

1.2.12. Motor de Corriente Continua

En este proyecto vamos a controlar un motor de corriente continua como los que se utilizan en multitud de proyectos de tecnología o juguetes como coches teledirigidos, Scalextric, etc. De hecho también se suelen utilizar en robots móviles muy a menudo. El robot Printbot que utilizamos en este libro no utiliza este tipo de motores pero otros robots a la venta en internet si los llevan (por ejemplo el de la Figura 1.2.12-1). Esta actividad la vamos a dividir en dos subactividades: una primera actividad donde moveremos los motores en una dirección u otra, y una segunda donde además controlaremos la velocidad a la que giran.

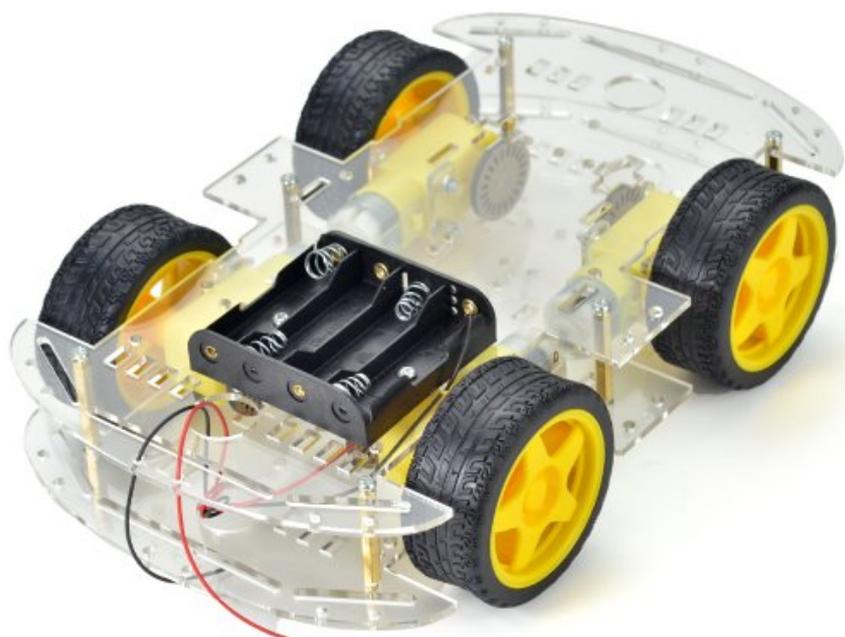


Figura 1.2.12-1 Robot con 4 motores de corriente continua de venta en internet, como el que vamos a controlar en esta actividad

Componentes:

- Placa ZumBT o Arduino compatible
- Motor de corriente continua (no contenido en el kit de robótica de BQ). Puede adquirirse en cualquier tienda de electrónica. A veces incorporan una reductora como el de la Figura 1.2.12-2.
- Placa de alimentación de motores basada en un integrado L298N (no contenida en el kit de robótica de BQ). Esta placa se utiliza en cualquier desarrollo con Arduino que necesite alimentar motores de continua o motores paso a paso. Puede adquirirse en tiendas de electrónica o en la web. (ver Figura 1.2.12-3)

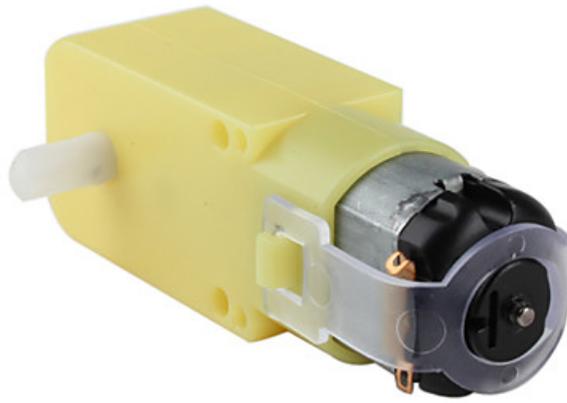


Figura 1.2.12-2 Motor de Corriente Continua con reductora acoplada

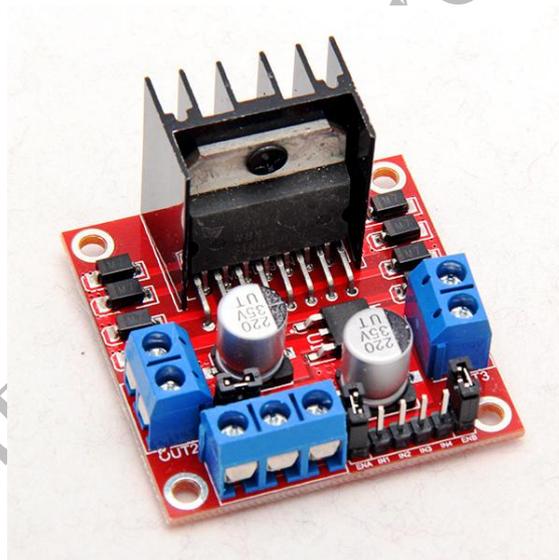


Figura 1.2.12-3 Placa de alimentación de motores basada en el integrado L298N

1.2.12.1. Movimiento sin control de velocidad

Como dijimos, vamos a empezar controlando simplemente la dirección de giro del motor. Por ello no necesitaremos las entradas ENA o ENB del integrado L298, que puentearemos a 5v.

Conexionado.

Vamos a seguir el conexionado mostrado en la Figura 1.2.12-4 entre el Arduino, la placa de alimentación y el motor. En concreto estas son las conexiones que debemos hacer.

Libro de Actividades de Robótica Educativa

el robot Printbot conectado a una de estas placas, como puede observarse el conexionado es similar al expuesto en la Figura 1.2.12-4. Con este conexionado podemos utilizar IN1, IN2 para definir la polaridad en la alimentación al motor y con ello el sentido de giro de motor. Por lo tanto con las salidas 5 y 6 de Arduino, que están conectadas a IN1 y IN2 controlaremos si movemos el motor y la dirección de giro:

Si Pin 5 =1 y Pin 6=0 entonces el motor gira en sentido horario.

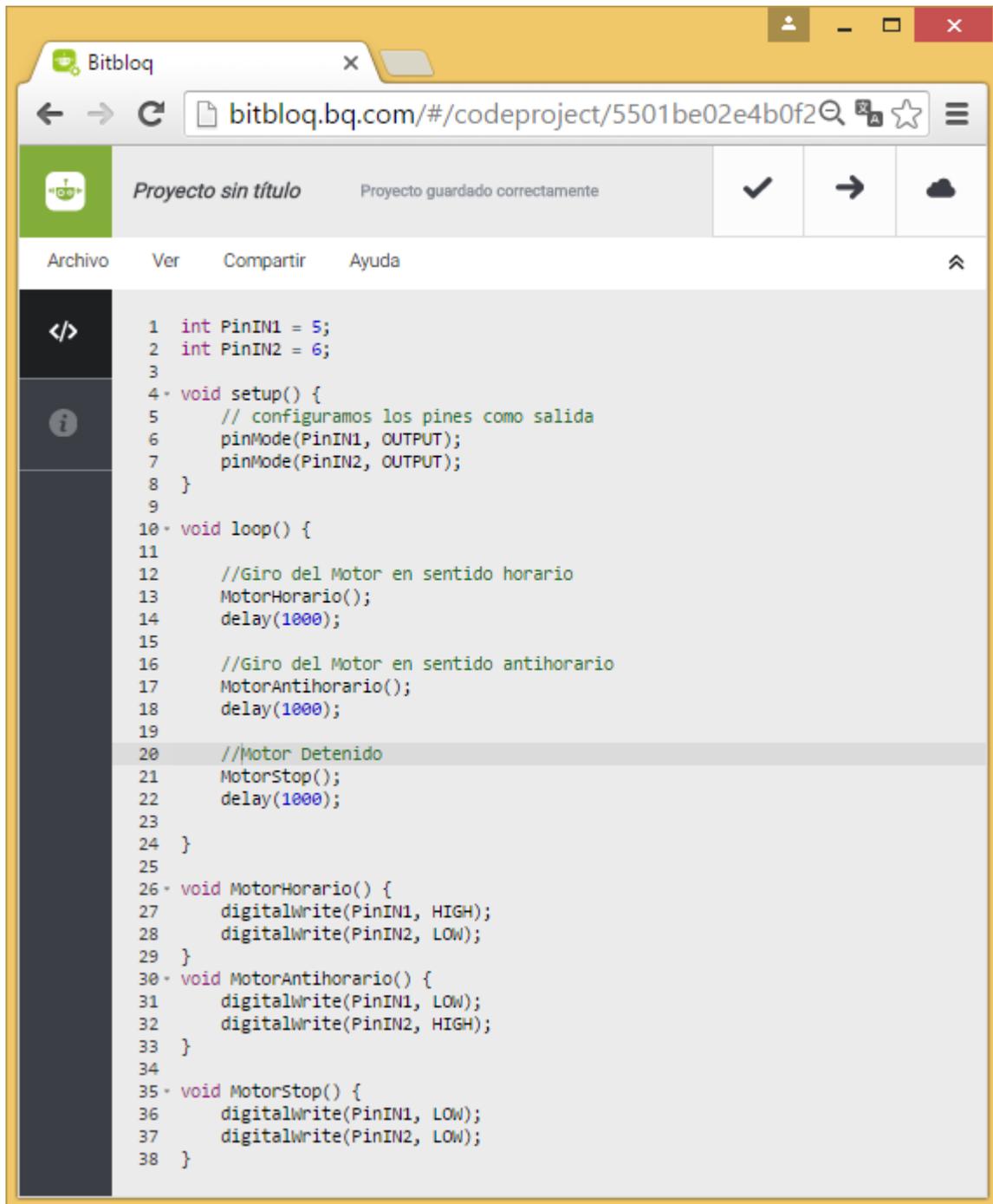
Si Pin 5 =0 y Pin6 = 1 entonces el motor gira en sentido antihorario.

Si Pin5=0 y Pin 6=0 entonces el motor está parado.

Programación

A fecha de edición de esta actividad no existe código de bloques en bitbloq que permita el control de estos motores, por lo que escribiremos el código directamente en Arduino (ver Figura 1.2.12-5):

www.automatycayrobotica.es



The image shows a screenshot of a web browser displaying the Bitbloq online code editor. The browser's address bar shows the URL `bitbloq.bq.com/#/codeproject/5501be02e4b0f2`. The editor interface includes a menu bar with options like 'Archivo', 'Ver', 'Compartir', and 'Ayuda'. The main area contains the following C++ code:

```
1 int PinIN1 = 5;
2 int PinIN2 = 6;
3
4 void setup() {
5     // configuramos los pines como salida
6     pinMode(PinIN1, OUTPUT);
7     pinMode(PinIN2, OUTPUT);
8 }
9
10 void loop() {
11     //Giro del Motor en sentido horario
12     MotorHorario();
13     delay(1000);
14
15     //Giro del Motor en sentido antihorario
16     MotorAntihorario();
17     delay(1000);
18
19     //Motor Detenido
20     MotorStop();
21     delay(1000);
22 }
23
24
25
26 void MotorHorario() {
27     digitalWrite(PinIN1, HIGH);
28     digitalWrite(PinIN2, LOW);
29 }
30 void MotorAntihorario() {
31     digitalWrite(PinIN1, LOW);
32     digitalWrite(PinIN2, HIGH);
33 }
34
35 void MotorStop() {
36     digitalWrite(PinIN1, LOW);
37     digitalWrite(PinIN2, LOW);
38 }
```

Figura 1.2.12-5 Código en Arduino para el movimiento de un motor CC sin control de velocidad

Como vemos, en el código nos hemos definido 3 funciones para movimiento horario, antihorario, y parada. En estas funciones lo único que hacemos es poner los pines 1 y 2 a HIGH o LOW según la lógica explicada anteriormente.

1.2.12.2. Movimiento con control de velocidad

Si la entrada ENA o ENB introducimos una señal PWM podemos controlar la velocidad de giro del motor. Básicamente estas entradas cortan la alimentación del motor y una señal PWM es una señal cuadrada en la que variamos el ancho de pulso, de manera que podemos activar el motor con pulsos más largos o más cortos. Es el mismo fundamento dar mayor o menor intensidad a un led con un

Libro de Actividades de Robótica Educativa

PWM que vimos en la actividad 1.2.2: Si tenemos pulsos largos el motor estará activo la mayoría del tiempo, por lo que su velocidad será mayor. Si utilizamos pulsos cortos el motor estará desactivo la mayoría del tiempo por lo que su velocidad será menor. Si quieres saber más de PWM puedes consultar XX.

Conexionado.

Vamos a añadir al conexionado anterior un pin más que nos sirva, mediante PWM, para el control de velocidad del motor. Con ello el conexionado se nos queda tal como se muestra en la Figura 1.2.12-6 En concreto estas son las conexiones que debemos hacer.

- Pin 5 Arduino → pin IN1 placa L298N
- Pin 6 Arduino → pin IN2 placa L298N
- Pin 9 Arduino → pin EN1 placa L298N (-Jumper de ENA quitado).
- Vin Arduino → 6-12v placa L298N
- GND Arduino → GND placa L298N
- Cables (+,-) motor → Motor 1 (+,-) placa L298N
- Cables (+,-) porta pilas → Jack alimentación Arduino

www.automatizarrobotica.es

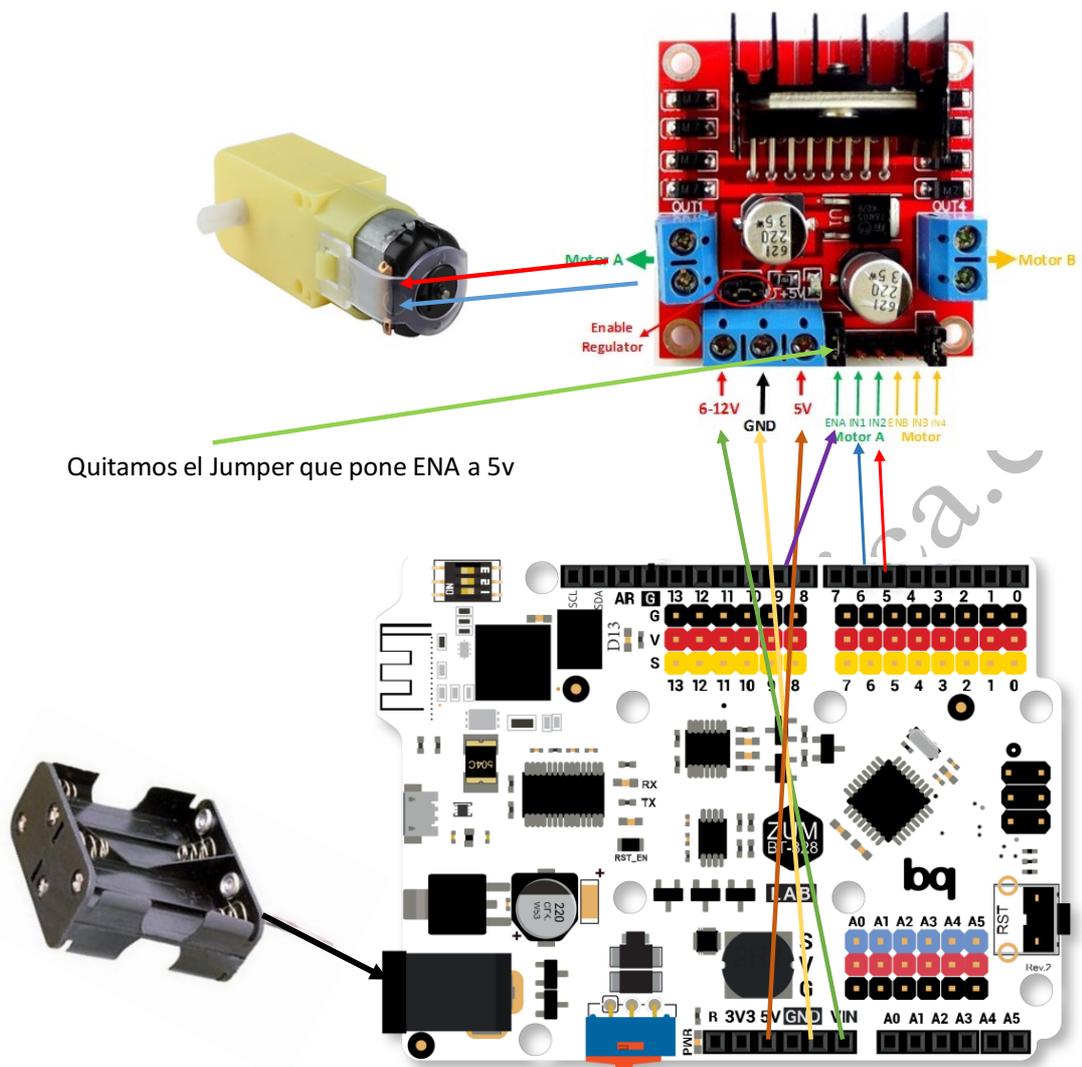


Figura 1.2.12-6 Conexión entre Arduino y placa de alimentación de motores.

Con este conexionado podemos utilizar IN1, IN2 para definir la polaridad en la alimentación al motor y con ello el sentido de giro de motor. Por lo tanto con las salidas 5 y 6 de Arduino, que están conectadas a IN1 y IN2 controlaremos si movemos el motor y la dirección de giro. Con EN1 lo que hacemos es activar o desactivar el motor, por lo que si lo conectamos a una salida PWM podremos controlar la velocidad de giro. En particular la lógica de control de este motor será:

Si Pin 5 =1 y Pin 6=0 entonces el motor gira en sentido horario.

Si Pin 5 =0 y Pin6 = 1 entonces el motor gira en sentido anti-horario.

Si Pin5=0 y Pin 6=0 entonces el motor está parado.

Pin 9 será analógico tomará valores entre 0 y 255, donde 255 es la máxima velocidad del motor.

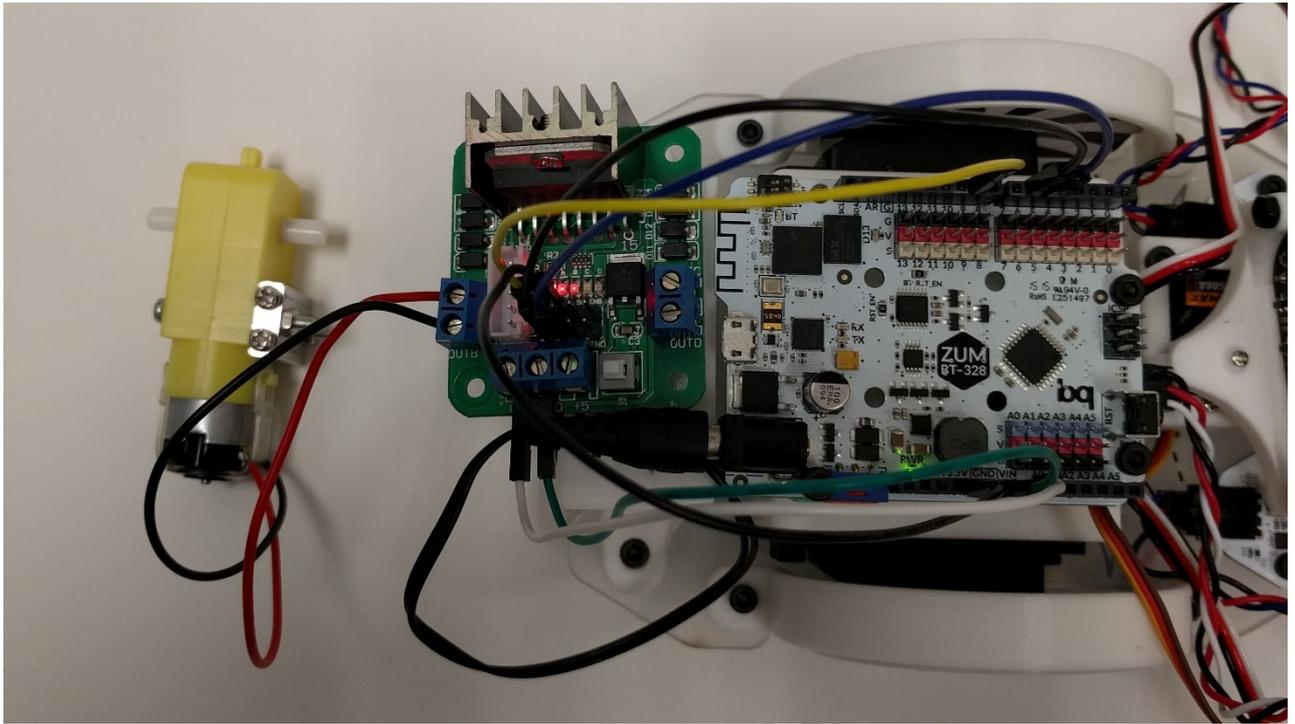
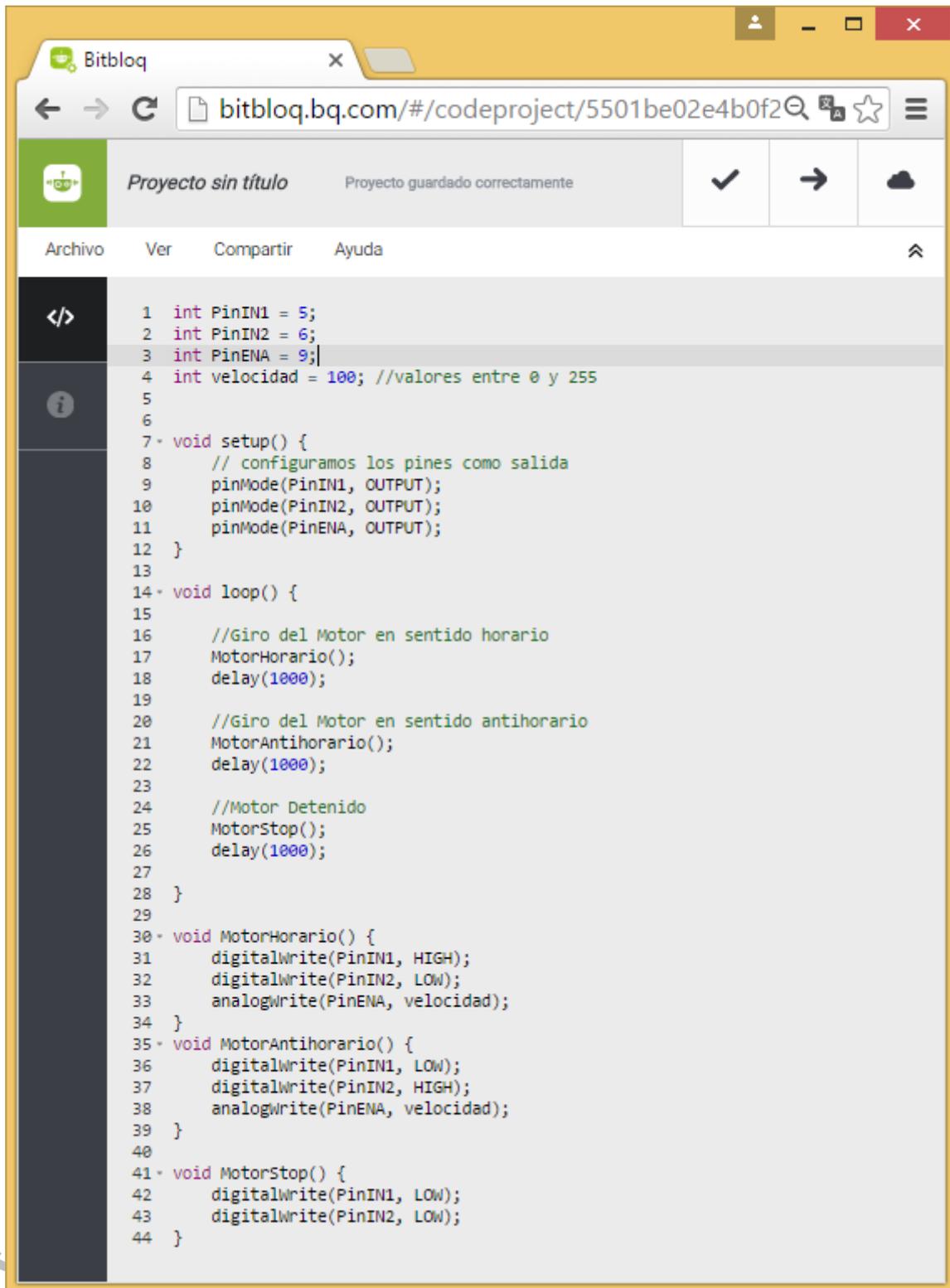


Figura 1.2.12-7. Printbot conectado a una placa L298N para el control de motor de corriente continua

Programación

El código para realizar esto en Arduino es:

www.automatycayron.com



The image shows a screenshot of the Bitbloq web IDE. The browser address bar shows the URL `bitbloq.bq.com/#/codeproject/5501be02e4b0f2`. The IDE interface includes a menu bar with 'Archivo', 'Ver', 'Compartir', and 'Ayuda'. The main area displays the following C++ code:

```
1 int PinIN1 = 5;
2 int PinIN2 = 6;
3 int PinENA = 9;
4 int velocidad = 100; //valores entre 0 y 255
5
6
7 void setup() {
8     // configuramos los pines como salida
9     pinMode(PinIN1, OUTPUT);
10    pinMode(PinIN2, OUTPUT);
11    pinMode(PinENA, OUTPUT);
12 }
13
14 void loop() {
15
16     //Giro del Motor en sentido horario
17     MotorHorario();
18     delay(1000);
19
20     //Giro del Motor en sentido antihorario
21     MotorAntihorario();
22     delay(1000);
23
24     //Motor Detenido
25     MotorStop();
26     delay(1000);
27
28 }
29
30 void MotorHorario() {
31     digitalWrite(PinIN1, HIGH);
32     digitalWrite(PinIN2, LOW);
33     analogWrite(PinENA, velocidad);
34 }
35 void MotorAntihorario() {
36     digitalWrite(PinIN1, LOW);
37     digitalWrite(PinIN2, HIGH);
38     analogWrite(PinENA, velocidad);
39 }
40
41 void MotorStop() {
42     digitalWrite(PinIN1, LOW);
43     digitalWrite(PinIN2, LOW);
44 }
```

Figura 1.2.12-8 Código en Arduino para el control en velocidad de un motor de CC.

Es importante darse cuenta que la única diferencia con el código sin el control de velocidad es que utilizamos “analogwrite” para definir una salida PWM sobre el pin 9 y este pin está conectado con EN1 por lo que podremos controlar la velocidad del motor.

Un poquito de explicación teórica

El **motor de corriente continua** (denominado también **motor de corriente directa**, **motor CC** o **motor DC**) es una máquina que convierte la energía eléctrica en mecánica, provocando un movimiento rotatorio, gracias a la acción que se genera del campo magnético [Wikipedia]. Se compone de dos partes: el estator y el rotor (ver Figura 1.2.12-9).

- El estator es la parte mecánica del motor donde están los polos del imán.
- El rotor es la parte móvil del motor con devanado y un núcleo, al que llega la corriente a través de las escobillas.

Cuando la corriente eléctrica circula por el devanado del rotor, se crea un campo electromagnético. Este interactúa con el campo magnético del imán del estator. Esto deriva en un rechazo entre los polos del imán del estator y del rotor creando un par de fuerza donde el rotor gira en un sentido de forma permanente. Por lo tanto, la forma de controlar el motor será cambiando la corriente eléctrica que circula con el devanado, pudiendo así cambiar la velocidad de giro o la dirección.

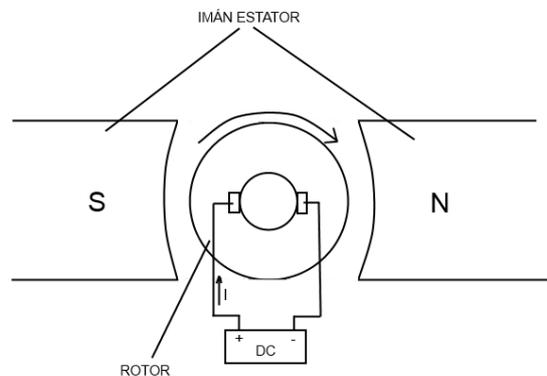


Figura 1.2.12-9 Esquema de un motor CC

Arduino solo proporciona 40mA en sus salidas, que no es suficiente para mover este tipo de motores. Hace falta por lo tanto un circuito que amplifique la señal. El L293N es un integrado para controlar motores DC que usa el sistema puente en H. Los puente en H se componen de 4 transistores que, además de proporcionar una alimentación externa superior a la de Arduino, se comportan como interruptores, cambiando la polarización del motor y por lo tanto su sentido de giro.

En la Figura 1.2.12-10 mostramos un ejemplo de puente en H. Si por ejemplo alimentamos el transistor Q1 y Q4 el motor girará hacia un sentido. Si por el contrario activamos el Q3 y el Q2 el motor girará al sentido contrario. Básicamente el circuito L293N lo que hace es conectar el Pin EN1 a los transistores Q1 y Q4 y el Pin EN2 a los transistores Q3 y Q2.

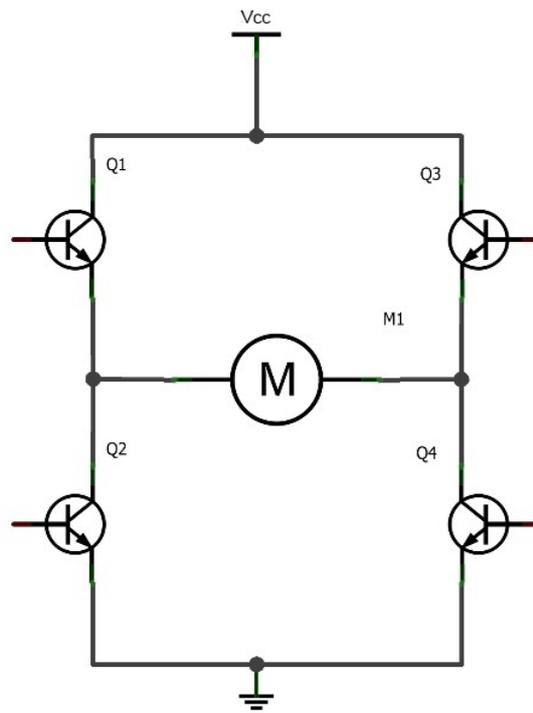


Figura 1.2.12-10 Puente en H para el control de un motor CC

www.automaticayrc.com/robotica.es