

# Taller Introducción a la Ingeniería Eléctrica

## Robot basado en Arduino

Instituto de Ingeniería Eléctrica

Taller 5: Actuadores

18 de Abril de 2024

## 1 Actuadores

- Motor CC -corriente continua- (*DC Motor -Direct current-*)
  - Controlador de motores (*Driver*)
- Motor Servo
  - Funcionamiento
  - Uso

## 2 Ejercicios

- Ejercicio 1
- Ejercicio 2
- Ejercicio 3 - opcional

## 1 Actuadores

- Motor CC -corriente continua- (*DC Motor -Direct current-*)
  - Controlador de motores (*Driver*)
- Motor Servo
  - Funcionamiento
  - Uso

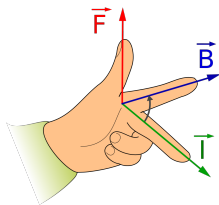
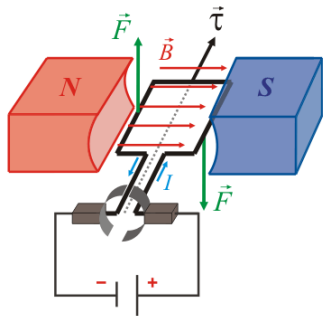
## 2 Ejercicios

- Ejercicio 1
- Ejercicio 2
- Ejercicio 3 - opcional

# DC Motor: Principio de Funcionamiento

Ley de Lorentz:

- Fuerza de giro proporcional a la corriente.
- Sentido de giro controlado por el sentido de la corriente.



<https://gfycat.com/idealistic hospitable harp seal>

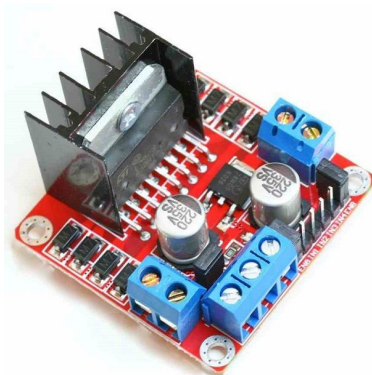
<https://gfycat.com/absolutely indignant glass fish>

<https://gfycat.com/lavish scented degu>

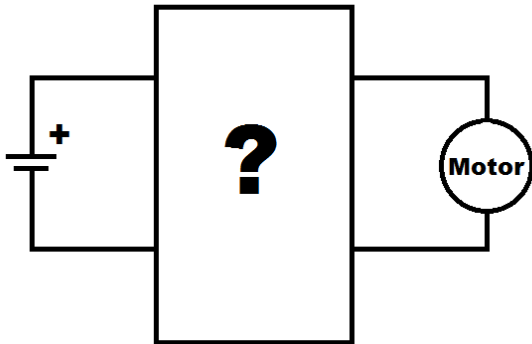
# Controlador de motores (*Driver*)

## ¿Por qué se necesita?

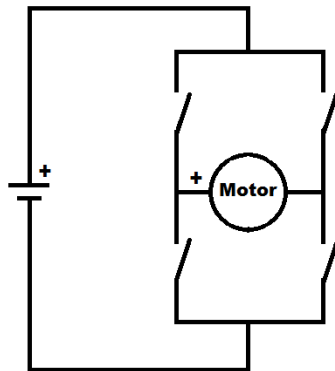
- Brinda la potencia necesaria para hacer girar los motores a la velocidad deseada.
- Facilita el manejo del sentido de giro de los motores.



¿Cómo invertir el giro?

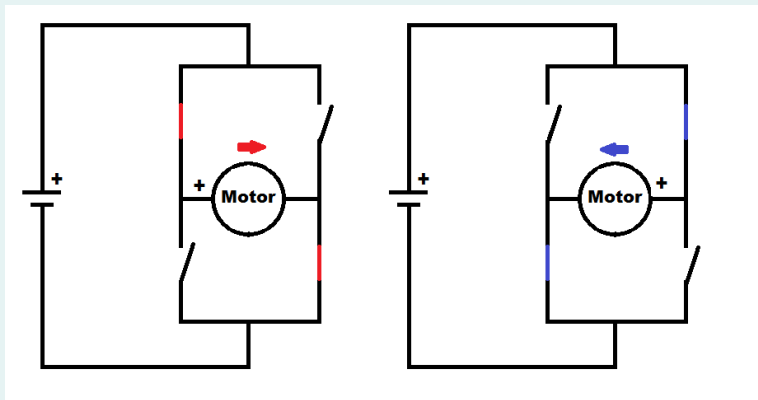


## Puente H



# Controlador de motores (Driver)

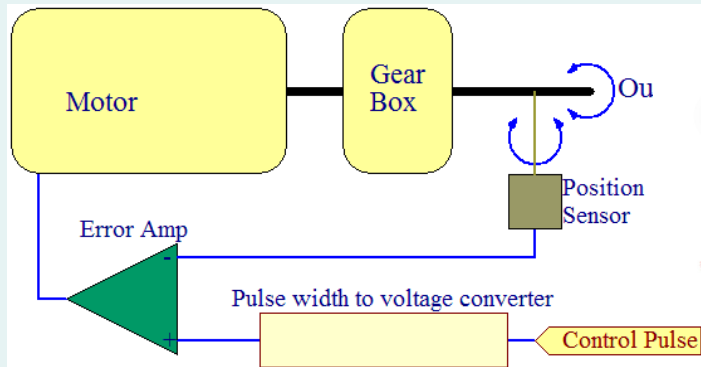
## Puente H



Variando el valor medio de la tensión de alimentación es posible variar la velocidad.



## Principio de funcionamiento



## ¿Cómo programamos al micro para mover el servo?

```
#include <Servo.h> // Incluye Librería del servo

#define SERVO_PIN 2 // Pin donde se conecta servo

Servo nuestroServo; // Crea objeto nuestroServo de tipo Servo

void setup(){
  nuestroServo.attach(SERVO_PIN); // Toma al SERVO_PIN como el
                                  // pin donde esta conectado el servo
}

void loop(){
  nuestroServo.write(90); // Coloca el servo en posición (entre 0 y 180 grados)
}
```

## 1 Actuadores

- Motor CC -corriente continua- (*DC Motor -Direct current-*)
  - Controlador de motores (*Driver*)
- Motor Servo
  - Funcionamiento
  - Uso

## 2 Ejercicios

- Ejercicio 1
- Ejercicio 2
- Ejercicio 3 - opcional

# Ejercicio 1

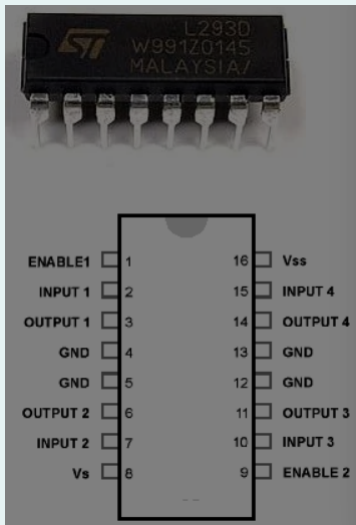
Utilizando un *DC Motor*, el *Driver* L293D y un *pack* de baterías se debe generar un programa que mediante la lectura de la posición de un botón controle el sentido de giro (horario / antihorario) del motor. Además, mediante un potenciómetro debe controlar la velocidad del motor.

Se pide:

- se deberá hacer uso del controlador de motores L293D presente en el simulador. Para lo cual, será necesario usar 3 pines por motor:
  - 2 para controlar el sentido de giro.
  - 1 para el control de la velocidad. Elegir un pin digital con salida PWM para así tener varios niveles de velocidad.
- además, el circuito integrado 293D necesita dos alimentaciones positivas: una desde los 5V que provee el Arduino y otra externa. Para esta última instancie una batería de 9V.
- luego de armar el hardware (ver guía de conexiones -13-) y de escribir el código, simular hasta obtener los resultados esperados.  
(Nota: recordar la posibilidad de crear **FUNCIONES!!!**)

# Ejercicio 1

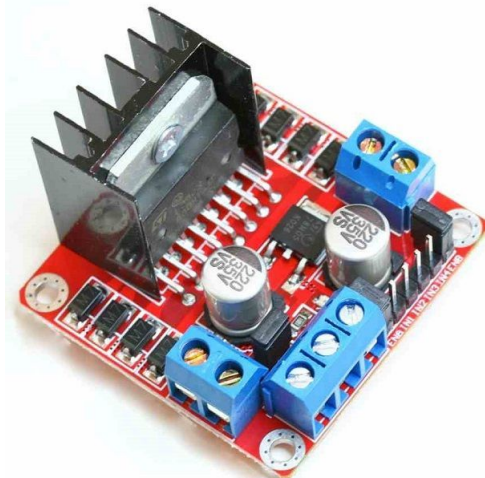
## L293D PinOut - Guía de conexiones



#PIN	ARDUINO	EXTERIOR
1	PWM D	
2	PIN DD	
3		MOTOR D +
4	GND	
5	GND	
6		MOTOR D -
7	PIN DI	
8		9V
9	PWM I	
10	PIN ID	
11		MOTOR I +
12	GND	
13	GND	
14		MOTOR I -
15	PIN II	
16	5V	

## Ejercicio 2

Una vez se hayan logrado los resultados esperados del Ejercicio 1 en el simulador, armar el sistema en la realidad y comprobar el correcto funcionamiento con los materiales entregados por los docentes (baterías, zócalo, Driver basado en el circuito integrado L298).



## Ejercicio 3 - opcional

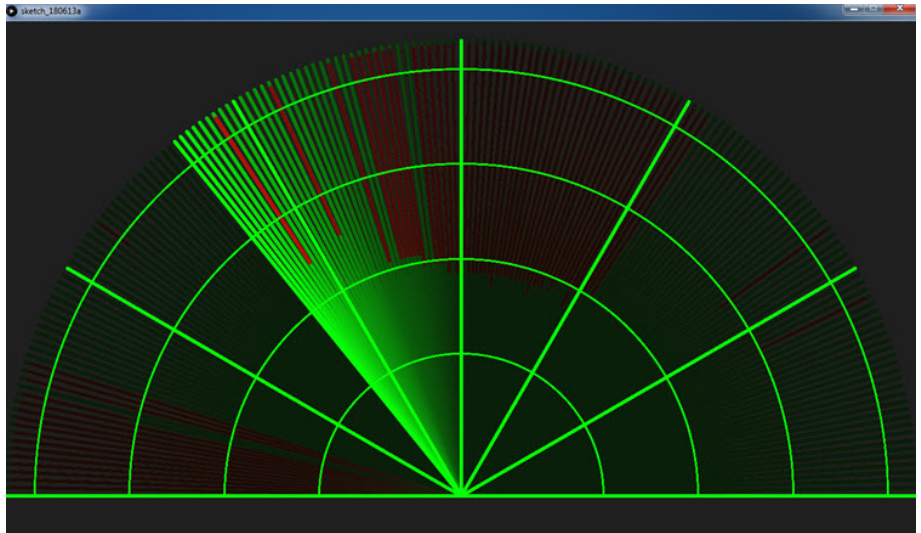
Obtener 10 medidas de distancia a través del sensor de ultrasonido. Las mismas deben ser tomadas con el sensor en una posición nueva cada vez. Logrando así cubrir mayor superficie a cubrir.

Para mover dicho sensor se lo puede colocar sobre un servomotor y a éste hacerlo girar 10 ángulos distintos tratando de cubrir un semiplano.

Se pide:

Construir un sonar desplegando data en el PC mediante el *IDE processing*.  
<https://playground.arduino.cc/Interfacing/Processing/>

# Sonar con processing





# Resumen para la próxima clase:

- 1 Terminar los ejercicios de clase. Traer uno preparado para mostrar en la siguiente clase.
- 2 Por dudas utilizar el *Foro de consultas*. Asistir a la clase de consulta los jueves a la tarde.
- 3 Queda disponible un cuestionario sobre esta clase, que deberá ser completado en el sitio EVA. Lo deberá hacer cada estudiante individualmente!!