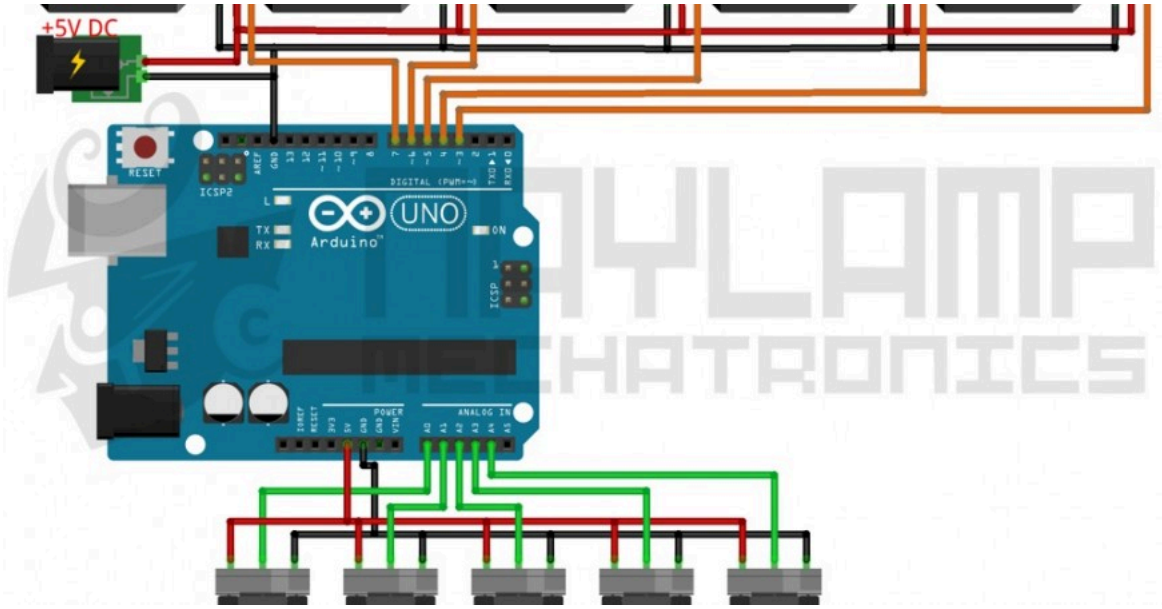
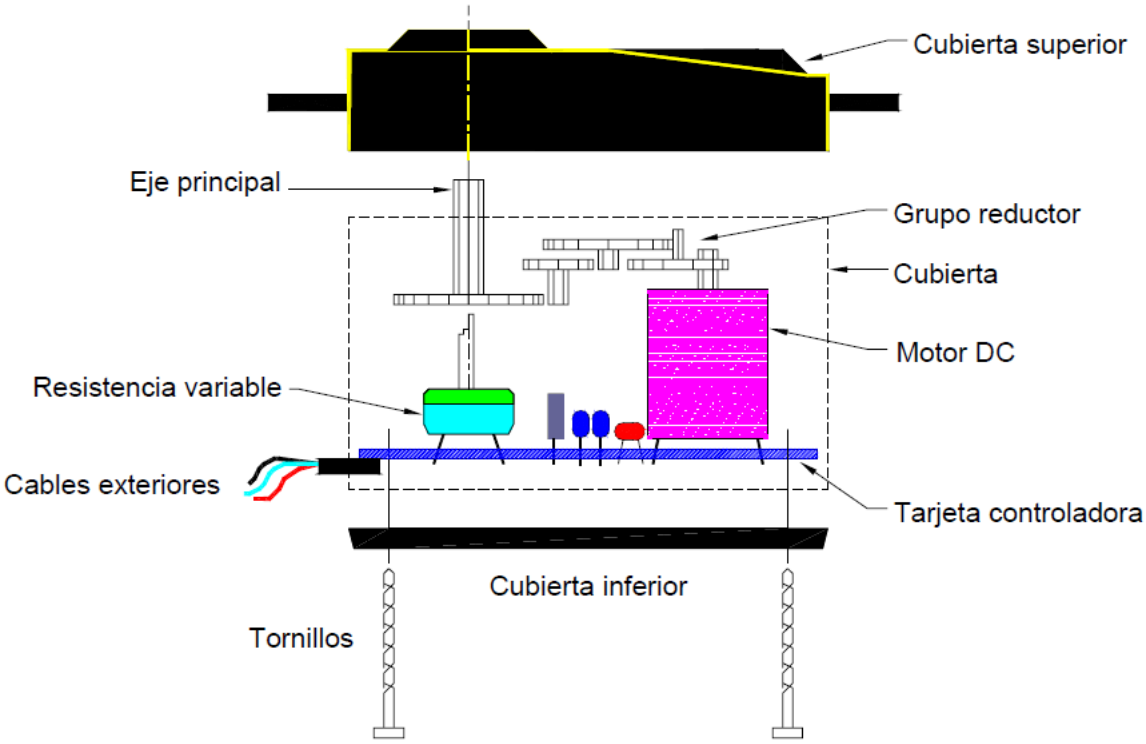
TUTORIAL USO DE SERVOMOTORES CON ARDUINO.

342405



En este tutorial se explicará el funcionamiento y características de un servomotor, los diferentes tipos que existen y como programar uno o varios servos.

Un servomotor o comúnmente llamado servo, es un motor DC con la capacidad de ubicar su eje en una posición o ángulo determinado, internamente tiene una caja reductora la cual le aumenta el torque y reduce la velocidad, un potenciómetro encargado de sensar la posición del eje y una pequeña tarjeta electrónica que junto al potenciómetro forman un control de lazo cerrado.

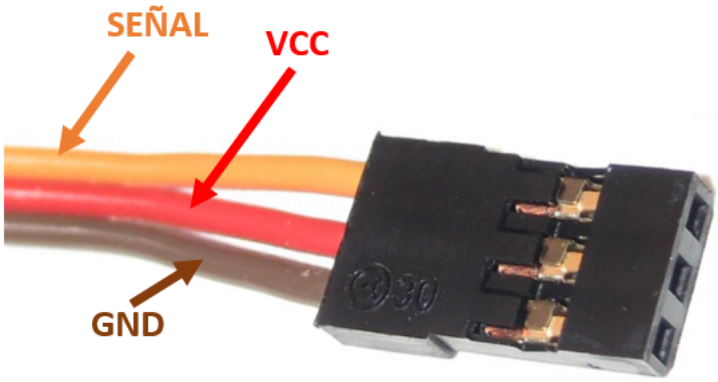


En el mercado existen diferentes modelos de servomotores, la principal diferencia entre ellos es el torque. Este punto es bueno tenerlo claro para elegir el servomotor adecuado. Es mejor elegir un servo con torque superior al que requerimos, pues el consumo de corriente es proporcional a la carga. En cambio sí sometemos un servomotor a cargas superiores a su torque, corremos el riesgo de malograr tanto la parte mecánica (engranes) y eléctrica del servo, incluso podemos generar ruido en la fuente.

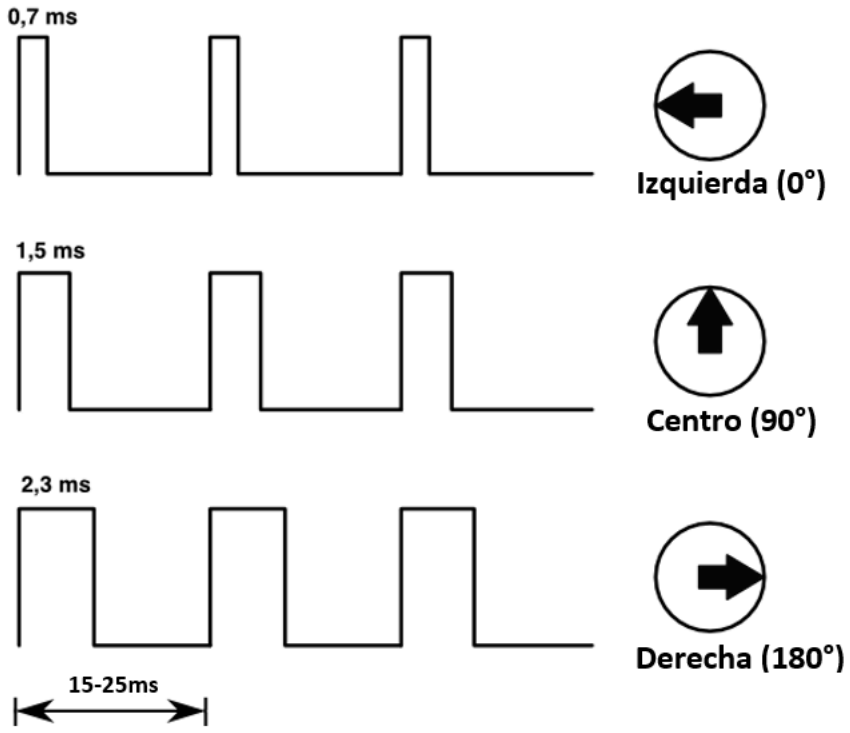
Servomotor	Modelo	Voltaje	Torque
	SG90	3V-7.2V	1-1.6 Kg/cm
	SG-5010	4.8V-6V	3.5K-6.5 Kg/cm
	MG995	4.8V – 7.2V	8.5-10 kg/cm
MG946	MG946	4.8V – 7.2V	10.5-14 Kg/cm

Servomotor	Modelo	Voltaje	Torque
	RDS3115	5V – 7.2V	13.5- 15Kg/cm

Todos los servos usados para robótica, tiene un conector de 3 cables. VCC (rojo), GND (Marrón) y Señal (Naranja):



La señal o dato que hay que enviarle al servo es una señal de PWM donde el tiempo en alto es equivalente al ángulo o posición del servo. Estos valores pueden variar y van desde 0.5 a 1 milisegundo para la posición 0° y 2 a 2.4 milisegundos para la posición de 180°, el periodo de la señal debe ser cercano a 20 milisegundos.



Librería servo de Arduino

El IDE Arduino trae una librería completa para el control de servomotores, la documentación completa lo encontramos en su página oficial: Servo Library

Expliquemos brevemente las funciones de la librería.

La biblioteca Servo admite hasta 12 motores en la mayoría de las placas Arduino y 48 en el Arduino Mega. En las placas que no sean los de Mega, el uso de la biblioteca desactiva la funcionalidad PWM en las patillas 9 y 10. En el Arduino Mega, hasta 12 servos pueden ser utilizados sin interferir con la funcionalidad PWM, pero si se usan de 12 a 23 motores desactivará el PWM en los pines 11 y 12.

A continuación se muestran las funciones de la librería Servo:

attach(Pin)

Establece el pin indicado en la variable servo. Ej: *servo.attach(3);*

attach(Pin,min,max)

Establece el pin indicado en la variable servo, considerando min el ancho de pulso para la posición 0° y máx. el ancho de pulso para 180°.Ej: *servo.attach(3,900,2100);*

write(angulo)

Envía la señal correspondiente al servo para que se ubique en el ángulo indicado, ángulo es un valor entre 0 y 180°. Ej: *servo.write(45);*

writeMicroseconds(tiempo)

Envía al servo el ancho de pulso=tiempo en microsegundos. Ej: *servo.writeMicroseconds(1500);*

read()

Lee la posición actual del servo, devuelve un valor entre 0 y 180. Ej: *angulo=servo.read();*

attached(Pin)

Verifica si la variable servo está unido al pin indicado, devuelve true o false. Ej: *if(servo.attached(3));*

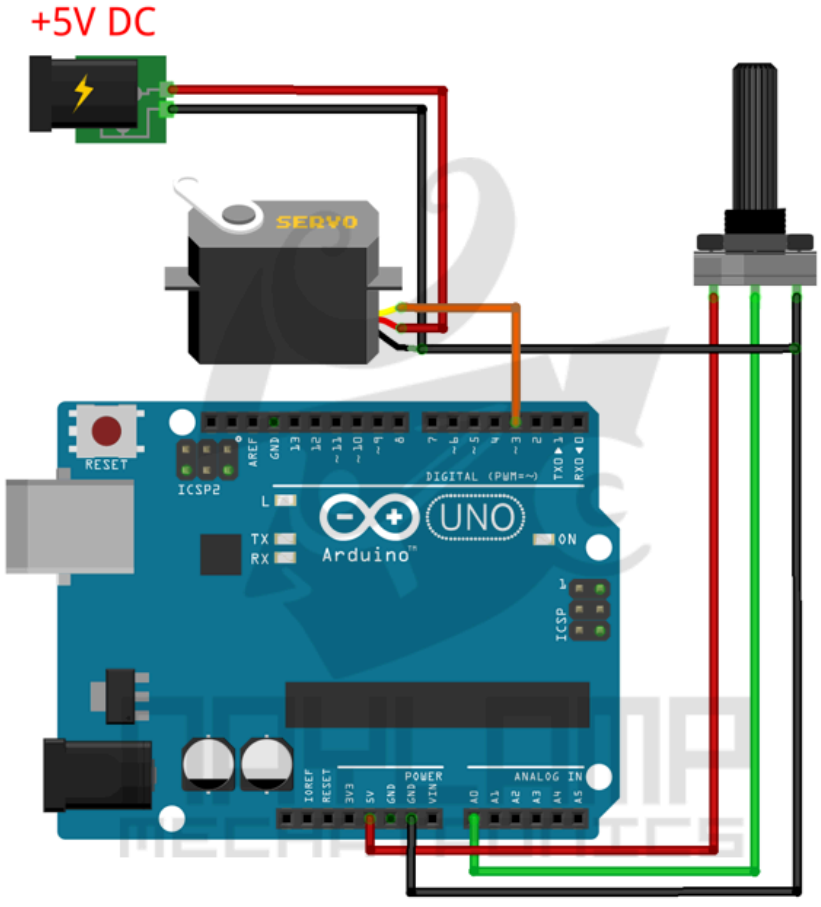
detach(pin)

Separa la variable Servo del pin indicado. Ej: *servo.detach(3)*;

Explicado lo anterior empecemos con la parte práctica realizando algunos ejemplos.

1. Control de un servomotor por ángulo:

Para esto realizaremos las siguientes conexiones:



**La alimentación del motor puede ser la misma que el Arduino siempre y cuando la fuente soporte la potencia del servo y sea de 5V. Los 5V del USB solo soporta un servo SG90, más servos o de otro tipo se necesita usar una fuente externa.*

Luego de realizar las conexiones procederemos a cargar("subir") el siguiente programa:

```
//LIBRARIES:
#include <Servo.h>

Servo myservo;  //creamos un objeto servo
int angulo;

void setup(){
  myservo.attach(3);  // asignamos el pin 3 al servo.
  Serial.begin(9600); // iniciamos el puerto serial
}


void loop() {

  angulo= 0;
  myservo.write(angulo);
  Serial.print("ángulo:  ");
  Serial.println(angulo);
  delay(2000);

  angulo= 90;
  myservo.write(angulo);
  Serial.print("ángulo:  ");
  Serial.println(angulo);
  delay(2000);

  angulo= 180;
  myservo.write(angulo);
  Serial.print("ángulo:  ");
  Serial.println(angulo);
  delay(2000);

  angulo= 90;
  myservo.write(angulo);
  Serial.print("ángulo:  ");
  Serial.println(angulo);
  delay(2000);
}
```

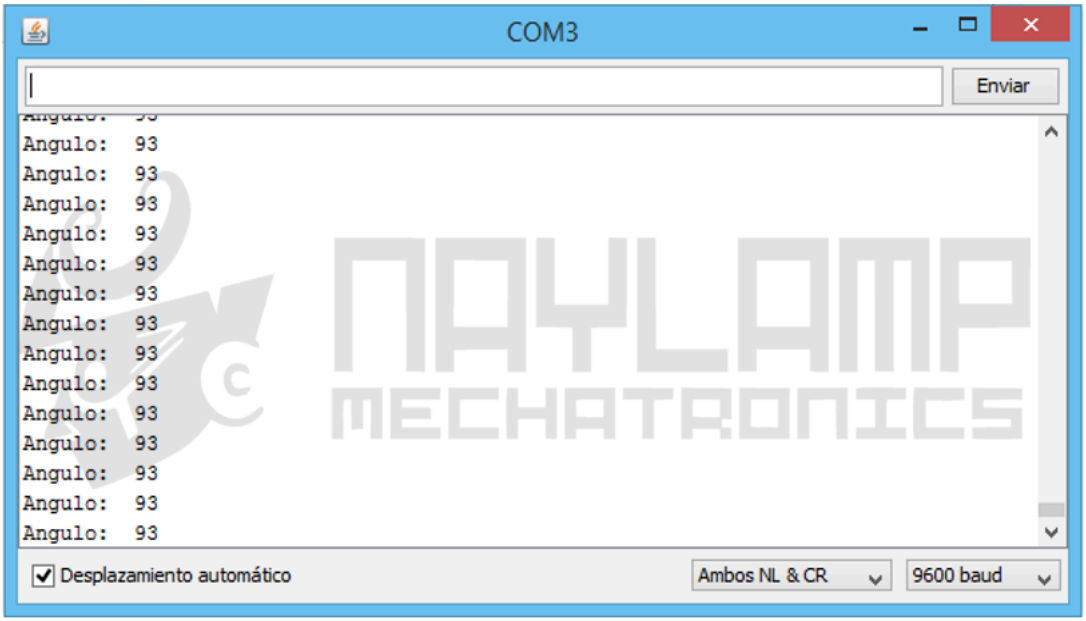
2. Control de un servomotor a través de un potenciómetro:

En este ejemplo controlaremos la posición del servo por medio de un potenciómetro, donde el servo se posicionará en función de la posición del potenciómetro.

```
#include <Servo.h>

Servo myservo;  //creamos un objeto servo

void setup() {
  myservo.attach(3);  // asignamos el pin 3 al servo.
   Serial.begin(9600);
}

void loop() {
  int adc = analogRead(A0);  // realizamos la lectura del potenciometro
  int angulo = map(adc, 0, 1023, 0, 180);  // escalamos la lectura a un valor entre 0 y 180
  myservo.write(angulo);  // enviamos el valor escalado al servo
  Serial.print("ángulo:  ");
  Serial.println(angulo);
  delay(10);
}
```

Como se observa en el código, básicamente se hace la lectura del potenciómetro y se lo escala entre valores de 0° y 180° para después enviarlo al servomotor.

Una vez cargado el programa el servo debe moverse cada vez que movemos le potenciómetro, y si abrimos el **Monitor Serie de Arduino IDE** debería mostrar el ángulo del servomotor.



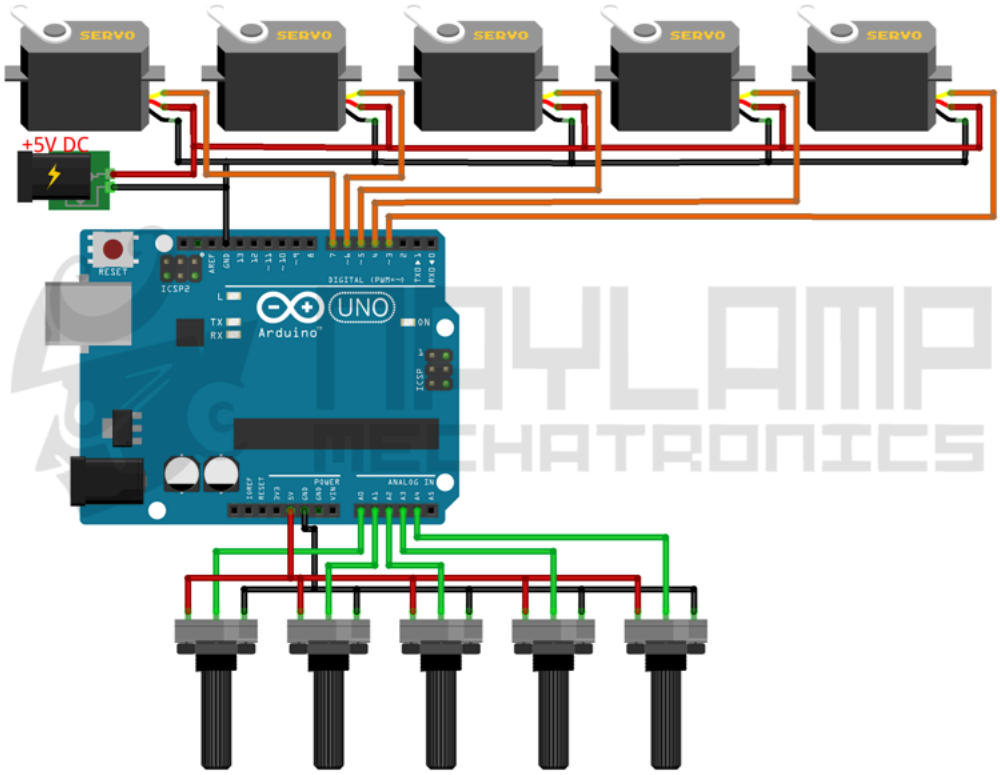
Si la poción del servomotor no coincide con el ángulo correspondiente, es porque el ancho del pulso que maneja la librería no coincide con los del servo, para esto es necesario indicarle el ancho de pulso min(para 0°) y el máximo (para 180°), la única modificación seria en la siguiente linea:

```
myservo.attach(3,900,2100);  // asignamos el pin 3 al servo, 900->0° y 2100->180°
```

Deben cambiar los valores 900 y 2100 hasta lograr las posiciones correctas.

3. Controlando varios Servomotores con potenciómetros:

Este ejemplo es una extensión del ejemplo anterior, las conexiones serían las siguientes

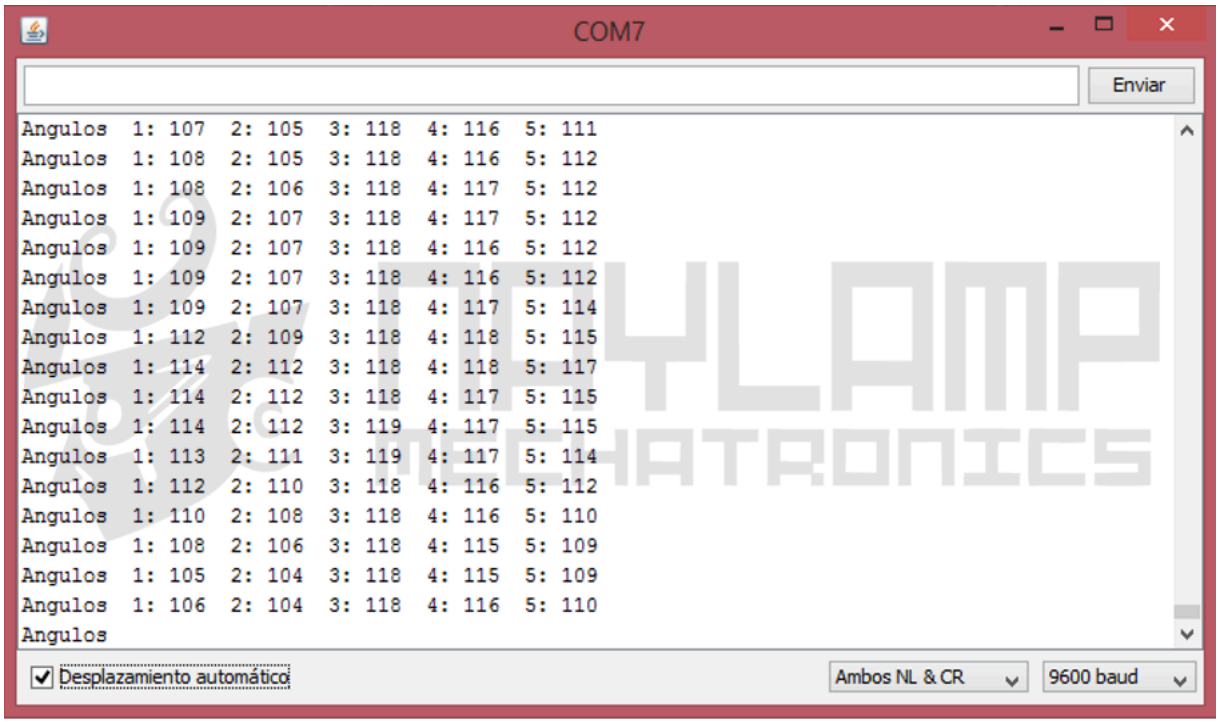


El código se muestra a continuación:

```
#include <Servo.h>

Servo servo1;  //creamos un objeto servo
Servo servo2;  //creamos un objeto servo
Servo servo3;  //creamos un objeto servo
Servo servo4;  //creamos un objeto servo
Servo servo5;  //creamos un objeto servo
void setup()
{
  servo1.attach(3);  // asignamos el pin al servo.
  servo2.attach(4);  // asignamos el pin al servo.
  servo3.attach(5);  // asignamos el pin al servo.
  servo4.attach(6);  // asignamos el pin al servo.
  servo5.attach(7);  // asignamos el pin al servo.
   Serial.begin(9600);
}

void loop()
{
  int angulo1 = map(analogRead(A0), 0, 1023, 0, 180);  // escalamos la lectura a un valor entre 0 y 180
  int angulo2 = map(analogRead(A1), 0, 1023, 0, 180);  // escalamos la lectura a un valor entre 0 y 180
  int angulo3 = map(analogRead(A2), 0, 1023, 0, 180);  // escalamos la lectura a un valor entre 0 y 180
  int angulo4 = map(analogRead(A3), 0, 1023, 0, 180);  // escalamos la lectura a un valor entre 0 y 180
  int angulo5 = map(analogRead(A4), 0, 1023, 0, 180);  // escalamos la lectura a un valor entre 0 y 180
  servo1.write(angulo1);  // enviamos el valor escalado al servo.
  servo2.write(angulo2);  // enviamos el valor escalado al servo.
  servo3.write(angulo3);  // enviamos el valor escalado al servo.
  servo4.write(angulo4);  // enviamos el valor escalado al servo.
  servo5.write(angulo5);  // enviamos el valor escalado al servo.
  //----Enviamos los ángulos serialmente--------------
  Serial.print("Angulos  1: ");
  Serial.print(angulo1);
  Serial.print("  2: ");
  Serial.print(angulo2);
  Serial.print("  3: ");
  Serial.print(angulo3);
  Serial.print("  4: ");
  Serial.print(angulo4);
  Serial.print("  5: ");
  Serial.println(angulo5);
  delay(10);
}
```

Como se observa es similar al primer ejemplo, se necesita declarar una variable (objeto) servo para cada servomotor.

Al abrir el **Monitor Serie de Arduino IDE** deben de visualizarse los ángulos correspondientes para cada servo



Se puede eliminar las últimas líneas de código si no necesitan visualizar los ángulos a través del monitor serial.

PRODUCTOS RELACIONADOS

En Stock

Micro Servo SG90 1.5Kg

SKU: 000021

S/ 10,00

ARTÍCULOS RELACIONADOS

Usando ESP8266 con el IDE de Arduino

66 427311

21 COMENTARIOS

Bi****** 10/10/2023 Responder