

# Control de servomotores

Una de las aplicaciones más utilizadas de los sistemas de control por ordenador y en la robótica están asociados con los motores, que permiten accionar o mover otros componentes, como puertas, barreras, válvulas, ruedas, etc. Uno de los tipos que vamos a ver en este capítulo son los servos, hay de dos tipos:

- El **servomotor** o **servos convencionales** que posee la capacidad de posicionar su eje en un ángulo determinado entre 0 y 180 grados en función de una determinada señal.
- **Servo de rotación continua** Son servos por fuera igual que los anteriores, pero pueden girar 360º y se controlan por tiempo

Por defecto cuando se dice **servo**, es un **servomotor o servo convenciona**

## Servos de rotación continua



Para controlar un servo de rotación continua, las instrucciones a realizar son :

- Incluye la librería de servos **#include <Servo.h>**
- Declaras una variable servo **Servo myservo;** //puedes poner el nombre que quieras p.e. miservo
- En *setup()* tienes que decir a qué pin está conectado **myservo.attach(9);** //por ejemplo pin 9
- Y en *loop()*
  - **myservo.write(90);** //significa servo parado
  - **myservo.write(180);** //significa servo funcionando al 100% en el sentido de las agujas del reloj
  - **myservo.write(0);** //significa servo funcionando al 100% en el sentido contrario de las agujas del reloj

Mira el vídeo, esta realizado con otra shield ECHIDNA y con bloques (curso Echidna <https://libros.catedu.es/books/arduino-con-echidna-y-mblock-scratch> ) fíjate como:

- Los extremos 0º y 180º es a máxima velocidad, pero un sentido u otro.
- 90º es parado.
- Un valor intermedio es menos velocidad (se ve el ejemplo 80º y 100º) -
- Si tiene deriva, (cosa frecuente) tienen un potenciómetro para ajustar.

<https://www.youtube.com/embed/Z-5SerXmRY0>

Si quieres saber más sobre servomotores te recomendamos estas paginas del Zaragozano Luis LLamas: [Servomotores](#) convencionales y Servomotores de [rotación continua](#)

## Servomotores o servos convencionales



Los servos son un tipo especial de motor en el que se añade una circuito lógico electrónico que permite un control mucho más preciso que a un motor normal de corriente continua. Esto les permite posicionar el eje en un ángulo determinado.

El hardware interno se compone de un potenciómetro y un circuito integrado que controlan en todo momento los grados que gira el motor. De este modo, en nuestro caso, desde Arduino, usando las salidas digitales PWM podremos controlar fácilmente un servo. Lo ideal es conectarlo a 6V pero trabajan bien en los 5V del Arduino.

Hay muchos modelos, en robótica educativa cuestan entre 1-5€, el más común es el SG90, muy barato, pero tiene muy poca fuerza, el MG90S tiene algo más, si queremos algo más, ya tiene que ser el MG996R pero ya este modelo **NO se puede conectar directamente al Arduino**, el pico de energía que necesita, provoca el reinicio de la placa. Incluso varios pequeños SG90.

Las instrucciones son las mismas que los servos de rotación continua, pero los valores que se proporcionan son los grados que se desean.

- Incluye la librería de servos **#include <Servo.h>**

- Declaras una variable servo **Servo myservo;** //puedes poner el nombre que quieras p.e. miservo
- En *setup()* tienes que decir a qué pin está conectado **myservo.attach(9);** //por ejemplo pin 9
- Y en *loop()*
  - **myservo.write(90);** //Posición 90º (posición por defecto)
  - **myservo.write(180);** //Posición 180º
  - **myservo.write(0);** // Posición 0º

La instrucción *myservo.write(angulo)* envía por el pin digital declarado en *myservo.attach()* pulsos cuadrados de 50Hz y de anchura el estado alto proporcional al ángulo que se desea.

- Un pulso de 0.5-1ms es 0º
- Un pulso de 1.5 ms es 90º
- Un pulso de 2-2.5ms es 180º

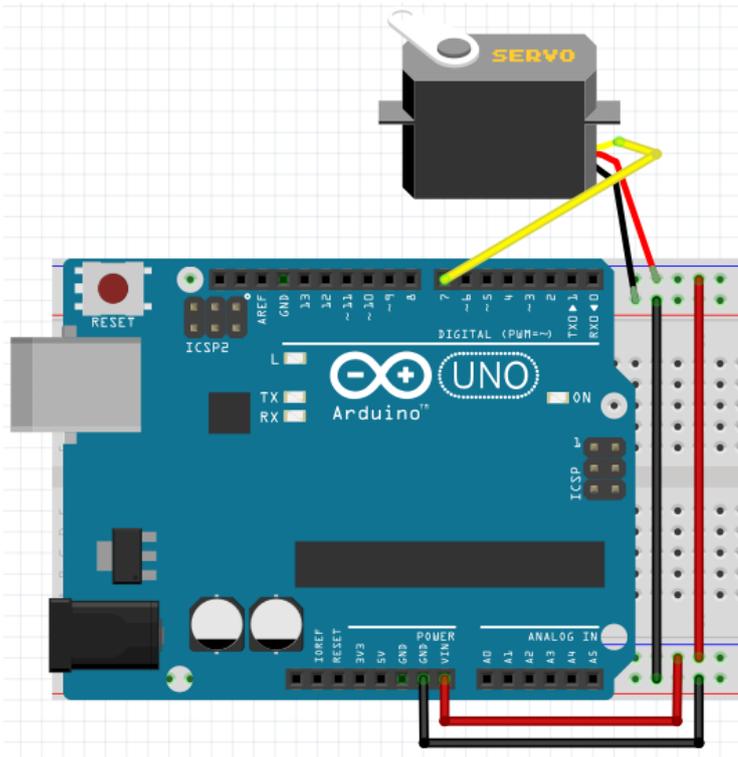
Si quieres saber más, te recomendamos <https://www.luisllamas.es/controlar-un-servo-con-arduino/>

## Montaje 25: Testea tu servo (servomotor)

En el siguiente programa de testeo ([fuente: forum arduino](#)) vamos a probar el servo.

- Conecta el servo al pin 7
- Utiliza el puerto serie para teclear el ángulo que quieras con el teclado de tu ordenador.
- No queremos que entiendas todo el código, pues el puerto serie lee es caracteres ASCII y tiene que convertir el carácter a ángulos.
- Si tecleas un valor más grande de 500 se le indica al servo no el ángulo que se tiene que mover, sino cuanto tiempo en ms se tiene que mover.

## Montaje 25: Conexión sin Edubásica

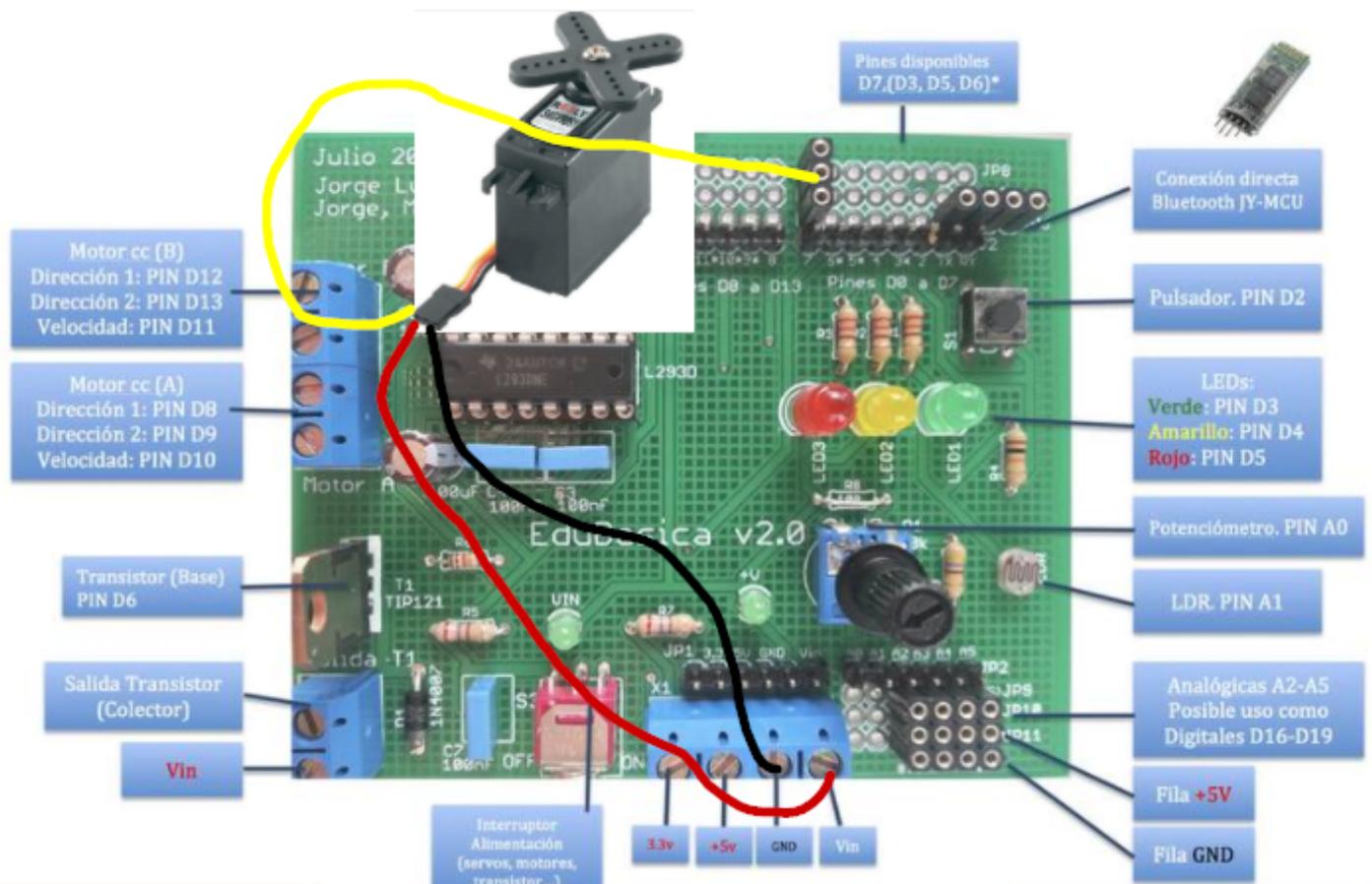


La conexión se realiza mediante 3 cables: 2 de alimentación (+5V/GND) y un tercero, conectado por ejemplo el 7 , donde indicaremos los grados que queremos que gire a través de un programa en Arduino.

## Montaje 25: Conexión con Edubásica

En Edubásica tenemos una forma muy sencilla de conectar un servo a la tarjeta. Lo puedes hacer mediante las clavijas identificadas con JP3. De arriba abajo las conexiones son:

- Señal (pin7)
- +Vin
- GND



Recuerda que siempre puedes utilizar los pines analógicos como E/S digitales, del pin 14 al 19.

Por ejemplo, puedes conectar el servo al pin analógico 5, pero declarado como digital en el 19.

## Montaje 25: video

<https://www.youtube.com/embed/b7JWiL-tucg?rel=0>

Por ejemplo en este Servo **HD-1440A** con el anterior programa se ve que es un servo barato:

- No puede hacer ángulos de +180° luego es un servo convencional
- No puede hacer ángulos de menos de 10° no llega a parar, o sea tiene deriva.

Si eliges uno un poco más caro como el **MG90S** no tiene estos problemas en los extremos. Ver

## Montaje 25: simulación

Aquí lo tienes simulado en Tinkercad [https://www.tinkercad.com/things/4FQNF0doS8Z-](https://www.tinkercad.com/things/4FQNF0doS8Z-25testeatuservo)

[25testeatuservo](https://www.tinkercad.com/things/4FQNF0doS8Z-25testeatuservo)



## Montaje 27 controlando el servo

Vamos a simular un motor paso a paso. El servo ahora tiene que pasar de 10° a 180° de 10 en 10 parando 1 segundo en cada paso

<https://www.youtube.com/embed/lfpjzziVuFw>

El programa lo puedes conseguir en el botón Code de la siguiente simulación

<https://www.tinkercad.com/embed/bJlRTA5Lsha?editbtn=1>

## Montaje 28 servo y potenciómetro

Podemos probar una aplicación muy importante que está basada en mover el servo según una determinada entrada analógica. Este nos puede ser muy útil si queremos controlar servos por medio de joysticks por ejemplo o cualquier dispositivo que cuente con potenciómetros para realizar un movimiento.

El código está obtenido desde los ejemplos que vienen incluido en la IDE de de Arduino (Knob) que encontrarás en: Archivo->Ejemplos->Servo, sólo hemos cambiado esta línea: **`myservo.attach(7)`** ;

Lo que hace este programa es variar la posición del servo en función de la posición del potenciómetro que leemos de manera analógica.

Sólo nos queda mapear la lectura para que se mueva de 0 a 180°.

[https://www.youtube.com/embed/\\_knaXSFBF\\_M](https://www.youtube.com/embed/_knaXSFBF_M)

En la siguiente simulación, puedes mover el potenciómetro y ver el resultado

<https://www.tinkercad.com/embed/gl6syqapJHe?editbtn=1>