

USO DE SERVOMOTORES CON ARDUINO

Por: *Pedro Ruiz Fernández*

Licencia: Creative Commons by-nc-sa 3.0

1. ¿Qué es un servomotor?

Un servomotor es un motor de corriente continua que sobre el cual podemos regular su sentido de giro, velocidad y posición del mismo. Lo hay de dos tipos:

- De rotación continua: mediante pulsos largos o cortos hacemos que el servo gire en un sentido u otro o bien se pare.
- Normal: el servomotor se puede mover entre dos posiciones separadas 180°.

2. Servomotores de rotación continua

Conexionado

Cable	Conexionar a ...
Negro o Marrón	Tierra (GND)
Rojo	5 V
Blanco	Salida digital que tengamos libre

Funcionamiento con librería Servo.h

La forma más sencilla de trabajar con servomotores de rotación continua es a través de la librería "Servo.h", se trata de crear un objeto de tipo servomotor (por ejemplo *Servo miservo*) al cual se le puede pasar el procedimiento *write*, dónde le pasamos un valor a *write* entre 0 y 180.

En función del valor que le des a *write* harás que pare, gire en un sentido o gire en el otro sentido.

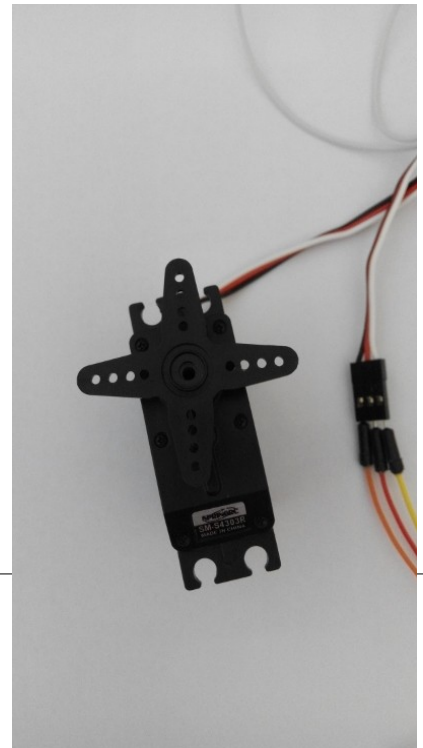
- *miservo.write (90)*: hace que el servo pare.
- *miservo.write (0)*: gira hacia izquierda a toda potencia.
- *miservo.write (entre 0 y 90)*: gira hacia izquierda a una velocidad intermedia, conforme nos acercamos a 90 va bajando la velocidad.
- *miservo.write (180)*: gira hacia derecha a toda velocidad.
- *miservo.write (entre 180 y 90)*: gira hacia derecha a una velocidad intermedia, conforme nos acercamos a 90 va bajando la velocidad.

Ejemplo de uso:

```
#include <Servo.h> // se incluye la librería en cuestión
Servo miservo; // se crea un objeto llamado miservo de tipo Servomotor

void setup () {
  miservo.attach (4); // conectamos el pin de control de servo al pin 4
}

void loop () {
  miservo.write (0); // el servo gira hacia izquierda a toda potencia
  miservo.write (90); // el servo se para
  miservo.write (180); // el servo gira hacia derecha a toda potencia
}
```



Funcionamiento sin librería servo (enviando pulsos)

Para manejar estos servomotores la técnica es enviar un pulso de cierta duración para provocar el giro en ambos sentidos o bien el paro.

Para provocar el giro en sentido horario (derecha) bastaría mandar un pulso alto entre 500 y 1400 microsegundos (en nuestro caso 900 microsegundos) seguido de un pulso bajo de 20000 microsegundos, este pulso bajo es común para todos los movimientos del motor, quedaría de la siguiente manera:

```
digitalWrite (pin,HIGH);
delayMicroseconds (900);
digitalWrite (pin,LOW);
```

```
delayMicroseconds (20000);
```

Para obtener un giro en el sentido antihorario (izquierda) bastaría dar un pulso alto entre 1600 y 2000 microsegundos (en nuestro caso 2100 microsegundos) seguido del pulso bajo de 20000 microsegundos, expresándose de la siguiente forma:

```
digitalWrite (pin,HIGH);  
delayMicroseconds (2100);  
digitalWrite (pin,LOW);  
delayMicroseconds (20000);
```

Para provocar el paro habría que dar un pulso alto de 1500 microsegundos, seguido como siempre del pulso bajo de 20000 microsegundos, y el código se manifestaría de la siguiente forma:

```
digitalWrite (pin,HIGH);  
delayMicroseconds (1500);  
digitalWrite (pin,LOW);  
delayMicroseconds (20000);
```

Funciones para facilitar el uso de servomotores de rotación continua

En este código tenemos definidas tres funciones, una para parar (*para*) otra para rotar en un sentido (*izquierda*) y una tercera para rotar en el sentido contrario (*derecha*).

A cada función se le pasan dos parámetros el pin al que está conectado el motor y el tiempo que quieres que esté funcionando en segundos.

Ejemplos, servomotor que esté conectado en el pin 4 y que gire hacia la izquierda 3 segundos y que pare durante 5 segundos:

- izquierda (4,3);
- para (4,5);

```
int x;  
float distancia;  
void setup ()  
{  
}  
void izquierda (int salida,int tiempo) {  
  unsigned long tiempoinicio =millis();  
  pinMode (salida,OUTPUT);  
  while (tiempoinicio+(tiempo*1000)>millis()) {  
    digitalWrite (salida,HIGH);  
    delayMicroseconds (2100);  
    digitalWrite (salida,LOW);  
    delayMicroseconds (20000);  
  }  
  digitalWrite (salida,HIGH);  
  delayMicroseconds (1500);  
  digitalWrite (salida,LOW);  
  delayMicroseconds (20000);  
}  
  
void para (int salida,int tiempo) {  
  unsigned long tiempoinicio =millis();  
  pinMode (salida,OUTPUT);  
  while (tiempoinicio+(tiempo*1000)>millis()) {  
    digitalWrite (salida,HIGH);  
    delayMicroseconds (1500);  
    digitalWrite (salida,LOW);  
    delayMicroseconds (20000);  
  }  
  digitalWrite (salida,HIGH);  
  delayMicroseconds (1500);  
  digitalWrite (salida,LOW);  
  delayMicroseconds (20000);  
}
```

```

}

void derecha (int salida,int tiempo) {
  unsigned long tiempoinicio =millis();
  pinMode (salida,OUTPUT);
  while (tiempoinicio+(tiempo*1000)>millis()) {
    digitalWrite (salida,HIGH);
    delayMicroseconds (900);
    digitalWrite (salida,LOW);
    delayMicroseconds (20000);
  }
  digitalWrite (salida,HIGH);
  delayMicroseconds (1500);
  digitalWrite (salida,LOW);
  delayMicroseconds (20000);
}

void derecha2 (int salida) {
  pinMode (salida,OUTPUT);
  digitalWrite (salida,HIGH);
  delayMicroseconds (900);
  digitalWrite (salida,LOW);
  delayMicroseconds (20000);
}

void para2 (int salida) {
  pinMode (salida,OUTPUT);
  digitalWrite (salida,HIGH);
  delayMicroseconds (1500);
  digitalWrite (salida,LOW);
  delayMicroseconds (20000);
}

void izquierda2 (int salida) {
  pinMode (salida,OUTPUT);
  digitalWrite (salida,HIGH);
  delayMicroseconds (2100);
  digitalWrite (salida,LOW);
  delayMicroseconds (20000);
}

void tiempo (int segundos) {
  delay (segundos*1000);
}

//avanza (int salida, int limite, distancia) {
//while (distancia > limite) {
//digitalWrite (salida,HIGH);
//delayMicroseconds (900);
//digitalWrite (salida,LOW);
//delayMicroseconds (20000);
//}
//}

void loop() {
  izquierda (4,5);
  para (4,5);
  derecha (4,5);
}

```

3. Servomotores de rotación no continua (“normales”)

Estos servomotores pueden rotar entre 0° y 180°, por tanto podemos fijar su posición entre estas dos.

Conexionado

Cable	Conexionar a ...
Negro o Marrón	Tierra (GND)
Rojo	5 V
Blanco o Amarillo	Pin que enviará los pulsos de configuración de la posición (en Arduino pines 9 o 10)

Funcionamiento con librería *Servo.h*.

Para ello debemos usar la librería *Servo.h* y lo haremos conforme al ejemplo siguiente:

```
#include <Servo.h> //se incluye la librería servo
Servo miservo; //se define el objeto miservo como tipo Servo
void setup(){ // configuración
  miservo.attach(4); // se conecta el servo miservo al pin 4
}
void loop(){ // bucle principal
  miservo.write (0); // pone el servo miservo a 0°
  delay (1000); // espera 1 seg.
  miservo.write (90); // pone el servo miservo a 90°
  delay (1000); // espera 1 seg.
}
```

