Motor User Manual

Last updated: 18 March 2017

Prepared by Jony Margelist [jony.margelist@stud.hslu.ch]

Motor:	(GWS) NARO Servo
URL:	https://www.parallax.com/product/900-00014
Cost per unit:	$8.99 USD

General Information

About Servo

big punch and are very energy-efficient. These features allow them to be used to operate remote-controlled or radio-controlled toy cars, robots and airplanes. Servo motors are also used in industrial applications, robotics, in-line manufacturing, pharmaceutics and food services. But how do the little guys work? The servo circuitry is built right inside the motor unit and has a positionable shaft, which usually is fitted with a gear (as shown below). The motor is controlled with an electric signal which determines the amount of movement of the shaft.

What's inside the servo?

To fully understand how the servo works, you need to take a look under the hood. Inside there is a pretty simple set-up: a small DC motor, potentiometer, and a control circuit. The motor is attached by gears to the control wheel. As the motor rotates, the potentiometer's resistance changes, so the control circuit can precisely regulate how much movement there is and in which direction. When the shaft of the motor is at the desired position, power supplied to the motor is stopped. If not, the motor is turned in the appropriate direction. The desired position is sent via electrical pulses through the signal wire. The motor's speed is proportional to the difference between

its actual position and desired position. So if the motor is near the desired position, it will turn slowly, otherwise it will turn fast. This is called proportional control. This means the motor will only run as hard as necessary to accomplish the task at hand, a very efficient little guy.

How is the servo controlled?

Servos are controlled by sending an electrical pulse of variable width, or pulse width modulation (PWM), through the control wire. There is a minimum pulse, a maximum pulse, and a repetition rate. A servo motor can usually only turn 90° in either direction for a total of 180° movement. The motor's neutral position is defined as the position where the servo has the same amount of potential rotation in the both the clockwise or counter-clockwise direction. The PWM sent to the motor determines position of the shaft, and based on the duration of the pulse sent via the control wire; the rotor will turn to the desired position. The servo motor expects to see a pulse every 20 milliseconds (ms) and the length of the pulse will determine how far the motor turns. For example, a 1.5ms pulse will make the motor turn to the 90° position. Shorter than 1.5ms moves it in the counter clockwise direction toward the 0° position, and any longer than 1.5ms will turn the servo in a clockwise direction toward the 180° position. When these servos are commanded to move, they will move to the position and hold that position. If an external force pushes against the servo while the servo is holding a position, the servo will resist from moving out of that position. The maximum amount of force the servo can exert is called the torque rating of the servo. Servos will not hold their position forever though; the position pulse must be repeated to instruct the servo to stay in position.

Types of Servo Motors

There are two types of servo motors - AC and DC. AC servo can handle higher current surges and tend to be used in industrial machinery. DC servos are not designed for high current surges and are usually better suited for smaller applications. Generally speaking, DC motors are less expensive than their AC counterparts. These are also servo motors that have been built specifically for continuous rotation, making it an easy way to get your robot moving. They feature two ball bearings on the output shaft for reduced friction and easy access to the rest-point adjustment potentiometer.

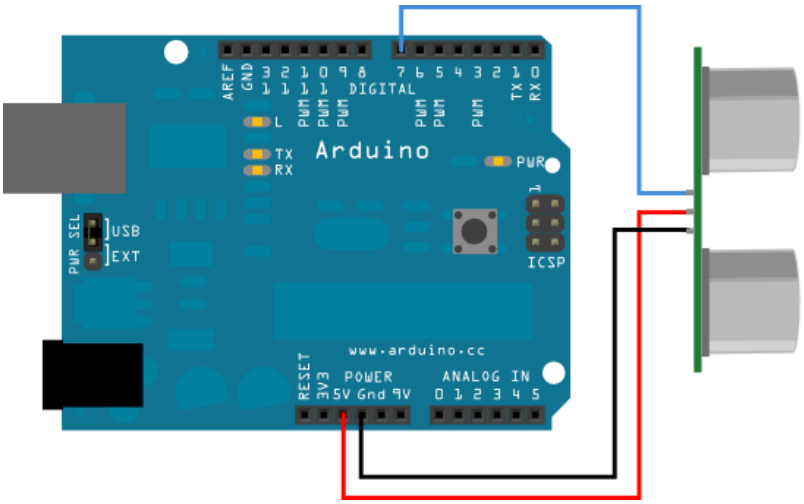
Servo Motor Applications

Servos are used in radio-controlled airplanes to position control surfaces like elevators, rudders, walking a robot, or operating grippers. Servo motors are small, have built-in control circuitry and have good power for their size.
In food services and pharmaceuticals, the tools are designed to be used in harsher environments, where the potential for corrosion is high due to being washed at high pressures and temperatures repeatedly to maintain strict hygiene standards. Servos

are also used in in-line manufacturing, where high repetition yet precise work is necessary.
Of course, you don't have to know how a servo works to use one, but as with most electronics, the more you understand, the more doors open for expanded projects and projects' capabilities. Whether you're a hobbyist building robots, an engineer designing industrial systems, or just constantly curious, where will servo motors take you?

Getting it to work!

Connection



5V	connected to	5V pin on Arduino
GND	connected to	GND on Arduino
SIG	connected to	Any digital input pin (example uses 11)

Example Code

```
#include <Servo.h>

Servo myServo1;
int pos;
int pinServo1=11;

void setup() {
  pos = 0;
  myServo1.attach(pinServo1);
}

void loop() {
  for(pos = 0; pos  <= 180; pos +=1) {
    myServo1.write(pos);
    delay(30);
  }
  delay(5000);
  for(pos =180; pos >=0; pos -=1) {
    myServo1.write(pos);
    delay(30);
  }
  delay(5000);
}

```

Code: in plain English

#include <Servo.h>
Include the servo library

Servo myServo1;
int pos;
int pinServo1=11;
Defines 3 variables for the servo motor, not needed, but makes the code much more readable

void setup() {
Setup method which is executed only once
pos = 0;
Set the pos variable to 0 for 0°.
myServo1.attach(pinServo1);
attach the servo motor at the pin 11(pinServo1 stored the number 11)

```
//loop()-method which goes on forever
void loop() {
```
loop()-method which runs infinite

```
for(pos = 0; pos  <= 180; pos +=1) {
myServo1.write(pos);
```
turn the servo from 0° to 180° degree

```
delay(30);
```
without the delay it would be instantaneous

```
delay(5000);
```
delay between one rotation to the the other rotation

```
for(pos =180; pos >=0; pos -=1) {
myServo1.write(pos);
```
rotate back to position 0

```
delay(30);
```
without the delay it would be instantaneous

```
delay(5000);
```
delay before the first rotation starts