

## **Práctica 9. Servomotor con Arduino UNO**

En esta práctica de implementación, consiste en utilizar un servomotor, experimentar y descubrir el funcionamiento de un servomotor, visualizar el movimiento e identificar qué tipos de proyecto se pueden realizar con este componente.

Un servomotor o servo. Es un motor eléctrico, pero con dos características especiales. Por un lado, nos permite mantener una posición que indiquemos, siempre que esté dentro del rango de operación del propio dispositivo. Por otro lado nos permite controlar la velocidad de giro, podemos hacer que antes de que se mueva a la siguiente posición espere un tiempo.

Hay varios modelos de servomotor con Arduino. En este caso vamos a utilizar un Micro Servo 9g SG90 de Tower Pro. Es necesario mirar el datasheet del servo. Todos tienen un funcionamiento muy parecido y la programación puede variar muy poco.



El ángulo de giro, en este caso nos permite hacer un barrido entre  $-90^\circ$  y  $90^\circ$ . Lo que viene a ser un ángulo de giro de  $180^\circ$ . Aunque el servo puede moverse con una resolución de más de 1 grado, este es el máximo de resolución que vamos a conseguir debido a la limitación de la señal PWM que es capaz de generar Arduino UNO.

Estos motores funcionan con una señal PWM, con un pulso de trabajo entre 1 ms y 2 ms y con un periodo de 20 ms (50 Hz). Este dato nos indica la velocidad máxima a la que podemos mover el servomotor con Arduino. Solo podremos cambiar de posición cada 20 ms. Esto dependerá del tipo y marca de nuestro servo.

El elegir una salida PWM u otra da lo mismo, todas las salidas de este tipo funcionan igual.

## Objetivo

Experimentar y descubrir el funcionamiento de un servomotor, visual el movimiento e identificar qué tipos de proyecto se pueden realizar con este componente.

## Equipo y Materiales

1 Arduino UNO

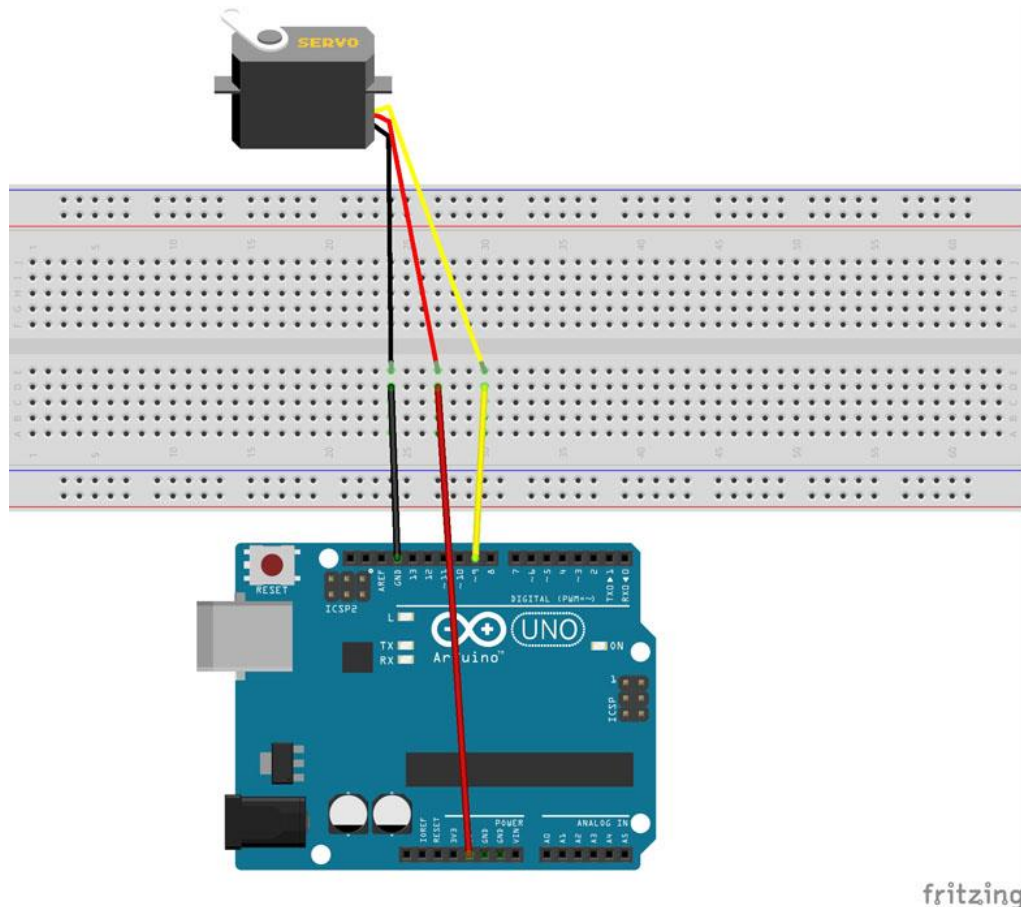
1 Protoboard

1 Servomotor

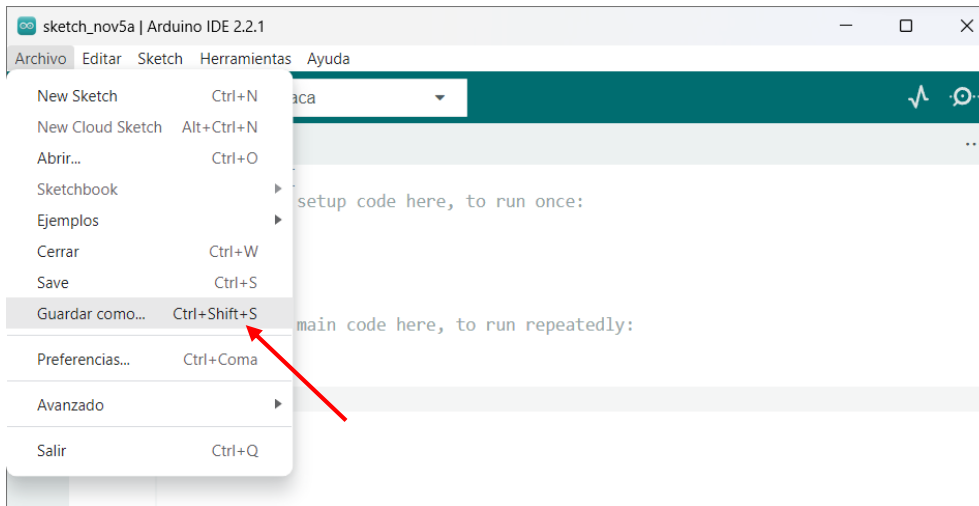
IDE Arduino

## Procedimiento

**PASO 1.** Realice el circuito con sus componentes físicos. A continuación, se muestra cómo deben de realizarse la comunicación física entre el servo y el Arduino.



**PASO 2.** Diríjase al IDE Arduino. Siempre al iniciar deberá de nombrar su proyecto. En la parte superior de click sobre “Archivo” y en “guardar como”, borre el nombre predefinido y sustitúyalo por “Servo.ino”

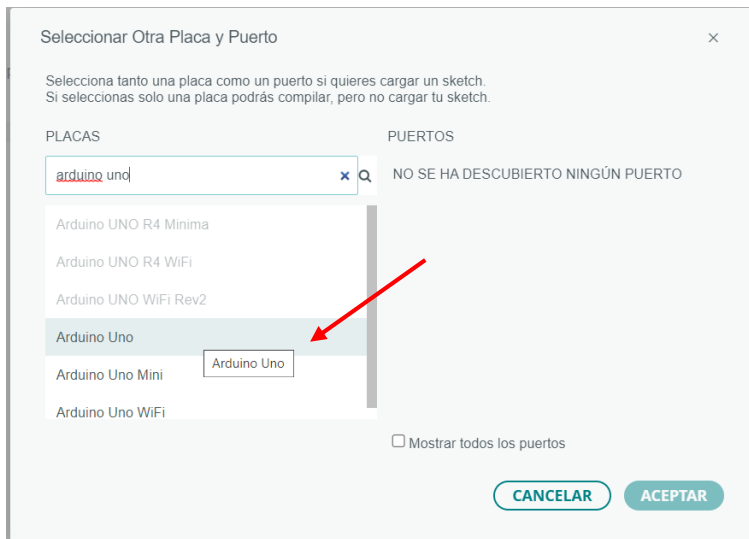
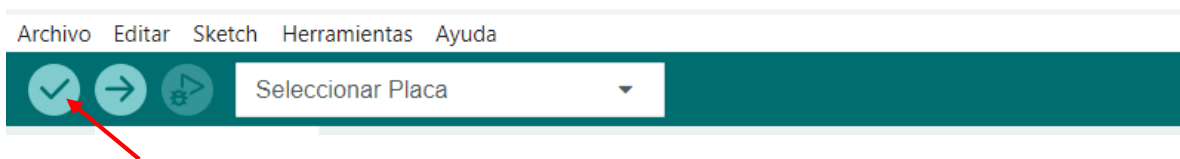
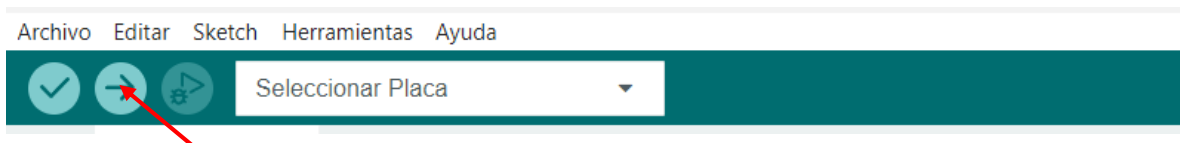


**PASO 3.** Una vez guardado el archivo, borre el contenido que le aparece y sustitúyalo por el siguiente código. Este es un código básico para posicionar el servo en los ángulos 0°, 90° y 180°.

```

6 // Incluimos la librería para poder controlar el servo
7 #include <Servo.h>
8
9 // Declaramos la variable para controlar el servo
10 Servo servoMotor;
11
12 void setup() {
13   // Iniciamos el monitor serie para mostrar el resultado
14   Serial.begin(9600);
15
16   // Iniciamos el servo para que empiece a trabajar con el pin 9
17   servoMotor.attach(9);
18 }
19
20 void loop() {
21
22   // Desplazamos a la posición 0°
23   servoMotor.write(0);
24   // Esperamos 1 segundo
25   delay(1000);
26
27   // Desplazamos a la posición 90°
28   servoMotor.write(90);
29   // Esperamos 1 segundo
30   delay(1000);
31
32   // Desplazamos a la posición 180°
33   servoMotor.write(180);
34   // Esperamos 1 segundo
35   delay(1000);
36 }

```

**PASO 4.** Seleccione la placa que está utilizando**PASO 5.** Verifique que su programa no tenga errores.**PASO 6.** Conecte el cable del Arduino a este y al puerto USB de su computadora.**PASO 7.** Cargue el programa

## CUESTIONARIO

1. ¿Qué librería se utiliza?
2. ¿Cómo se busca y se incluye una librería? Describa los pasos.
3. Para realizar la misma acción de posicionar el servo a los ángulos 0º, 90º y 180º, ¿Cómo lo realizaría sin la librería? Investíguelo.
4. En el código, ¿Cómo se puede posicionar el servomotor a 45º utilizando la librería?
5. Ahora queremos que el servo haga un barrido desde el 0 al 180º y luego en sentido contrario. Cambie el código por el siguiente.

```
6 // Incluimos la librería para poder controlar el servo
7 #include <Servo.h>
8
9 // Declaramos la variable para controlar el servo
10 Servo servoMotor;
11
12 void setup() {
13     // Iniciamos el monitor serie para mostrar el resultado
14     Serial.begin(9600);
15
16     // Iniciamos el servo para que empiece a trabajar con el pin 9
17     servoMotor.attach(9);
18
19     // Inicializamos al ángulo 0 el servomotor
20     servoMotor.write(0);
21 }
22
23 void loop() {
24
25     // Vamos a tener dos bucles uno para mover en sentido positivo y otro en sen
26     // Para el sentido positivo
27     for (int i = 0; i <= 180; i++)
28     {
29         // Desplazamos al ángulo correspondiente
30         servoMotor.write(i);
31         // Hacemos una pausa de 25ms
32         delay(25);
33     }
34
35     // Para el sentido negativo
36     for (int i = 179; i > 0; i--)
37     {
38         // Desplazamos al ángulo correspondiente
39         servoMotor.write(i);
40         // Hacemos una pausa de 25ms
41         delay(25);
42     }
43 }
```

6. En el código, ¿Cómo se realiza el regreso del servomotor?
7. ¿Qué es un bucle for?
8. Explique como el código hace para que el servomotor de hacia delante y luego hacia atrás sin tener que mover nada en el hardware, solo con la programación.
9. ¿Qué proyecto harías con este servomotor? (Al menos 3 ideas)
10. Busca en internet otro proyecto simple con este módulo y realízalo.