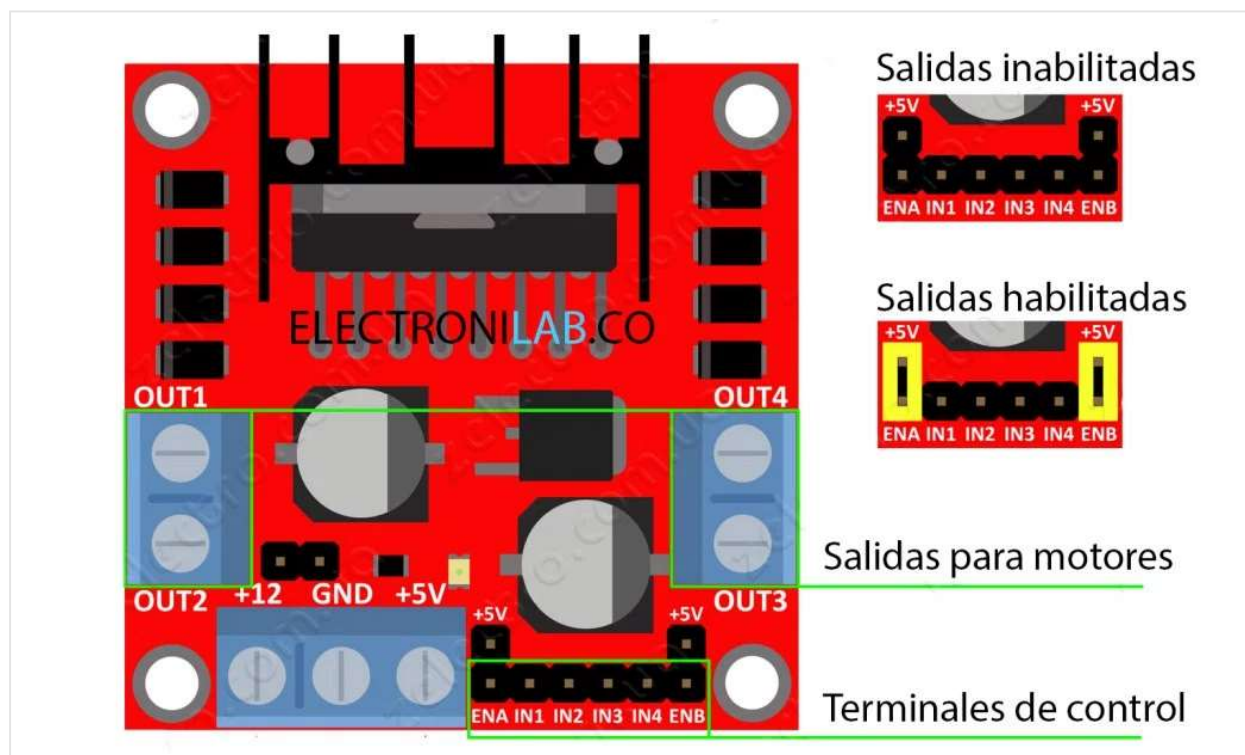


Tutorial: Uso de Driver L298N para motores DC y paso a paso con Arduino

El siguiente tutorial esta basado en el [Driver dual para motores \(Full-Bridge\) – L298N](#), ofrecido por ELECTRONILAB.CO. Puedes adquirir este módulo en nuestra [tienda](#).

Este módulo basado en el chip L298N te permite controlar dos motores de corriente continua o un motor paso a paso bipolar de hasta 2 amperios.

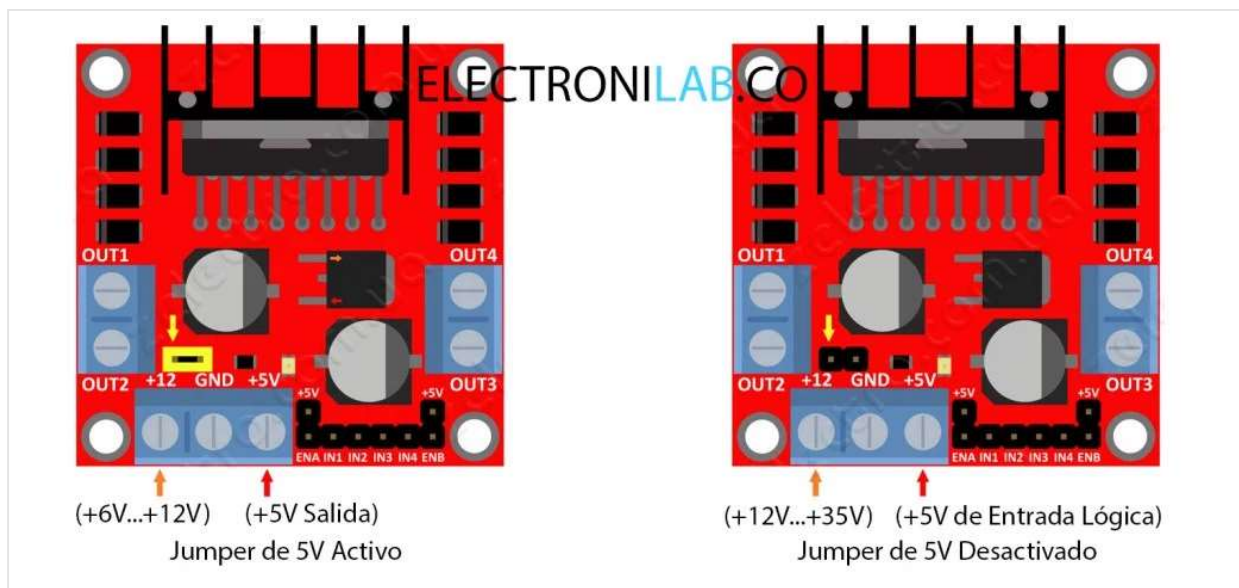
El módulo cuenta con todos los componentes necesarios para funcionar sin necesidad de elementos adicionales, entre ellos diodos de protección y un regulador **LM7805** que suministra 5V a la parte lógica del integrado L298N. Cuenta con jumpers de selección para habilitar cada una de las salidas del módulo (A y B). La **salida A** esta conformada por **OUT1** y **OUT2** y la **salida B** por **OUT3** y **OUT4**. Los pines de habilitación son **ENA** y **ENB** respectivamente.



En la parte inferior se encuentran los pines de control del módulo, marcados como **IN1**, **IN2**, **IN3** e **IN4**.

Conexión de alimentación

Este módulo se puede alimentar de 2 maneras gracias al regulador integrado LM7805.



Cuando el jumper de selección de 5V se encuentra **activo**, el módulo permite una alimentación de entre **6V a 12V DC**. Como el regulador se encuentra activo, el pin marcado como +5V tendrá un voltaje de 5V DC. Este voltaje se puede usar para alimentar la parte de control del módulo ya sea un microcontrolador o un Arduino, pero recomendamos que el consumo no sea mayor a 500 mA.

Cuando el jumper de selección de 5V se encuentra **inactivo**, el módulo permite una alimentación de entre **12V a 35V DC**. Como el regulador no está funcionando, tendremos que conectar el pin de +5V a una tensión de 5V para alimentar la parte lógica del L298N. Usualmente esta tensión es la misma de la parte de control, ya sea un microcontrolador o Arduino.

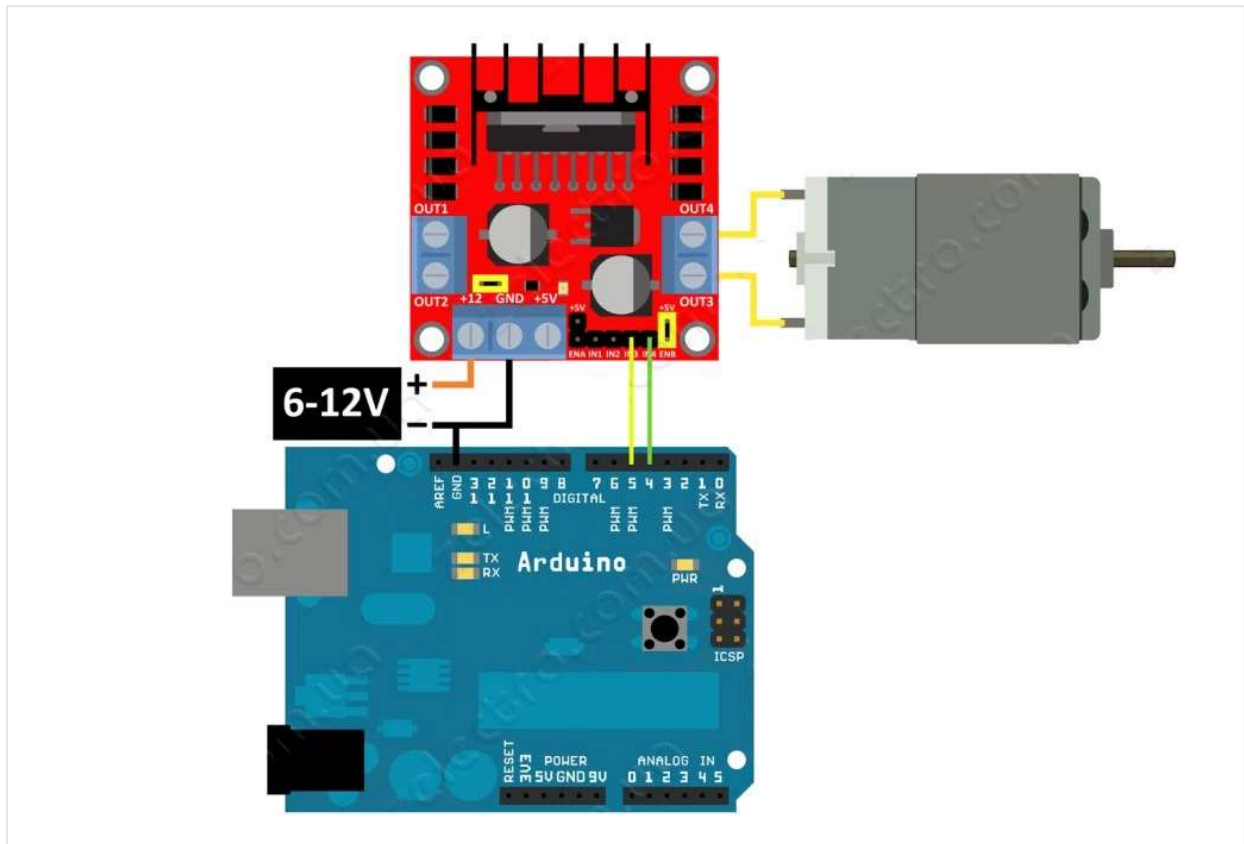
Recomendamos nunca conectar una tensión de entrada al pin de +5V, cuando el jumper de selección de 5V se encuentre activado. Esto provocaría un corto y podría dañar permanentemente el módulo.

Control de un motor DC

Como demostración, vamos a controlar un motor DC a través de la salida B del módulo. El pin **ENB** se conectará con el jumper a +5V.

El ejemplo está desarrollado en **Arduino UNO**, pero el código es compatible con cualquier **Arduino** o pinguino.

Esquema de conexión



Código en Arduino

El programa básicamente activa el motor en un sentido por 4 segundos, luego detiene el motor por 0.5 segundos, después activa el motor en sentido inverso por 4 segundos y por último detiene el motor por 5 segundos. Luego repite la acción indefinidamente.

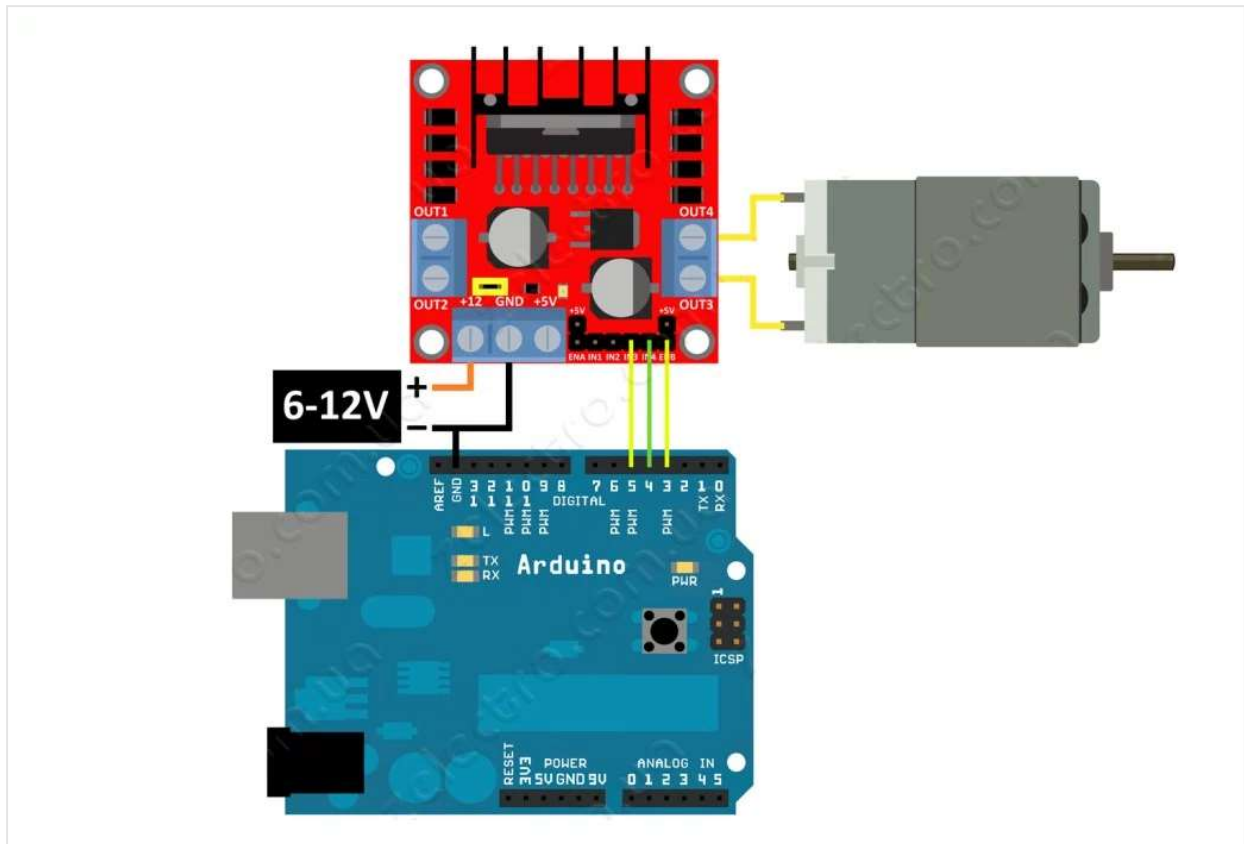
```
1  /*
2   Ejemplo de control de motor DC usando modulo L298
3   http://electronilab.co/tienda/driver-dual-para-motores-full-bridge-l298n/
4
5   El programa activa el motor en un sentido por 4 segundos,
6   para el motor por 500 ms, activa el motor en sentido inverso por 4 segundos
7   y se detiene por 5 segundos. Luego repite la acción indefinidamente.
8
9   Creado 16/05/14
10  por Andres Cruz
11  ELECTRONILAB.CO
12  */
13
14  int IN3 = 5;
15  int IN4 = 4;
16
17  void setup()
18  {
19    pinMode (IN4, OUTPUT); // Input4 conectada al pin 4
20    pinMode (IN3, OUTPUT); // Input3 conectada al pin 5
```

```
21 }
22 void loop()
23 {
24     // Motor gira en un sentido
25     digitalWrite (IN4, HIGH);
26     digitalWrite (IN3, LOW);
27     delay(4000);
28     // Motor no gira
29     digitalWrite (IN4, LOW);
30     delay(500);
31     // Motor gira en sentido inverso
32     digitalWrite (IN3, HIGH);
33     delay(4000);
34     // Motor no gira
35     digitalWrite (IN3, LOW);
36     delay(5000);
37 }
38
```

Control de un motor DC variando su velocidad

Si queremos controlar la velocidad del motor, tenemos que hacer uso de PWM. Este PWM será aplicado a los pines de activación de cada salida o pines ENA y ENB respectivamente, por tanto los jumper de selección no serán usados.

Esquema de conexión



Código en Arduino

El programa controla la velocidad de un motor DC aplicando PWM al pin ENB del módulo L298N.

```

1  /*
2  Ejemplo de control de motor DC usando modulo L298
3  http://electronilab.co/tienda/driver-dual-para-motores-full-bridge-1298n/
4
5  Creado 16/05/14
6  por Andres Cruz
7  ELECTRONILAB.CO
8  */
9
10 int IN3 = 5;    // Input3 conectada al pin 5
11 int IN4 = 4;    // Input4 conectada al pin 4
12 int ENB = 3;    // ENB conectada al pin 3 de Arduino
13 void setup()
14 {
15   pinMode (ENB, OUTPUT);
16   pinMode (IN3, OUTPUT);
17   pinMode (IN4, OUTPUT);
18 }
19 void loop()
20 {
21   //Preparamos la salida para que el motor gire en un sentido
22   digitalWrite (IN3, HIGH);

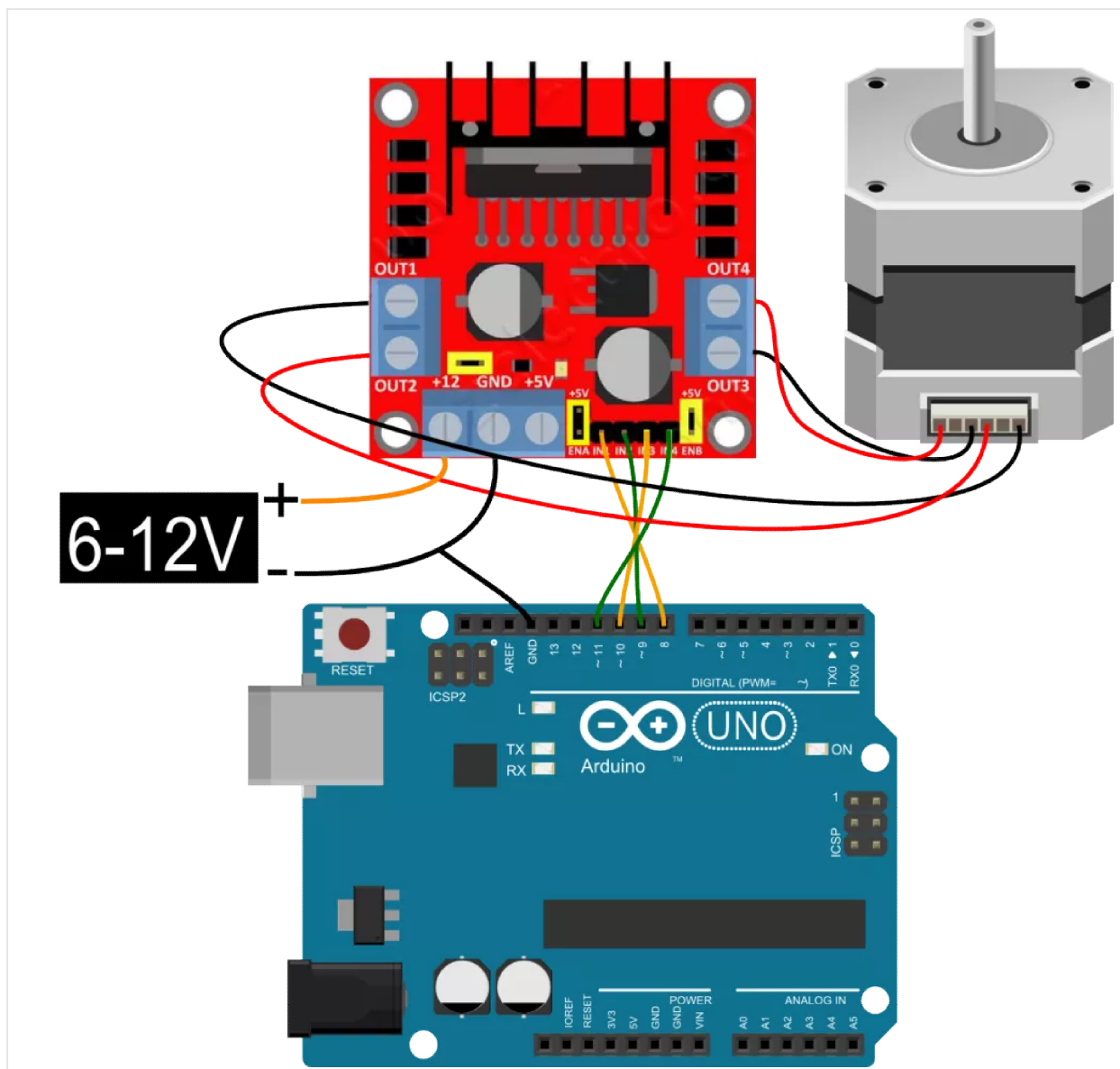
```

```
23 digitalWrite (IN4, LOW);
24 // Aplicamos PWM al pin ENB, haciendo girar el motor, cada 2 seg aumenta la velocidad
25 analogWrite(ENB,55);
26 delay(2000);
27 analogWrite(ENB,105);
28 delay(2000);
29 analogWrite(ENB,255);
30 delay(2000);
31 // Apagamos el motor y esperamos 5 seg
32 analogWrite(ENB,0);
33 delay(5000);
34 }
```

Control de un motor paso a paso bipolar

Los motores paso a paso pueden ser bipolares o unipolares. En este ejemplo trabajamos con un motor paso a paso bipolar, sin embargo este driver es capaz de controlar un motor paso a paso unipolar si se conecta en configuración bipolar, obviando los cables del *tab central* de las bobinas del motor. El máximo consumo permitido es de 2 amperios.

Esquema de conexión



El esquema de conexión muestra la conexión utilizada entre el módulo L298N y el motor paso a paso. Cada una de las bobinas del motor está conectada a una salida del módulo. Para identificar las bobinas de un motor paso a paso utilizamos un multímetro en modo de continuidad. Los cables que dan continuidad son los extremos de cada bobina.

En este caso, como el motor paso a paso es de 12 VDC, utilizamos el jumper de selección de +5V, para activar el regulador interno del módulo y solo hacer uso de una fuente de 12 VDC para alimentar el motor.

Los jumper de activación ENA y ENB los hemos activado de igual manera.

Código de Arduino

El código de Arduino hace girar el motor paso a paso una vuelta en un sentido y luego ejecuta otra vuelta en sentido opuesto. Este código hace uso de la librería '*Stepper.h*', que se instala por defecto en las últimas versiones del IDE de Arduino.

El valor de la variable **stepsPerRevolution** depende del número de pasos del motor paso a paso. Este valor se encuentra en las especificaciones de la hoja de datos del motor. En nuestro caso el motor paso a paso utilizado es de 48 pasos/vuelta.

```
1  /*
2  Stepper Motor Control - one revolution
3
4  Este programa impulsa un motor paso a paso unipolar o bipolar.
5  El motor está conectado a los pines digitales 8 - 11 de la Arduino.
6
7  El motor debe girar una vuelta en una dirección, a continuación,
8  una revolución en la otra dirección.
9
10 Created 11 Mar. 2007
11 by Tom Igoe
12
13 Modificado
14 16/05/14
15 por Andres Cruz
16 ELECTRONILAB.CO
17 */
18
19 #include <Stepper.h>
20
21 const int stepsPerRevolution = 48; // cambie este valor por el numero de pasos de su motor
22
23 // inicializa la libreria 'stepper' en los pines 8 a 11
24 Stepper myStepper(stepsPerRevolution, 8,9,10,11);
25
26 void setup() {
27     // establece la velocidad en 60rpm
28     myStepper.setSpeed(60);
29     // inicializa el puerto serial
30     Serial.begin(9600);
31 }
32
33 void loop() {
34     // gira una vuelta en una direccion
35     Serial.println("clockwise");
36     myStepper.step(stepsPerRevolution);
37     delay(500);
38
39     // gira otra vuelta en la otra direccion
40     Serial.println("counterclockwise");
41     myStepper.step(-stepsPerRevolution);
42     delay(500);
43 }
```