCB que usa un IC L298, esta placa posee dos puentes H por lo que podremos controlar dos motores de forma independiente. La figura 2 se muestra la placa y sus conexiones.

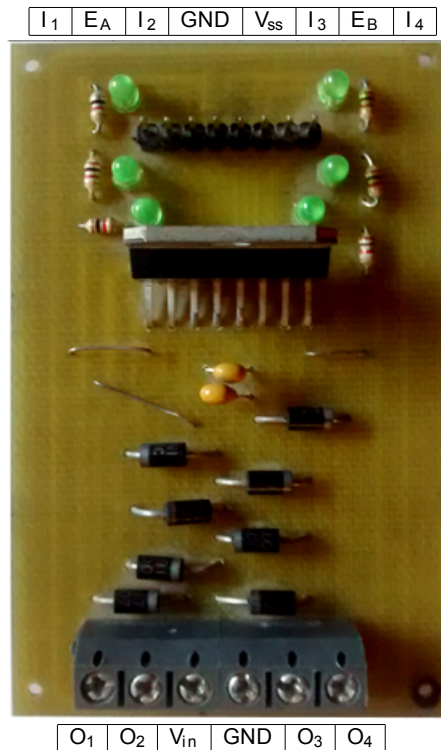


Figura 2: PCB con IC L298.

Por simplicidad describiremos solo uno de los puentes H de la placa, el otro funciona de forma análoga. Las entradas  $I_1$ ,  $I_2$  y  $E_A$  controlan el motor conectado a los terminales  $O_1$  y  $O_2$ . El funcionamiento del

motor depende de los valores lógicos de las entradas como indica el cuadro 1.

	$I_1$ HIGH	$I_1$ LOW
$I_2$ HIGH	Freno	Retroceso
$I_2$ LOW	Avance	Libre

Cuadro 1: Entradas y estado del motor.



### Información

Los valores lógicos corresponden a los valores 0 y 1, a nivel eléctrico (sistema digital) esto se traduce a un voltaje 5 V para 1 (estado HIGH) y 0 V para 0 (estado LOW). Arduino permite manejar salidas digitales, estas solo tienen dos estados (HIGH o LOW), mediante el comando `digitalWrite(pin, HIGH/LOW)`.

Los pines  $I_1$  e  $I_2$  establecen el sentido de giro y el estado, para controlar la rapidez de giro usamos el pin  $E_A$  con una entrada del tipo PWM, donde el ciclo de trabajo será proporcional a la velocidad del motor. El cuadro 2 señala la función de cada uno de los terminales de entrada y salida.

Terminal	Descripción	Tipo de señal
$I_1, I_2$	Control motor #1	HIGH o LOW
$E_A$	Habilitar puente H #1	Señal PWM
$V_{ss}$	Voltaje de alimentación circuito lógico	5 V DC
GND	Tierra ( <i>ground</i> )	-
$I_3, I_4$	Control motor #2	HIGH o LOW
$E_B$	Habilitar puente H #2	Señal PWM
$O_1, O_2$	Salida motor #1	Voltaje de salida para motor #1
$O_3, O_4$	Salida motor #2	Voltaje de salida para para motor #2
$V_{in}$	Voltaje de alimentación motores	5 – 30 V DC dependiendo del motor

Cuadro 2: Terminales de entrada y salida de la placa puente H.

El voltaje  $V_{ss}$  puede obtenerse directamente del pin  $V_{cc}$  de Arduino, pues los circuitos lógicos tienen un bajo consumo. Notar que las conexiones a tierra de ambos circuitos, control y motores, deben estar conectadas.



### Atención

El IC L298 soporta una corriente máxima de 2 A por puente H, dado que no posee disipador de calor, límitese a usar motores con un **consumo no superior a 1 A** y verifique que el IC no sufra sobrecalentamiento.

## 4. Conexión de Arduino a la placa puente H

La conexión entre Arduino y la placa puente H para el control de un motor es bastante simple y se muestra en la figura 3. En este caso se ha conectado un motor de 12 V a uno de los puentes H de la placa.

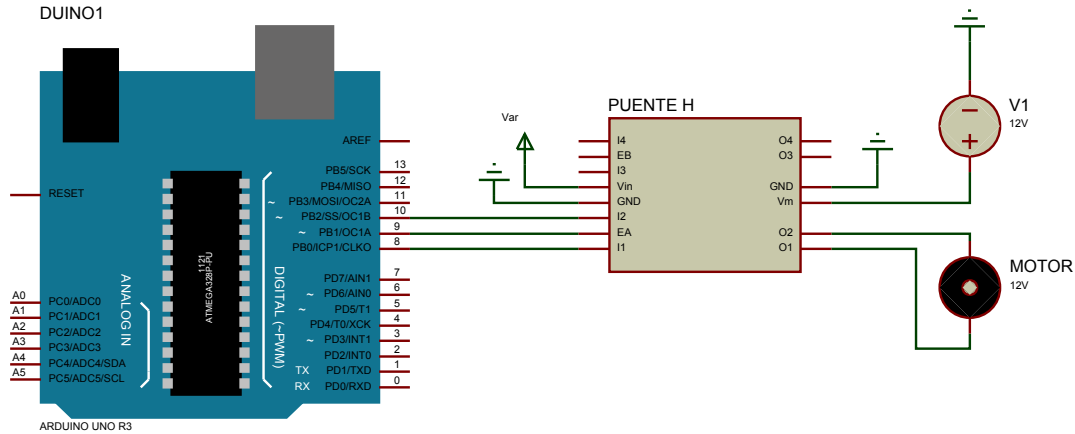


Figura 3: Esquemático de conexión entre Arduino UNO y la placa puente H

Para la configuración del sentido de giro se emplea el método `digitalWrite(pin, HIGH/LOW)` sobre los pines 8 y 10. Para el control de velocidad se usa una señal PWM por el pin 9, esto se implementa con el método `analogWrite(pin, duty)`. El código 1 muestra un ejemplo donde se cambia el sentido y velocidad de giro del motor.

```
1  /*
2  Control de motores de CC con CI L298
3  EI2001 - Robótica y Mecatrónica
4  */
5
6  // Pines de entrada para CI L298
7  #define MOTOR_CTL1 8    // I1 Input 1
8  #define MOTOR_CTL2 10   // I2 Input 1
9  #define MOTOR_PWM 9     // EA Enable A
10
11 void setup()
12 {
13     // Configuración de pines para control del motor
14     // Control de sentido de giro
15     pinMode(MOTOR_CTL1, OUTPUT);
16     pinMode(MOTOR_CTL2, OUTPUT);
17     // Control de velocidad
18     pinMode(MOTOR_PWM, OUTPUT);
19 }
20
21 void loop()
22 {
23     // Movimiento en avance
24     digitalWrite(MOTOR_CTL1, HIGH);
25     digitalWrite(MOTOR_CTL2, LOW);
```

```

26 // PWM para controlar velocidad
27 analogWrite(MOTOR_PWM, 200);
28 delay(2000);
29
30 //Cambiar sentido de giro
31 digitalWrite(MOTOR_CTL1, LOW);
32 digitalWrite(MOTOR_CTL2, HIGH);
33 delay(1000);
34 analogWrite(MOTOR_PWM, 255); //Velocidad Max
35 delay(1000);
36 }

```

Código fuente 1: Código de ejemplo IC L298

#### 4.1. Experiencia práctica

Se adjunta el *sketch* `motorcc.ino` donde se han implementado métodos de control para el motor #1, que usa la misma conexión anterior. Modifique el programa añadiendo funcionalidades que le permitan controlar dos motores de forma independiente usando ambos puentes H del IC L298. Para llevar a cabo esto considere:

- Definir pines de control para el otro puente H.
- Modificar los métodos para controlar ambos motores.



#### Correcciones

Si encuentras algún error envía un correo a [rmunozriff@ing.uchile.cl](mailto:rmunozriff@ing.uchile.cl). Puedes descargar el código fuente desde [esta dirección](#).