

ARDUMOTO SHIELD Y L298

Controlando motores de un cierto consumo con Arduino

Home ([Http://Www.Prometec.Net](http://www.prometec.net)) • Ardumoto Shield Y L298

OBJETIVOS

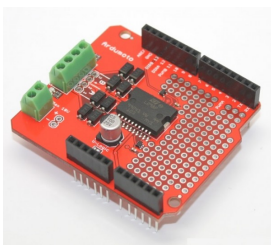
- ★ Presentar el chip **L298**, dual full bridge para control de motores.
- ★ Presentar el **Shield Ardumoto**.
- ★ Consumo de hasta 2 Amperios por cada motor.
- ★ Ver la forma de programar su uso.

MATERIAL REQUERIDO.



(<http://www.prometec.net/producto/arduino-uno/>)

Arduino Uno (<http://www.prometec.net/categoria-producto/arduinos/>) o equivalente.



(<http://www.prometec.net/producto/ardumoto-shield/>)

Un Shield Ardumoto
(<http://www.prometec.net/producto/ardumoto-shield/>)



(<http://www.prometec.net/wp-content/uploads/2015/03/wheel-2.png>)

Un par de ruedas y motores CC
(<http://www.prometec.net/producto/rueda-motor/>)

*Esta sesión es poco mas que una traducción libre de la página de **Sparkfun Ardumoto Hookup guide** (<https://learn.sparkfun.com/tutorials/ardumoto-shield-hookup-guide>)*

MÁS SOBRE MOTORES DE CC

Ya vimos en varias sesiones anteriores como manejar motores de corriente continua (CC) directamente mediante señales **PWM** y también como usar un sistema un poco más cómodo para manejar 4 motores mediante el **Motor Shield V1**.

Vimos que este **Motor Shield V1**, era cómodo y práctico para mover hasta 4 motores CC, y la pregunta es entonces *¿Por qué ver más chips / Shields de control de motores?*

La realidad es que el Motor Shield V1 es un juguete esplendido para iniciaros con el control de motores en Arduino y por muy poco dinero, pero enseguida encontrareis que este **Shield V1** está un poco limitado por la intensidad máxima que toleraba en su diseño, y que no podía sobrepasar 0,6 Amp, lo que no da para mucho en cualquier proyecto en el que tengáis que usar motores de continua con algo más de consumo.

En cuanto empecéis a buscar motores de una cierta potencia veréis que el consumo se dispara con rapidez impidiendo usar drivers sencillos como los que hemos visto hasta ahora.

Por eso el mercado, siempre atento a las oportunidades, nos brinda otras soluciones cuando la carga a mover crece, y una solución es utilizar drivers cuya capacidad de carga sea un poco superior.

No es difícil encontrar chips y Shields para **Arduino** que resuelven este problema, pero la documentación suele ser el primer defecto de muchas de estas soluciones (*Por su inexistencia*) y el precio puede crecer muy rápidamente sino se elige con cuidado.

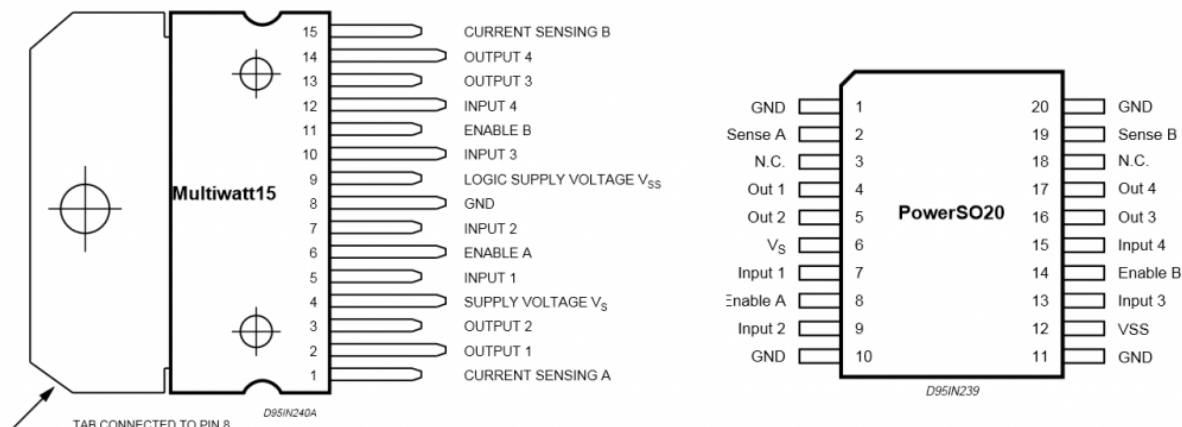
Por eso, entre las muchas soluciones disponibles, hemos optado por presentaros un chip típico para estas ocasiones, el **L298**, que es muy interesante porque nos permite gobernar dos motores de continua, con una carga de hasta 2 Amperios por cada uno, e incluye todo lo necesario para invertir la dirección de movimiento de estos y para controlar la velocidad de giro. Todo en un único chip.

Es muy frecuente encontrar este chip gobernando pequeños robots de dos ruedas y en esta sesión presentaremos este chip **L298** y también un **Shield** llamado **Ardumoto** que nos permitirá controlar con comodidad un par de motores de corriente continua de un modo conveniente sin necesidad de grandes desembolsos.

FULL-BRIDGE MOTOR DRIVER DUAL **L298**

Vamos a empezar viendo un poco las características del **L298**. Básicamente este chip es un controlador **doble full bridge**, es decir que permite manejar dos motores (*De corriente continua o de paso a paso*) y controlar tanto la velocidad como el sentido de giro de ambos motores.

Se presenta en dos versiones, en formato para soldar a una PCB (O protoboard) y en formato SMD, de soldadura en superficie que resulta un poco más complicado de soldar en casa:



(<http://www.prometec.net/wp-content/uploads/2015/09/L298N.png>)

Según su data sheet, puede entregar hasta 50V a los motores manteniendo 2Amp de carga y hasta 3Amp de pico (*Que mejor no sea sostenido, si os gusta vuestro chip*)

Symbol	Parameter	Value	Unit
V_S	Power Supply	50	V
V_{SS}	Logic Supply Voltage	7	V
V_i, V_{en}	Input and Enable Voltage	-0.3 to 7	V
I_O	Peak Output Current (each Channel) – Non Repetitive ($t = 100\mu s$) – Repetitive (80% on –20% off; $t_{on} = 10ms$) – DC Operation	3 2.5 2	A A A
V_{sens}	Sensing Voltage	-1 to 2.3	V
P_{tot}	Total Power Dissipation ($T_{case} = 75^\circ C$)	25	W
T_{op}	Junction Operating Temperature	-25 to 130	$^\circ C$
T_{stg}, T_j	Storage and Junction Temperature	-40 to 150	$^\circ C$

2 Amperios sostenidos es ya un consumo medio y con una cierta capacidad para desarrollar trabajo, por lo que este chip es muy apreciado en esos pequeños proyectos que tienen que mover por ejemplo un par de ruedas, o elevar un cierto peso. También se recomienda si tenéis que excitar solenoides o relés de un cierto tamaño cuyo consumo excede los mínimos que hemos visto hasta ahora.

Por eso os lo encontrareis una y otra vez en diferentes proyectos **Arduino**, porque es un chip competente, muy extendido y de precio moderado, y en muchas ocasiones se encuentra ubicado en un Shield como el que vamos a usar en esta sesión...



(<http://www.prometec.net>)

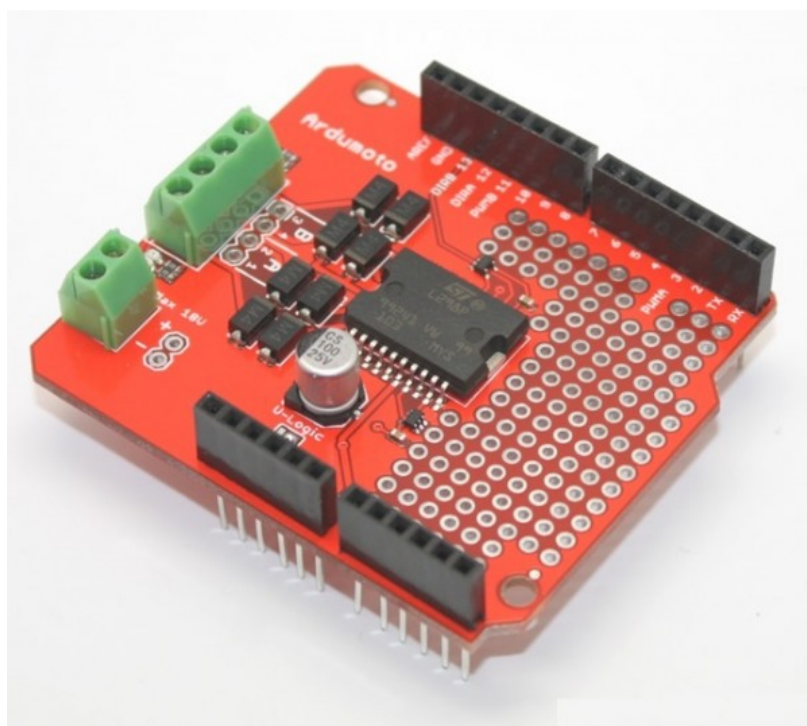
TIENDA (<http://www.prometec.net/tienda/>)

ARDUINO (<http://www.prometec.net/indice-tutoriales/>)

La manera más cómoda de usar este chip es mediante un pequeño **shield** que nos permita conectarlo a nuestros **Duinos**, sin demasiada complicación y el **Ardumoto Shield** es uno diseñado con esa intención. Originalmente diseñado por SparkFun, ha sido rápidamente copiado por los fabricantes chinos que ofrecen replicas directas del mismo.

FORO (<http://www.prometec.net/foros-prometec/>)

CONTACTO (<http://www.prometec.net/contact/>)

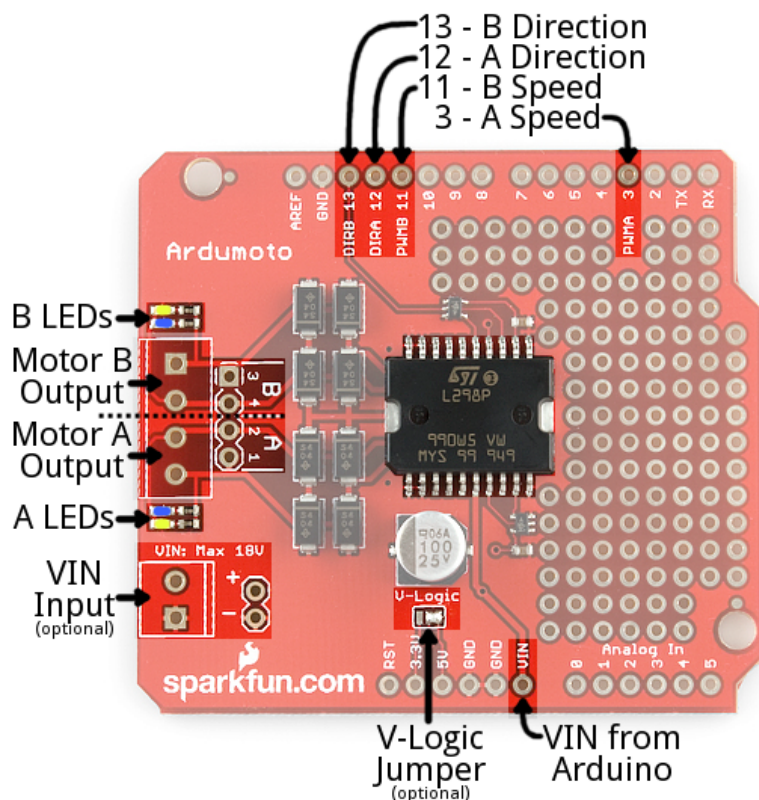


(<http://www.prometec.net/wp-content/uploads/2015/09/ardumoto-motor-surucu-shield.jpg>)

Incluye dos únicos bridge, lo que nos limita a dos motores CC o a un único **paso a paso**, pero a cambio tenemos todo el control mediante pines directos de **Arduino**.

También podéis apreciar que dispone de una pequeña área de pruebas y prototipos, donde podréis soldar componentes adicionales, como LEDS, Bluetooth o WIFI o cualquier otra cosa que se os pueda ocurrir, y que siempre viene bien cuando montas motores.

El shield utiliza 2 pines de Arduino para controlar cada motor. Uno regula la dirección de giro y el otro la velocidad mediante una señal **PWM**, lo que hacen un total de 4 pines dejando libres el resto.

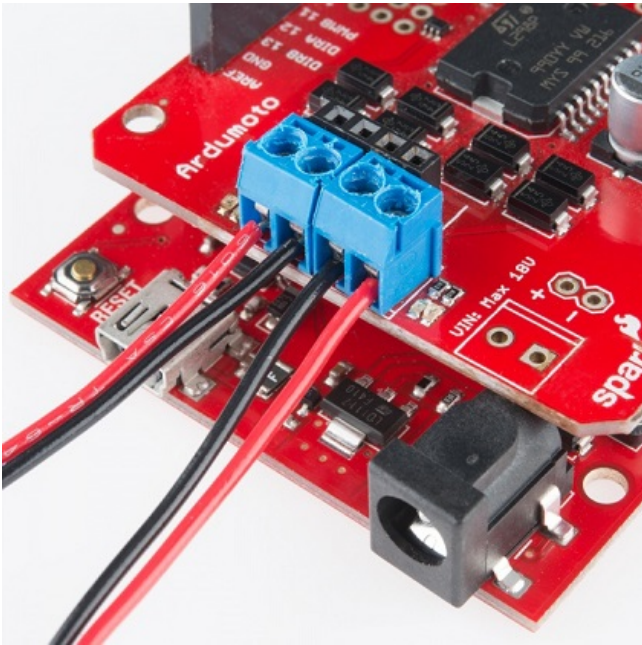


(<http://www.prometec.net/wp-content/uploads/2015/09/ardumoto-control.png>)

El **Ardumoto Shield** se reserva los pines 12 y 13 para controlar la velocidad, y los pines 3 y 11, para controlar el sentido de giro de los motores A y B respectivamente.

PIN ARDUINO	USO	NOTAS
3	PWM Motor A	Valores de 0 a 255
11	PWM Motor B	Valores de 0 a 255
12	Sentido Motor A	Es un Bool True / False
13	Sentido Motor B	Es un Bool True / False

A mano izquierda podéis ver las conexiones para los motores A y B, así como un conector para alimentar los motores independientemente de la alimentación de Arduino.



(http://www.prometec.net/wp-content/uploads/2015/09/Arдумoto_Shield_Hookup_Guide1.jpg)
No tiene demasiada importancia en qué orden conectáis los cables de los motores, siempre y cuando lo tengáis claro. Os recomiendo que os fijéis en los colores de los cables y conectéis ambos pares de igual modo para evitar sorpresas.
Fijares también, que a los laterales de donde conectamos los motores hay un par de LEDs de colores que indican el estado del motor y su sentido de giro (*En el video del final se aprecian bastante mejor*).

EL PROGRAMA DE CONTROL

El **shield Arдумoto** usa simplemente los pines de control que mostramos en la tabla más arriba, por lo que no requiere ninguna librería especial para mover los motores sino que usamos **Arduino** y sus pines para controlar el **L298N** y por tanto la velocidad y sentido de los motores conectados.
El programa que nos presenta SparkFun es de lo más sencillo y le vamos a usar como ejemplo de control. Empezamos con algunas definiciones para implicar el programa:

```

#define CW 0                //Sentido ClockWise
#define CCW 1               //Sentido Counter ClockWise

#define MOTOR_A 0
#define MOTOR_B 1

const byte PWMA = 3;        // PWM control (speed) for motor A
const byte PWMB = 11;       // PWM control (speed) for motor B
const byte DIRA = 12;       // Direction control for motor A
const byte DIRB = 13;       // Direction control for motor B

```

Y ahora vamos a definir un par de funciones sencillas de manejo. La primera la usaremos para arrancar uno de los motores con un sentido y velocidad dados:

```

void driveArдумoto(byte motor, byte dir, byte spd)
{
  if (motor == MOTOR_A)
  {
    digitalWrite(DIRA, dir);
    analogWrite(PWMA, spd);
  }
  else if (motor == MOTOR_B)
  {
    digitalWrite(DIRB, dir);
    analogWrite(PWMB, spd);
  }
}

```

Y la segunda es aún más sencilla, para parar el motor:

```

void stopArдумoto(byte motor)
{
  driveArдумoto(motor, 0, 0);
}

```

Y como tenemos que programar los pines, porque no tenemos una librería que nos haga el trabajo sucio, podemos definir una función que lo haga:

```

void setupArдумoto()
{
  pinMode(PWMA, OUTPUT);
  pinMode(PWMB, OUTPUT);
  pinMode(DIRA, OUTPUT);
  pinMode(DIRB, OUTPUT);

  digitalWrite(PWMA, LOW);
  digitalWrite(PWMB, LOW);
  digitalWrite(DIRA, LOW);
  digitalWrite(DIRB, LOW);
}

```

EL resto del programa tiene poco mérito. Básicamente vamos llamando a Arдумoto() para hacer girar los motores en diferente sentido y con distinta velocidad.

No pretende ser más que un pequeño ejemplo de como podéis realizar el control de los motores conectados al **Arдумoto Shield**. Prog_201 (http://www.prometec.net/wp-content/uploads/2015/09/Prog_201.rar)

```
void loop()
{ // Mueve solo el Motor A, a diferentes velocidades
  driveArdumoto(MOTOR_A, CCW, 255); // Motor A, avance a toda maquina
  delay(1000); // Motor A girará 1 segundo

  driveArdumoto(MOTOR_A, CW, 127); // Motor A, avance a media potencia
  delay(1000);

  stopArdumoto(MOTOR_A); // Parar Motor A

  // Mueve solo el Motor B a varias velocidades
  driveArdumoto(MOTOR_B, CCW, 255); // Motor B, retroceso a toda maquina
  delay(1000);
  driveArdumoto(MOTOR_B, CW, 127); // Motor B, avance media
  delay(1000);

  stopArdumoto(MOTOR_B); // Para Motor B

  // Ahora girar ambos motores
  driveArdumoto(MOTOR_A, CW, 255); // Motor A avance toda
  driveArdumoto(MOTOR_B, CW, 255); // Motor B avance toda
  delay(1000);

  // Ahora retroceso a media maquina
  driveArdumoto(MOTOR_A, CCW, 127); // Motor A at max speed.
  driveArdumoto(MOTOR_B, CCW, 127); // Motor B at max speed.
}
```

Y solo nos queda un pequeño video mostrando el resultado final:



RESUMEN DE LA SESIÓN

- ★ Hemos presentado el **L298**, un chip de doble bridge completo muy habitual en los montajes de pequeños robots móviles.
- ★ Hemos visto cómo usar el **Ardumoto Shield**.
- ★ Presentamos un pequeño ejemplo de programación
- ★ El **Ardumoto Shield** se puede usar para controlar dos motores CC o uno paso a paso.

Anterior
(<http://www.prometec.net/motorshield-servos/>)



(<https://www.facebook.com/prometecnet-1541207239447373/timeline/>)

Siguiente (<http://www.prometec.net/adafruit-motor-shield-v2/>)

Copyright Prometec (<http://www.prometec.net>)

