



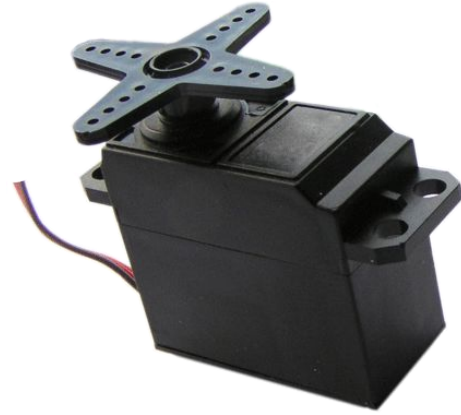
VISUALINO

APUNTES Y PROYECTOS IV

por Aurelio Gallardo Rodríguez BY – SA – NC

SERVOS

(DE ROTACIÓN CONTINUA)



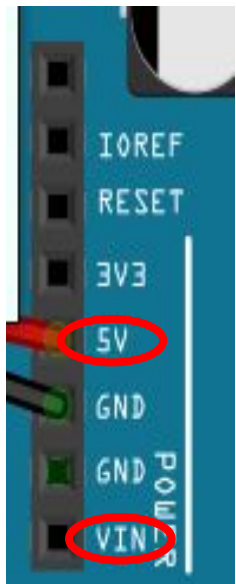
¿QUÉ SON LOS SERVOS?

Un servo es un motor de corriente continua especial; gracias a un sistema de engranajes reductores y electrónica de control integrada, seremos capaces de controlar su posición (ángulo), como su velocidad.

Algunos servos (servos de rotación continua) se usan como motores de corriente continua especiales en los que es fácil programar el sentido de la marcha y su velocidad, pero no su posición. Suelen consumir poco y ser capaces de tener un par motor relativamente elevado. Otros están mejor diseñados para girar en ángulos precisos (entre 0° y 180° , e incluso 0° y 360°).

Los servos se usan mucho como motores para robots (por ejemplo, el printbot de BQ) o para posicionamiento de elementos.

PASO PREVIO: DIFERENCIAS ENTRE 5V Y VIN



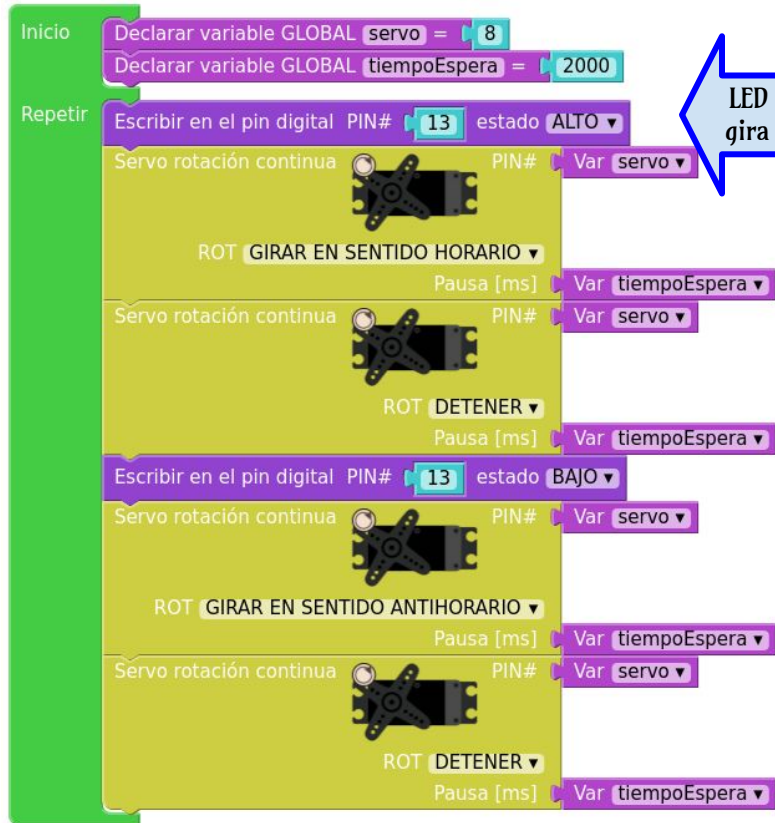
ARDUINO puede alimentarse de dos formas: bien a través de la tensión proporcionada por el ordenador a través del puerto USB (5V) o bien a través de unas pilas o baterías, admitiendo pilas o baterías que proporcionen entre 7 - 12 Voltios (recomendado).

Bien, en la primera opción no hay diferencia. Ambos pines 5V, y Vin te proporcionarán aproximadamente la misma tensión (suele ser algo menor de 5V, alrededor de 4.5V) pero en el segundo caso sí tiene importancia. 5V seguirá siendo esa tensión (siempre algo menos) y *Vin será el valor de tensión de las pilas o la batería*. ¿Por qué es importante cuando uso motores? Porque los motores tienen una tensión de trabajo (puede ser 5V o puede ser más) que hay que aplicar, y según la carga, que necesiten un consumo mayor o menor (intensidad en mA). *Es posible que algunos motores no tengan suficiente con lo que proporciona el puerto USB y necesiten baterías o pilas más potentes.*

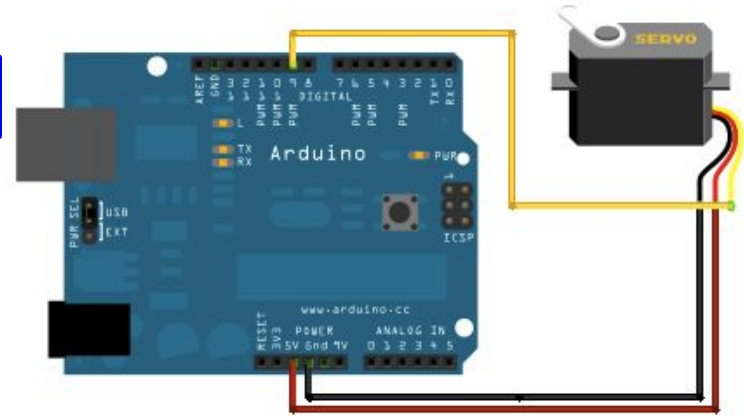
Por eso, y según la aplicación, al usar servos y/o motores de corriente continua, necesitaré pilas o baterías potentes y circuitos añadidos (ULN2003, un puente H como el SN75441 o un motor driver shield).

Ejemplo en: <http://computointegradoits.blogspot.com.es/2012/05/puente-h.html>

ACTIVAR UN SERVO DE ROTACIÓN CONTINUA. ADELANTE, PARA, ATRÁS, PARA.



LED 13 enciende cuando gira en sentido horario.

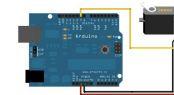


- ❑ El sentido horario o antihorario **hay que verlo desde la trasera del servo**, no desde donde gira el eje.
- ❑ Si el servo “tiembla” cuando se da la orden de paro, hay que calibrarlo.
- ❑ Calibración: <https://youtu.be/krCLMx88gvc>

Aurelio Gallardo Rodríguez



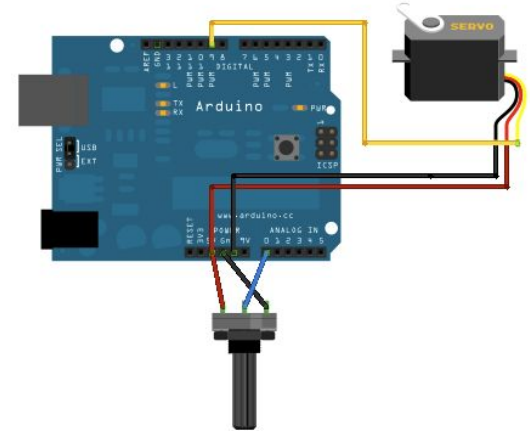
CONSIDERACIONES DE LOS SERVOS DE ROTACIÓN CONTINUA



- Los servos de rotación continua, a diferencia de otros servos, “no saben” parar a un ángulo determinado. Cuando se le proporciona un ángulo de 0° giran en sentido horario a máxima velocidad. Por tanto los bloques de la derecha son equivalentes.
- Si uso el segundo bloque con 90° el efecto es el de pararse.
- Si uso el segundo bloque con 180° el efecto es el de girar en sentido contrario.
- Si el segundo bloque se usa con números entre 1° y 89° , contra más cerca de 90° más despacio gira en sentido horario. Entre 91° y 179° ocurre algo idéntico en sentido antihorario.
- Si no estoy satisfecho con la velocidad de un motor, quizás necesite considerar servos más potentes u otros motores de corriente continua.



CONTROLA VELOCIDAD Y DIRECCIÓN DE UN SERVO R.C. CON UN POTENCIÓMETRO



SERVOS

(POSICIONAMIENTO)



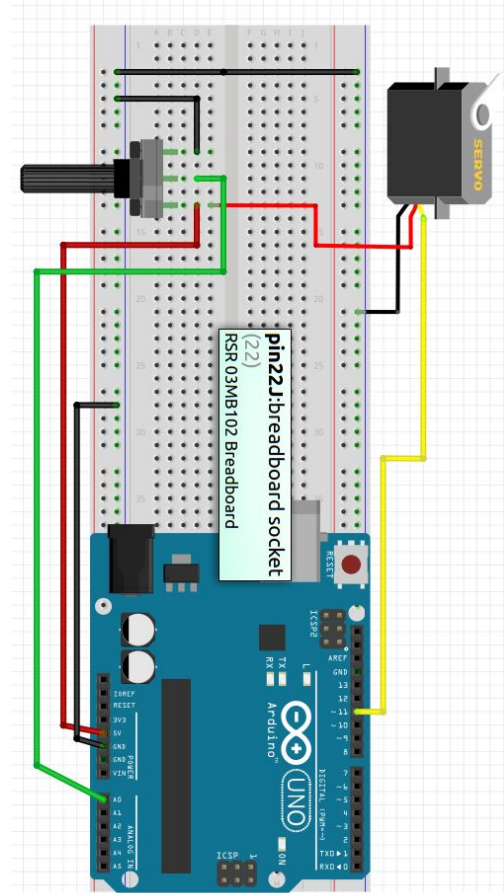
CONTROLA EL ÁNGULO DEL SERVO

```
Inicio
  Declarar variable GLOBAL servo = 11
  Declarar variable GLOBAL angulo = 0

Repetir
  Var angulo = Potenciómetro PIN# Pin analógico A5
  Var angulo = Mapear Var angulo De [ 0 - 1023 ] a [ 0 - 180 ]
  Imprimir por puerto serie con salto de línea Var angulo
  Servo PIN# Var servo
  Grados (0~180) Var angulo
  Pausa [ms] 10
  si Var angulo ≤ 90
    ejecutar Escribir en el pin digital PIN# 13 estado ALTO
  de lo contrario Escribir en el pin digital PIN# 13 estado BAJO
```

Muy parecido al del control de velocidad con servos de rotación continua.

Según el modelo, con 0° o 180° puede “temblar”. Evitarlo poniendo un rango de 1° a 179°



PROYECTOS CON SERVOS

- Con dos servos de corriente continua, y algún tipo de base y un par de ruedas, montadas y atornilladas sobre el eje de los servos, construye un pequeño coche. Hazlo moverse adelante, atrás, girar... (Para que tenga estabilidad, un tornillo largo en la parte trasera puede hacer de contrapeso).
- Con dicho coche, intenta los siguientes retos
 - ◆ **Proyecto 1:** Que avance unos 5 metros atravesando la puerta de una habitación, se gire y vuelva a la habitación.
 - ◆ **Proyecto 2:** Que le dé un par de vueltas a una mesa.
- **Proyecto 3:** Con un servo normal, de posicionamiento (atornillar algún elemento indicador), y con dos sensores LDR, el servo debe apuntar donde se esté recibiendo más cantidad de luz.
- **Proyecto 4:** En el coche anterior, con dos sensores LDR, el coche debe girar hacia donde haya más luz.

POSIBLE SOLUCIÓN PROYECTO 4 CON LDR

```
Inicio
  Declarar variable GLOBAL luz1 = 0
  Declarar variable GLOBAL luz2 = 0
  Declarar variable GLOBAL luzMaxima = 100

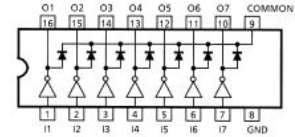
Repetir
  Var luz1 = Sensor de Luz PIN# Pin analógico A3
  Var luz2 = Sensor de Luz PIN# Pin analógico A2
  Var luz1 = (Var luz1 x 100) ÷ (Var luz1 + Var luz2)
  Var luz2 = (Var luz2 x 100) ÷ (Var luz1 + Var luz2)
  Imprimir por puerto serie con salto de línea crear texto con Var luz1
  " "
  Var luz2
  Var luz1 = Mapear Var luz1 De [ 0 - Var luzMaxima ] a [ 90 - 45 ]
  Servo PIN# Pin digital 9
  Grados (0-180) Var luz1
  Pausa [ms] 30
  Var luz2 = Mapear Var luz2 De [ 0 - Var luzMaxima ] a [ 90 - 135 ]
  Servo PIN# Pin digital 8
  Grados (0-180) Var luz2
  Pausa [ms] 30
```



Proyecto susceptible de ser mejorado

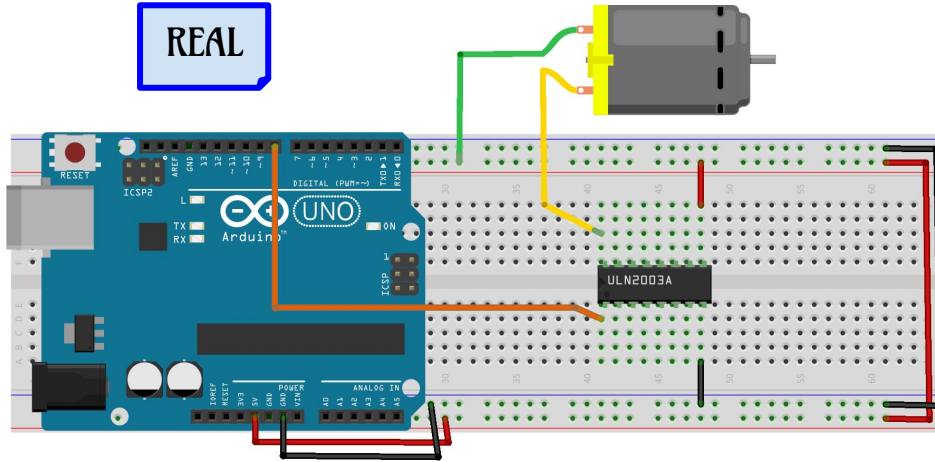
MOTORES CORRIENTE CONTINUA

(CON ULN2003)

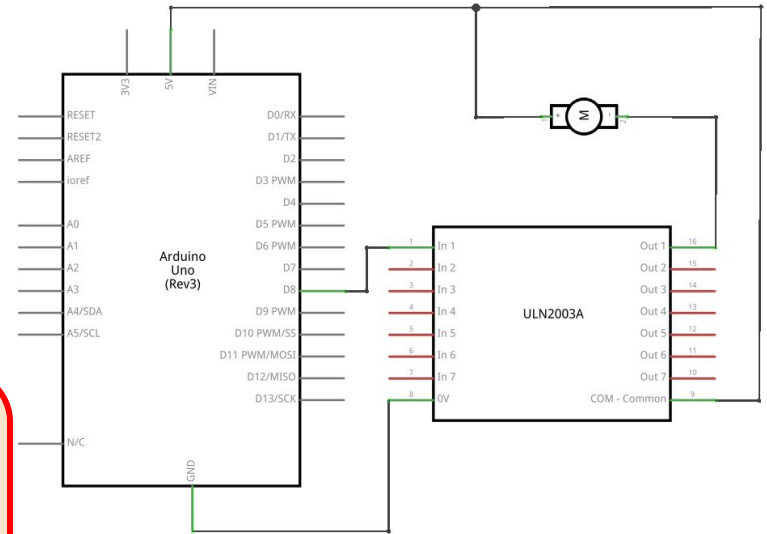


ACTIVADO SIMPLE DEL MOTOR

REAL



ESQUEMA



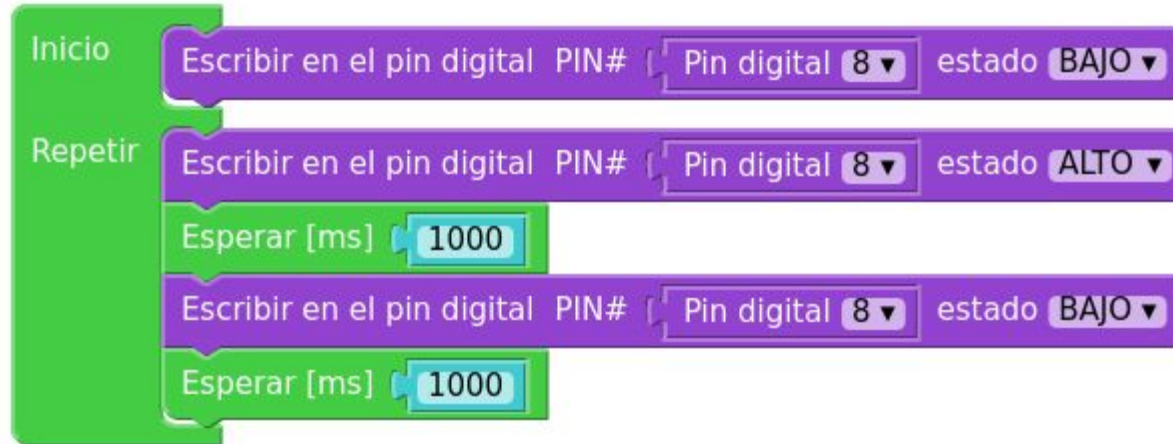
NOTA: El motor se conecta a los 5V de tensión del ARDUINO. Si requiere otra fuente de tensión externa, el pin 9 del ULN deberá conectarse a ella e interconectar todos los negativos al GND. Cada salida (colector par darlington) del ULN2003 admite 500mA máximo de corriente.

fritzing

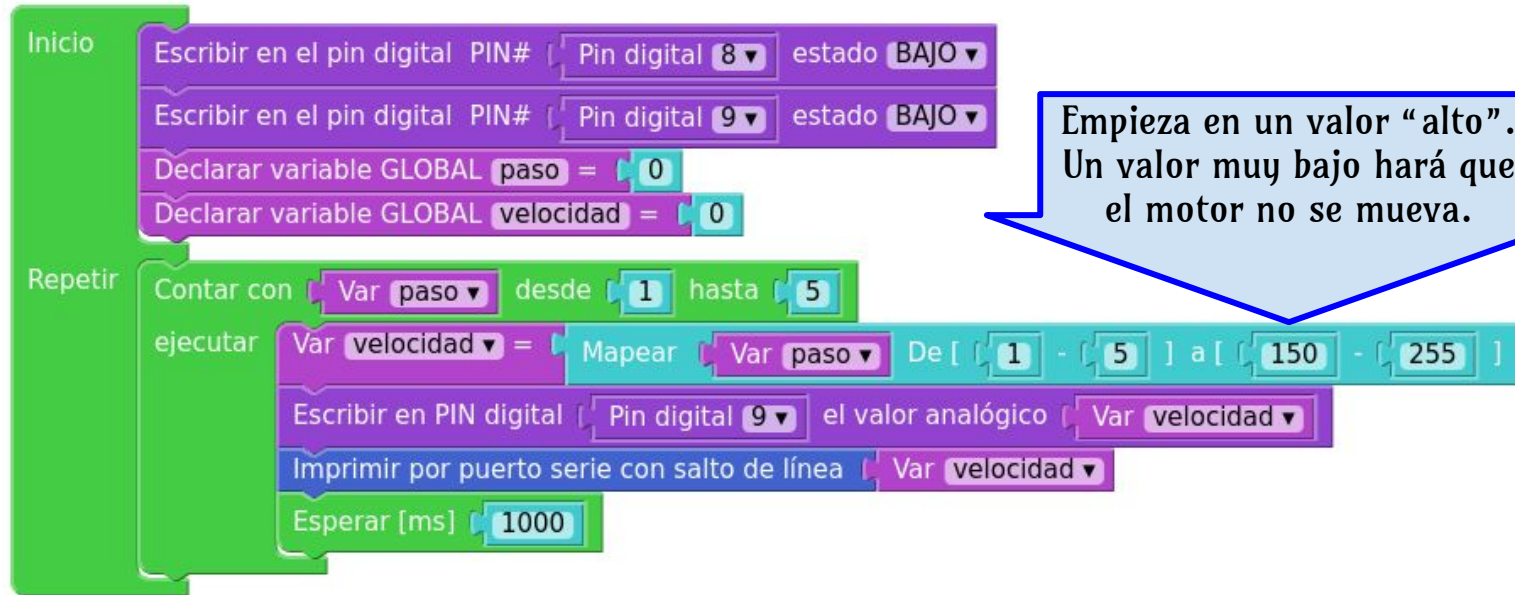
Aurelio Gallardo Rodríguez



ACTIVADO SIMPLE DEL MOTOR.



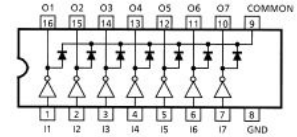
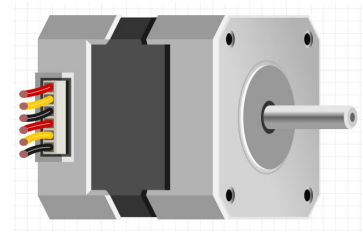
CONTROL DE LA VELOCIDAD DEL MOTOR



Empieza en un valor "alto".
Un valor muy bajo hará que
el motor no se mueva.

MOTORES PASO A PASO

(CON ULN2003)



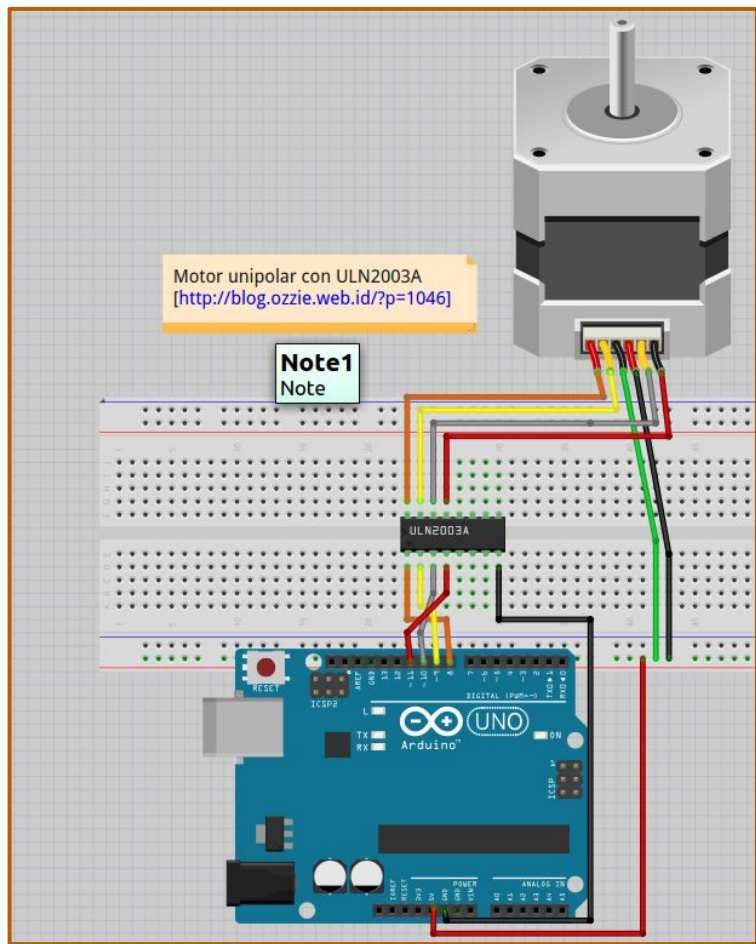
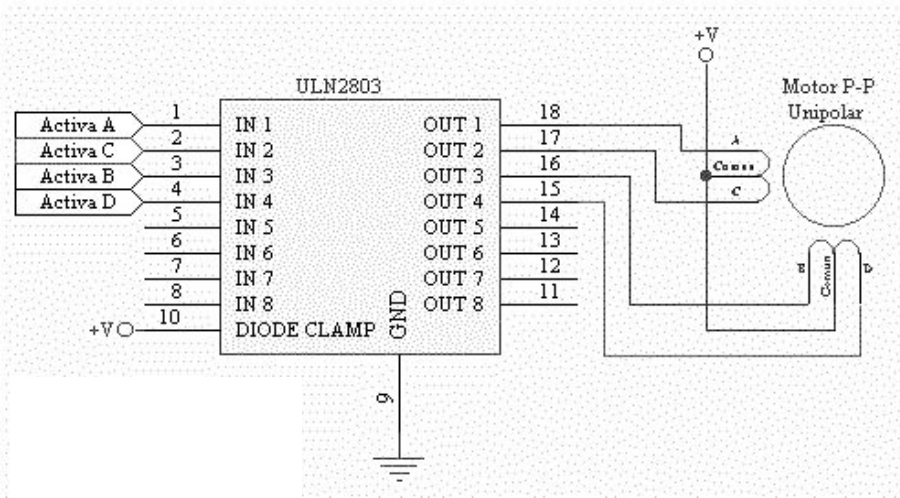
SOBRE MOTORES PASO A PASO.

No es nuestro objetivo estudiar a fondo este tipo de motores. Simplemente daremos alguna información sobre ellos. Tener en cuenta que:

1. En lo que sigue tendremos como ejemplo un motor paso a paso unipolar de 6 cables; podemos aprender más sobre ellos en esta web: <http://www.todorobot.com.ar/tutorial-sobre-motores-paso-a-paso-stepper-motors/>
2. En la web anterior, leer al final, si lo necesitáis, cómo identificar cada cable.
3. Usaremos para activarlos nuestra placa ARDUINO más un micro ULN2003.
4. Un motor paso a paso nos servirá para poder controlar con precisión la posición de nuestro motor así como la velocidad y sentido de rotación.
5. Los motores paso a paso se pueden encontrar fácilmente en algunos dispositivos electrónicos e informáticos, como impresoras, grabadoras o discos duros.
6. Se supondrá que se tienen identificados las conexiones. Los pines A, B, C y D se conectarán, a través del ULN2003, a los pines 8, 9, 10 y 11 de ARDUINO.

MOTOR PASO A PASO.



En el caso de usar una fuente externa, conectar dicha fuente al pin 9, así como a los cables comunes. El motor del esquema funciona con los 5 Volt. de Arduino.



Aurelio Gallardo Rodríguez



MOTOR PASO A PASO. SECUENCIA NORMAL.

PA-SO	Bobina A	Bobina B	Bobina C	Bobina D	
1	ON	ON	OFF	OFF	
2	OFF	ON	ON	OFF	
3	OFF	OFF	ON	ON	
4	ON	OFF	OFF	ON	

- Giro antihorario.
- Función `activarPaso` que recibe un paso de 0 a 3

```
★ activarPaso int cual
ejecutar
★ si
  Var cual = 0
  ejecutar
    Escribir en el pin digital PIN# Var A estado ALTO
  de lo contrario
    Escribir en el pin digital PIN# Var A estado BAJO
★ si
  Var cual = 0
  ejecutar
    Escribir en el pin digital PIN# Var B estado ALTO
  de lo contrario
    Escribir en el pin digital PIN# Var B estado BAJO
★ si
  Var cual = 1
  ejecutar
    Escribir en el pin digital PIN# Var C estado ALTO
  de lo contrario
    Escribir en el pin digital PIN# Var C estado BAJO
★ si
  Var cual = 2
  ejecutar
    Escribir en el pin digital PIN# Var D estado ALTO
  de lo contrario
    Escribir en el pin digital PIN# Var D estado BAJO
```

MOTOR PASO A PASO. SECUENCIA NORMAL.

```
Inicio
  Declarar variable GLOBAL A = 11
  Declarar variable GLOBAL B = 10
  Declarar variable GLOBAL C = 9
  Declarar variable GLOBAL D = 8
  Escribir en el pin digital PIN# Var A estado BAJO
  Escribir en el pin digital PIN# Var B estado BAJO
  Escribir en el pin digital PIN# Var C estado BAJO
  Escribir en el pin digital PIN# Var D estado BAJO
  Declarar variable GLOBAL paso = 0
  Declarar variable GLOBAL t = 1000

Repetir
  activarPaso
  cual Var paso
  Esperar [ms] Var t
  Var paso = Var paso + 1
  Var paso = Var paso % 4
```

- Variable t, tiempo de espera entre activaciones. ¡Cuidado! Una t muy pequeña o cero puede hacer que no funciones bien. En cualquier caso, experimenta.
- Declarar variables y decir en qué pin están conectadas.
- Empezar en el paso 0
- Sumar 1 al paso, y tomar el resto de la división entre cuatro; cuando paso = 3 al sumar 1 se convertirá en 4, pero al calcular el resto se volverá otra vez 0.

Pregunta: ¿Qué modificación o añadido harías al programa para que gire en el otro sentido?

Ampliación: diseña un programa, que, con dos botones, avance un poco el motor a izquierda o derecha.

MOTOR PASO A PASO. SECUENCIA WAVE DRIVE



PA-SO	Bobina A	Bobina B	Bobina C	Bobina D	
1	ON	OFF	OFF	OFF	
2	OFF	ON	OFF	OFF	
3	OFF	OFF	ON	OFF	
4	OFF	OFF	OFF	ON	

MOTOR PASO A PASO. SECUENCIA MEDIO PASO

```

activarPaso int cual
ejecutar
  si
    Var cual ≤ 1 o Var cual = 7
    ejecutar
      Escribir en el pin digital PIN# Var A estado ALTO
    de lo contrario
      Escribir en el pin digital PIN# Var A estado BAJO
  si
    Var cual ≤ 3 y Var cual ≠ 0
    ejecutar
      Escribir en el pin digital PIN# Var B estado ALTO
    de lo contrario
      Escribir en el pin digital PIN# Var B estado BAJO
  si
    Var cual ≤ 5 y Var cual ≥ 3
    ejecutar
      Escribir en el pin digital PIN# Var C estado ALTO
    de lo contrario
      Escribir en el pin digital PIN# Var C estado BAJO
  si
    Var cual ≥ 5
    ejecutar
      Escribir en el pin digital PIN# Var D estado ALTO
    de lo contrario
      Escribir en el pin digital PIN# Var D estado BAJO
  
```

```

Repetir
  cual 7 - Var paso
  activarPaso
  Esperar [ms] Var t
  Var paso = Var paso + 1
  Var paso = Var paso % 8
  
```

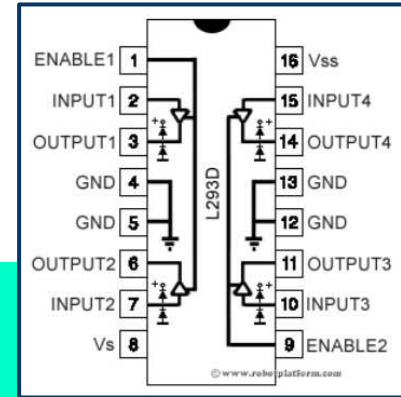
PA- SO	Bobina A	Bobina B	Bobina C	Bobina D	
1	ON	OFF	OFF	OFF	
2	ON	ON	OFF	OFF	
3	OFF	ON	OFF	OFF	
4	OFF	ON	ON	OFF	
5	OFF	OFF	ON	OFF	
6	OFF	OFF	ON	ON	
7	OFF	OFF	OFF	ON	
8	ON	OFF	OFF	ON	

Aurelio Gallardo Rodríguez

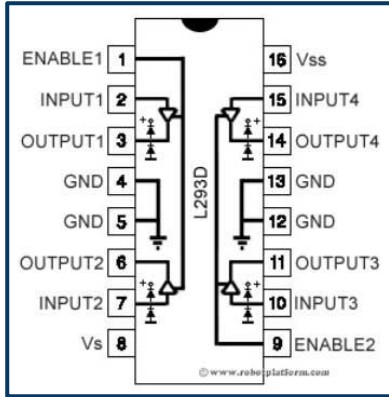


MOTORES CORRIENTE CONTINUA

(CON L293D)



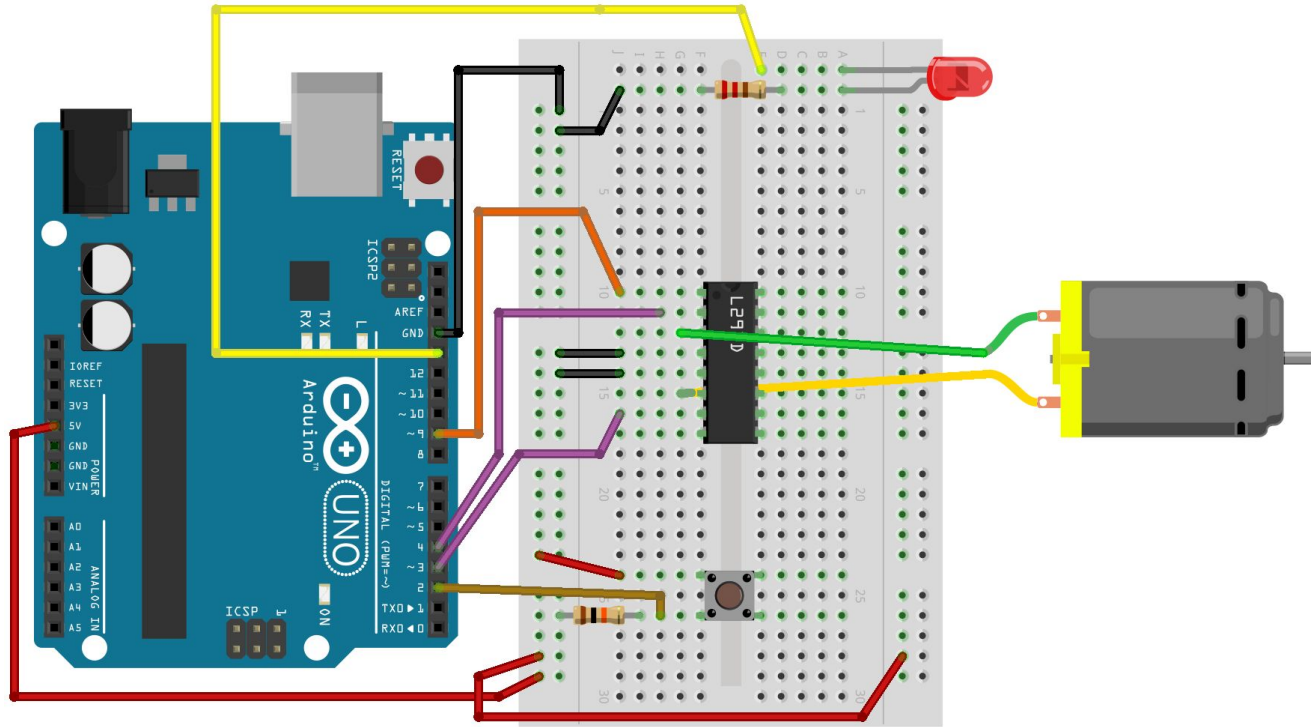
L293D



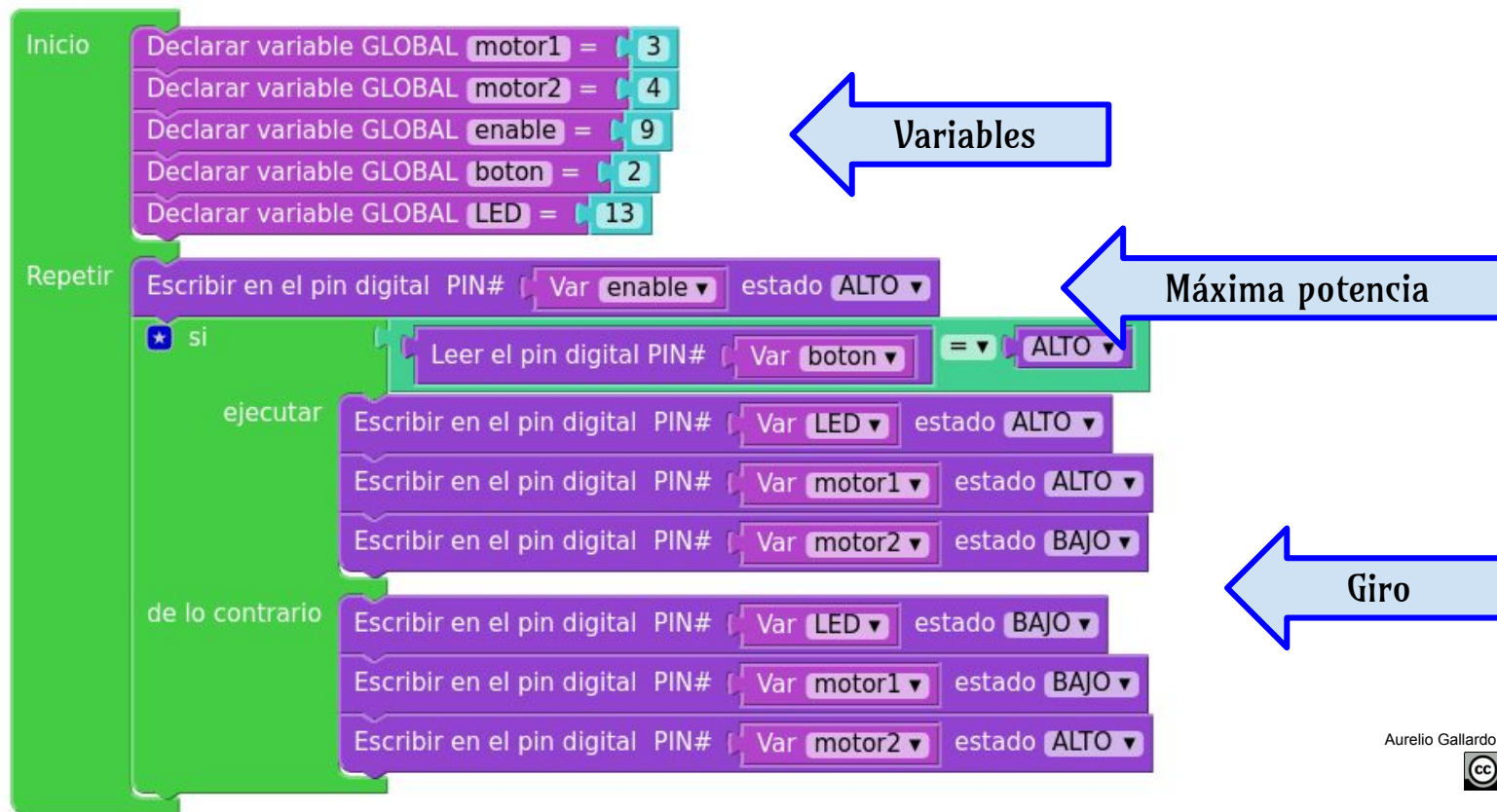
<https://alonsodub.wordpress.com/2012/06/08/control-de-motor-cc-velocidad-y-direccion/>

- Este microchip es capaz de controlar dos motores de corriente continua con un consumo de 600mA en cada motor, y una tensión entre 4,5V y 36V.
- Vs y Vss debe estar a la tensión de trabajo del motor. Si es una fuente externa habrá que conectar su positivo a ambos pines, y el negativo conectarlo con el GND de la placa.
- Para pequeñas potencias con motores de 5V, se puede probar con la tensión del Arduino.
- El pin ENABLE controla la velocidad, conectando a un pin de salida analógica.
- Los pines 2 (INPUT1) y 7 (INPUT2) controlan el sentido de giro. Para ello, uno tiene que estar en bajo y el otro en alto.
- Para el segundo motor, las conexiones son análogas.

CONEXIONADO



CAMBIAR SENTIDO DE GIRO PULSANDO UN BOTÓN



CAMBIO DE VELOCIDAD ENVIANDO POR PUERTO SERIE N^{OS} DEL 0 AL 9

```
Inicio
  Declarar variable GLOBAL motor1 = 3
  Declarar variable GLOBAL motor2 = 4
  Declarar variable GLOBAL enable = 9
  Declarar variable GLOBAL boton = 2
  Declarar variable GLOBAL LED = 13
  Declarar variable GLOBAL leer de tipo Entero = 0
  Declarar variable GLOBAL velocidad = 0

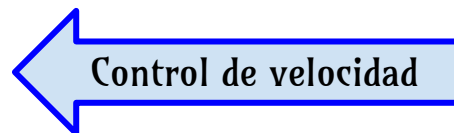
Repetir
  Puerto Serie Disponible
  ejecutar Var leer = Leer entero por el puerto serie

  * si
    Var leer >= 0 y Var leer <= 9
    ejecutar Var velocidad = Mapear Var leer De [ 0 - 9 ] a [ 0 - 255 ]

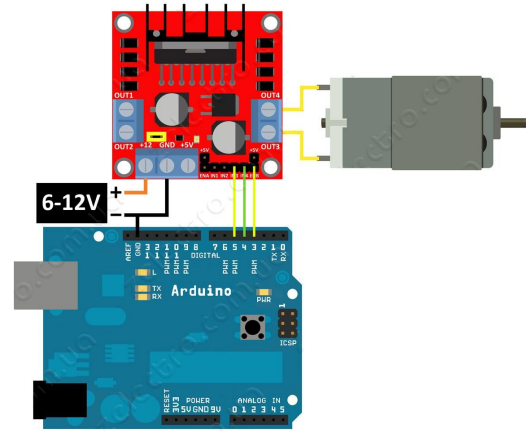
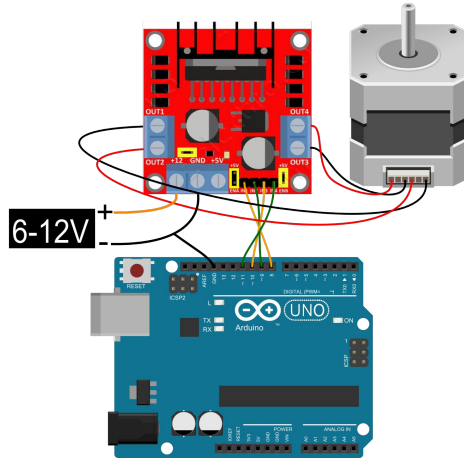
  Escribir en PIN digital Var enable el valor analógico Var velocidad
  Imprimir por puerto serie con salto de línea Var velocidad

  * si
    Leer el pin digital PIN# Var boton = ALTO
    ejecutar
      Escribir en el pin digital PIN# Var LED estado ALTO
      Escribir en el pin digital PIN# Var motor1 estado ALTO
      Escribir en el pin digital PIN# Var motor2 estado BAJO
    de lo contrario
      Escribir en el pin digital PIN# Var LED estado BAJO
      Escribir en el pin digital PIN# Var motor1 estado BAJO
      Escribir en el pin digital PIN# Var motor2 estado ALTO
```

- Controlado el giro mediante un botón conectado al pin 2.
- Velocidad controlada mediante un número introducido por teclado del 0 al 9. 0 → parada total y 9 → máxima potencia.



L298N-L293D



- <http://electronilab.co/tutoriales/tutorial-de-uso-driver-dual-l298n-para-motores-dc-y-paso-a-paso-con-arduino/>
- <http://www.instructables.com/id/Bi-Polar-Stepper-Motor-with-L293D-and-Arduino/>