Aprendiendo Arduino

Aprendiendo a manejar Arduino en profundidad

ARCHIVO DE LA ETIQUETA: L298N

Motores

Como ya se ha visto anteriormente, un pin de Arduino solo puede tener los valores de 0 y 5 voltios y dar hasta 40 mA. Esto es insuficiente para mover un motor del tipo que sea, por lo tanto si queremos que Arduino maneje un motor, deberemos usar un driver.

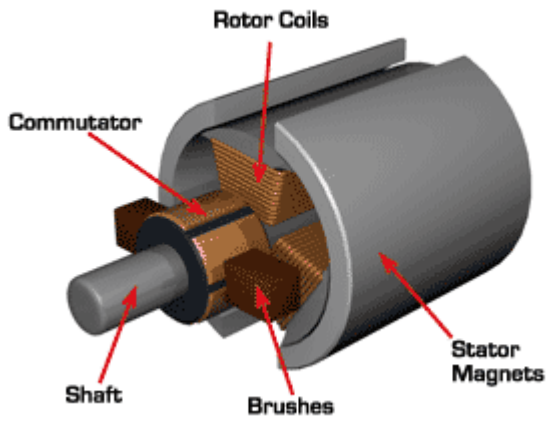
Un motor driver es un amplificador de corriente cuya función es tomar una pequeña señal de control de baja corriente y convertirla en una señal de alta corriente que pueda alimentar el motor.

Hay muchos tipos de motor drivers en función del motor a manejar, máximo voltaje, máxima corriente de salida, etc...

Más información: http://www.futureelectronics.com/en/drivers/motor-driver.aspx

Motor DC

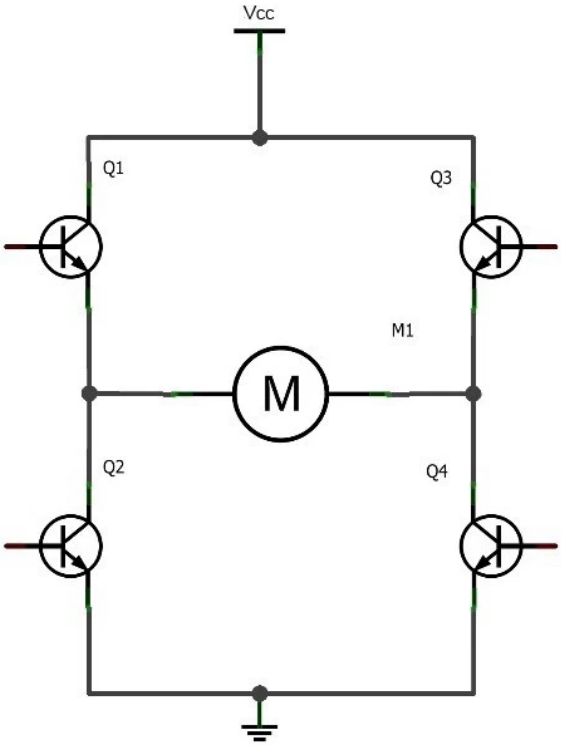
Un motor de corriente continua convierte la energía eléctrica en mecánica. Se compone de dos partes: el estator y el rotor. El estator es la parte mecánica del motor donde están los polos del imán. El rotor es la parte móvil del motor con devanado y un núcleo, al que llega la corriente a través de las escobillas. Si queremos cambiar el sentido de giro del rotor, tenemos que cambiar el sentido de la corriente que le proporcionamos al rotor, basta con invertir la polaridad de la pila o batería.

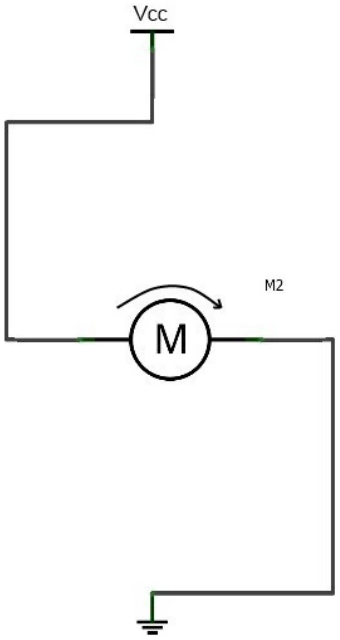


Para controlar un motor DC desde Arduino, tendremos que usar un driver para motores para proporcionar más corriente al motor ya que las salidas del Arduino sólo dan hasta 40mA. Con el driver podemos alimentar el motor con una fuente de alimentación externa.

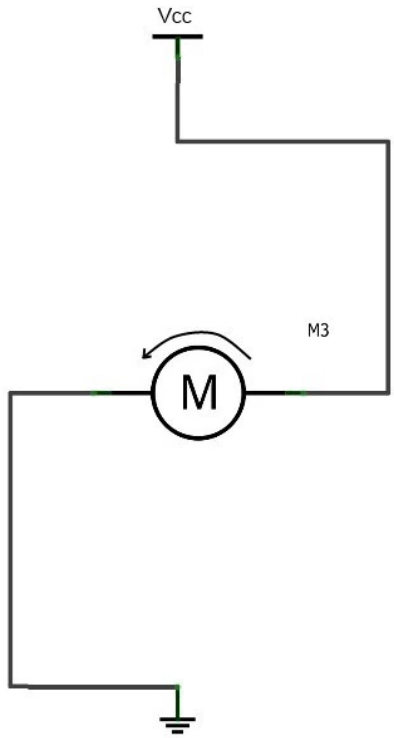
Más información: http://www.prometec.net/motorcc/

El L293D es un integrado para controlar motores DC que usa el sistema puente en H. Es un sistema para controlar el sentido de giro de un motor DC usando cuatro transistores y también la velocidad del motor.
En la imagen vemos que los transistores se comportan como interruptores y dependiendo que transistores conducen y cuáles no cambia la polarización del motor, y con esto el sentido de giro.



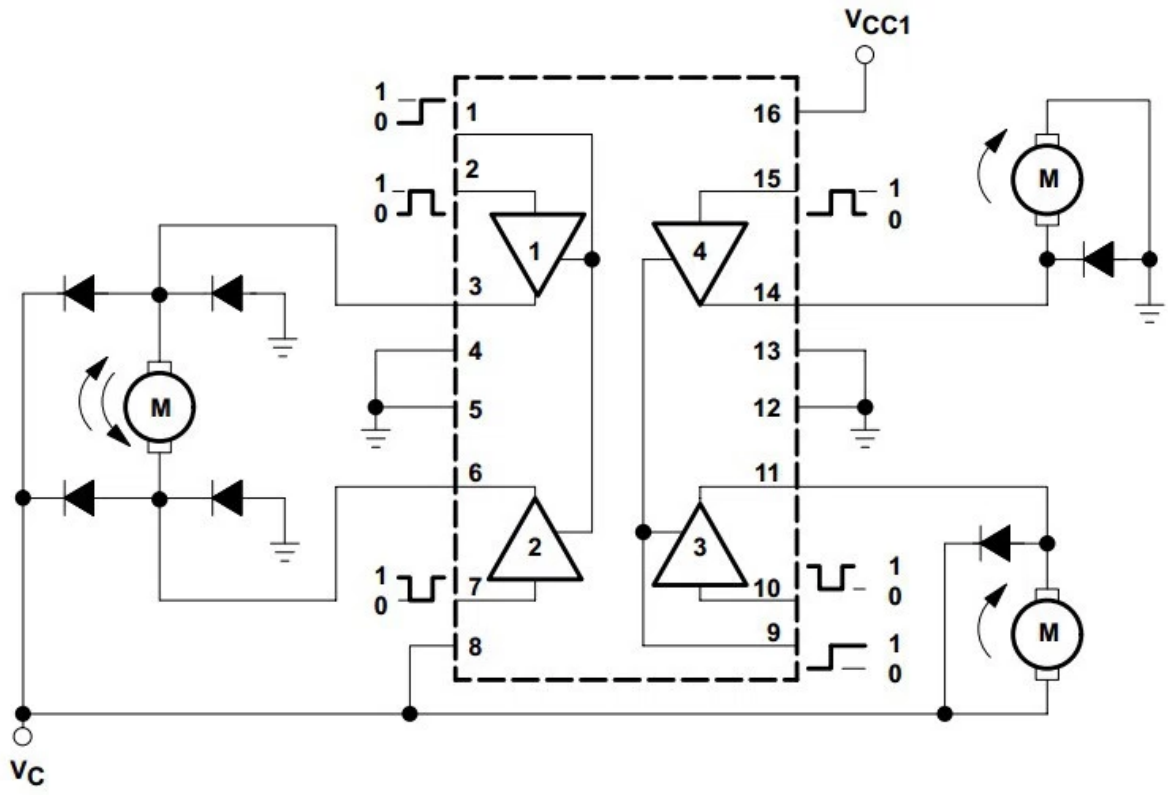


Q1 y Q4 conducen
Q2 y Q3 en corte



Q2 y Q3 conducen
Q1 y Q4 en corte

El L293D tiene dos puentes H y proporciona 600mA al motor y soporta un voltaje entre 4,5V y 36V tal y cómo pone en el datasheet: http://www.ti.com/lit/ds/symlink/l293d.pdf

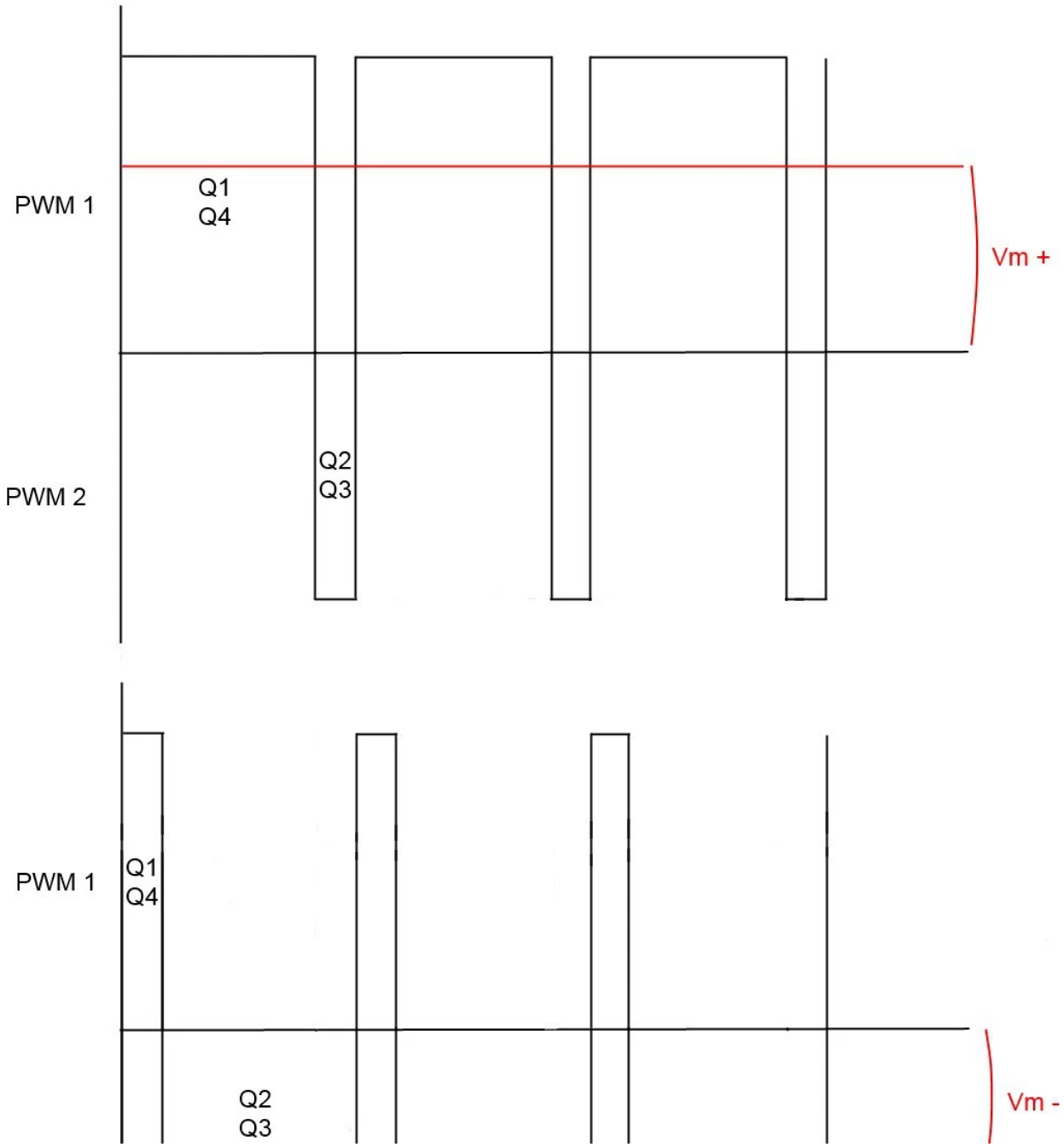


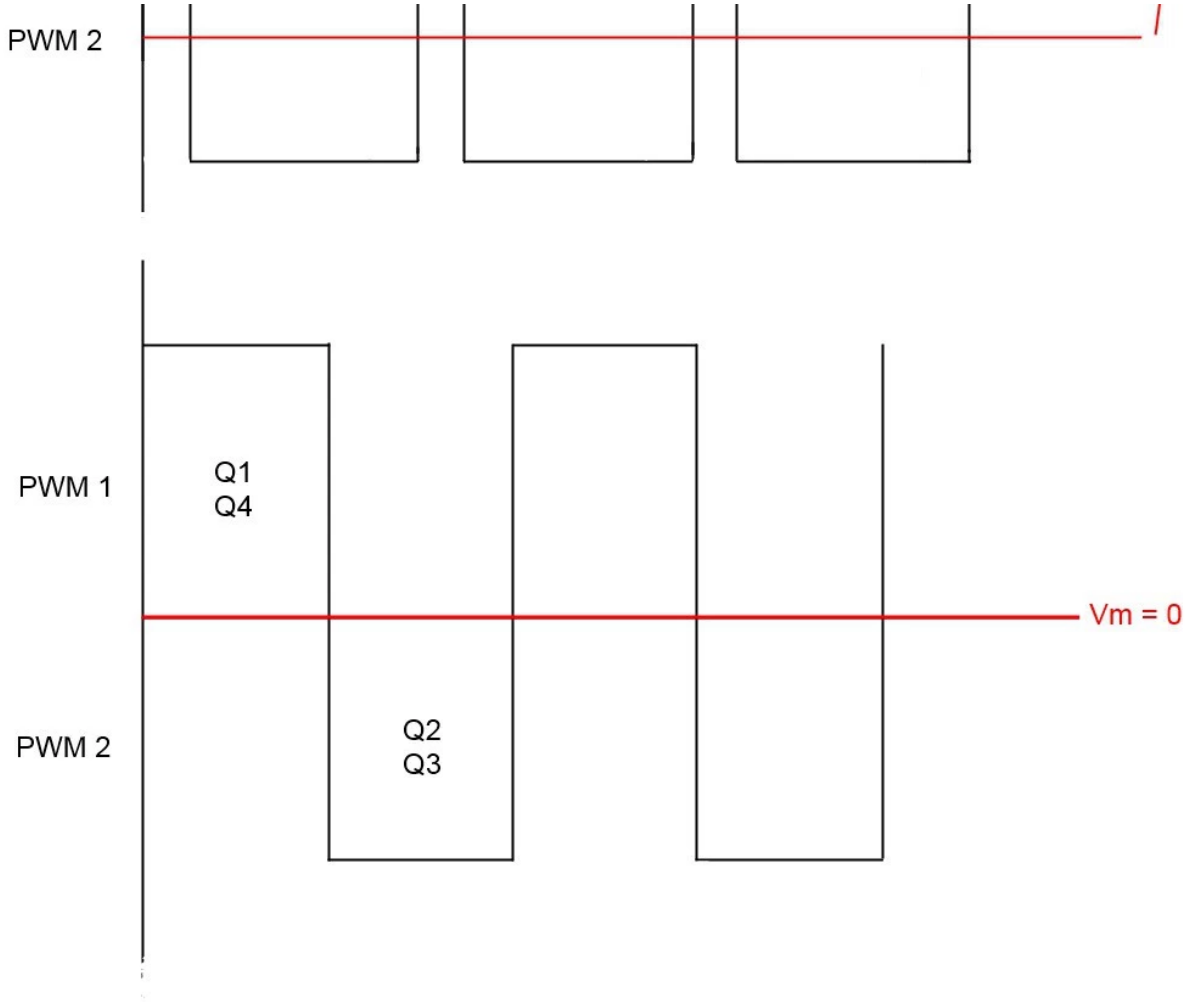
Nosotros usaremos la parte de la izquierda (los diodos externos hay que ponerlos para evitar las corrientes inducidas del motor si usamos el modelo L293 y están incluidos en el integrado si usamos el L293D). Como

se aprecia en la imagen, los pins 3 y 6 son las salidas y se conectan a los bornes del motor. Y los pins 2 y 7 son las entradas donde conectaremos las salidas del Arduino. Dependiendo qué valor ponemos entre los pines 2 y 7 el motor girará en un sentido o en otro.

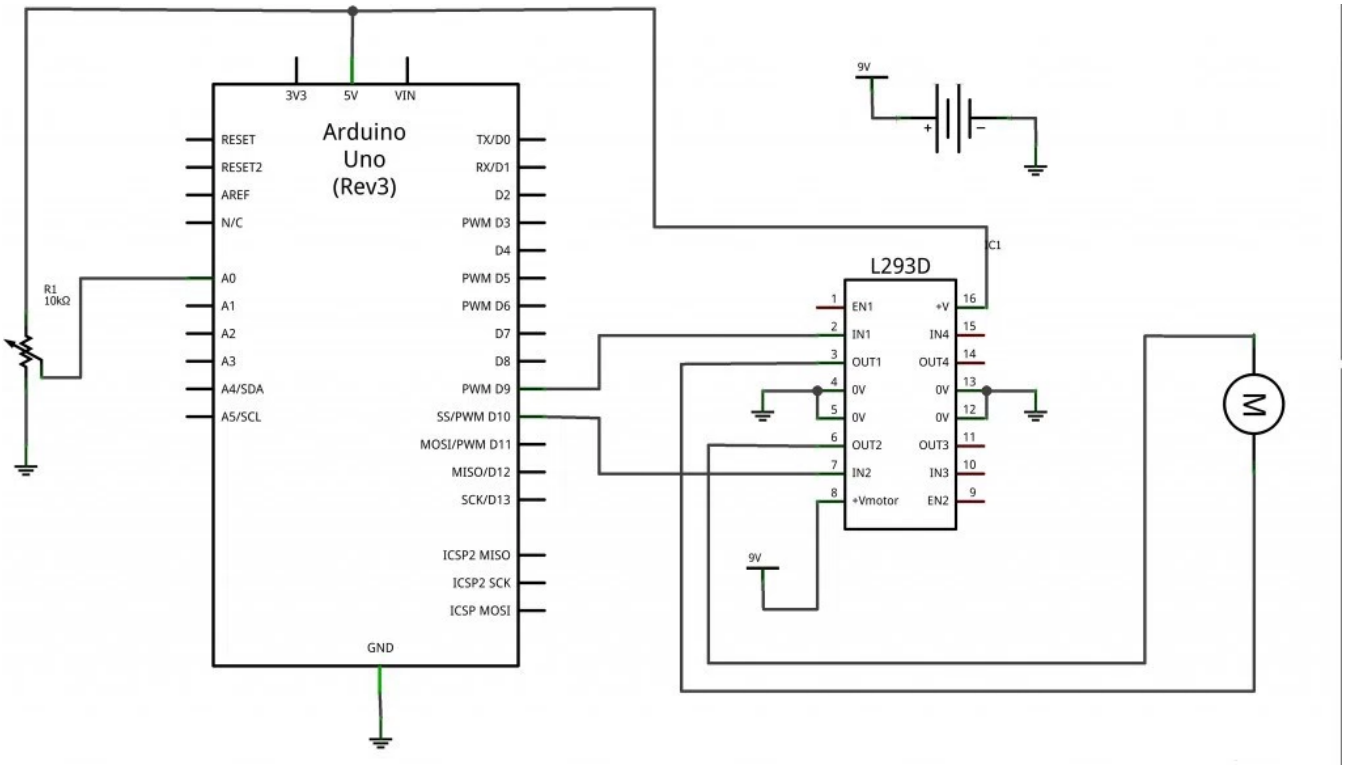
Es muy IMPORTANTE que se utiliza el driver L293, hay que poner diodos para evitar dañar el integrado con las corrientes parásitas generadas por los propios solenoides de las cargas. No obstante el modelo L293D no los necesita, ya que, los lleva incorporados el propio integrado, lo que se hace “más sencillo” y “económico” su uso. También es cierto que L293 al no llevarlos integrados nos permite escoger los que mejor se adapten a nuestras cargas o necesidades.

Para controlar la velocidad del motor se usa la técnica de PWM. Sabemos que hay que atacar los pins 2 y 7 del L293D desde dos salidas del Arduino. En estas dos salidas habrá un PWM a cada una. Pero tenemos que invertir un PWM. ¿Qué quiere decir invertir? Pues que cuando en un PWM tengamos un pulso a un valor alto, en el otro PWM el mismo pulso sea valor bajo. En la imagen lo entenderemos de una manera más gráfica.





Montaje:



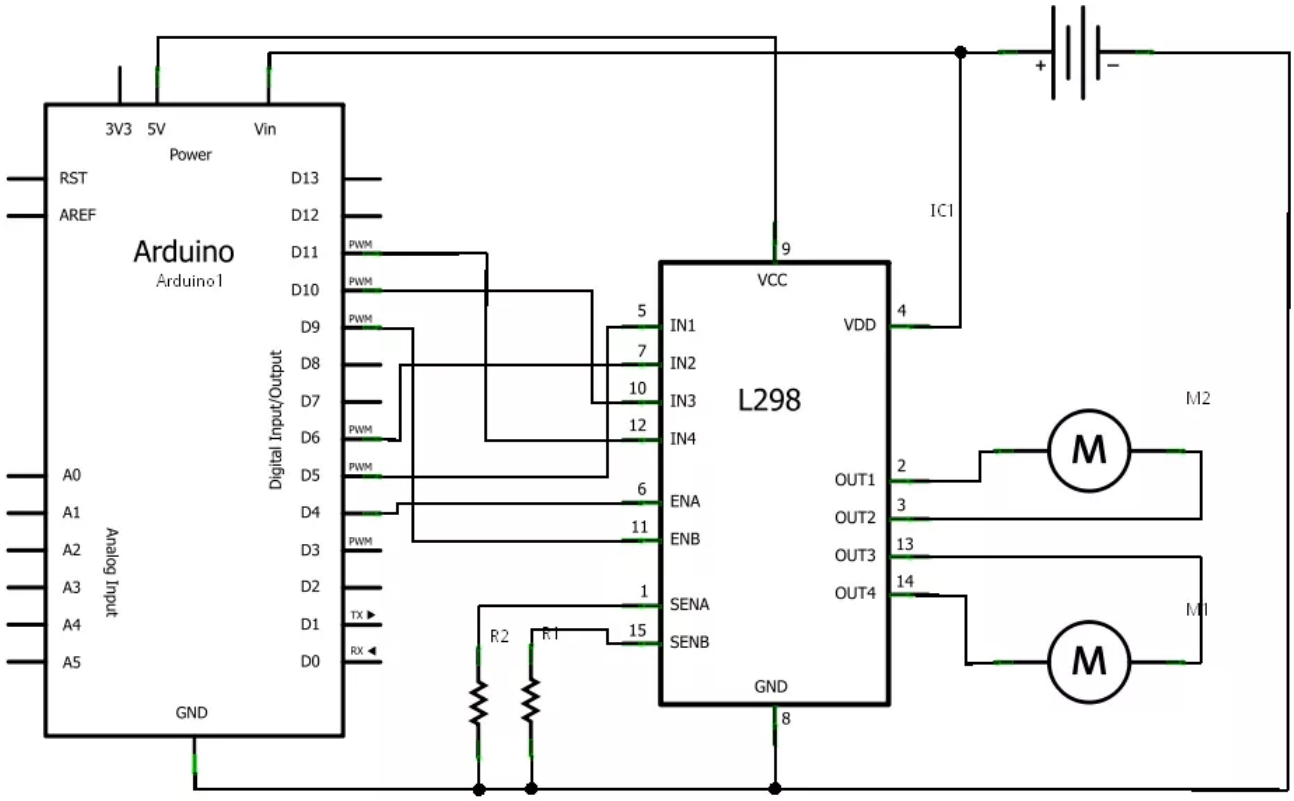
Más información y código en: http://diymakers.es/control-velocidad-y-sentido-de-motor-dc/

Tutorial para controlar un motor con Arduino: http://www.allaboutcircuits.com/projects/use-an-arduino-to-control-a-motor/

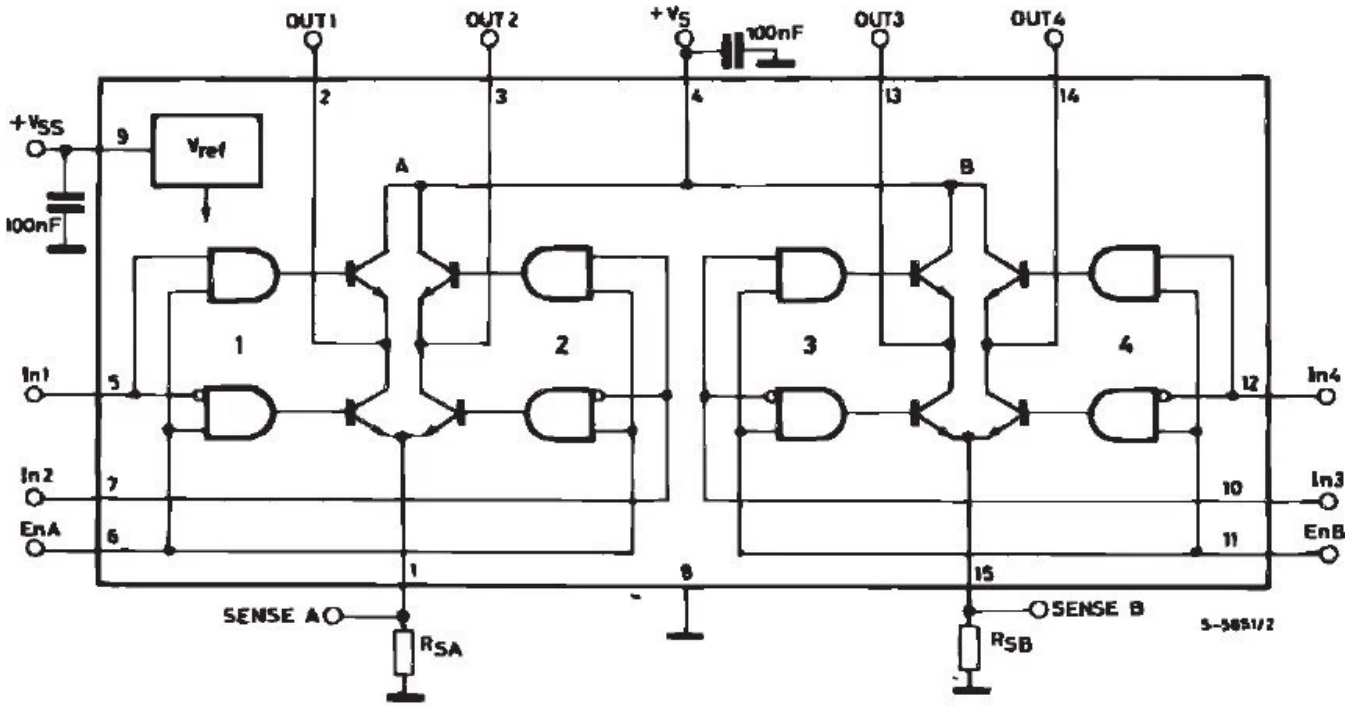
Ejemplo del playground de Arduino:
http://playground.arduino.cc/Main/DirectionalMotorControlWithAL293D

Otro driver de motor muy utilizado es el L298 (datasheet) que es el utilizado por el motor shield. Este driver es similar en funcionamiento al anterior pero posee un sensor de corriente muy útil.

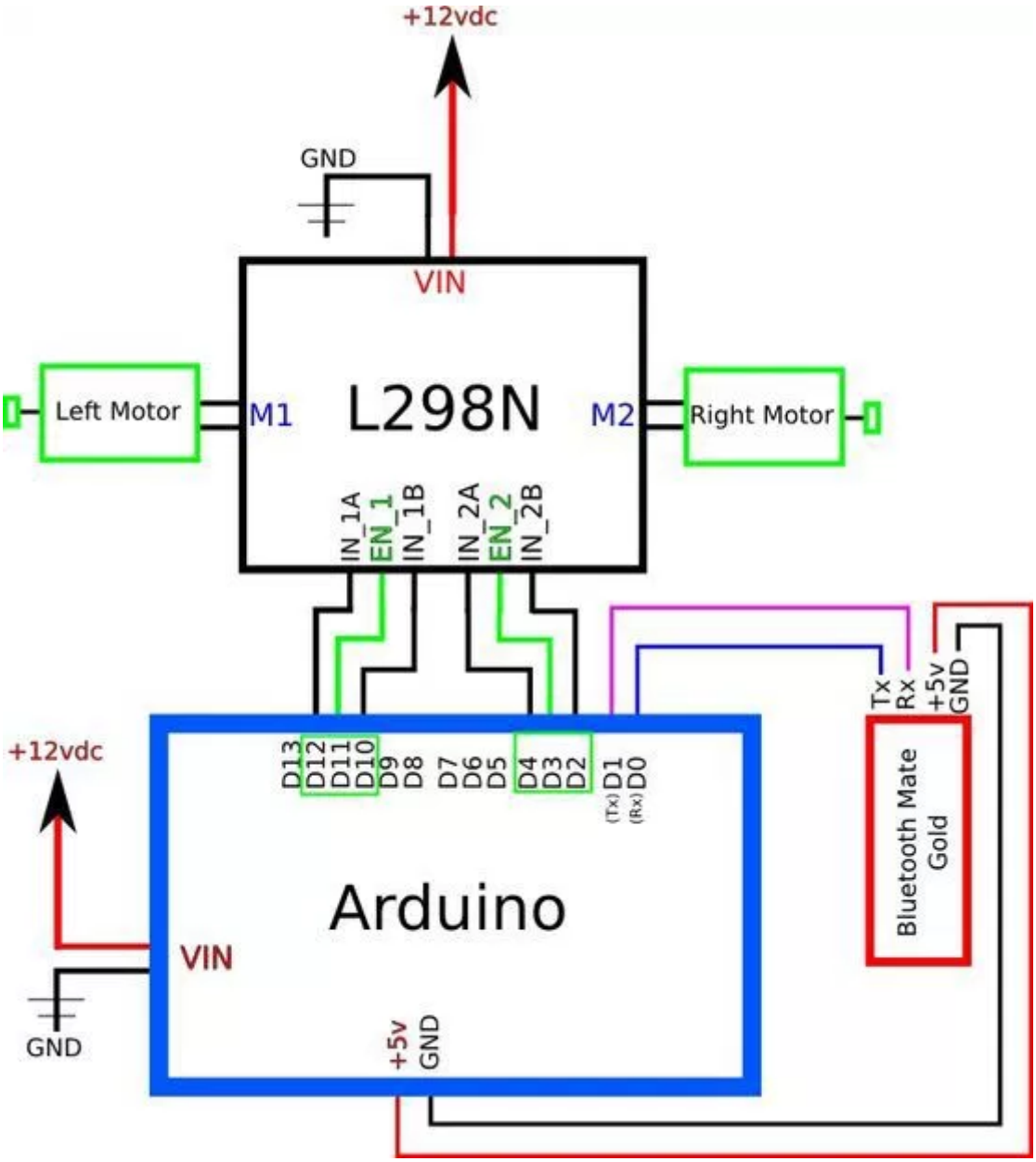
Montaje



Diagrama



Esquema de conexión

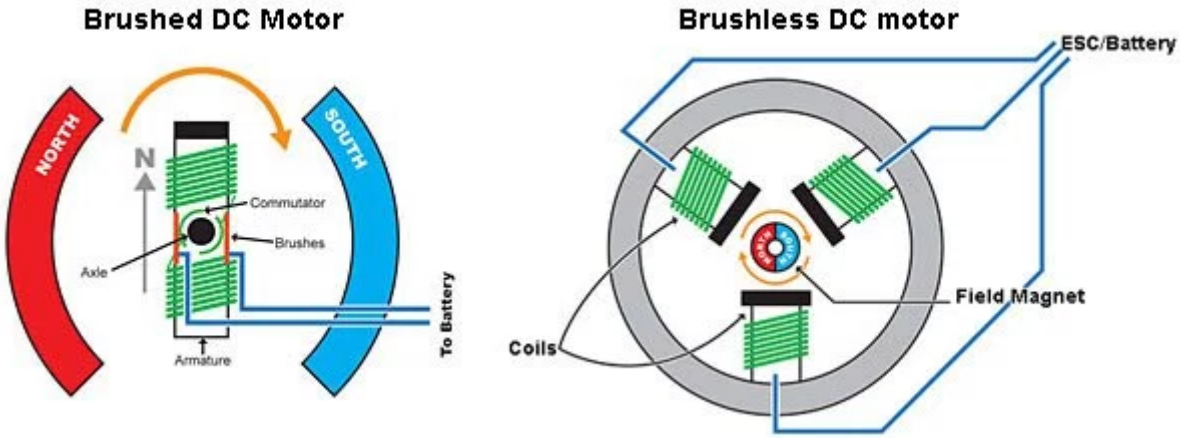


Diferencias entre el L293 y el L298:

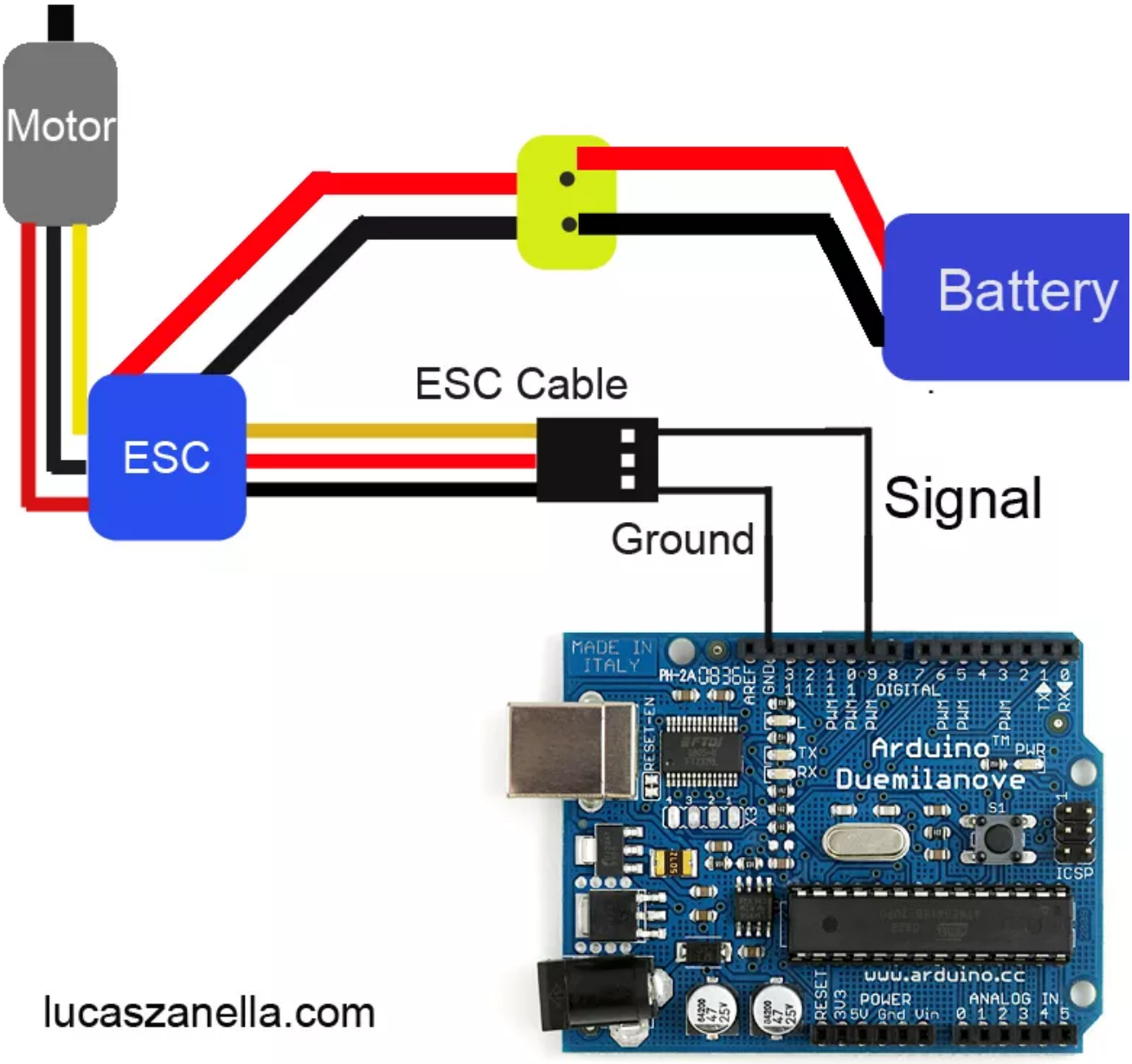
- L293 es un quadruple half-H driver y el L298 es un dual full-H driver
- El L293 al tener 4 canales permite manejar motores paso a paso de 4 hilos y el L298 solo tiene dos canales perfecto para manejar dos motores DC. En el L293 las cuatro lineas de entrada/salida son independientes.
- La corriente de salida por canal en el L293 es de 1A, mientras que en el L298 es de 2A, por ese motivo el L298 puede llevar un disipador.
- Los diodos protectores del L293D deben ponerse externamente en el L293 y L298

Motores DC Brushless

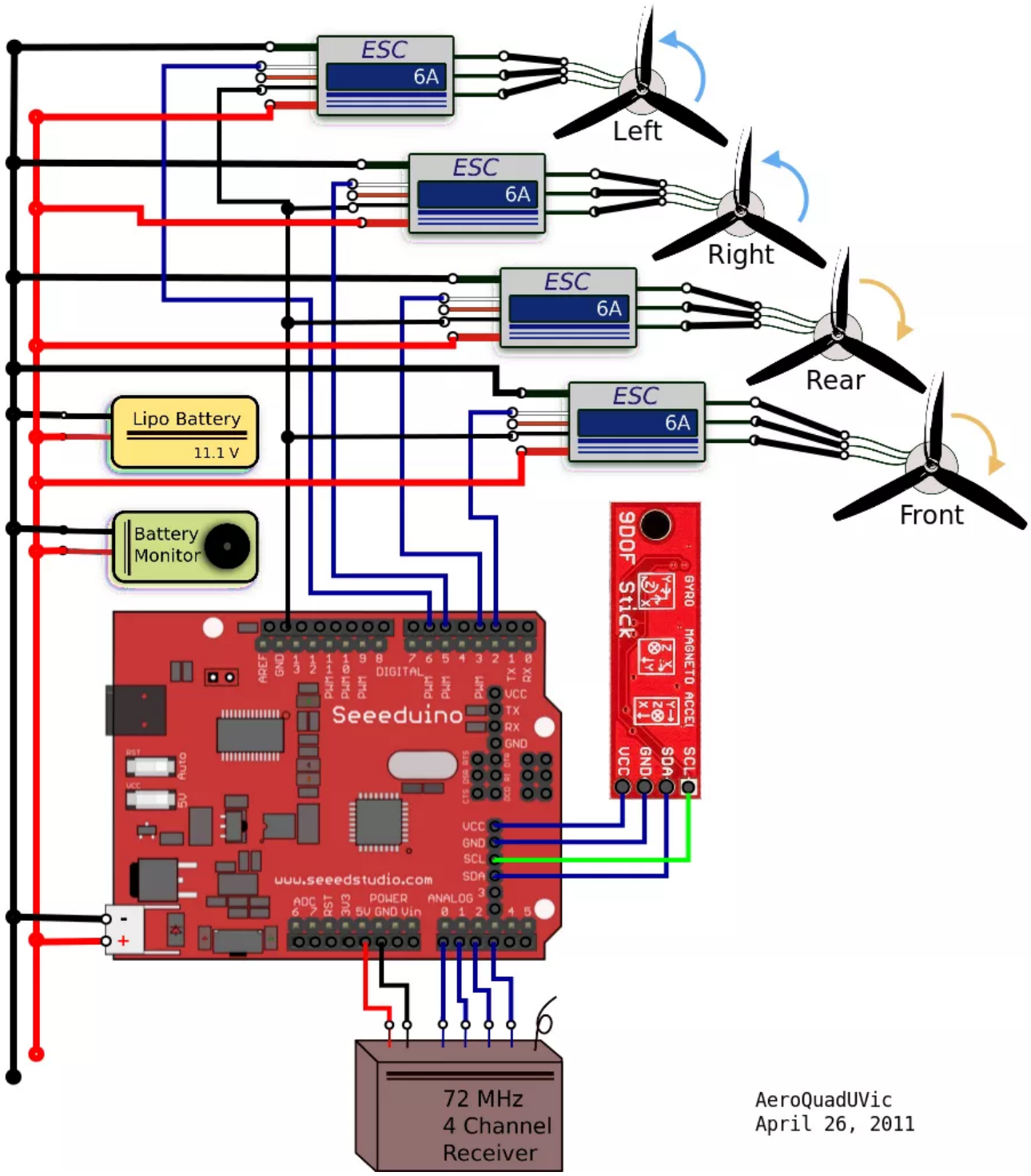
Hay motores DC brushless: https://en.wikipedia.org/wiki/Brushless_DC_electric_motor



Para cotrolar los motores brushless necesitaremos un ESC (Electronic Speed Control)
https://en.wikipedia.org/wiki/Electronic_speed_control



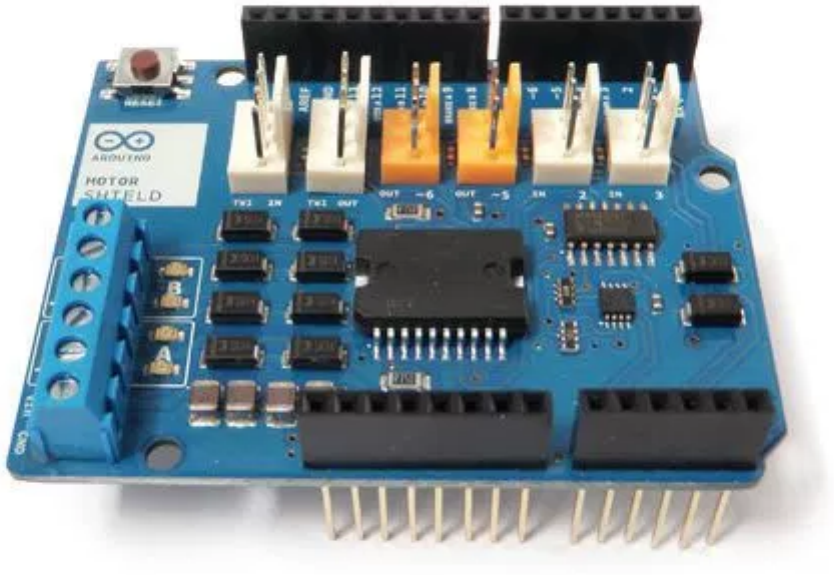
Los motores brushless se usan habitualmente en los drones son trifásicos con un variador para controlar de forma muy exacta la velocidad del motor.



Más información: https://learn.adafruit.com/adafruit-motor-selection-guide/brushless-dc-motor-control

Arduino Motor Shield

El Arduino motor shield está basado en el L298 que es un dual full-bridge diseñado para cargas inductivas como relés, solenoides y motores DC o paso a paso. Me permite manejar dos motores DC controlando las velocidad y dirección de cada motor de forma independiente. También permite medir la corriente absorbida por cada motor entre otras características. La corriente máxima que puede manejar con 2A por canal y el voltaje máximo de suministro es de 50V.



Esquemático: https://www.arduino.cc/en/uploads/Main/arduino_MotorShield_Rev3-schematic.pdf

Arduino motor shield puede ser alimentado por una fuente externa, por lo tanto debemos alimentar al Arduino conectado al motor shield mediante el alimentador y no es posible alimentarlo por USB porque la corriente del motor puede exceder la corriente máxima del estándar USB. El L298 también tiene un alimentación lógica que toma de los 5V de Arduino.

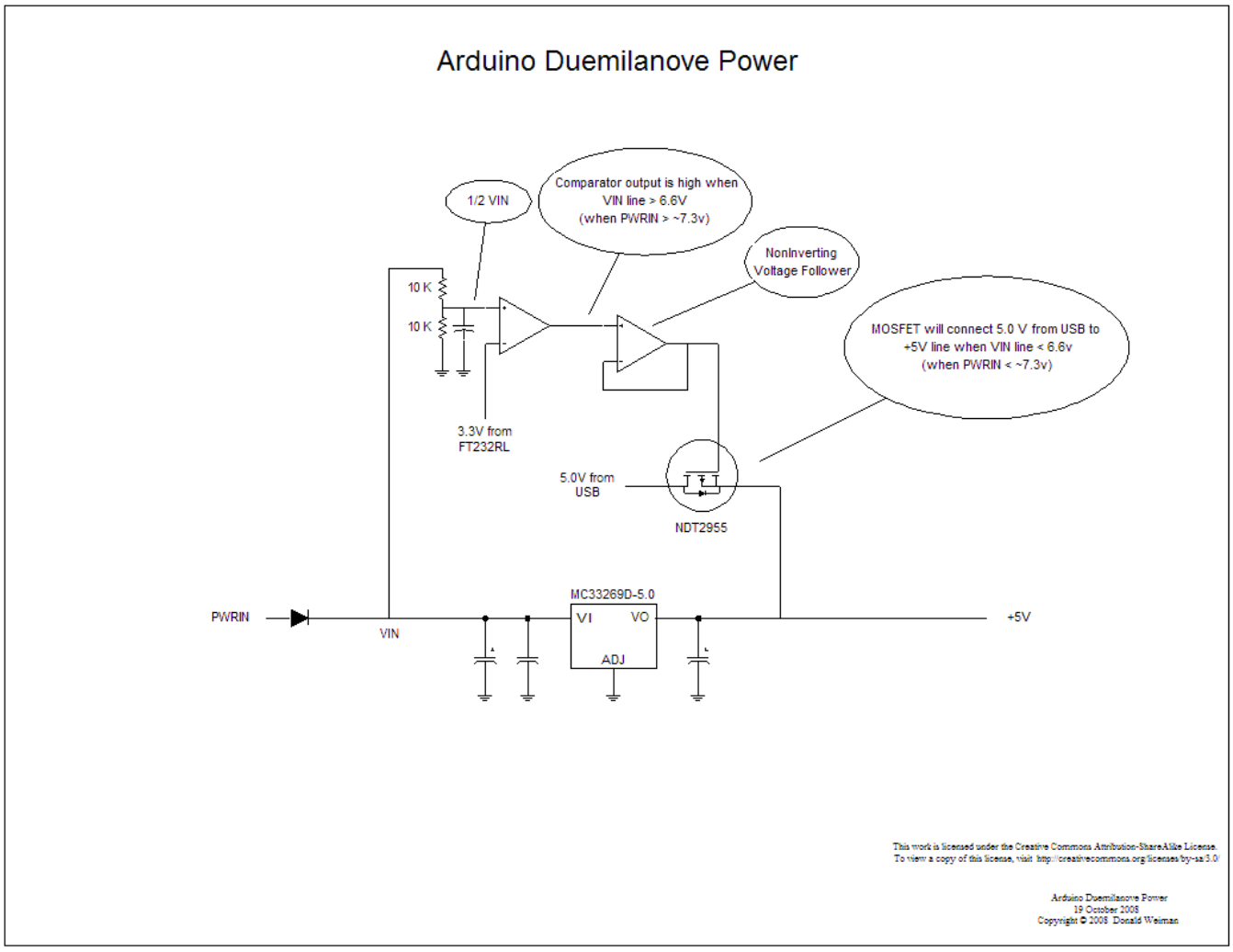
El L298 necesita dos alimentaciones, una para la parte de control (5V) y otra para los motores (12V). La de los motores en tu caso la coge del Vin al que conectas la fuente conmutada de 12V.

Al alimentar desde LSP5 (Borna marcada con Vin MAX 12V), es el VMOT que alimenta Vin según el esquemático de motor shield y al L298P. La alimentación lógica del L298P se hace a través del regulador de tensión del Arduino.

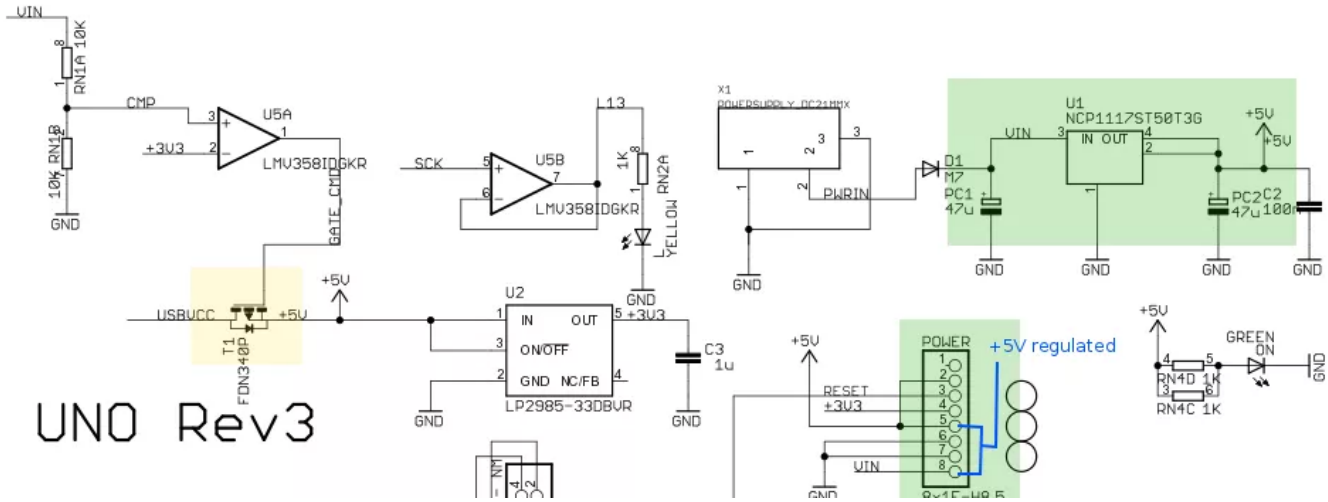
Al alimentar el Arduino mediante el Vin el Motor Shield, puede ser problemático, puesto que al alimentar Arduino a través del Vin no tenemos la protección del diodo M7 ni el condensador para estabilizar el voltaje, por lo que en caso de intercambiar la polaridad se puede dañar Arduino y además si al arrancar un motor el voltaje de Vin cae por debajo de un valor puede ocurrir que el regulador de tensión NCP1117 no sea capaz de dar 5V a su salida y en ese caso Arduino se reiniciaría. Además el NCP1117 tiene una caída de tensión, por lo que para alimentar el Arduino a través de Vin como indica la documentación de Arduino debe ser con un valor entre 7 y 12 V.

Tener en cuenta que si alimento a través de Vin, pero tengo conectado el USB a Arduino, entonces, el MOSFET FDN340P permitirá alimentar el microcontrolador cuando Vin/2 < 3.3V y no se produce el reinicio

de la MCU, actuando como una fuente de alimentación de backup.



Esquema del Arduino UNO: https://www.arduino.cc/en/uploads/Main/arduino-uno-schematic.pdf



Para evitar posibles daños al Arduino sobre el que va montado el motor shield, si el motor requiere más de 9V, es recomendable separar las líneas de alimentación del shield y del Arduino, esto es posible cortando el jumper “Vin Connect” que está en la parte trasera del shield.

En caso de problemas, la solución es alimentar por separado Arduino y el motor shield y cortar el Vin-Connect.

Los pines de alimentación en el Arduino motor shield son:

- Vin en la borna de conexiones, es la entrada de alimentación a los motores conectados al shield. Una fuente de alimentación externa conectada a esta borna también alimenta a la placa Arduino sobre la que va montada el shield. Cortando el jumper "Vin Connect" se elimina esta alimentación a Arduino.
- GND es la masa de los motores que va unida a la masa del Arduino.

Este shield dispone de dos canales llamados A y B y cada uno usa 4 de los pines de Arduino para manejar o monitorizar el motor. En total se usan 8 pines. Se pueden usar los canales separados para manejar dos motores DC o combinarlos para manejar un motor paso a paso bipolar.

Los pines son:

Function	pins per Ch. A	pins per Ch. B
Direction	D12	D13
PWM	D3	D11
Brake	D9	D8
Current Sensing	A0	A1

Si no se necesita el freno y el sensor de corriente, es posible deshabilitar estas características cortando los correspondientes jumpers en la parte trasera del shield.

Los pines de freno al ponerlos a HIGH frenan el motor en lugar de dejarlos correr libremente al cortar la alimentación.

Mediante la función analogRead() es posible leer la corriente en los pines de current sensing como una entrada analógica normal. Está calibrado para medir 3.3V para la corriente máxima de 2A.

Esquema: https://www.arduino.cc/en/uploads/Main/arduino_MotorShield_Rev3-schematic.pdf

Más información: https://www.arduino.cc/en/Main/ArduinoMotorShieldR3

Ejemplos de uso:

- https://www.arduino.cc/en/Tutorial/DueMotorShieldDC
- http://www.arduino.org/learning/tutorials/boards-tutorials/arduino-motor-shield-two-dc-motors-example
- http://www.instructables.com/id/Arduino-Motor-Shield-Tutorial/

NOTA: El Motor Shield de arduino.cc ha sido retirado, pero arduino.org comercializa el Arduino Motor Shield que es el mismo: http://www.arduino.org/products/shields/arduino-motor-shield

Otros shields y breakout board para motores DC

Adafruit Motor Shield:

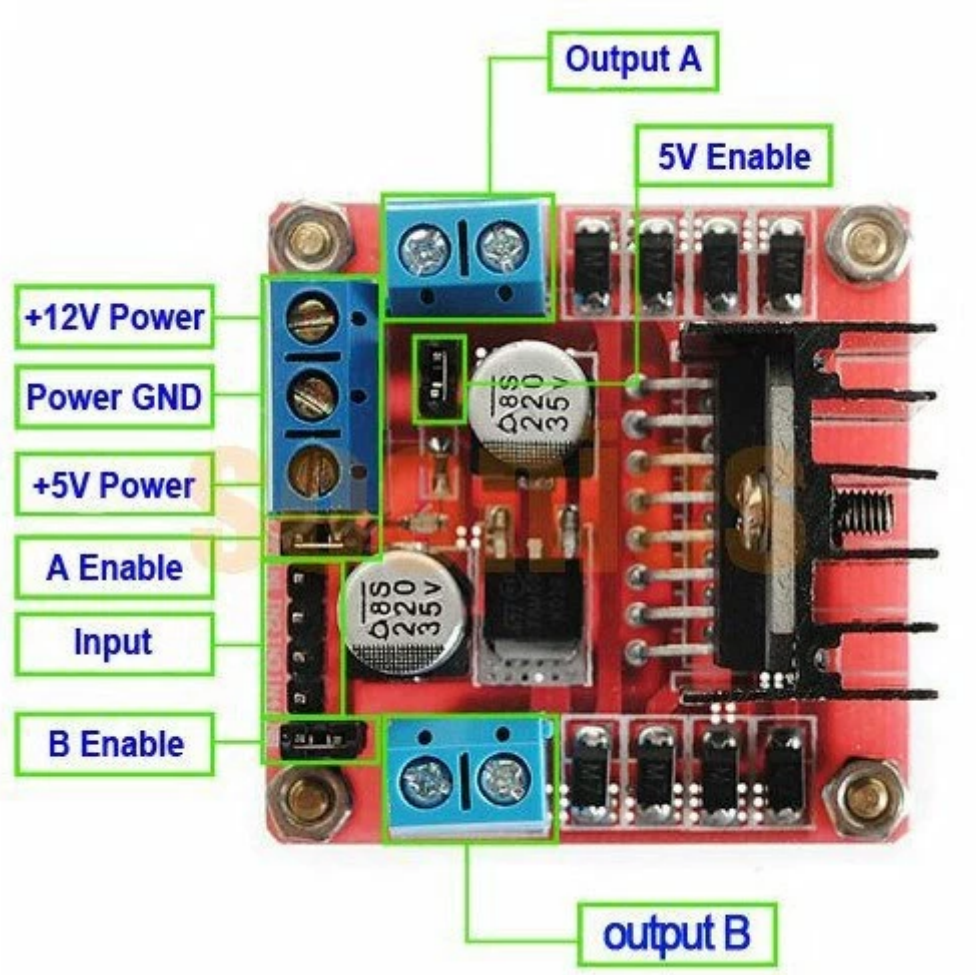
- Producto: https://www.adafruit.com/products/1438
- Tutorial: https://learn.adafruit.com/adafruit-motor-shield-v2-for-arduino
- Librería: https://github.com/adafruit/Adafruit-Motor-Shield-library/

L298N Breakout Board (Esta breakout board es muy sencilla y bien documentada)**:**

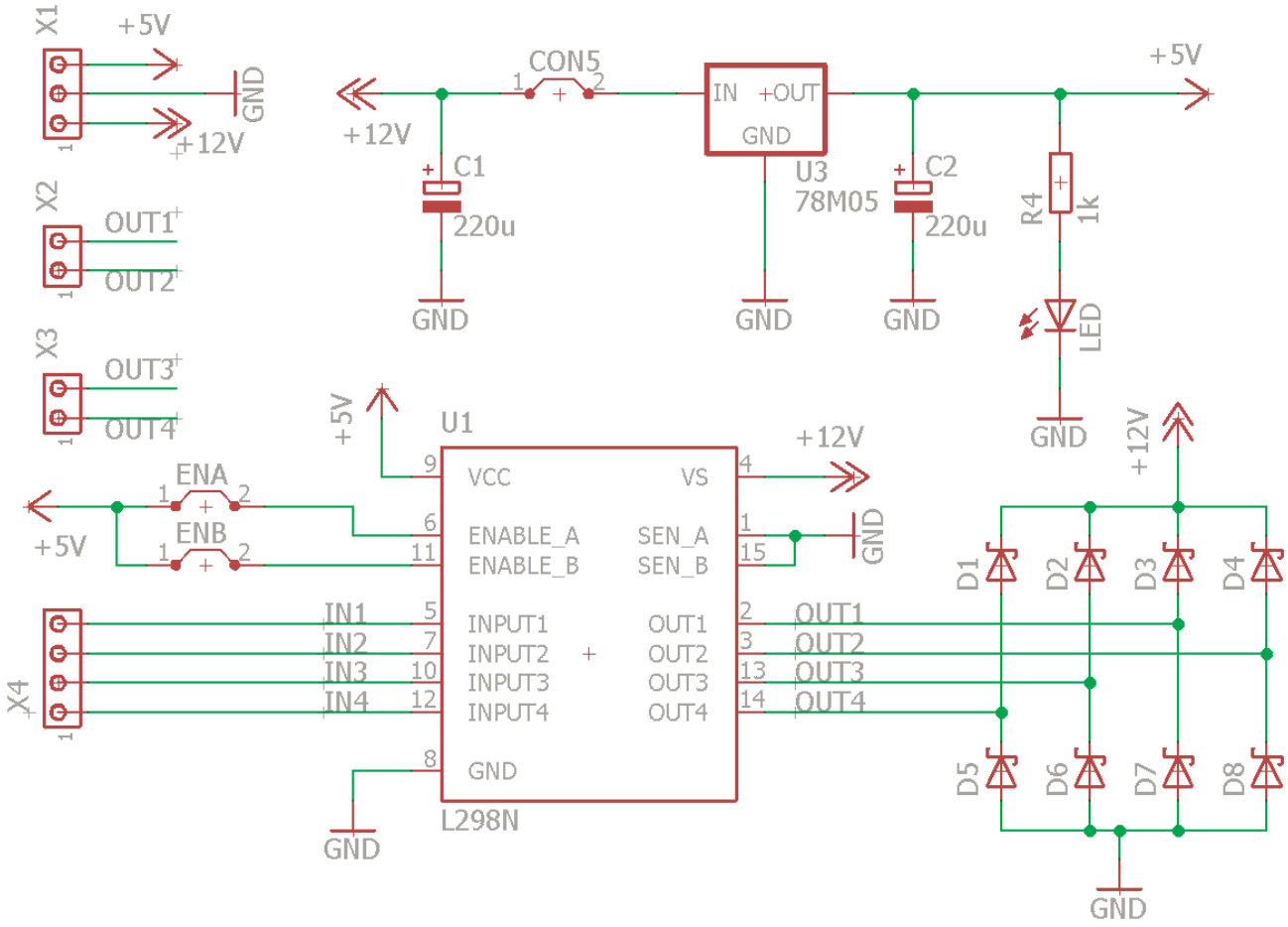
- Producto: http://www.geeetech.com/l298n-stepper-motor-driver-board-p-567.html
- Wiki: http://www.geeetech.com/wiki/index.php/L298N_Motor_Driver_Board (con esquemático)
- El que usa el coche: http://www.vetco.net/catalog/product_info.php?products_id=14337
- Test: https://developer.mbed.org/teams/TVZ-Mechatronics-Team/code/L298N-Breakout-Test/

Specification:

- chipset: L298N
- Driving power supply voltage Vs: +5V to +46V
- Peak current of driving power supply Io: 2A
- Vss: +5V to +7V
- Current of logic power supply: 0 – 36mA
- PWM control signal range:
  - Low level: -0.3V < Vin < 1.5V
  - High level: 2.3V < Vin< Vss
- Enable signal range:
  - Low level: -0.3V
  - High level: 2.3V < Vin< Vss
- Maximum power consumption: 25W
- Working temperature: -25C to 130C
- Regulador de tensión para los 5V.



Esquemático:

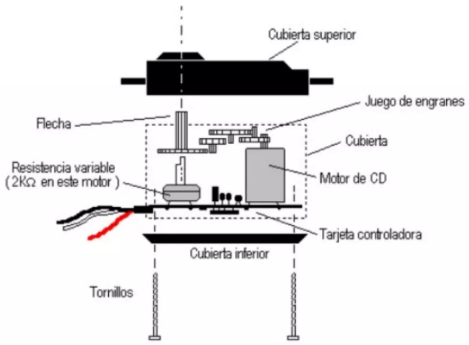


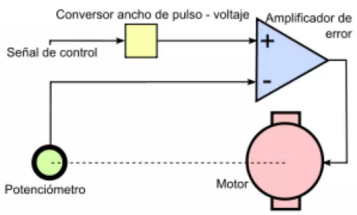
Servomotor

Servomotor (o también llamado servo) es similar a un motor de corriente continua pero con la capacidad de posicionarse en una posición determinada y permanecer fija en esta. Normalmente el ángulo es de 0 a 180 grados, y se alimentan a 5 voltios mínimo.

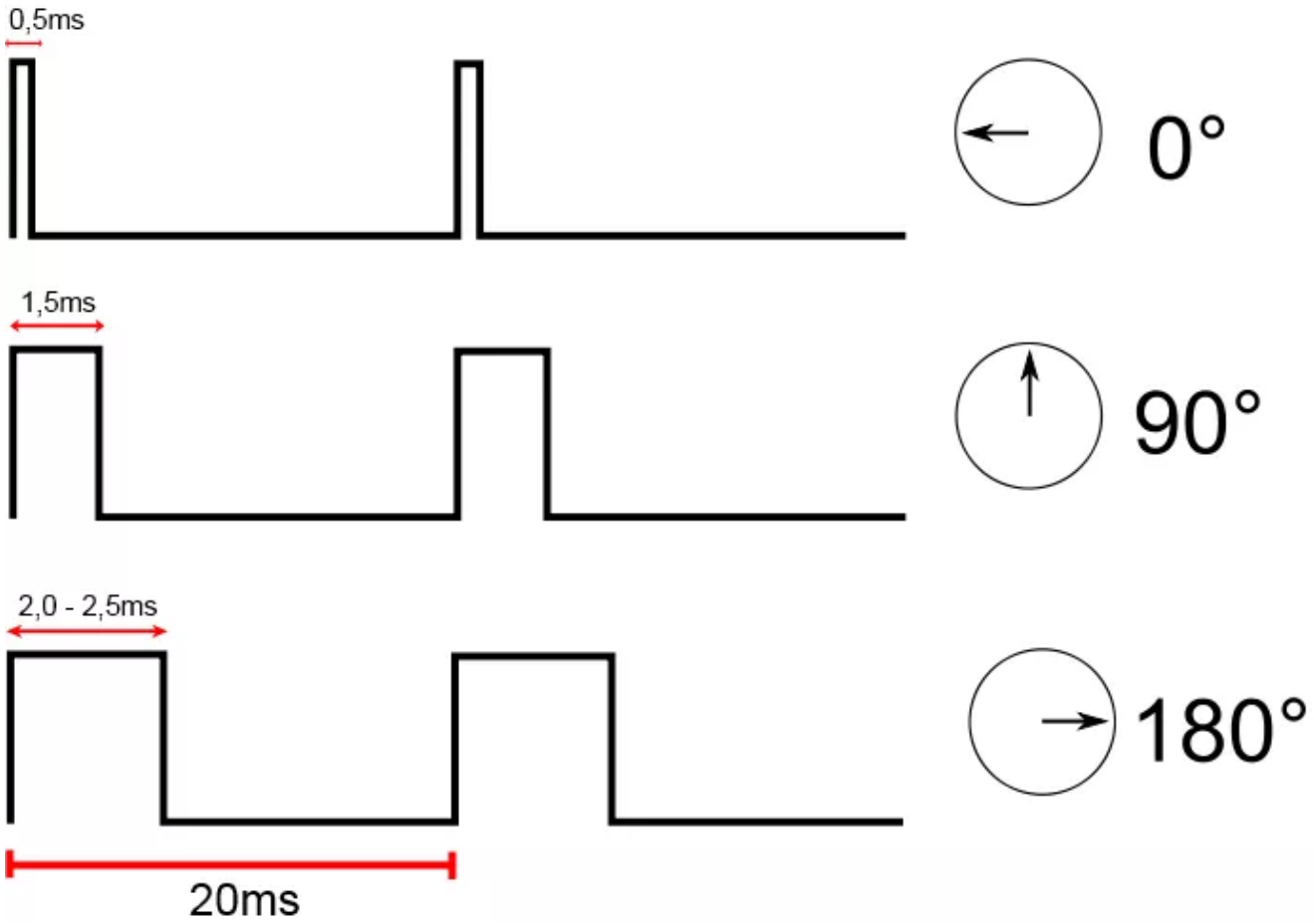
Algunos servos servos pueden alimentarse directamente desde Arduino sin necesidad de un driver, lo que supone una ventaja en algunos casos.

Un servomotor está formado por un motor de corriente continua, una caja reductora, un juego de engranajes, un potenciómetro y un circuito de control. Puede aguantar cierto peso a través del par o torque del servo indicado en sus características. Normalmente se indica con Kg/cm, que quiere decir los kilos que aguanta a 1 cm de distancia.





Para controlar un servo, se usa el PWM. La mayoría trabaja en una frecuencia de 50 Hz (20ms). Cuando se manda un pulso, la anchura de este determina la posición angular del servo. La anchura varía según el servomotor pero normalmente es entre 0,5ms a 2,5ms.



El Arduino utiliza la librería <Servo.h> para controlar los servos y usa las siguientes funciones:

- http://arduino.cc/en/Reference/Servo
- http://arduino.cc/en/Reference/ServoAttach
- http://arduino.cc/en/Reference/ServoWrite
- http://arduino.cc/en/Reference/ServoRead

Fuente, más información y código en: http://diymakers.es/controlar-servomotor-con-mando-ir/

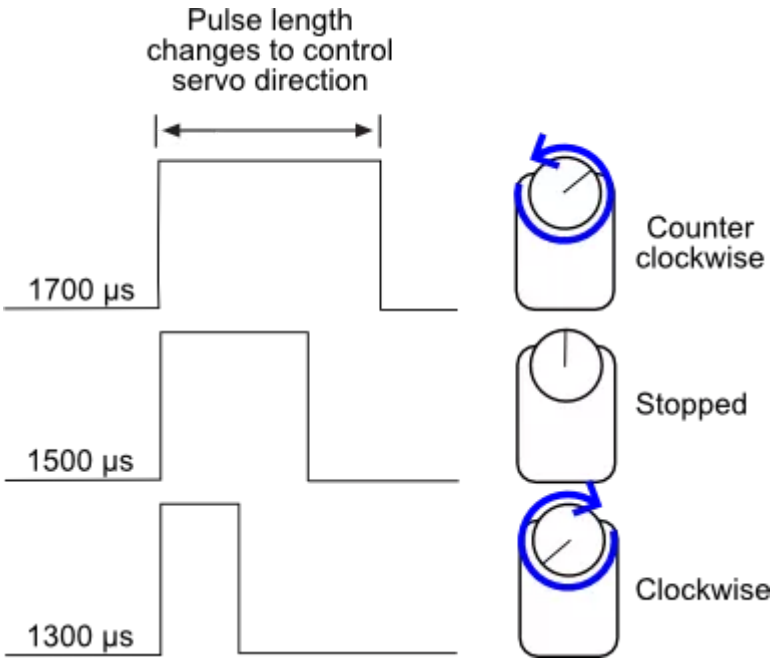
Más información: http://en.wikipedia.org/wiki/Servomotor

Si queremos tener control de bucle cerrado con un motor DC como tenemos con los servos, podemos hacerlo incorporando un encoder al motor DC.

Servo de rotación continua

Un servo de rotación continua es un motor cuyo circuito electrónico nos permite controlar la dirección de giro. A diferencia del servos anteriormente mencionados, no se detiene en una posición, sino que gira continuamente. Son muy utilizados en robótica y en muchas aplicaciones electrónicas, como en lectores de DVD.

En un servo de rotación continua, la función write() configura la velocidad del servo en lugar del ángulo de posición. En este caso 0 es máxima velocidad en giro contrario al sentido horario, 180 es máxima velocidad en sentido horario y 90 motor parado.



Trucar un servo a rotación continua: http://www.ardumania.es/trucar-servo-a-rotacion-continua/

Servo de rotación continua comerciales:

- http://tienda.bricogeek.com/motores/118-servomotor-de-rotacion-continua-sm-s4303r.html
- https://www.servocity.com/html/hsr-2645cr_servo__continuous_r.html#.VqiH0_nhDct
- https://www.pololu.com/product/1248

Tutorial sencillo de BQ: http://diwo.bq.com/muevete-el-servo-de-rotacion-continua/

Tutorial más completo de servos de rotación continua:

- http://www.element14.com/community/community/design-challenges/enchanted-objects/blog/2015/04/03/review-3-digital-continuous-rotation-360-servo
- http://www.element14.com/community/community/design-challenges/enchanted-objects/blog/2015/04/04/review-4-digital-continuous-rotation-360-servo-part-2

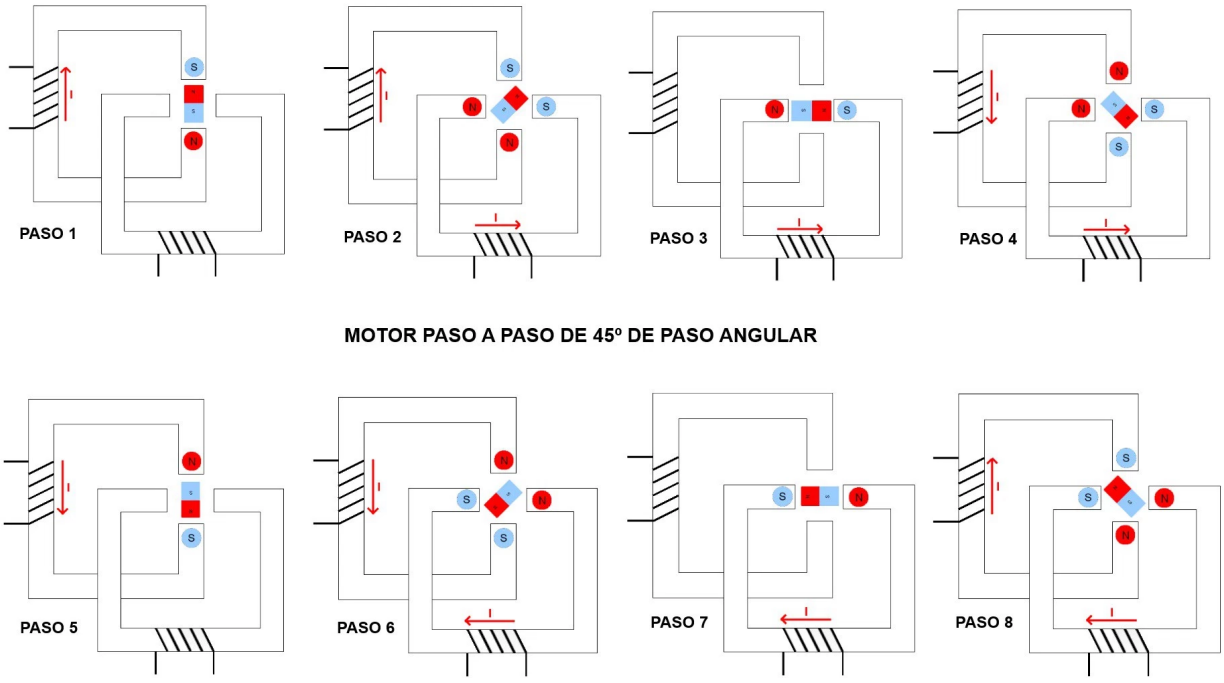
Calibrador de servo: http://www.instructables.com/id/Servo-calibrator-GUI/

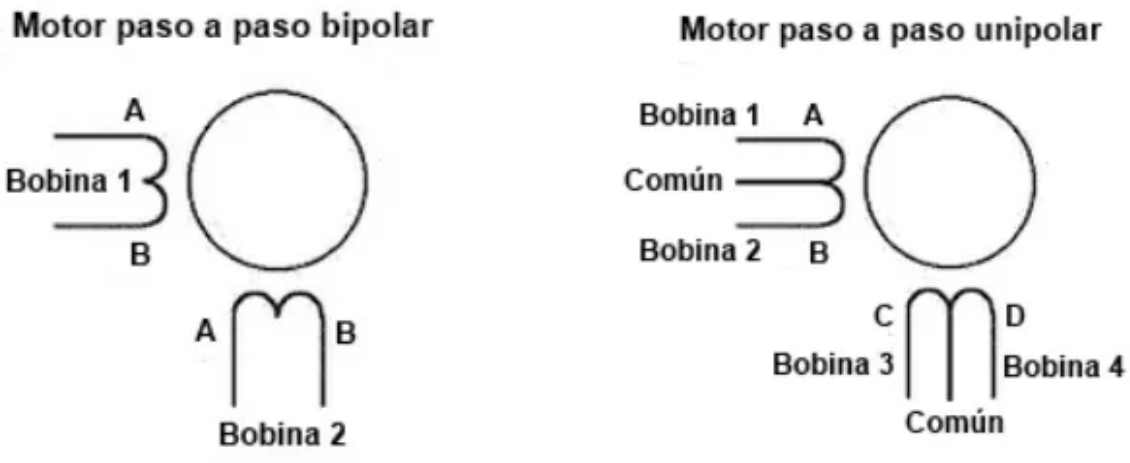
Servo driver para Raspberry Pi: https://learn.adafruit.com/adafruit-16-channel-servo-driver-with-raspberry-pi/overview

Motor paso a paso

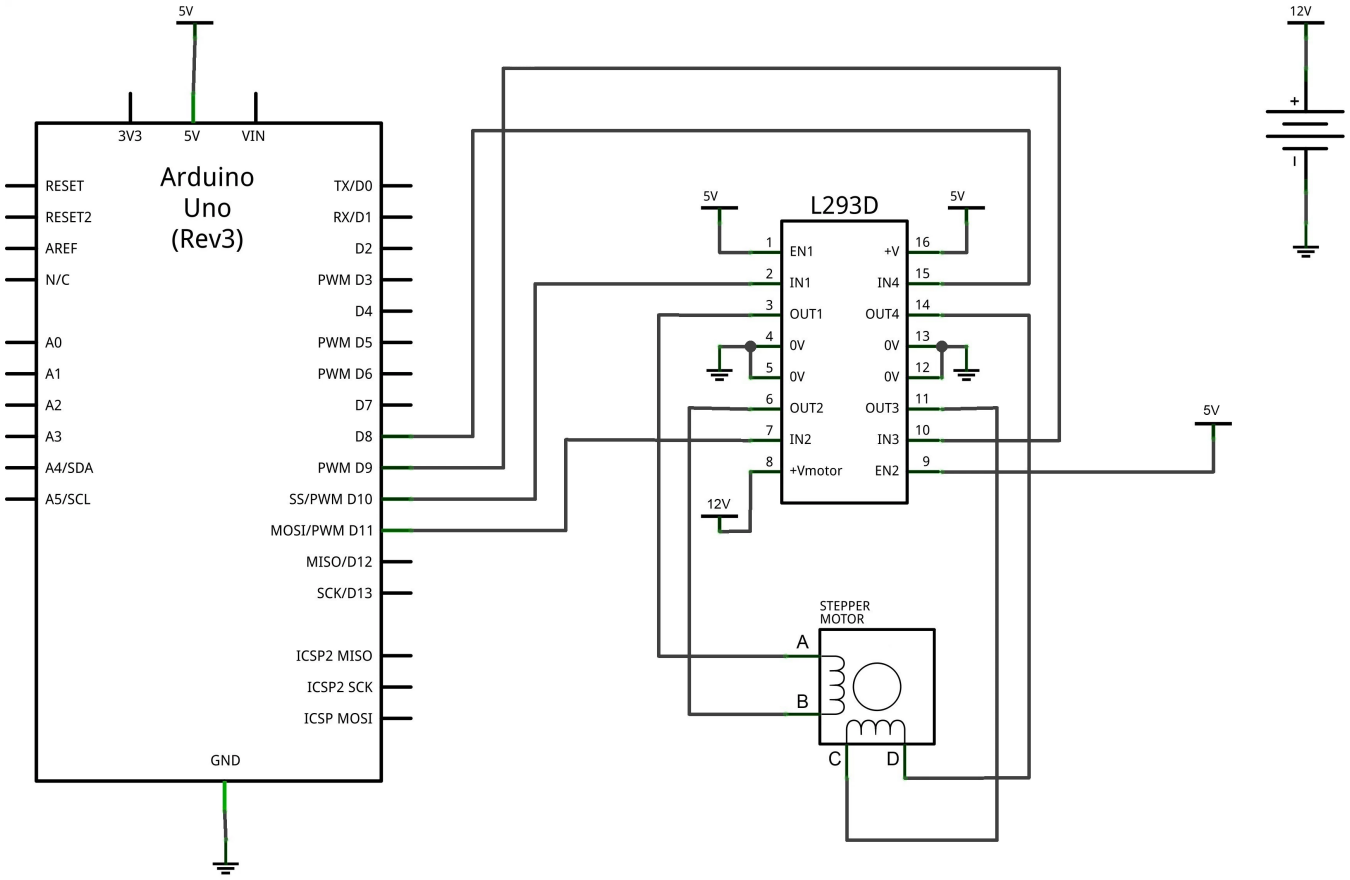
Un motor paso a paso (también llamado stepper) es un dispositivo electromagnético que convierte impulsos eléctricos en movimientos mecánicos de rotación. La principal característica de estos motores es que se mueven un paso por cada impulso que reciben. Normalmente los pasos pueden ser de 1,8º a 90º por paso, dependiendo del motor. Son motores con mucha precisión, que permiten quedar fijos en una posición (como un servomotor) y también son capaces de girar libremente en un sentido u otro (como un motor DC).

Cuando circula corriente por una o más bobinas del estator se crea un campo magnético creando los polos Norte-Sur. Luego el rotor se equilibrará magnéticamente orientando sus polos Norte-Sur hacia los polos Sur-Norte del estator. Cuando el estator vuelva a cambiar la orientación de sus polos a través de un nuevo impulso recibido hacia sus bobinas, el rotor volverá a moverse para equilibrarse magnéticamente. Si se mantiene esta situación, obtendremos un movimiento giratorio permanente del eje. El ángulo de paso depende de la relación entre el nombre de polos magnéticos del estator y el nombre de polos magnéticos del rotor.





Los motores bipolares son más complejos de controlar ya que el flujo de corriente tiene que cambiar de dirección a través de las bobinas con una secuencia determinada. Para esto debemos conectar cada una de las dos bobinas en un puente en H (H-Bridge). Para esto, utilizaremos el integrado L293 que contiene dos H-Bridge (datasheet).



Para controlar motores paso a paso con Arduino, utilizaremos la librería <Stepper.h>:

- http://arduino.cc/en/Reference/Stepper/
- http://www.tigoe.net/pcomp/code/circuits/motors/stepper-motors/

Fuente, más información y código en:

- http://diymakers.es/mover-motores-paso-paso-con-arduino/
- http://es.wikipedia.org/wiki/Motor_paso_a_paso

Un ejemplo de driver de un motor paso a paso https://www.pololu.com/product/2133

Datasheet del driver: https://www.pololu.com/file/download/drv8825.pdf?file_id=0J590

Este tipo de motor es más lento, su rotación es más precisa, es de fácil configuración y control, además mientras que los servos requieren un mecanismo de retroalimentación y circuitos de soporte para accionamiento de posicionamiento, un motor paso a paso tiene control de posición a través de su naturaleza de rotación por incrementos fraccionales.

Los motores paso a paso son adecuados para las impresoras 3D y dispositivos similares en los que la posición es fundamental.

Además de los drivers de motores paso a paso como los puente H y los darlingtong array, es posible usar otros drivers como Easydriver que funciona dando los pasos y la dirección. Más información:

- http://www.schmalzhaus.com/EasyDriver/
- https://www.sparkfun.com/products/12779

Para los motores paso a paso podemos usar las mismas shields que para los motores DC:

- https://www.arduino.cc/en/Main/ArduinoMotorShieldR3
- https://learn.adafruit.com/adafruit-motor-shield-v2-for-arduino
- https://learn.adafruit.com/adafruit-motor-shield

Más Motores

Los de alterna monofásicos podemos controlar su encendido, apagado y sentido de giro con un relé o un contactor y estos a su vez controlados con un Arduino como se ha visto anteriormente.

Los motores trifásicos habrá que usar un arrancador para controlar su encendido y apagado y un variador para controlar la velocidad y podríamos controlarlos estos mediante señales adecuadas desde Arduino como modbus.

Disponemos de shields para controlar múltiples motores, ya sean DC, servos o paso a paso.

A la hora de seleccionar un motor, hay varios factores que deben evaluarse antes de decidir qué tipo utilizar. Estos factores incluyen velocidad, par, bucle abierto o cerrado, resolución, precio y mantenimiento.

La velocidad es uno de los criterios más importantes a tener en cuenta al elegir entre servo y motor paso a paso. Hay una relación inversa entre velocidad y par en los motores por pasos. Cuando la velocidad se

incrementa, el par decrece. Los motores servo tienen un par constante hasta la velocidad nominal. Como criterio general, por encima de 1000 rpm, debe seleccionarse servo. Si la velocidad está por debajo de 500 rpm, los motores por pasos son una buena elección porque producen un par más alto que el servomotor de tamaño equivalente. Los servomotores tienen la capacidad de producir un par pico en cortos periodos de tiempo que es hasta 8 veces su par nominal continuo. Esto es particularmente útil cuando la resistencia del movimiento no es constante. Los motores por pasos no tienen esta capacidad.

En algunos casos que requieren la alta velocidad, alta aceleración o aplicaciones críticas, el bucle cerrado es importante. Esto ocurre por ejemplo cuando se trabaja con accesorios caros en los que los fallos no son aceptables.

Guía de selección de motores: https://learn.adafruit.com/adafruit-motor-selection-guide/types-of-motors

Web como muchos motores: https://www.servocity.com/

Por ejemplo de un motor CC con reductora y encoder:
https://www.servocity.com/html/26_rpm_planetary_gearmotor_w__.html#.VpfFM_nhDct

Esta entrada se publicó en Arduino, Motores y está etiquetada con Arduino, Arduino Motor Shield, L298N, Motor DC, Motor Paso a Paso, Motores, Servo, Servo Rotación Continua, Servomotor, Stepper en 4 julio, 2016 [https://aprendiendoarduino.wordpress.com/2016/07/04/motores/] .