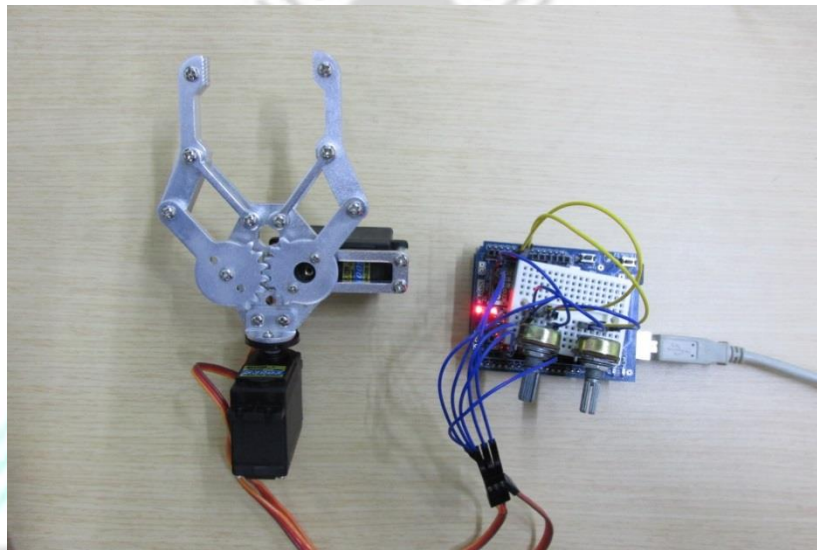


# Control de pinza robótica con Arduino

*Por William David Galvis*

Agosto 10 del 2013

Esta guía describe como controlar una pinza robótica con dos servomotores, usando un módulo de conversión análogo a digital (ADC) y de PWM del Arduino Duemilanove.



La pinza robótica consta de dos servomotores que generan los movimientos, el primero controla la apertura y cierre de las pinzas y el otro se encarga del giro sobre su mismo eje.

Para hacer que estas se muevan de acuerdo a los deseos del operador, se usan dos potenciómetros como dispositivos de mando, que al moverlos, harán que los servomotores lo hagan también.




Para que todo funcione como se describió arriba, se debe programar el Arduino, de tal forma que al cambiar el valor del voltaje que entregan los potenciómetros cuando son movidos, mande la orden de cambiar el ángulo de posición de los servomotores respectivos, esto se hace "leyendo" el voltaje que cambia en las conexiones de los potenciómetros y "escribiendo" su valor equivalente en la salida PWM, haciendo la siguiente relación: cuando el potenciómetro entrega 0 voltios, el




Calle: 41 No. 51 15, local 141, Teléfono: 232 77 84, E-Mail: [www. microkitselectronica.com](http://www.microkitselectronica.com)

Centro Comercial Paseo Bolívar, Carabobo con los Huesos, Medellín Antioquia, Colombia.

servomotor respectivo estará a 0° y si el voltaje que entrega el potenciómetro es 5 voltios, el servomotor se moverá a 180°.

### Lista de Elementos

	Referencia	Cantidad
	Pinza metálica simple REF: PMR	1
	Servo motor 995G REF:S995G	2
	Arduino Duemilanove REF: ARMILANOVE	1

	<p>Arduino prototype shield mini REF: ARSHMINI</p>	<p>1</p>
	<p>Cable Header 12cm REF: HD12</p>	<p>15</p>
	<p>Potenciometro 10k Ohm</p>	<p>2</p>

### CIRCUITO (Hardware)

Los servomotores son actuadores rotatorios (motores) acoplados a un sensor de posición que censa el ángulo en el que se encuentra el motor, ellos permiten el control preciso de los ángulos de posición, velocidad y aceleración, sus características y complejidad de control dependen del campo de aplicación, en

este caso, se usan dos servomotores destinados a la robótica con control de posición.

La conexión del servomotor se hace con tres líneas, identificadas por colores como se relaciona en la siguiente tabla.

Color	Nombre	Descripción
Marrón	GND	Cable que se debe conectar a la referencia, (-) o tierra.
Rojo	VCC	Voltaje de alimentación positivo.
Amarillo	Señal	Es donde se ingresa la señal PWM que sale del Arduino para controlar el servo.

TABLA 1

Siguiendo la tabla 1, se conectan los dos servomotores al Arduino, teniendo en cuenta que las señales de control salen de los pines D6 y D5, una para cada servo, como se observa en el esquema de la figura 1 y el diagrama de montaje en la Proto Board de la figura 2.

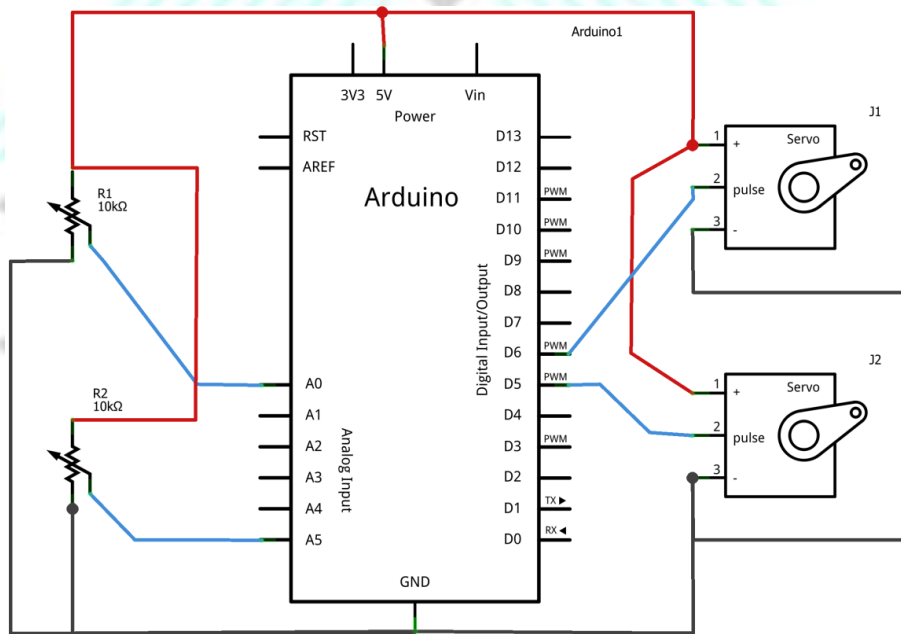


FIGURA 1

En la siguiente imagen (figura 2), se detalla el diagrama final de montaje en la Proto Board.

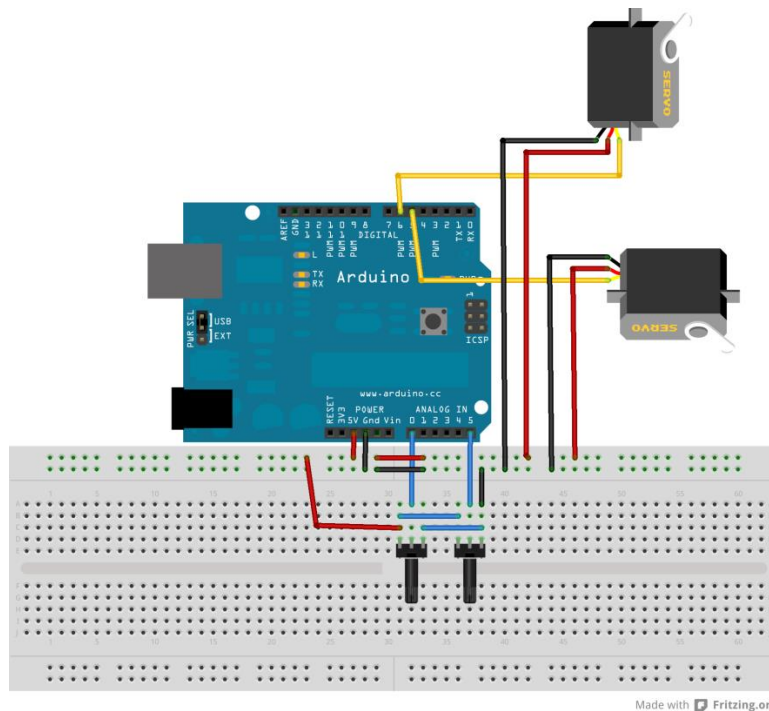


FIGURA 2

Como se observa en las dos figuras anteriores los potenciómetros van conectados de la siguiente forma, los dos pines externos a voltaje +5voltios del Arduino y a GND, el conector del medio va a la entrada para la lectura del voltaje análogo del Arduino, o módulo ADC, con múltiples canales, de los cuales se elige la entrada análoga 0 y la 5.

Una vez montado el circuito se procede a realizar la programación, se ejecuta el entorno de desarrollo de Arduino y se escribe el siguiente código:

```
#include <Servo.h> // se llama la librería servo
Servo servo1; // Se crea un objeto servomotor para controlarlo
Servo servo2; // Se crea otro objeto llamado servo2

void setup()
{
  servo1.attach(5); // Se asigna el pin 5 el objeto servo
                  // para controlar el servomotor servo1
  servo2.attach(6); // se le asigna el pin 6 al servomotor servo2
}

void loop() // ciclo infinito
{
  int angle1 = analogRead(0); // Define la variable angle1 y la
```

```
// Inicializa al valor leído en el pin análogo 0

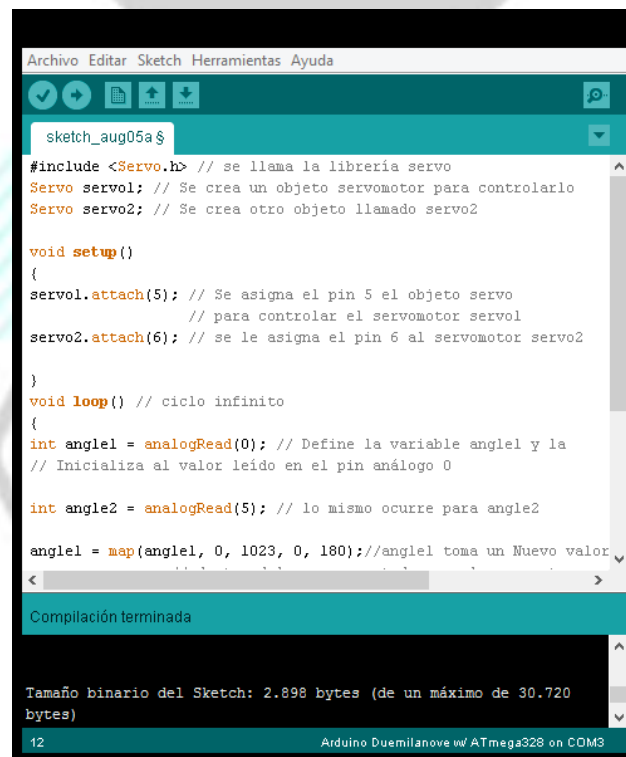
int angle2 = analogRead(5); // lo mismo ocurre para angle2

angle1 = map(angle1, 0, 1023, 0, 180); // angle1 toma un Nuevo valor
// dentro del rango aceptado por el servomotor

angle2 = map(angle2, 0, 1023, 0, 180);

servo1.write(angle1); // Se usa la función write para
// escribir el valor escalizado del ángulo al pin de salida
// PWM asignado al objeto servo1

servo2.write(angle2);
delay(15);
}
```



```
Archivo Editar Sketch Herramientas Ayuda

sketch_aug05a $

#include <Servo.h> // se llama la librería servo
Servo servo1; // Se crea un objeto servomotor para controlarlo
Servo servo2; // Se crea otro objeto llamado servo2

void setup()
{
  servo1.attach(5); // Se asigna el pin 5 el objeto servo
                  // para controlar el servomotor servo1
  servo2.attach(6); // se le asigna el pin 6 al servomotor servo2
}

void loop() // ciclo infinito
{
  int angle1 = analogRead(0); // Define la variable angle1 y la
  // Inicializa al valor leído en el pin análogo 0

  int angle2 = analogRead(5); // lo mismo ocurre para angle2

  angle1 = map(angle1, 0, 1023, 0, 180); // angle1 toma un Nuevo valor

  servo1.write(angle1);
  servo2.write(angle2);
  delay(15);
}

Compilación terminada

Tamaño binario del Sketch: 2.898 bytes (de un máximo de 30.720
bytes)

12 Arduino Duemilanove w/ ATmega328 on COM3
```

FIGURA 3

Arduino incluye una librería para Modulación de Ancho de pulso, para poder usarla se debe incluir al programa usando la macro `#include`, seguida del nombre de la librería encerrado entre los signos mayor y menor que `<Servo.h>`.

Al incluir esta librería se pueden usar sus funciones que permitirán controlar el servomotor fácilmente.

Esta librería soporta hasta 12 servomotores en muchas tarjetas Arduino y 48 en Arduino Mega.

Cuando se añade esta librería y se escribe la palabra `Servo` se nota inmediatamente que esta sale de color naranja (Figura 3) lo que indica que es una palabra reservada, que para este caso, pertenece a la librería `servo.h` y su función es definir un objeto al cual se le va a asignar un canal de salida PWM del Arduino. La forma en como se le asigna un canal al objeto `Servo` creado es como se muestra en los comentarios del código usando la función: `.attach(x)`; donde `x` es el canal de PWM en el cual se quiere que salga la señal de control.

```
int angle1 = analogRead(0);
```

En esta línea de código se define la variable entera (`int`), `angle1` que tomará como valor la lectura hecha por el ADC, al igualarlo a la función `analogRead(0)`; donde el 0 es el canal análogo elegido para la lectura del voltaje de uno de los potenciómetros. Lo mismo ocurre para el ángulo 2.

Como el valor entregado por el microcontrolador interno del Arduino para representar los voltajes leídos no encajan con el rango de números que se debe escribir a la función que escribe la señal PWM para los servomotores, entonces este valor de voltaje debe ser escalizado o convertido a otra escala de números que represente el ángulo al que se quiere que esté el servo, para ello se usa la función: `angle1 = map(angle1, 0, 1023, 0, 180)`; donde la variable `angle1` tomará otro valor linealmente proporcional al que tenía, es decir esta función actúa como una regla de tres, la relación entre estos valores se entenderá mejor en la siguiente tabla.

Voltaje entregado por el potenciómetro	Valor entregado por el Microcontrolador del Arduino (ATMEGA328)	Valor entregado a la función que escribe la señal PWM al servo motor
0v	0	0°
5v	1023	180°

TABLA 2

Finalmente se usa la función para escribir el valor leído al PWM, seguida de un retardo para garantizar que el servo se pueda mover a la posición programada antes de que se le indique una nueva y así mover el servomotor.

```
servo1.write(angle1);
```



FIGURA 4

Una vez terminado el programa, se procede a descargarlo al Arduino, haciendo click en el botón cargar y teniendo conectado el Arduino al computador, si ocurre algún problema vale la pena asegurarse que si haya reconocido la tarjeta buscando en herramientas y puerto serial, si aparece como en la imagen 5<sup>a</sup> (superior), podría ser que no está correctamente conectado el dispositivo o no se tiene instalado el driver del chip FTDI para la conversión serial USB, si todo está bien se vería como la figura 5b (inferior).

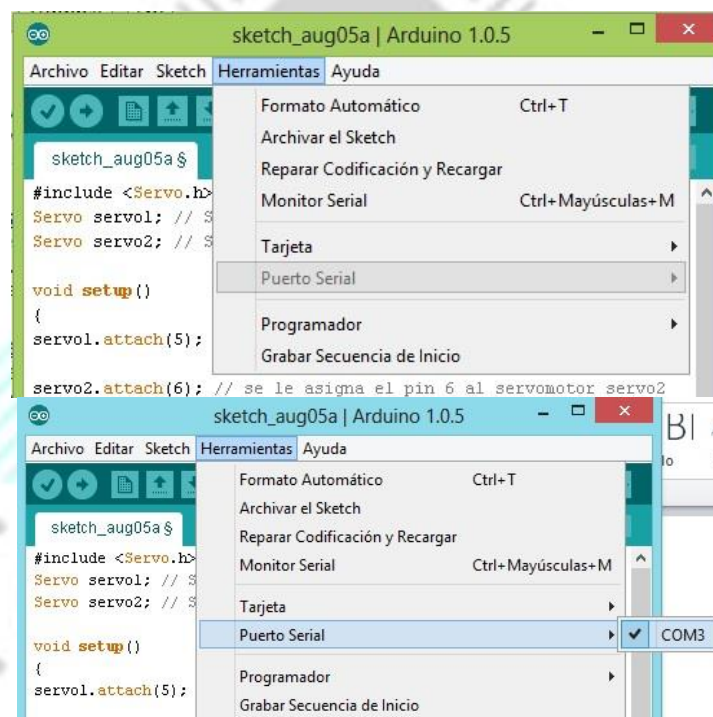


FIGURA 5

Siguiendo los pasos anteriores todo queda listo para disfrutar del movimiento de la pinza robótica MICROKITS con el Arduino Duemilanove.

## REFERENCIAS

- Librería para Servomotores de Arduino: <http://arduino.cc/en/Reference/Servo>  
 Ejemplo para manejo de Servomotores: <http://arduino.cc/en/Tutorial/Knob>  
 Video de respaldo [http://www.youtube.com/watch?feature=player\\_embedded&v=rWiQAUYRspk](http://www.youtube.com/watch?feature=player_embedded&v=rWiQAUYRspk)  
 Control Pinza Robótica con Aduino by [William Galvis](#) is licensed under a [Creative Commons Reconocimiento-CompartirIgual 3.0 Unported License](#).