**Raspberry Pi Servo Motor control**

In addition to[stepper motors](https://tutorials-raspberrypi.com/how-to-control-a-stepper-motor-with-raspberry-pi-and-l293d-uln2003a/), there are also small and cheap servo motors. The control of Raspberry Pi servo motors is very simple and thanks to the small size and weight they are used in many areas.  
Unlike stepper motors, servomotors can be controlled with a single GPIO.

In this tutorial, I’ll show an example of how to use Python to control one or more servo motors.

**Required Hardware Parts**

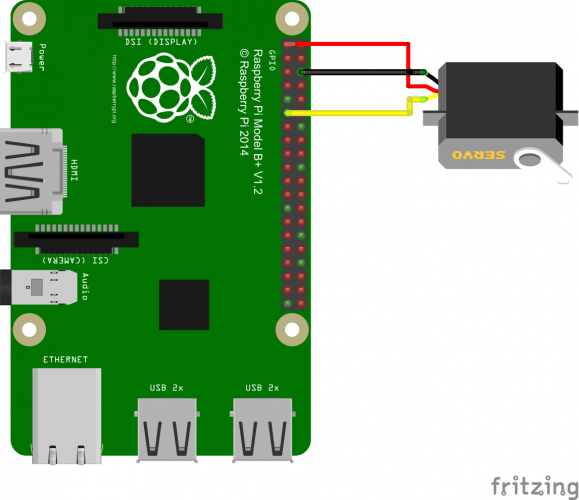
* [Servo motor](https://www.amazon.com/s/ref=nb_sb_noss_2?url=search-alias%3Daps&field-keywords=servo+motor&tag=754u-20)
* [Jumper cable](https://www.amazon.com/s/ref=nb_sb_noss_1?url=search-alias%3Daps&field-keywords=pin+jumper+cable&rh=i%3Aaps%2Ck%3Apin+jumper+cable&tag=754u-20)
* ([Breadboard](https://www.amazon.com/s/ref=nb_sb_ss_c_1_10?url=search-alias%3Daps&field-keywords=breadboard&sprefix=Breadboard%2Caps%2C705&crid=1VPK5G0A1NR3O&rh=i%3Aaps%2Ck%3Abreadboard&tag=754u-20))
* if necessary, a [battery holder](https://www.amazon.com/s/ref=nb_sb_ss_c_1_14?url=search-alias%3Daps&field-keywords=battery+holder&sprefix=battery+holder%2Caps%2C906&crid=275VAOLK68DW5&rh=i%3Aaps%2Ck%3Abattery+holder&tag=754u-20) + (rechargeable) [batteries](https://www.amazon.com/s/ref=nb_sb_ss_i_1_13/146-9480658-8062642?url=search-alias%3Daps&field-keywords=rechargeable+batteries&sprefix=rechargeable+%2Caps%2C234&crid=IO7GLUSJ5EFP&rh=i%3Aaps%2Ck%3Arechargeable+batteries&tag=754u-20), as an external power supply

Of course, it is possible to supply the servo motor with an external power source, but this only makes sense when using several motors. For that, [rechargeable batteries](https://www.amazon.com/s/ref=nb_sb_noss_1?url=search-alias%3Daps&field-keywords=rechargeable+batteries&rh=i%3Aaps%2Ck%3Arechargeable+batteries&tag=754u-20)/regular batteries would still be needed.

## Setup

In most cases, the colors of the servo are as follows and are connected to the Pi:

Black – comes to GND (pin 6) from the Pi  
Red – comes to 3V3 (pin 1) from the Pi  
Yellow/Orange – to a free GPIO pin (e.g., GPIO17, pin 11)



If you want to play it safe, you can set a ~ 1kΩ resistor between the data pin (yellow/orange) and the Pi. Normally this is not necessary.

If the servo motor does not rotate correctly, this may also influence the power supply of the Raspberry Pi (just look at the datasheet, what the engine consumes). In such a case an external power source makes sense (usually it is 4 to 6V).

## Software for controlling the Raspberry Pi servo motors

Unlike stepper motors, servo motors do not occupy quite a few GPIO pins to command a movement. For this, the rotation is controlled by the length of the pulse.

The angle of the motor is set along the length of the pulse, so [PWM](https://sourceforge.net/p/raspberry-gpio-python/wiki/PWM/) is particularly useful, which sends repetitive signals at even intervals (the Raspberry Pi [Python library](https://pypi.python.org/pypi/RPi.GPIO) must be installed).

We either start python (sudo python) or open a new script (sudo nano servomotor.py) with the following content:

import RPi.GPIO as GPIO

import time

servoPIN = 17

GPIO.setmode(GPIO.BCM)

GPIO.setup(servoPIN, GPIO.OUT)

p = GPIO.PWM(servoPIN, 50) # GPIO 17 for PWM with 50Hz

p.start(2.5) # Initialization

try:

while True:

p.ChangeDutyCycle(5)

time.sleep(0.5)

p.ChangeDutyCycle(7.5)

time.sleep(0.5)

p.ChangeDutyCycle(10)

time.sleep(0.5)

p.ChangeDutyCycle(12.5)

time.sleep(0.5)

p.ChangeDutyCycle(10)

time.sleep(0.5)

p.ChangeDutyCycle(7.5)

time.sleep(0.5)

p.ChangeDutyCycle(5)

time.sleep(0.5)

p.ChangeDutyCycle(2.5)

time.sleep(0.5)

except KeyboardInterrupt:

p.stop()

GPIO.cleanup()

If the servo motor shakes a bit while it is not moving, you can pause the pulse with ChangeDutyCycle(0)

By

ilangkaviyan