# Application Note # 106

## All About Servos

**Release Date: November, 2003**

**Version 1.0**

## Copyrights & Trademarks



## Disclaimer of Liability

HVW Technologies Inc. does not warrant the information; circuits, code or products described herin to be suitable for any application. This document is provided for informative and educational purposes only. HVW Technologies Inc. is not responsible for special, incidental, or consequential damages from any breach of warranty, or under any legal theory, including lost profits, downtime, goodwill, damage to or replacement of equipment or property, and any costs of recovering, reprogramming, or reproducing and data stored in or used with HVW Technologies products or any other manufacturers product or products used or referred to in this document.

HVW Technologies Inc. does not manufacture or sell any products that are certified for life support or mission-critical applications where the malfunction of product may either directly or indirectly risk the health or safety of any person or creature, or lay damage to any property. The use of any product or information is at the sole discretion and risk of the user.

## Errata

While every effort has been made to ensure the accuracy of the information contained within this document. Changes to firmware, product specifications or availability and other variables may effect or negate results obtained are beyond our control.

## Contact

Technical Support: Support@HVWTech.com
General Information: Info@HVWTech.com
Sales: Sales@HVWTech.com

# Application Note#106

## All about Servos

An explanation of how servos work and are controlled.

**Intended Audience**

This appnote is intended primarily for those interested in using servos for robotics applications. However, most of the material presented is applicable to servos in general.

**Objective**

To demystify servos and their abilities as well as clarify their limitations.

**Parts List**

No parts are required.

**Tools**

- Small Phillips screwdriver

# All About Servos

Servos have been used in the Remote Control (R/C) hobby industry for many years. They are robust, reliable and economical. Many people reading this will be using servos for purposes other than for which they were originally designed; something you should keep in mind if you're planning on using servos as drive motors for a 20 pound robot.

What's Inside The case

A servo consists of a small motor, a gearset, a feedback potentiometer (variable resistor), and some control electronics. The motor spins at variable speeds (much faster than the output shaft) and is coupled to a (reduction) gearset that converts the motor's high speed into something that is more usable for our purpose. When you reduce the motor's speed through a reduction gearset, you gain torque. Torque is the 'twisting' power of the servo -the more torque, the heavier the object the servo can move.

**How Servos Work**

A servo is a classic example of a closed-loop feedback system. The potentiometer is coupled to the output gear. Its resistance is proportional to the position of the servo's output shaft (0° to 180 °). This resistance is used by the control electronics to generate an error signal when the desired position isn't the same as the current position. If you send a servo a command to place itself at 90° and the output shaft is actually at 45°, an error signal will cause the motor to move the head (via the gears) until the error signal is 0 (when the head has reached 90°). If the head had been at 180°, an error signal of opposite polarity (and double the amplitude) would have been generated and the motor would have turned in the oposite direction to bring the head 'back' to 90°. As you can see, the current position is 'fed-back' to the control system in a loop to maintain a zero error signal. The farther the actual position from the desired position, the faster the motor turns to bring the error back to zero.

**How are They Controlled**

Servos have 3 wires coming out of them: Power, Ground and Signal. Power is normally 7.2 VDC but most servos run quite well on voltages as low as 5 volts. The Signal lead is used to send the positioning signal to the servo. Servos are controlled using a system called Pulse Code Modulation (PCM). In order to understand this, you need to understand the terms “milliseconds"(ms) and “microseconds" (µs). 1 ms is 1/1000th of a second; or put another way, there are 1000 ms in every second. 1 µs is 1/1,000,000th (one -one millionth) of a second, or, there are 1 million µs in each second. Servo manufacturers usually specify pulse-widths in µs, so it's handy to be able to convert between µs and ms.

The servo's electronics work in 20 ms blocks (50 of them every second). For each 20 ms block, the servo needs to see a positive-going pulse whose period (width in ms) tells it where to position the head (output shaft).

The period (width) of the pulse determines where to place the head (0° to 180°). Different servo manufacturers require somewhat different signals to do the same thing, so you have to experiment a little to find the pulse-widths that correspond to each position. For example, the Futaba servo sold by HVW Technologies has a 90° position (middle) pulse-width of about 1.5 ms. This means that if you send a 1.5 ms pulse to the servo at least once every 20 ms, the servo will move to, and hold at, it's 90° position. If you try to turn the head with your hand you will feel the servo forcing against you, trying to keep the 90° position. If you have a similar servo, made by another company, you may have to send a 1.45 ms pulse, for example, to get the same 90° position. It should be noted that even servos of the same manufacturer and model can have slight differences.

Sending a Control Signal

The pulses needed to control a servo are easily generated with a BASIC Stamp 2 (or equivalent), with a single line of code:

```
PULSOUT pin, 750
```

Where ‘pin’ is the BASIC Stamp pin connected to the servo's control line. The PULSOUT command works in multiples of 2 µs, so we need 750 x 2µs = 1500 µS which is 1.5 ms

In practice, you can send pulses more often that once every 20 ms; you can send them less often as well, but if you don't send any pulses for about 50 ms or so, the control electronics in the servo will “go to sleep" (enter a power-saving mode). Don't forget, servos were designed for R/C airplanes and cars, where battery life is important. When a servo powers-down, it no longer works against an applied force to maintain it's position. You will find that even then the head is fairly difficult to turn.

**Continuous Rotation**

For many applications, modified servos make excellent drive wheels for mobile robots -but how do you get them to keep going around instead of stopping at 180° ? There are a couple of problems:

The servos output gear is connected to the feedback potentiometer (so it can track the output shafts position and “feed-back" this information to the control electronics). The potentiometer has a finite resistance -it can't keep increasing with successive rotations of the head gear so there is a mechanical stop to prevent it from turning too far.

The solution is to make the servo think that it's output shaft is always at the 90° (middle) point and cut-off the mechanical stop (plastic tab). When you give it a command to go to 0°, the control circuitry will see that the output shaft is at 90° and will run the motor to try and bring it to 0° (and zero the error signal). What happens though is that since the resistance isn't changing as the output shaft rotates, the control system will keep the motor running forever. Conversely, telling the servo to go to the 180° position will cause the motor to spin in the opposite direction. Since the tab no longer mechanically stops the head, you now have a very nifty little gearmotor.

**The Modification Process**

There are 2 ways to make the servo think it's head is always at 90°:

Method 1: Remove the potentiometer and replace it with 2 resistors who's values match the "center" resistance of the potentiometer.

Method 2: Leave the potentiometer in, but stop it from turning with the output shaft.

Which method should you use ? -That depends on your application, but generally, we prefer the second method as it is much faster and simpler. Some things to consider:

| Method | Pros | Cons |
|---|---|---|
| Fixed Resistors | - Immune to vibrations<br>- You have a nice little pot left over | - Can't be easily un-done<br>- Requires you find suitable resistors to replace pot.<br>- Requires soldering |
| Disconnect Pot from Output Shaft | - Fast<br>- Easily reversed (for most servos)<br>- No soldering | - Prone to vibrations |

**Modification Instructions**

See AppNote AN-107 for step-by-step instructions on modifying a servo for continuous rotation.

**Limitations**

Servos do have limitations. Most relate to the differences between what the servo was designed for and the application you are actually using it for. Depending on your application, some of those listed may or may not be an issue.

- They're not incredibly powerful although they are certainly strong enough for most people's needs.

- Unless you spend considerably more for servos with ball bearings, the servo's lifespan may be less than what you might hope for. Back to our earlier comment about what servos are designed for, they are NOT designed to support several pounds of side force while continuously rotating. Realize that a few minutes of continuous rotation on a robot weighing 2 pounds is the equivalent in wear to many, many, many hours of use in it's intended area of use (R/C Planes, cars etc.). That said, you can still expect most servos -even those without ball bearings, to last several years of intermittent experimental or competition use. If you have an application that will see extended, continuous, or heavy-duty use buy ball-bearing servos -they cost more but will last much longer and provide smoother performance.

-Servos require constant attention from the control system. A pulse must be sent every 20 ms, so the control electronics must continuously allocate resources to do this. Depending on the processor being used, this may consume a considerable portion of the processing power available. HVW Technologies has a number of servo controller available that will

**Resources**

Futaba FP-S148 Servo –A very good quality general purpose servo that is easy to modify.

Economoy Servo (Modified) –A reasonable quality servo that has already been modified for continuous rotation.

Phidget Servo Controller – Control a single servo from your PC's USB port.

Phidget Quad Servo – Control up to 4 servos from your PC's USB port