

INFORMACIÓN BÁSICA DE UN SERVOMOTOR PARA ARDUINO

El servomotor tiene tres cables: Rojo, Negro/Marrón y Amarillo/Naranja.

- El **negro** o marrón es GND (ground, “masa” o “negativo”, según el barrio).
- El **rojo** es la alimentación +5V. ...
- El **amarillo/naranja** es el control/señal, y por acá va a llegar la información que modifique la posición del servo.

Características técnicas de un servomotor con Arduino

Hay varios modelos de servomotor con Arduino. En este caso vamos a utilizar un Micro Servo 9g SG90 de Tower Pro. Como siempre digo, hay que mirar la [ficha técnica](#) del producto. Todos tienen un funcionamiento muy parecido y la programación puede variar muy poco.

Cosas a tener en cuenta con este dispositivo. Lo primero, el ángulo de giro, en este caso nos permite hacer un **barrido entre -90° y 90°**. Lo que viene a ser un **ángulo de giro de 180°**.

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). ¿Qué quiere decir todo esto? 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.

Conexionado con Arduino

Las conexiones dependerán del tipo de servomotor con Arduino que estemos utilizando.


Todos deben tener 3 cables. Uno irá a tierra, otro a la alimentación de 5 Voltios y el tercero a un pin PWM.

¿Qué es PWM en Arduino?

PWM (Modulación por Ancho de Pulso, o *Pulse Width Modulation*) es una técnica para simular una señal analógica usando una salida digital. En Arduino, puedes usar PWM para **controlar la velocidad de un motor, el brillo de un LED, la posición de un servo**, etc.

Una señal PWM alterna entre encendido (*HIGH*) y apagado (*LOW*), pero puedes variar el “ciclo de trabajo” (*duty cycle*), que es el porcentaje de tiempo que está encendida.

Pines PWM en Arduino

En placas como el **Arduino UNO**, los pines que soportan PWM están marcados con un .