CAPÍTULO 8: INICIANDO CON MOTORES

Iniciando con motores

Muchos esperamos este capítulo. Ver girar un motor es una sensación satisfactoria. Ver rodar una cubierta y saber que tenés el poder de hacerla girar cuanto quieras, o el tiempo que quieras, es un pensamiento agradable y superador.

Disfrutemos este proyecto, juguemos con las variables, hagamos ejercicios, pruebas, desafíos y proyectemos nuevas formas para hacer de esta experiencia, muchos proyectos más.

Consejo:

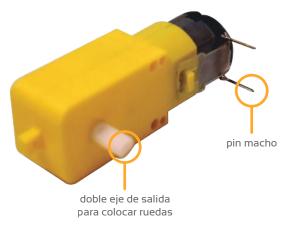
Prestá mucha atención a la polaridad de los motores. No realices cambios con los cables, sino con la programación.



Motores de corriente continua

El motor de corriente continua también se conoce como motor de corriente directa, motor CC o motor DC (por *direct current*, las iniciales en inglés de corriente directa) y es una máquina capaz de convertir energía eléctrica en energía mecánica, provocando un movimiento rotatorio gracias a la acción de un campo magnético.

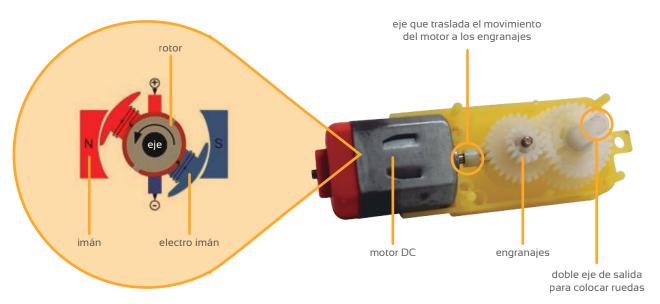
Los motores de nuestro robot son de corriente continua y poseen un eje de doble salida, donde podemos colocar nuestras ruedas, y dos pines macho para su alimentación.



NOTA: Cuida los pines macho, ya que están soldados con estaño. Evita manipularlos en exceso, y no los conectes y desconectes permanentemente, realiza el cambio de polaridad de los mismos por medio del código de programación.

¿Cómo funcionan?

Para que comprendas mejor su funcionamiento, veamos qué hay debajo de la carcasa de nuestros motores. A simple vista podemos distinguir dos partes importantes; por un lado tenemos el motor propiamente dicho y por otro lado tenemos los engranajes que se encargan de la conversión para que, en última instancia, podamos mover una rueda.



El motor DC es el encargado de recibir la energía eléctrica y, a través de sus imanes internos, crea un campo electromagnético. Este campo electromagnético hace que se produzca un movimiento rotatorio, el cual pone en funcionamiento el eje que se conecta directamente con los engranajes. A su vez, estos engranajes imprimen ese movimiento en la rueda.

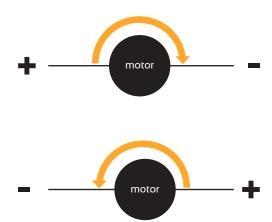




¿Cómo cambiar el sentido en que gira el motor?

Como hemos visto en el punto anterior, el motor es capaz de generar un movimiento rotatorio cuando recibe voltaje en sus terminales (lo que se conoce como los bornes del motor). Existen dos sentidos posibles de giro de los motores; por ejemplo, en un auto es posible ir hacia adelante o manejar en reversa.

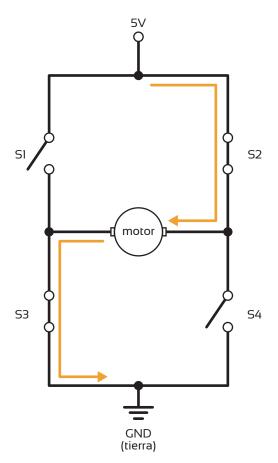
Para lograr que un motor gire en uno u otro sentido lo que hace falta es cambiar la polaridad de sus bornes, una operación sencilla pero que requiere de mucho cuidado para evitar hacer un cortocircuito que podría estropear nuestro arduino. Para realizar esta inversión de polaridad de manera segura vamos a utilizar un puente H.

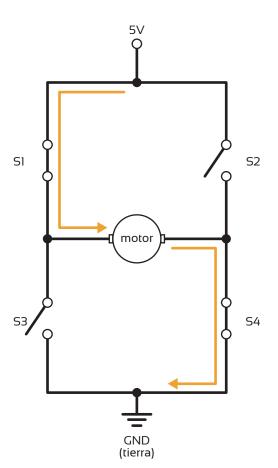


Puente H

El puente H es un circuito sumamente sencillo para el control de giro en motores DC y está compuesto principalmente por cuatro interruptores, los cuales se accionan de dos en dos.

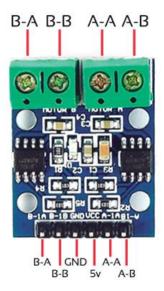
En la primera imagen activamos los interruptores 1 y 4, así el positivo le llega al motor por la borna derecha provocando que gire en sentido de las agujas del reloj. En la segunda imagen, desactivamos los interruptores 1 y 4 y activamos los interruptores 2 y 3, de esta forma el positivo le llega al motor por la borna izquierda y el motor gira en sentido antihorario.











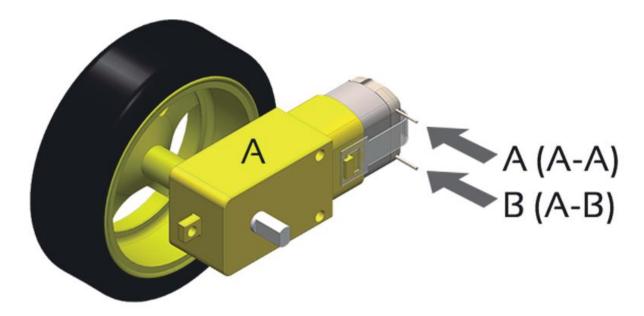
En nuestro caso, el kit de BOTS viene con un **puente H dual** ya que permite controlar los dos motores al mismo tiempo.

En el puente H dual las cuatro entradas superiores (los pines hembra B-A / B-B / A-A / A-B) son para el ingreso de las terminales de los dos motores.

Por su parte, los pines machos conectan el puente H con el Arduino, conectando las cuatro entradas de los motores, la alimentación y la tirada a tierra (GND).

Entonces, con ayuda del puente H, el funcionamiento de los motores se vuelve muy simple. Por ejemplo, tomemos de referencia el motor A:

- Si alimentamos la entrada A-A el motor gira en un sentido.
- Si alimentamos A-B gira en el sentido opuesto.
- Si alimentamos ambas entradas generamos un cortocircuito intencional que provoca el freno del giro.



Lo mismo ocurriría con el motor B. Si alimentamos la entrada B-A el motor girará en un sentido, si alimentamos B-B girará en el sentido opuesto, y si alimentos ambos a la vez el cortocircuito provocará que se detenga de inmediato.





Recomendación

Si tu robot se mueve conectado con el cable USB, pero con la batería no, es un problema de amperaje. Puede ser que la batería ya esté medio gastada o no sea de buena calidad. Te recomendamos Duracell.

Otra opción es que pruebes con un power bank (Cargador portátil de celular). Si no tenés, hay algunos modelos pequeños que cuestan más o menos lo mismo que una batería, pero son recargables, la carga dura mucho más, y tienen mucho mejor amperaje.



Una posible solución por software es poner un delay(50); entre el arranque de ambos motores en cada función de movimiento. El momento en el que un motor debe vencer la inercia para comenzar a girar es el de mayor consumo, por lo que defasando un poco el arranque del segundo motor, logramos reducir la exigencia sobre la placa arduino. Las funciones deberían quedarte algo así:

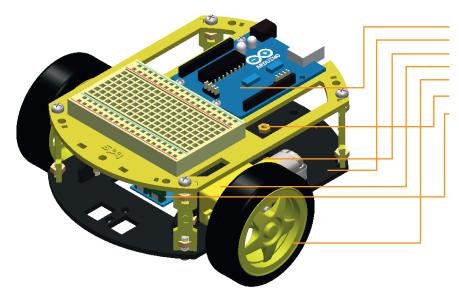
```
void adelante (){
  analogWrite(AIA,255);
  analogWrite(AIB,O);
  delay(50);
  analogWrite(BIA,255);
  analogWrite(BIB,O);
}
```





Ensamble

A continuación veremos la estructura y componentes necesarios para hacer nuestra primer prueba de motores DC.

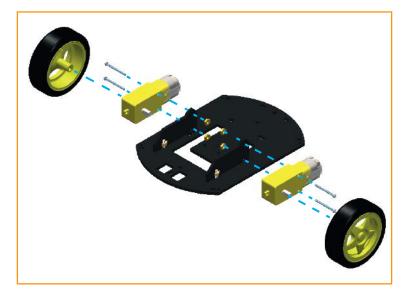


Materiales:

- -Estructura de Hola Mundo
- -Chasis B
- -Piezas H x 2 unidades
- -2 motores DC
- -2 ruedas
- -Ruedo multidireccional
- -Puente H

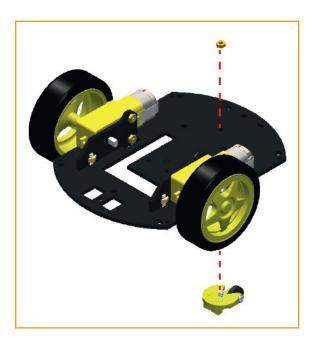


Paso 1: Abulonar las piezas H al chasis B





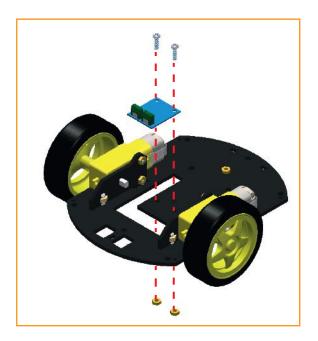




Paso 3: Tomar la rueda multidireccional armada, y ubicarla en el chasis B

Paso 4: Con 2 bulones 3x12, ubicar el Puente H en las perforaciones del chasis B, con cuidado de no apretar mucho la placa para no dañar el circuito del controlador de los motores.





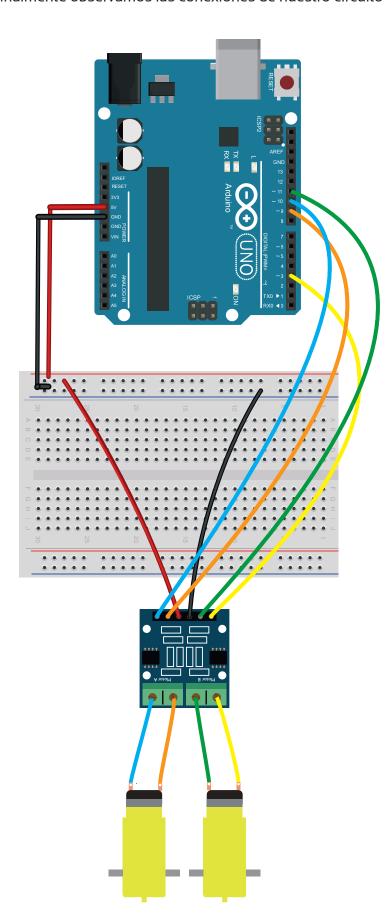
Paso 5:Unir toda la estructura del chasis B con el ensamble de "Hola mundo".





Circuito electrónico

Finalmente observamos las conexiones de nuestro circuito electrónico.

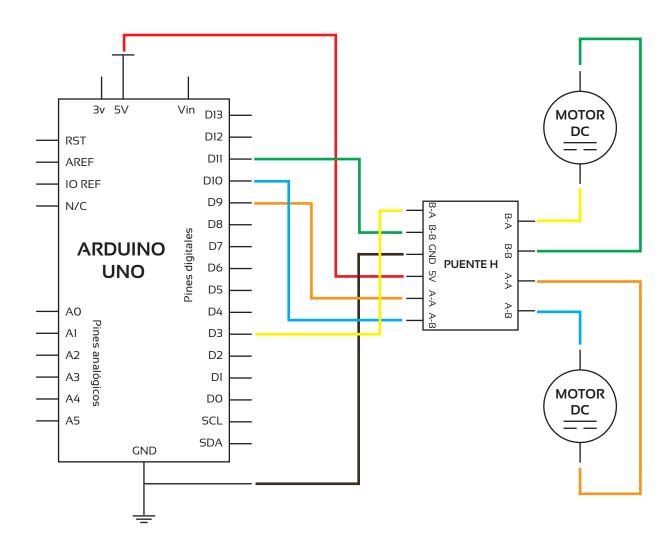


Componentes electrónicos:

- -Puente H
- -2 motores DC
- -2 cables M-M
- -10 cables M-H



Circuito electrónico en símbolos





Código de programación

Desarrollaremos el código de programación para realizar distintos tipos de movimiento con nuestros motores.

```
//Definimos los pines que se van a conectar al
                                                motor lo hacemos con "const" para definir una
                                                constante, es decir no va a cambiar.
 1 const int AIA = 3;
                                                //pin 3 connectado al pin A-IA
 2 const int AIB = 11;
                                                //pin 11 connectado al pin A-IA
 3 const int BIA = 9;
                                                //pin 9 connectado al pin A-IA
 4 const int BIB = 10;
                                                //pin 10 connectado al pin A-IA
 5 const int ledTest = 13;
 6
 7
    void setup() {
     pinMode(AIA, OUTPUT);
 8
 9
     pinMode(AIB, OUTPUT);
10
     pinMode(BIA, OUTPUT);
                                                //Configuramos las entradas y salidas
 11
     pinMode(BIB, OUTPUT);
12
     pinMode(ledTest, OUTPUT);
13
14
     digitalWrite(AIA, LOW);
15
     digitalWrite(AIB, LOW);
16
     digitalWrite(BIA, LOW);
17
     digitalWrite(BIB, LOW);
18
     digitalWrite(ledTest, LOW);
19
20
     digitalWrite(ledTest, HIGH);
21
     delay(500);
22
     digitalWrite(ledTest, LOW);
                                                //Saludo inicial
23
     delay(500);
24
     digitalWrite(ledTest, HIGH);
25
     delay(500);
     digitalWrite(ledTest, LOW);
26
27
     delay(500);
     digitalWrite(ledTest, HIGH);
28
29
     delay(500);
     digitalWrite(ledTest, LOW);
30
31
     delay(500);
32 }
33 void loop() {
                                                //nombramos la funciónes "parpadeo", "adelante",
34
     parpadeo(1);
                                                "atras", "derecha", "izquierda" y "parar" que creare-
35
     adelante();
                                                mos más adelante
36
     delay(1500);
37
     parar();
38
     parpadeo(2);
39
40
     atras();
41
     delay(1500);
42
     parar();
43
44
     parpadeo(3);
45
     derecha();
46
     delay(1000);
47
     parar();
48
49
     parpadeo(4);
50
     izquierda();
     delay(1000);
51
52
     parar();
53 }
```



```
54
      void adelante()
                                                 //colocamos en alto los IA y en bajo los IB de los
55 {
                                                 dos motores - definimos "adelante".
56
      digitalWrite(AIA, HIGH);
 57
      digitalWrite(AIB, LOW);
58
      digitalWrite(BIA, HIGH);
59
      digitalWrite(BIB, LOW);
60 }
 61
62 void atras()
                                                 //colocamos en alto los IB y en bajo los IA de los
                                                 dos motores - definimos "atras".
63 {
64
      digitalWrite(AIA, LOW);
65
      digitalWrite(AIB, HIGH);
66
      digitalWrite(BIA, LOW);
 67
      digitalWrite(BIB, HIGH);
68 }
69
70 void derecha()
                                                 //Combinamos adelante y atras, para hacer que las
 71 {
                                                 ruedas giren invertidas - definimos asi "derecha" e
 72
      digitalWrite(AIA, HIGH);
                                                 "izquierda".
 73
      digitalWrite(AIB, LOW);
 74
      digitalWrite(BIA, LOW);
 75
      digitalWrite(BIB, HIGH);
 76 }
 77
 78 void izquierda()
79 {
80
      digitalWrite(AIA, LOW);
      digitalWrite(AIB, HIGH);
 81
 82
      digitalWrite(BIA, HIGH);
83
      digitalWrite(BIB, LOW);
84 }
85
86 void parar()
                                                 //Colocamos los dos valores en alto y por corto
87 {
                                                 circuito frena - definimos "parar".
88
      digitalWrite(AIA, HIGH);
89
      digitalWrite(AIB, HIGH);
90
      digitalWrite(BIA, HIGH);
 91
      digitalWrite(BIB, HIGH);
 92
      delay(*500);
93 }
94
95 void parpadeo(int veces) {
                                                 //Definimos "parpadeo"
96
     for (int j = 0; j < veces; j++) {
97
       digitalWrite(ledTest, HIGH);
98
       delay(200);
99
       digitalWrite(ledTest, LOW);
100
       delay(200);
101
     }
102 }
```

ACTIVIDADES Y EJERCITACIÓN



Cap. 8: Actividades complementarias

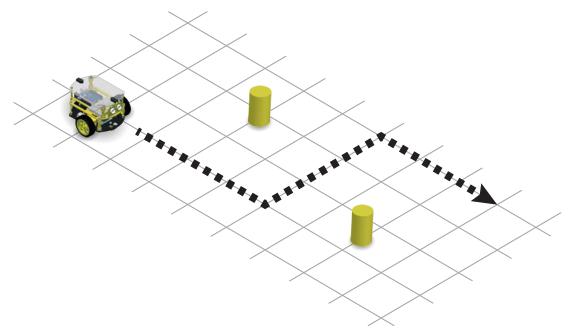
En base a los contenidos vistos en el "Capítulo 8: Iniciando motores" realiza las siguientes actividades complementarias.

Actividad 1

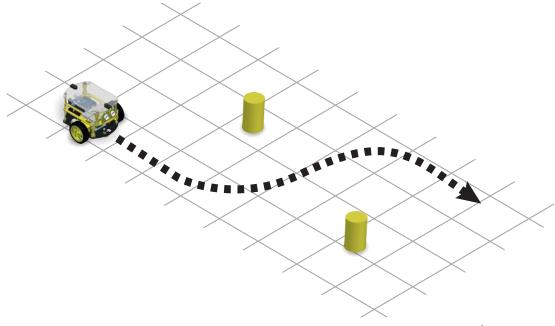
Desafío práctico:

Pon obstáculos en el piso y logra que el robot los esquive haciendo zigzag entre ellos. Para ello necesitarás medir el tiempo que precisa el motor para ir de un punto A a uno B, y hacer reglas de tres para calcular las distancias a programar. ¡Importante! Recuerda que de esta forma siempre debe usarse los motores a la misma potencia.

a. Intenta hacer el ejercicio con líneas y ángulos rectos.



b. Una vez que hayas podido programar el zigzagueo con ángulos rectos, inténtalo ahora con líneas curvas usando diferentes potencias para cada motor.







Cap. 8: Actividades complementarias

Actividad 1

a. Giro en ángulos rectos.

```
const int AIA = 3;
const int AIB = 11;
const int BIA = 9;
const int BIB = 10;
const int ledTest = 13;
void setup() {
 pinMode(AIA, OUTPUT);
 pinMode(AIB, OUTPUT);
 pinMode(BIA, OUTPUT);
 pinMode(BIB, OUTPUT);
 pinMode(ledTest, OUTPUT);
 analogWrite(AIA, O);
 analogWrite(AIB, O);
 analogWrite(BIA, O);
 analogWrite(BIB, O);
 digitalWrite(ledTest, LOW);
 digitalWrite(ledTest, HIGH);
 delay(500);
 digitalWrite(ledTest, LOW);
 delay(500);
 digitalWrite(ledTest, HIGH);
 delay(500);
 digitalWrite(ledTest, LOW);
 delay(500);
 digitalWrite(ledTest, HIGH);
 delay(500);
 digitalWrite(ledTest, LOW);
 delay(500);
}
void loop() {
 adelante();
 delay(2000);
 izquierda();
 delay(550);
 adelante();
 delay(2000);
 derecha();
 delay(550);
 adelante();
 delay(2000);
}
```





```
void adelante() {
 analogWrite(AIA, 255);
 analogWrite(AIB, O);
 analogWrite(BIA, 255);
 analogWrite(BIB, O);
}
void derecha() {
 analogWrite(AIA, 255);
 analogWrite(AIB, O);
 analogWrite(BIA, O);
 analogWrite(BIB, 255);
}
void izquierda() {
 analogWrite(AIA, O);
 analogWrite(AIB, 255);
 analogWrite(BIA, 255);
 analogWrite(BIB, O);
b. Giro en líneas curvas.
const int AIA = 3;
const int AIB = 11;
const int BIA = 9;
const int BIB = 10;
const int ledTest = 13;
void setup() {
 pinMode(AIA, OUTPUT);
 pinMode(AIB, OUTPUT);
 pinMode(BIA, OUTPUT);
 pinMode(BIB, OUTPUT);
 pinMode(ledTest, OUTPUT);
 analogWrite(AIA, O);
 analogWrite(AIB, O);
 analogWrite(BIA, O);
 analogWrite(BIB, O);
 digitalWrite(ledTest, LOW);
 //Saludo Inicial
 digitalWrite(ledTest, HIGH);
 delay(500);
 digitalWrite(ledTest, LOW);
 delay(500);
 digitalWrite(ledTest, HIGH);
 delay(500);
 digitalWrite(ledTest, LOW);
 delay(500);
 digitalWrite(ledTest, HIGH);
 delay(500);
 digitalWrite(ledTest, LOW);
 delay(500);
}
```





```
void loop() {
    izquierda();
    delay(3000);
    derecha();
    delay(3000);
}

void derecha() {
    analogWrite(AIA, 255); // los pines que activamos son los mismos que para ir hacia adelante,
    analogWrite(AIB, 0); // solo que con diferente potencia cada uno.
    analogWrite(BIA, 80);
    analogWrite(BIB, 0);
}

void izquierda() {
    analogWrite(AIA, 80);
    analogWrite(AIA, 80);
    analogWrite(BIB, 0);
    analogWrite(BIB, 0);
    analogWrite(BIB, 0);
}
```