

Interfacing a Servo to the Basic Stamp

Purpose

- To introduce the servo as a mechatronic actuator
- To discover how to control the movement of a servo using the Basic Stamp
- To practice interfacing power electronics and switches

Components

<u>Qty.</u>	<u>Item</u>
1	OEM Basic Stamp II microcontroller and serial port cable
2	680 Ω resistors
2	tact switches
1	green LED
1	red LED
1	3-pin double-ended male header
1	Futaba FP-S148 servo

Introduction

In this lab you will explore the workings of a radio control (R/C) servo and how the Stamp can be used to control one.

An R/C servo consists of a dc motor, gear train, potentiometer, and some control circuitry all mounted compactly in a case. R/C servos are commonly used in radio-controlled cars, airplanes, and boats to provide limited rotational motion to steer, move control surfaces, etc. R/C servos are attractive for educational use in mechatronics, because they are relatively inexpensive (about \$12-\$20), they can put out about 42 oz/in of torque, they can easily be modified to produce continuous shaft rotation at relatively slow speeds, and they can easily be controlled by a microcontroller. There are three wires, white, red, and black on the servo leading from a 3-pin female connector to the case. These carry the control signal, power, and ground return respectively.

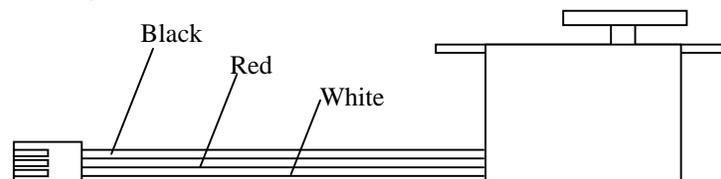
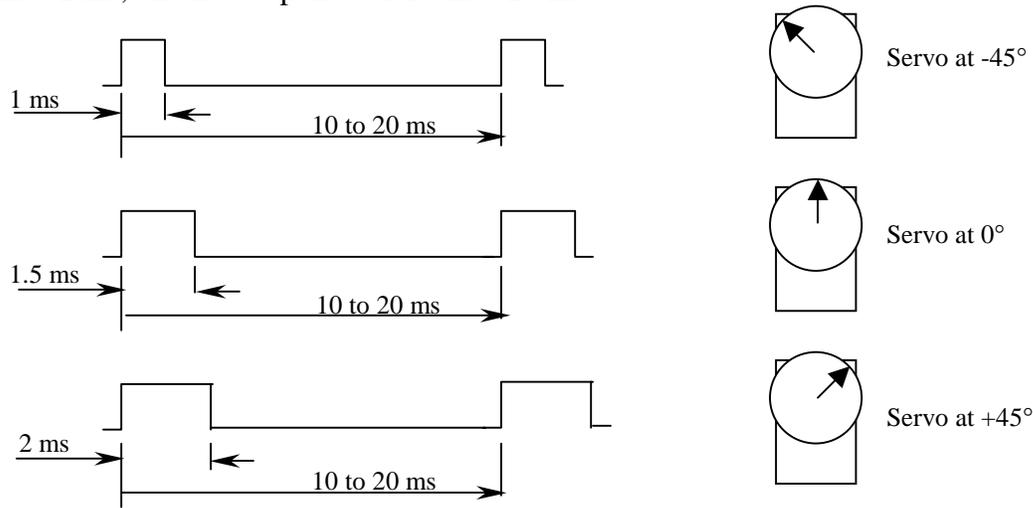


Figure 1 R/C Servo. The R/C servo uses three wires: white carries the control signal, red carries power (usually 4.8 V to 6 V), black is ground.

Figure 2 shows how an R/C servo is made to rotate. The control circuitry inside the servo must receive a stream of pulses whose widths may vary between about 1 ms and 2 ms, and these pulses must occur at intervals of about 10 to 20 ms. A potentiometer coupled to the rotation of the output shaft produces a voltage corresponding to the angle of the shaft. The control circuitry compares the “average” voltage of the control signal with the voltage from the potentiometer, and the shaft rotates until the two voltages are the same.

Figure 2 R/C Servo Operation. The R/C servo needs to see a pulse-width modulated control signal in order to position the output shaft. Pulse-widths vary between approximately 1ms - 2 ms, and have a period of 10 ms - 20 ms



Procedure

Servo Control Circuit

Figure 3 shows the circuit you will use in this lab.

1. Build the section of the circuit shown in the dashed rectangle first. **Do not** connect the servo to the Stamp yet! Don't forget to supply power and ground to the Basic Stamp. Be careful when you insert the tact switches into the solderless breadboard that you get the leads oriented properly. Make sure that each pair of legs that are tied together internally plug into the *same* row of 5 holes on the breadboard. Look for the dot on the underside of the switch. The legs on the side closest to the dot are tied together internally. Test this sub-circuit by entering and running the following program:

```
'Switch Control Test Circuit
'Put your name here
'Put the date here
DIR6=1      'sets the direction of pins 6 and 7 to outputs
DIR7=1
loop:
IF IN8=1 AND IN9=1 THEN off      'turn off LED's if no switch is pressed
IF NOT IN8=1 AND IN9=1 THEN red  'turn on red LED if switch 1 is pressed
OUT7=1      'else green
OUT6=0
goto loop
off:
OUT6=1
OUT7=1
goto loop
red:
OUT6=1
OUT7=0
goto loop
```

Your program should allow you turn on the red or green LED's when the corresponding switch is pressed. Save this program to your floppy disk.

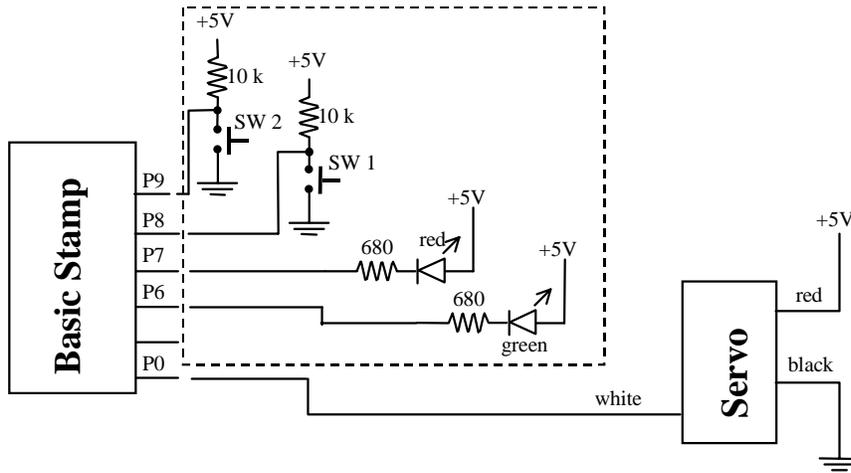


Figure 3 Servo Control Circuit.

2. After you have proven that the switches and LED's have been wired correctly, add the servo. Connect the servo to the breadboard using the 3-pin double-ended male header. You will have to push the short pins into the servo connector and the long pins into the breadboard. The 5 V to power the servo must come from an external supply and not from the Stamp, because the servo will draw upwards of 100 mA, far too much for the Stamp to supply. Make sure that the common grounds for the servo and Stamp are tied together.
3. Enter and run the following program:

```

'Servo test program
'Put your name here
'Put the date here
x var word
output 0      'equivalent to OUT0=1
here:
for x=1 to 100
  pulsout 0, 500
  pause 10
next
pause 500
for x=1 to 100
  pulsout 0, 1000
  pause 10
next
pause 500
goto here
    
```

What does the servo do? The command, PULSOUT n, t sends a pulse of duration 2t microseconds out pin n. PAUSE t pauses the program for t milliseconds.

Add some code to the previous program to turn on one of the LED's when the servo rotates in one direction and to turn on the other when the servo rotates in the opposite direction.

Experiment with the pulse durations by varying them between 500 and 1000. **What value of pulse duration will put the servo at 0°?**

4. Write a program that will cause the servo to rotate to one extreme when one of the switches is pressed and will rotate to the other extreme when the other button is pressed. If both switches are pressed, then the servo should rotate to its central position. Save your program to your floppy, and include it in your lab report.

End Notes

It is relatively easy to modify a servo so that it can rotate continuously. An excellent step-by-step guide for doing so can be found at:

<http://www.seattlerobotics.org/guide/servohack.html>

If you use the Futaba FP-S148, it is especially easy. You don't even need to replace the potentiometer as noted in the link above. You can manually position the pot shaft to its center position.

Once you've hacked the servo, by varying the pulse-width as you did in this lab, you then have a low-cost, relatively powerful variable speed dc motor drive.

Using a servo can be a low-cost and efficient approach for giving mobility to your robot. Tower Hobbies (<http://www.towerhobbies.com/>) has the best prices around for Futaba servos (3 for \$39.47).