## Mando

Controle la posición de un [servomotor](http://en.wikipedia.org/wiki/Servo_motor#RC_servos) RC (hobby) con su Arduino y un potenciómetro.

Este ejemplo hace uso de la [**biblioteca de servos**](https://www.arduino.cc/en/Reference/Servo) Arduino .

### Hardware requerido

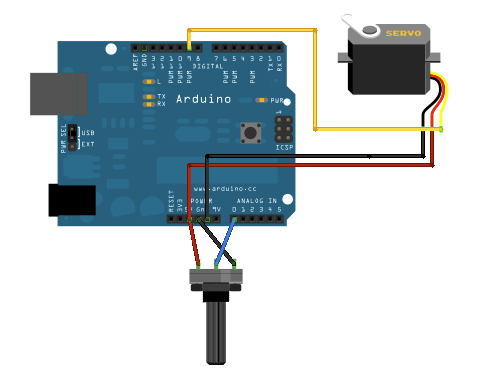
* Arduino o Tablero Genuino
* Servo motor
* Potenciómetro de 10k ohmios
* cables de conexión

### Circuito

Los servomotores tienen tres cables: alimentación, tierra y señal. El cable de alimentación suele ser rojo y debe conectarse al pin de 5 V en la placa Arduino o Genuino. El cable de tierra es típicamente negro o marrón y debe conectarse a un pin de tierra en el tablero. El pin de señal es típicamente amarillo o naranja y debe conectarse al pin 9 de la placa.

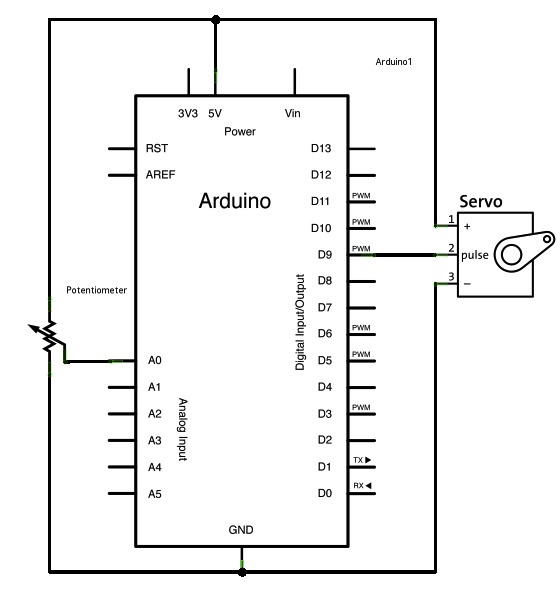
El potenciómetro debe estar cableado de modo que sus dos pines externos estén conectados a la alimentación (+ 5V) y a tierra, y su pin central esté conectado a la entrada analógica 0 en la placa.

haga clic en las imágenes para ampliar

[](https://www.arduino.cc/en/uploads/Tutorial/knob_BB.png)

imágenes desarrolladas usando [Fritzing](http://www.fritzing.org/" \t "_blank) . Para obtener más ejemplos de circuitos, consulte la [página del proyecto Fritzing](http://fritzing.org/projects/)

### Esquemático

[](https://www.arduino.cc/en/uploads/Tutorial/knob_schem.png)

### Código

/\*

Controlling a servo position using a potentiometer (variable resistor)

by Michal Rinott <http://people.interaction-ivrea.it/m.rinott>

modified on 8 Nov 2013

by Scott Fitzgerald

http://www.arduino.cc/en/Tutorial/Knob

\*/

#include <**Servo**.h>

**Servo** myservo;  // create servo object to control a servo

int potpin = 0;  // analog pin used to connect the potentiometer

int val;    // variable to read the value from the analog pin

void setup() {

 myservo.attach(9);  // attaches the servo on pin 9 to the servo object

}

void loop() {

 val = analogRead(potpin);            // reads the value of the potentiometer (value between 0 and 1023)

 val = map(val, 0, 1023, 0, 180);     // scale it to use it with the servo (value between 0 and 180)

 myservo.write(val);                  // sets the servo position according to the scaled value

 delay(15);                           // waits for the servo to get there

}

### Ver también

* [adjuntar](https://www.arduino.cc/en/Reference/ServoAttach) ()
* [escribir](https://www.arduino.cc/en/Reference/ServoWrite) ()
* [mapa](https://www.arduino.cc/en/Reference/Map) ()
* [analogRead](https://www.arduino.cc/en/Reference/AnalogRead) ()
* [Referencia de la biblioteca de servos](https://www.arduino.cc/en/Reference/Servo)
* [Barrido](https://www.arduino.cc/en/Tutorial/Sweep) : barra el eje de un servomotor de un lado a otro.

# **map()**

[Matemáticas]

### Descripción

Mapea un número de un rango a otro. Es decir, un valor de **fromLow** se **asignaría** a **toLow** , un valor de **fromHigh** a **toHigh** , valores intermedios a valores intermedios, etc.

No restringe los valores dentro del rango, porque los valores fuera de rango a veces son intencionales y útiles. La función constrain()puede usarse antes o después de esta función, si se desean límites a los rangos.

Tenga en cuenta que los "límites inferiores" de cualquiera de los rangos pueden ser mayores o menores que los "límites superiores", por lo que la map()función se puede utilizar para invertir un rango de números, por ejemplo

y = map(x, 1, 50, 50, 1);

La función también maneja bien los números negativos, por lo que este ejemplo

y = map(x, 1, 50, 50, -100);

También es válido y funciona bien.

La función map()usa matemática entera, por lo que no generará fracciones, cuando la matemática podría indicar que debería hacerlo. Los restos fraccionarios se truncan y no se redondean ni promedian.

### Sintaxis

map(value, fromLow, fromHigh, toLow, toHigh)

### Parámetros

value: el número a mapear.  
fromLow: el límite inferior del rango actual del valor.  
fromHigh: el límite superior del rango actual del valor.  
toLow: el límite inferior del rango objetivo del valor.  
toHigh: el límite superior del rango objetivo del valor.

### Devoluciones

El valor mapeado.

### Código de ejemplo

/\* Map an analog value to 8 bits (0 to 255) \*/

void setup() {}

void loop() {

int val = analogRead(0);

val = map(val, 0, 1023, 0, 255);

analogWrite(9, val);

}

### Apéndice

Para los matemáticamente inclinados, aquí está la función completa

long map(long x, long in\_min, long in\_max, long out\_min, long out\_max) {

return (x - in\_min) \* (out\_max - out\_min) / (in\_max - in\_min) + out\_min;

}

### Notas y advertencias

Como se mencionó anteriormente, la función map () usa matemática entera. Por lo tanto, las fracciones podrían suprimirse debido a esto. Por ejemplo, fracciones como 3/2, 4/3, 5/4 serán devueltas como 1 desde la función map (), a pesar de sus diferentes valores reales. Entonces, si su proyecto requiere cálculos precisos (p. Ej., Voltaje con precisión de 3 decimales), considere evitar map () e implementar los cálculos manualmente en su código.