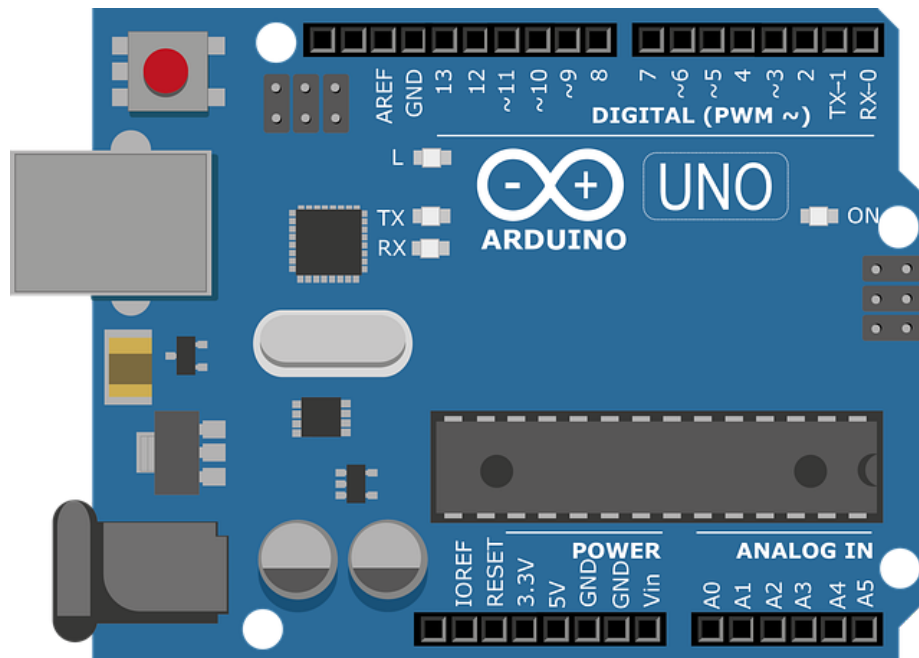


PRÁCTICAS CON ARDUINO



Práctica 10: Control de un servomotor de rotación continua. (Snap4arduino)

Grupo de Trabajo

Beatriz Antunez.

Departamento de Tecnología.

9 abril de 2018.



Finalidad de la práctica

En esta práctica aprenderemos a controlar un servomotor de rotación continua.

El control de un servo de rotación continua es idéntico al de un servo convencional, sólo varía el significado de la señal de control, que en lugar de transformarse en ángulo de posición se interpreta como velocidad angular, en ambos sentidos de giro.

Información

Un servo de rotación continua es una variante de los servos normales, en los que la señal que enviamos al servo controla la velocidad y el sentido de giro en lugar de la posición angular como ocurre en los servos convencionales.

Otra diferencia con los servos convencionales, que tienen un rango limitado de movimiento de 0 a 180°, es que un servo de rotación continua puede girar 360 grados en ambos sentidos de forma continua.



Las características y el control de un servo de rotación continua son similares a los de un servo convencional, de hecho, es posible modificar un servo para convertirlo en rotación continua simplemente eliminando los topes internos y sustituyendo el potenciómetro interno por dos resistencias iguales. No obstante, el rendimiento será inferior a un servo de rotación continua comercial.

Los servos de rotación continua son una forma sencilla de conseguir un motor con control de velocidad sin tener que añadir dispositivos adicionales, como ocurre en

el caso de motores DC o paso a paso, ya que el control está integrado en el propio servo.

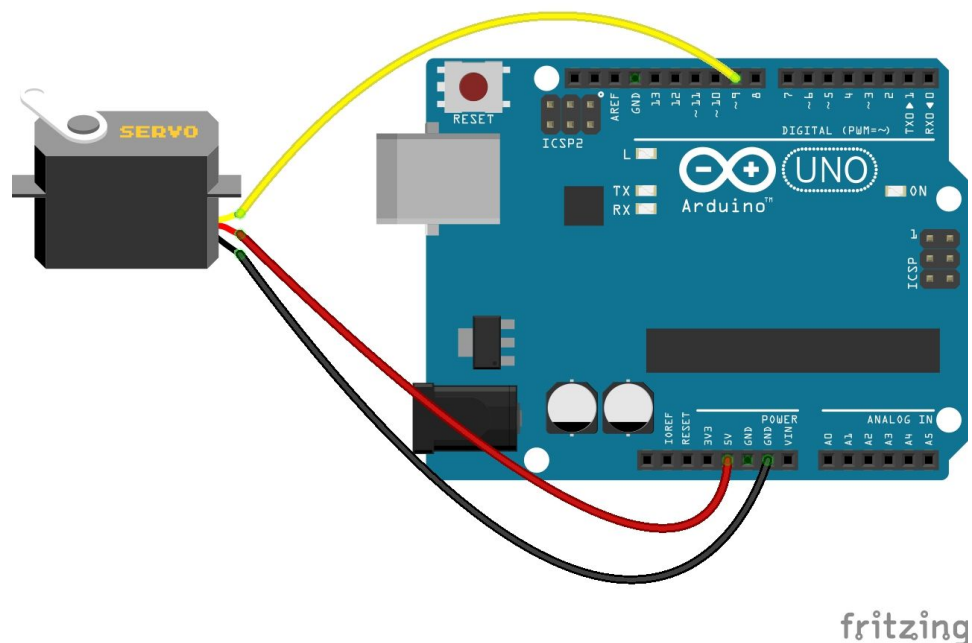
La principal desventaja de los servos de rotación continua es que sacrificamos el control de posición. Si queremos con precisión el ángulo girado tendremos que añadir un encoder.

Los servos de rotación continua son una opción sencilla para construir robots de tamaño pequeño o mediano, especialmente cuando implican el movimiento de varias ruedas. También son útiles en otros sistemas donde necesitemos controlar de forma sencilla la velocidad de giro.

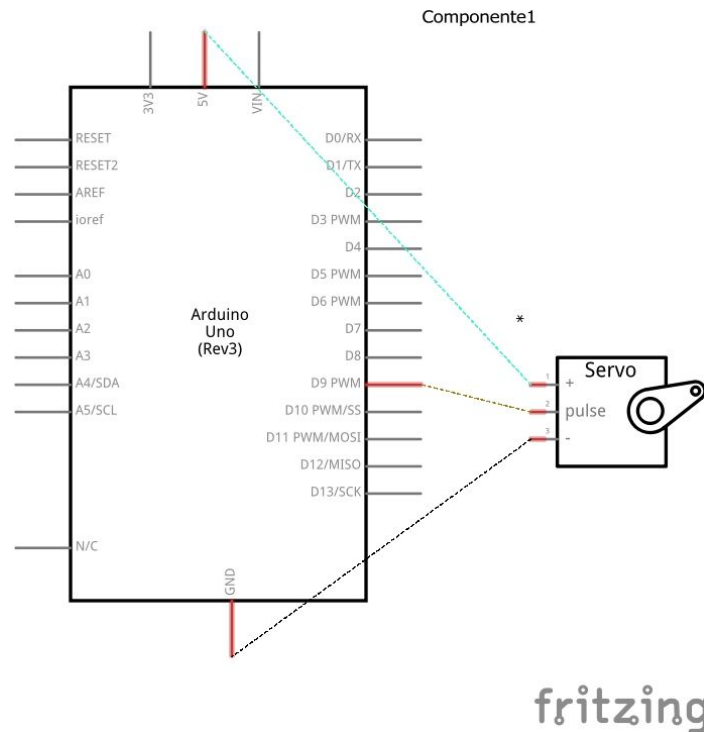
Hardware necesario

- Placa Arduino uno
- Servomotor de rotación continua
- Jumpers de colores

I. Esquema de conexiones



II. Esquema electrónico



Programación

Antes de empezar a escribir el código en la IDE Arduino, tenemos que incluir la librería para los servomotores. Para ello, pulsamos en el menú Programa de la IDE de arduino, a continuación opción "Incluir librería" y por último, pinchamos en "Servo", y ya podemos escribir nuestro programa.

Al igual que en el servo simple, la función “attach()” vincula el objeto servo con el pin de la placa al que lo vamos a conectar, mientras que la función “write()” configura la velocidad del servo en lugar del ángulo de posición. En este caso 0 es máxima velocidad en giro horario, 180 es máxima velocidad en sentido antihorario y 90 motor parado.

Vamos a declarar la variable “vel” para indicarle al servo el sentido de giro, siendo:

- vel = 90 servo parado (valor que hay que calibrar)
- vel = 0 máx. velocidad girando en sentido horario
- vel = 180 máx. velocidad girando en sentido antihorario

Puede pasar que ante la orden vel = 90 el servo no esté parado sino que gire levemente, lo que quiere decir que **necesitamos calibrarlo**.

Para calibrarlo ponemos en arduino el siguiente código:

```
#include <Servo.h>

//Declaramos las variables:
Servo servocontMotor; // creamos el objeto
int vel = 0; // velocidad del servo

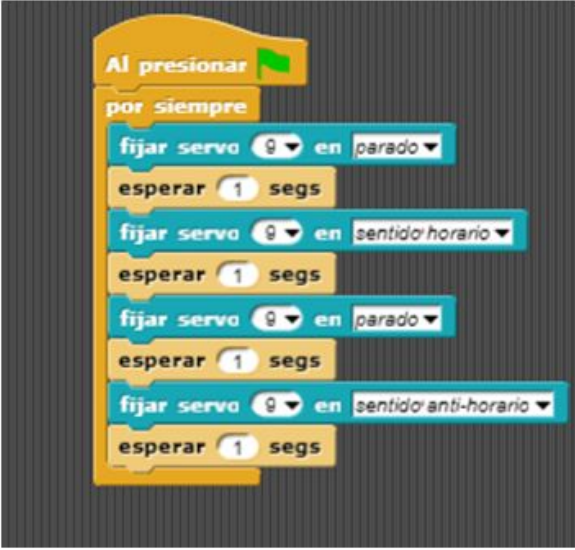
void setup() {
  //Iniciamos el servo para que se conecte al pin 9:
  servocontMotor.attach(9);
}


void loop() {
  vel = 90; //servo parado
  servocontMotor.write(vel);
  delay(1000);
}
```

Vamos a ir variando el valor vel = 90 poco a poco, empezando por 91 y probando si el servo se para; si sigue girando pondremos 92 y probaremos, así hasta el valor en el que se pare, momento en el que tendremos calibrado el aparato.

En nuestro ejemplo, el servomotor se paró en el valor vel=92 y ese es el valor que le voy a poner en el programa.

(Otra forma de calibrarlo, si el servomotor tiene el orificio del potenciómetro, es moviendo con un destornillador dicho potenciómetro y ponerlo a cero con el valor "write(90)").

Snap4Arduino	IDE arduino
	<pre> #include <Servo.h> //Declaramos las variables: Servo servocontMotor; // creamos el objeto int vel = 0; // velocidad del servo void setup() { //Iniciamos el servo para que se conecte al pin 9: servocontMotor.attach(9); } void loop() { vel = 92; //servo parado, este valor que calibrarlo servocontMotor.write(vel); delay(1000); vel = 0; // servo 100% sentido horario servocontMotor.write(vel); delay(1000); vel = 93; //servo parado, este valor calibrarlo servocontMotor.write(vel); delay(1000); vel = 180; // servo 100% sentido antihorario servocontMotor.write(vel); delay(1000); } </pre>



Como se observa en el programa, el valor de “vel” para parar el servo varía según vengamos de girar en sentido horario (vel=93) o antihorario (vel=92). Al ejecutar el programa con los valores estándar, si observamos que las paradas no son en seco sino que el servo se mueve levemente, pues calibramos los valores de “vel” hasta obtener las paradas completas después de los giros.

Actividades y propuestas de mejora

Como actividad de mejora se propone al alumnado que el control del servo se realice con algún sensor.