

Práctica de manejo de motores CC con arduino

Antonio Martínez

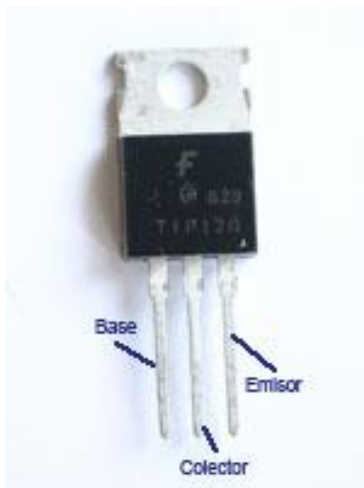
Motor CC



Existen varios tipos de motores (servos, Motores de paso..), el motor que tenemos en el aula normalmente es un motor DC. Son los motores que encontramos en juguetes, DVD's, etc. Al aplicarle un voltaje el motor gira y al quitarlo se detiene. Para cambiar la dirección solo hay que cambiar su polaridad.

Para controlarlo necesitamos más componentes: un transistor de potencia y un Diodo.

Transistor



Transistor

Tiene 3 patas: base, colector y emisor. Hay muchos tipos, nosotros utilizaremos el TIP 120. Será un interruptor electrónico, (no mecánicos) que abrirá y cerrará el "interruptor" de nuestro motor.

Diodo

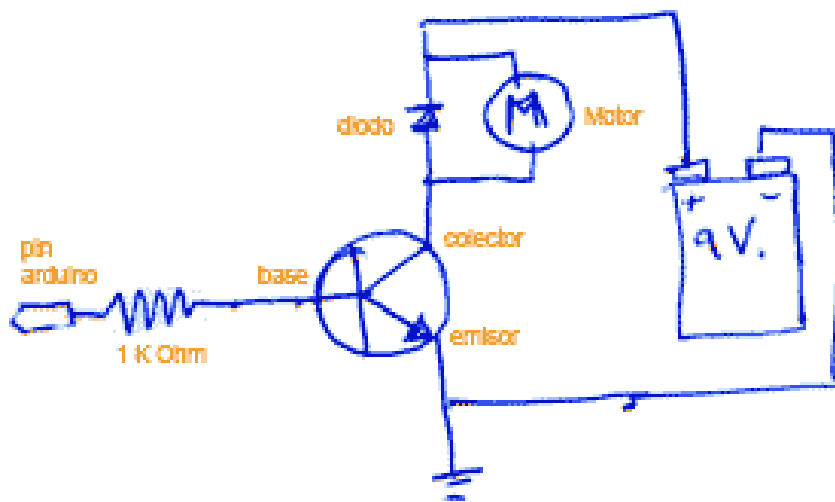


Diodo

El diodo es un semiconductor. La función de los diodos es dejar pasar la corriente en un solo sentido. Tiene polaridad, la banda oscura indica el cátodo(-).

¿Cómo se conecta a Arduino?

En este montaje utilizamos el pin digital PWM (modulación de Ancho de Pulso).



Necesitamos una fuente de alimentación externa, en este montaje una pila de 9 Voltios. El diodo nos sirve de protección para el Arduino que se alimenta con 5 Voltios, dejando que la corriente de 9 Voltios sólo circule para alimentar al motor. En cambio la Tierra debe estar conectada a la Tierra del Arduino, pues si no el circuito se desestabiliza. Esto es una regla en todos los montajes: las tierras o masas deben estar siempre conectadas.

El transistor además está protegiendo el Arduino, funcionando como interruptor para este circuito. También se utiliza una resistencia para protegerlo.

¿Cómo se programa?

Para enviar un pulso al motor se utiliza la instrucción `analogWrite(pin, pulso);` Los valores serán de 0 a 255, ya que PWM es una "simulación" de analógico, en la que 0 es 0 Voltios y 255, 5 Voltios.

En este programa enviamos un pulso al Motor que lo hace ir de velocidad o a las máximas revolución:

```
//  
  
int pulso = 0; // variable donde almacenamos el valor del pulso  
int pinMotor = 10; // Pin 10 que puede generar PWM  
  
void setup()  
{ // No tenemos que declarar que es una salida analógica  
}  
void loop()  
{  
for(pulso = 0 ; pulso <= 255; pulso ++)// ciclo para ir subiendo el voltaje desde 0 a 5  
voltios  
{  
analogWrite(pinMotor, pulso); // enviamos el pulso al motor vía PWM  
delay(15);  
}  
delay(600);  
}
```

Puente en H

Para controlar un motor de corriente continua se utiliza la técnica de puente en H.

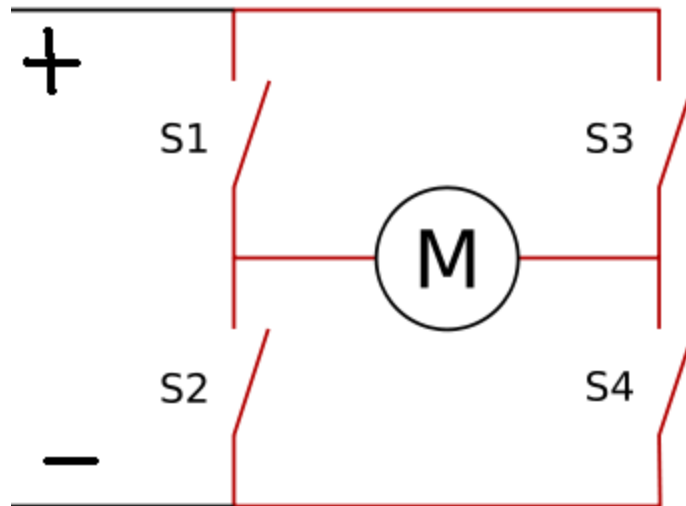


Figura 1A

Si cerramos los interruptores S1 y S4 el motor gira en un sentido, si cerramos S3 y S2 el motor gira en sentido contrario.

Chip L293D/B(puente H):

Es un circuito integrado o chip, que puede ser utilizado para controlar simultáneamente la velocidad y dirección de dos motores de continua (contiene dos puentes H). La diferencia entre el modelo L293D y L293B, es que el primero viene con diodos de protección que evita los daños producidos por los picos de voltaje que puede producir el motor.

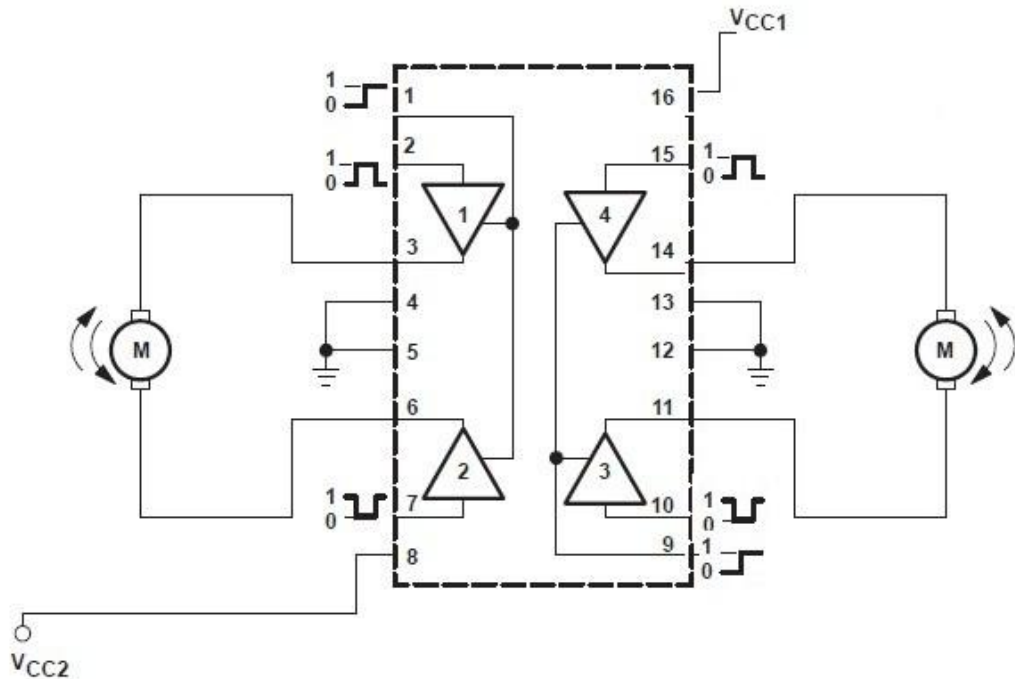


Figura 1. Diagrama de conexiones del circuito L293D con dos motores

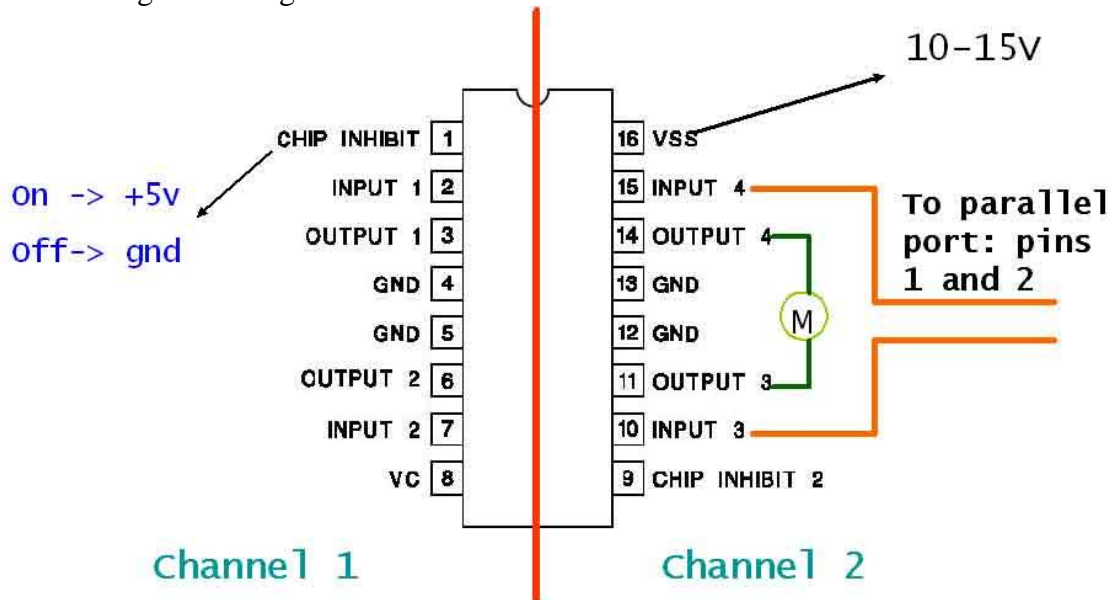


Figura 2. Diagrama de conexiones del circuito L293D (vista encapsulado)

Contiene 4 pines digitales (2,7,10, 15) para controlar la dirección de los motores. Los pines "enable" (1,9) admiten como entrada una señal PWM, y se utiliza para controlar la velocidad de los motores con la técnica de modulación de ancho de pulso. Los motores van conectados entre uno de los pines 3, 6, 11, o 14. La tensión V_{ss} es la que alimentará o dará potencia al motor.

:

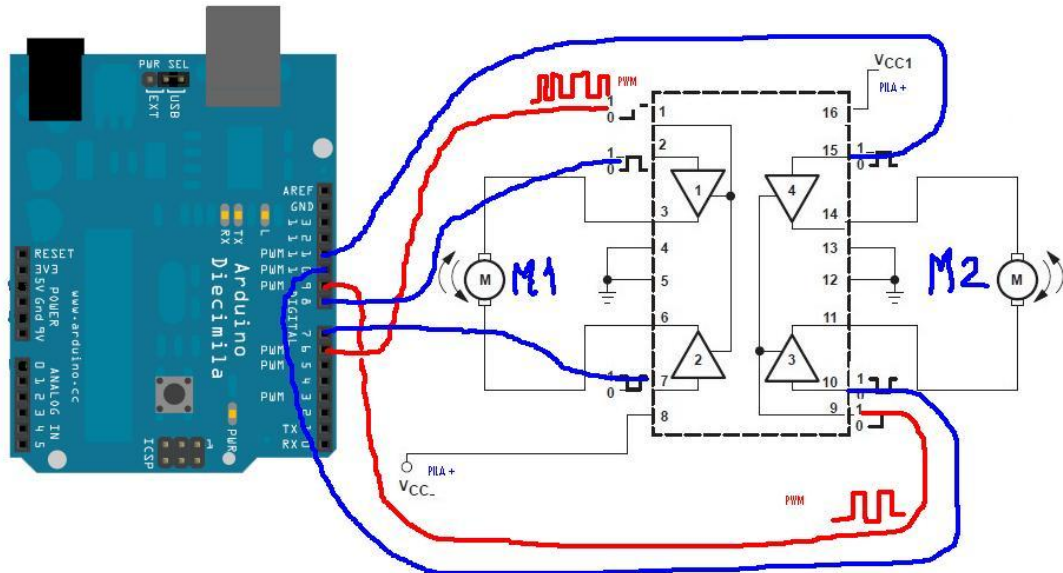


Figura 3. Conexiones del arduino al circuito L293D para control de dos motores.

<i>Conexiones arduino</i>	<i>Conexiones circuito L293</i>	<i>Funciones</i>
Pin 6 , salida PWM motor A (Tren de pulsos modulados en anchura)	Entrada Pin 1, habilita la transferencia de potencia a motor A	Para controlar velocidad Motor A
Pin 7 (dirección borna 1 motor A)	Pin 7	(dirección borna 1 motor A)
Pin 8 (dirección borna 2 motor A)	Pin 2	(dirección borna 2 motor A)
Pin 9 , salida PWM motor B (Tren de pulsos modulados en anchura)	Entrada Pin 9, habilita la transferencia de potencia a motor A	Para controlar velocidad Motor B
Pin 10 (dirección borna 1 motor B)	Pin 10	(dirección borna 1 motor B)
Pin 11 (dirección borna 2 motor B)	Pin 15	(dirección borna 2 motor B)
Negativo arduino (gnd)	4,5,12 y 13 unidas al gnd del arduino y al negativo de la pila	Los negativos del arduino, L293D y Pila deben estar unidos.
	Pines 8 y 16	Conectar al + de la pila, no conectar al + del Arduino pues crea interferencias y el usb del ordenador no tiene suficiente potencia para mover los motores
	Salidas 3 y 6 del L293D al motor A	Conexión del motor A
	Salidas 14 y 11 del L293D al motor B	Conexión del motor B

Tabla1 de conexiones entre Arduino y L293D

Código para controlar mediante ordenador robot con 2 motores CC desde el puerto serie
Las conexiones aparecen en la figura 3 y en la Tabla1.

```
char val;
int PWMA = 6; //velocidad motor A
int dirmotorA1 = 7; // direccion motor a borna 1
int dirmotorA2= 8; // direccion motor a borna2

int PWMB = 9; //velocidad motor B
int dirmotorB1 = 10; // direccion motor b borna 1
int dirmotorB2= 11; // direccion motor b borna2

int velocidad = 120;

void adelante(){

digitalWrite (dirmotorA1,HIGH); // gira motor A derecha
digitalWrite (dirmotorA2,LOW);
analogWrite (PWMA, velocidad);

digitalWrite (dirmotorB1,LOW); // gira motor B izquierda
digitalWrite (dirmotorB2,HIGH);
analogWrite (PWMB, velocidad);

}

void atras(){

digitalWrite (dirmotorA1,LOW); // gira motor A izquierda
digitalWrite (dirmotorA2,HIGH);
analogWrite (PWMA, velocidad);

digitalWrite (dirmotorB1,HIGH); // gira motor B derecha
digitalWrite (dirmotorB2,LOW);
analogWrite (PWMB, velocidad);

}

void izquierda(){

digitalWrite (dirmotorA1,HIGH); // gira motor A derecha
digitalWrite (dirmotorA2,LOW);
analogWrite (PWMA, velocidad);

digitalWrite (dirmotorB1,HIGH); // gira motor B derecha
digitalWrite (dirmotorB2,LOW);
```



```
void loop()
{
  if( Serial.available() )
  {
    val = Serial.read();
  }
  switch (val) {

    case 's':{
      paro();
      break;

    }

    case 'a':{
      adelante();
      break;
    }

    case 'r':{
      atras();
      break;
    }

    case 'i':{
      izquierda();
      break;
    }
    case 'd':{
      derecha();
      break;
    }
  }
}
```