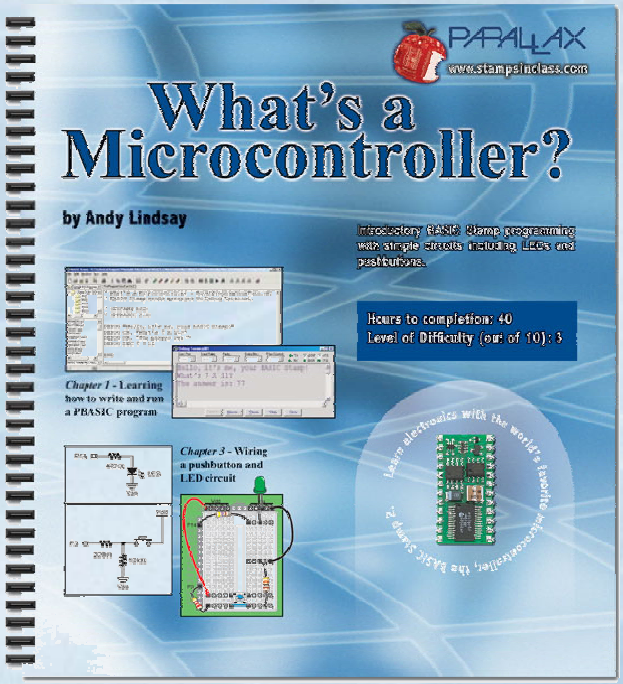
Chapter 4: Controlling Motion

What's a Microcontroller?



Presentation based on:

"What's a Microcontroller ?"

By Andy Lindsay

Parallax, Inc

Presentation developed by:

Martin A. Hebel

Southern Illinois University Carbondale

College of Applied Sciences and Arts

Electronic Systems Technologies

9/02/03

Presentation Index

What's a Microcontroller?

Use and Copyright

This presentation supplements **"What's a Microcontroller"** by Andy Lindsay. (Link to text at Parallax)

- ✓ This presentation is not a replacement for the text.
- ✓ Important concepts of the text are highlighted.
- ✓ In some cases, additional material has been added to augment the text. Denoted by titles colored gold.
- ✓ Full program listings are generally not provided in the presentation.

Distribution:

COPYRIGHTS AND TRADEMARKS

DISCLAIMER OF LIABILITY

Microcontroller Motion

Microcontrollers control the motion of many things in our daily lives:

- ✓Printer head movement.
- ✓DVD and VCR mechanisms.
- ✓Grocery store automatic doors.
- ✓Robotic movement.

Instead of being simply ON-OFF, many of these motion devices require very fast pulses of signals for position control or movement.

Examples of motors and motion devices:

- ✓ DC Motors
- ✓ AC Motors
- ✓ Stepper Motors
- ✓ Servos

All of these can be controlled from the BASIC Stamp, though many need additional electronic circuitry or additional mechanical components. The BASIC Stamp cannot directly drive a 25 amp DC motor, but it could with some additional components!

The hobby servo is the simplest and most directly-useful of all DC motors to control from the BASIC Stamp and will be the topic of this chapter.

The hobby servo is easy to connect and control, and has a useful mechanical output.

Activity #1: Connecting and Testing the Servo

What's a Microcontroller?

Servo Parts:

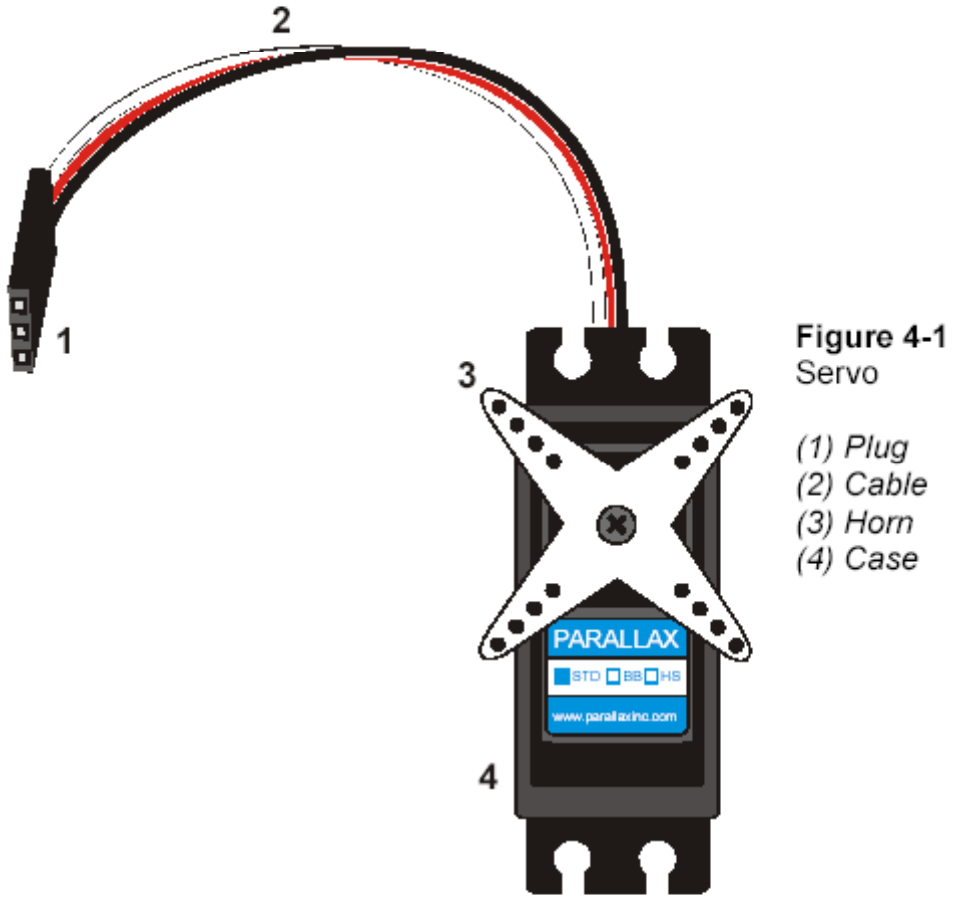


Figure 4-1
Servo

(1) Plug
(2) Cable
(3) Horn
(4) Case

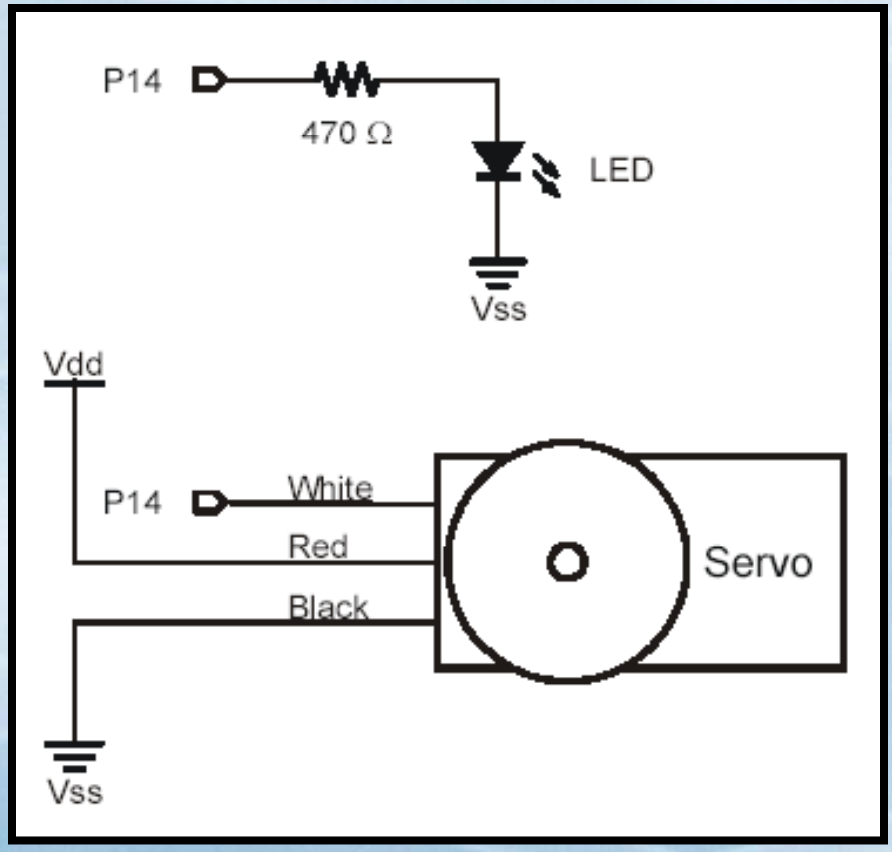
Servo Connector:

Black – Vss

Red – Vdd or Vin

White – Signal

Circuit to be built:

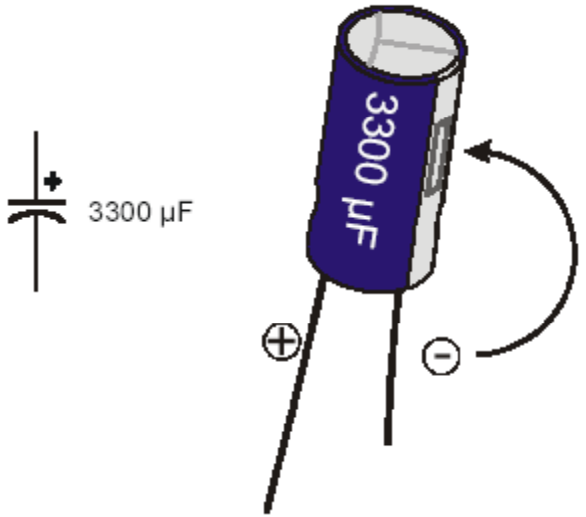


The servo can be damaged with voltages above 9V. A wall-transformer may be stated to be 9V, but output 12V. If using anything but batteries, verify the supply is no more than 9V. See your text.

Connecting to the servo to the board depends on which board you have.

Servo on Board of Ed. Rev A or HomeWork Board

A capacitor is needed to act as a 'surge volume' for electrons. It will supplement the voltage regulator when the servo draws power.



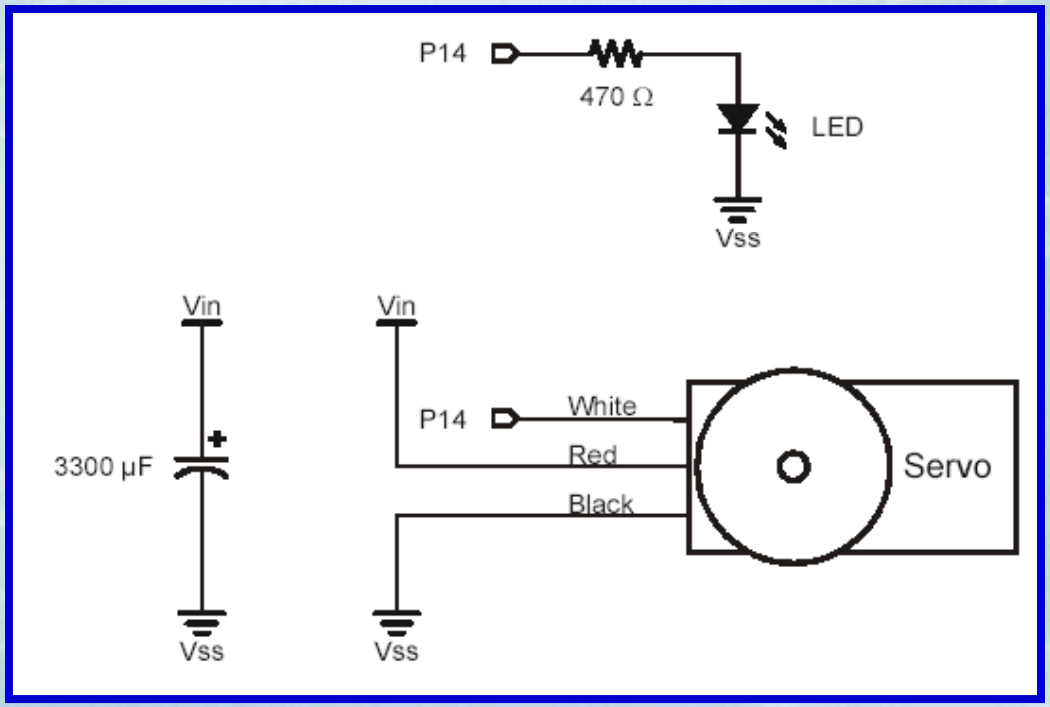
IMPORTANT: This capacitor has a positive and negative terminal. The negative terminal is the lead that comes out of the metal canister closest to the stripe with a negative (–) sign.

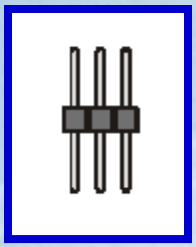
Electrolytic capacitors can explode if connected reverse polarity. Take care connecting and applying power.

Wear Safety Glasses!

Servo on Board of Ed. Rev A or HWB (cont)

- ✓Remove power
- ✓Connect circuit

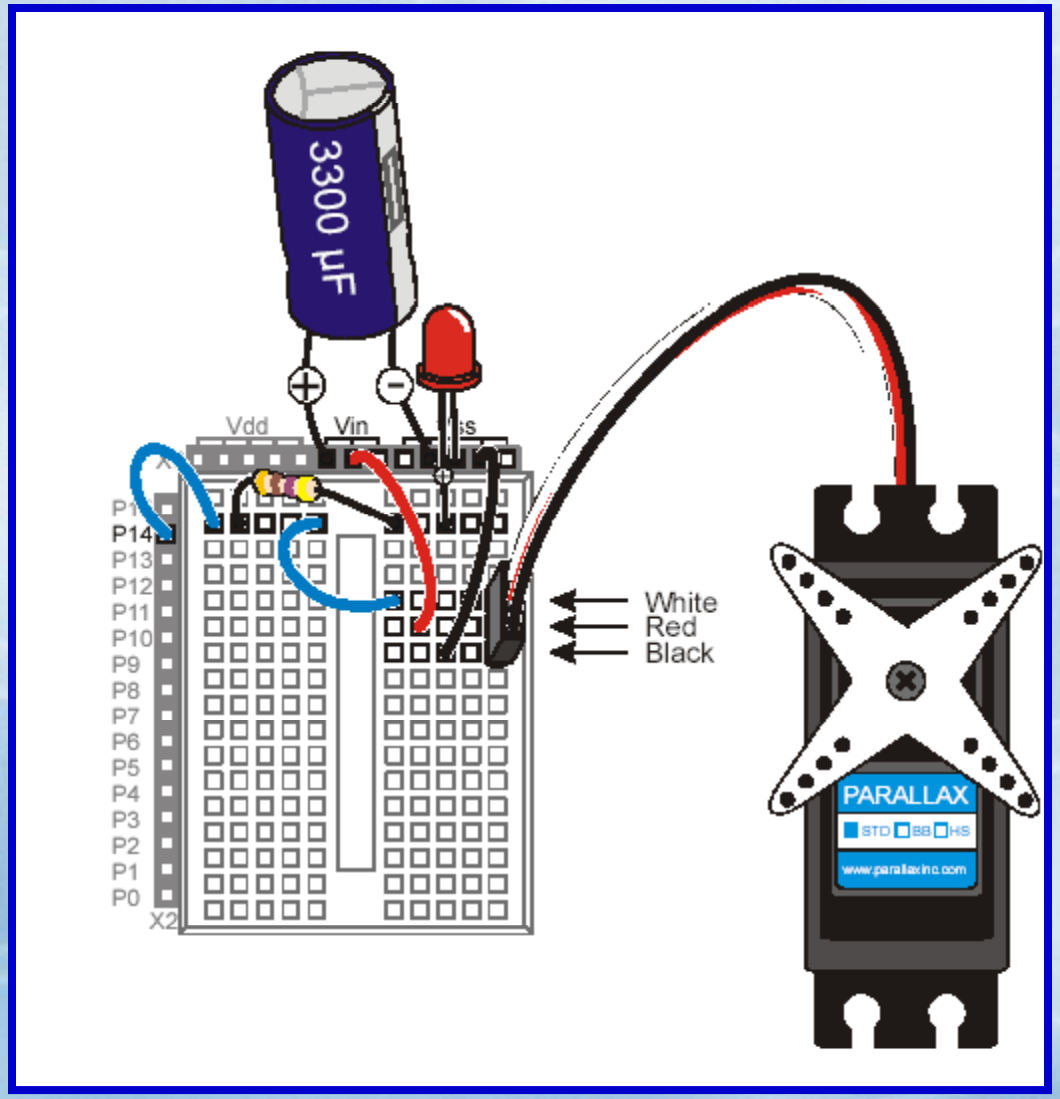




Use a connector like this

Servo on Board of Ed. Rev A or HWB (cont)

✓Connect Servo



Programming Servo Control

The servos is controlled by bursts of signals **spaced 20mS** apart. A high signal can last between **1mS to 2mS**.

The PULSOUT instruction is used to send the signals:

PULSOUT *pin, duration*

pin: Defines which I/O pin to use.

duration: defines how long the pulse should last, but it in NOT in mS.

The PULSOUT duration is in 2 microsecond (uS) increments.

1 μS = .000001 seconds.
1 mS = 1000μS

For a command of: PULSOUT 14,750

This would be sending a pulse that lasts 750 x 2 μS = 1500 μS or 1.5mS on pin 14.

The ServoTest program sends out groups of pulses with durations of:

500 = 1 mS

750 = 1.5 mS

1000= 2 mS

Example Code:

```
FOR counter = 1 TO 150
   PULSOUT 14, 1000
   PAUSE 20
NEXT
```

When ran, the pulse of 1000 is sent 150 times with 20mS pauses.

```
counter VAR Word
DEBUG "Counterclockwise 10 o'clock", CR
FOR counter = 1 TO 150
  PULSOUT 14, 1000
  PAUSE 20
NEXT

DEBUG "Clockwise 2 o'clock", CR
FOR counter = 1 TO 150
  PULSOUT 14, 500
  PAUSE 20
NEXT

DEBUG "Center 12 o'clock", CR
FOR counter = 1 TO 150
  PULSOUT 14, 750
  PAUSE 20
NEXT

DEBUG "All done."
END
```

As the program is ran, the servo's horn turns from the 10 o'clock to 2 o'clock to 12 o'clock positions.

The pulse width defines the absolute position the servo should move to over its 90 degree range.

There a variety of servos, and the range of travel may vary.

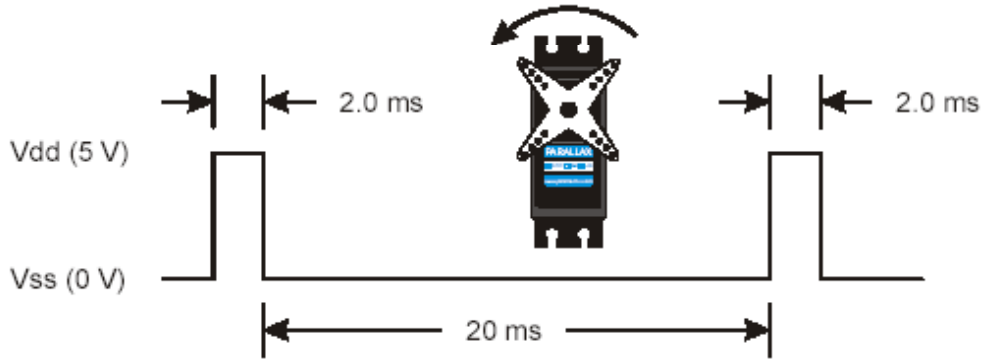


Figure 4-15
Timing Diagram for 2.0 ms Pulses Every 20 ms

Servo horn in 10 o'clock position.

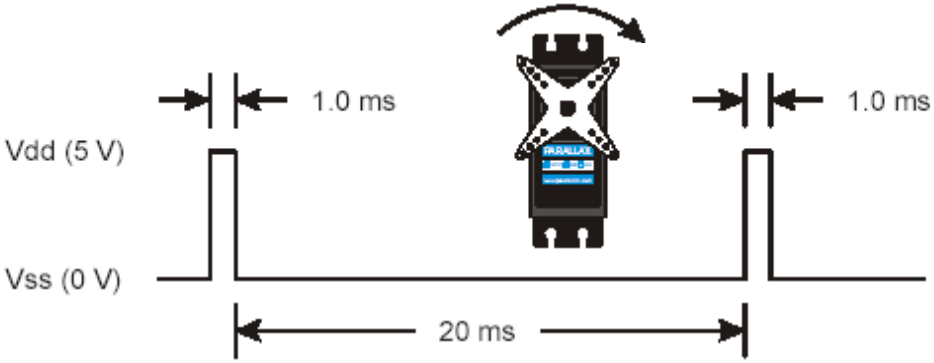


Figure 4-16
Timing Diagram for 1.0 ms Pulses Every 20 ms

Servo horn in 2 o'clock position.

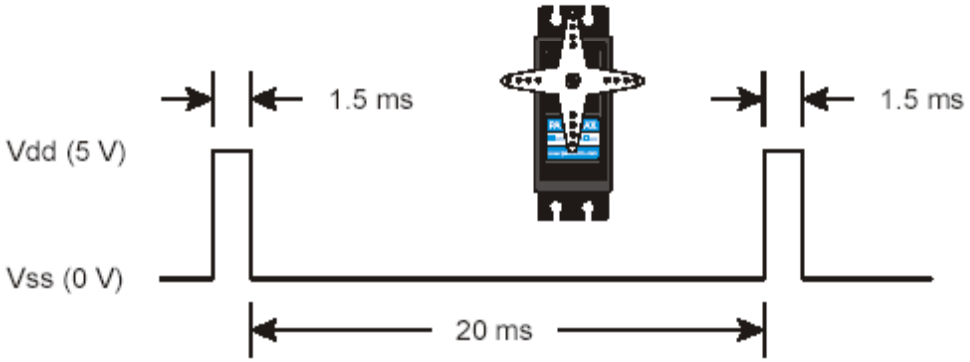


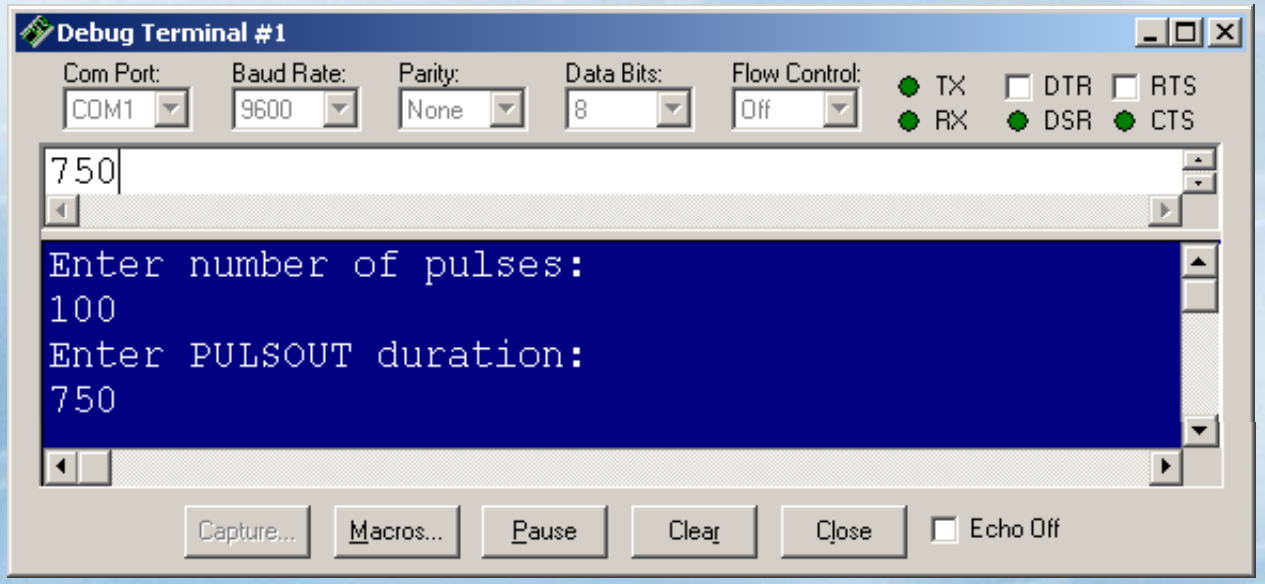
Figure 4-17
Timing Diagram for 1.5 ms Pulses Every 20 ms

Servo horn is in 12 o'clock position.

Activity #2: Controlling position with your Computer

The Servo Control With Debug program uses the DEBUG debug window to allow the user to enter the number of pulses and the duration.

Once prompted, the user enters the data:



The **DEBUGIN** command is used to accept data FROM the computer sent TO the BASIC Stamp when it is entered in the text box.

DEBUGIN DEC Pulses

Experiment with the values. Be sure to use durations between 500 and 1000 to prevent damage to the servo.

```
' {$STAMP BS2}
' {$PBASIC 2.5}
counter VAR Word
pulses VAR Word
duration VAR Word

DO
  DEBUG CLS, "Enter number of pulses:", CR
  DEBUGIN DEC pulses
  DEBUG "Enter PULSOUT duration:", CR
  DEBUGIN DEC duration
  DEBUG "Servo is running...", CR
  FOR counter = 1 TO pulses
    PULSOUT 14, duration
    PAUSE 20
  NEXT
  DEBUG "DONE"
  PAUSE 1000
LOOP
```

Of course, users make mistakes, so it would be a good idea to ensure the user enters durations only within the legal range!

If you make a mistake entering a number, press the reset button on the board. The backspace key sends data which the BASIC Stamp uses the same as the Enter key.

Activity #3: Converting Position to Motion

By changing the horn's position each pass through a loop, you can get motion. The **STEP** part of the FOR...NEXT defines how much to add each repetition.

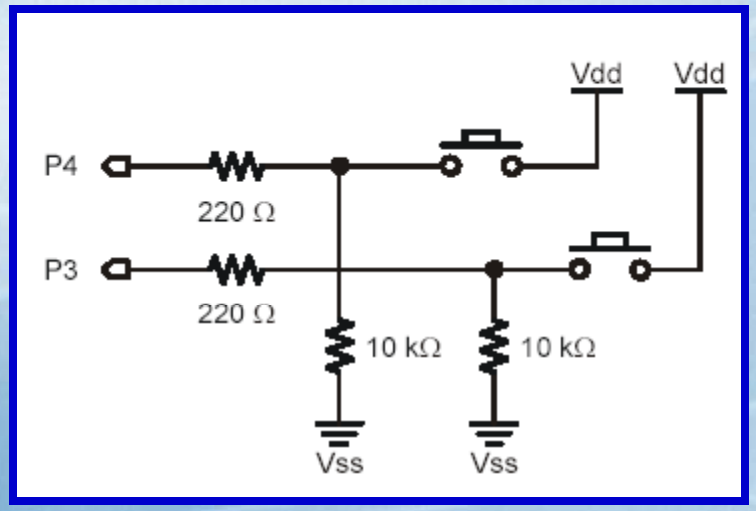
By modifying the STEP value the velocity of the servo can be controlled.

```
DEBUG CLS, "Pulse width increment by 4", CR
FOR counter = 500 TO 1000 STEP 4
  PULSOUT 14, counter
  PAUSE 20
  DEBUG DEC5 counter, CR, CRSRUP
NEXT
```

Activity #4: Servo Control with Pushbuttons

What's a Microcontroller?

Add two pushbuttons to the circuit to control the Servo's position.




```
duration VAR Word
duration = 750
DO
  IF IN3 = 1 THEN
    IF duration > 500 THEN
      duration = duration - 25
    ENDIF
   ENDIF

  IF IN4 = 1 THEN
    IF duration < 1000 THEN
      duration = duration + 25
    ENDIF
  ENDIF

  PULSOUT 14, duration
  PAUSE 10
  DEBUG HOME, DEC4 duration, " = duration"
LOOP
```

Standard and Continuous Rotation Servos

The standard servo has a feedback mechanism connected to the horn shaft. As the pulse is read, the actual position is compared to the setting, and the horn will turn until the two match.

The servo has a very high torque when being positioned.

The continuous rotation servo, or 'modified servo' has the feedback and mechanical stops removed. The feedback mechanism is adjusted so the a center pulse (750 or 1.5mS) stops the servo.

Pulses above and below center allow it to rotate freely at varying speeds in both directions. This makes a popular wheel drive system for table-top robots such as the Boe-Bot.

Chapter #4 Review

- ✓ Motors, stepper motors and _____ perform mechanical motion that can be controlled by the BASIC Stamp.
- ✓ The _____ is easiest to control with no additional hardware or electronics.
- ✓ The Servo's position is controlled by high pulses lasting from _____ – _____.
- ✓ Pulses are required to have a _____ low time between them.
- ✓ The _____ command is used to send pulses. The duration is in _____ increments.
- ✓ How can you control a servo's velocity?

Links

- ✓ BASIC Stamp Home
- ✓ Stamps In Class Home
- ✓ BASIC Stamp Software
- ✓ BASIC Stamp Robots
- ✓ BASIC Stamp Yahoo Group
- ✓ Stamps In Class Yahoo Group
- ✓ SIUC EST Degree