Chapter 4 Motion

Movement

- **Electrical Motors**
  - **DC Motors**
  - **Stepper motor**
  - **Servo**
  - **Modified servo**
- **Pneumatic or Hydraulic systems**

Nov 5 - 5:18 PM

In this chapter you will be experimenting with and programming the movement of a standard servo.

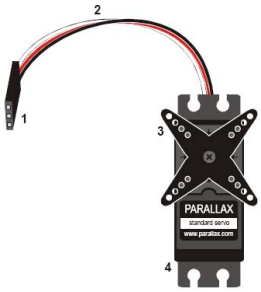

Figure 4-1
The Parallax Standard Servo

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

Nov 5 - 5:20 PM

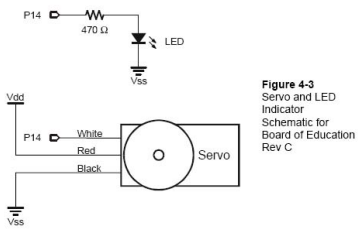

Figure 4-3
Servo and LED Indicator Schematic for Board of Education Rev C

Nov 5 - 5:20 PM

Timing Diagram

The second **FOR...NEXT** loop delivers 150 pulses, but this time, each pulse only lasts 1.0 ms. This instructs the servo to turn to the 2 o'clock position for about 3.15 seconds.

```
FOR COUNTER = 1 TO 150
  PULSOUT 14, 500
  PAUSE 20
NEXT
```

PULSOUT 14, 500 sends a pulse that lasts 500 × 2 µs. That's 1000 µs or 1 ms.

Figure 4-15 shows the timing diagram for this pulse train. The pauses between pulses

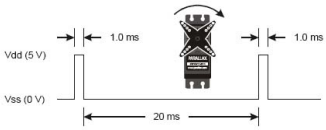

Figure 4-16
Timing Diagram for 1.0 ms Pulses Every 20 ms

Servo horn in 2 o'clock position.

The last **FOR...NEXT** loop delivers 150 pulses, each of which lasts 1.5 ms. This instructs the servo to go to its center position (12 o'clock) for about 3.23 seconds.

```
FOR counter = 1 TO 150
  PULSOUT 14, 750
  PAUSE 20
NEXT
```

Nov 5 - 5:20 PM

The last **FOR…NEXT** loop delivers 150 pulses, each of which lasts 1.5 ms. This instructs the servo to go to its center position (12 o'clock) for about 3.23 seconds.

```
FOR counter = 1 TO 150
  PULSOUT 14, 750
  PAUSE 20
NEXT
```

PULSOUT 14, 750 sends a pulse that lasts 750 × 2 µs. That's 1500 µs or 1.5 ms.

Figure 4-17 shows the timing diagram for these pulses. While the low time is still 20 ms, the pulse now lasts for 1.5 ms.

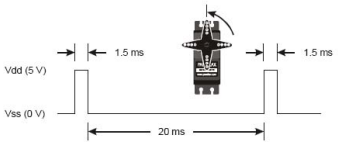

Figure 4-17
Timing Diagram for 1.5 ms Pulses Every 20 ms

Servo horn is in 12 o'clock position.

Do the Math

If you want to convert time from milliseconds to a **Duration** you can use for **PULSOUT**, use this equation.

Duration = number of ms × 500

Nov 5 - 5:22 PM

PULSOUT 14, 1000 sends a pulse that lasts 1000 × 2 µs. That's 2000 µs or 2 ms.

Figure 4-15 is called a timing diagram. It shows a picture of the high and low signals and how long they last. The timing diagram does not show how many pulses are delivered, but it does give you information about how long the high and low signals last. Each pulse (high signal) lasts for 2.0 ms. Each pulse is separated by a 20 ms delay while the signal is low.

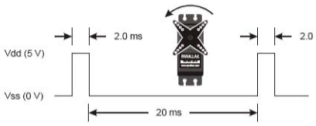

Figure 4-15
Timing Diagram for 2.0 ms Pulses Every 20 ms

Servo horn in 10 o'clock position.

The second **FOR…NEXT** loop delivers 150 pulses, but this time, each pulse only lasts 1.0 ms. This instructs the servo to turn to the 2 o'clock position for about 3.15 seconds.

```
FOR COUNTER = 1 TO 150
  PULSOUT 14, 500
  PAUSE 20
NEXT
```

Nov 5 - 5:20 PM

New Basic Commands:

Pulsout

Debugin

Nov 8 - 5:49 AM

Summary

Pulsout port, time
port = pin number (0-15)
time = number of 2 microsecond 'clicks'
must be between 500 and 1000

m = milli = 10^{-3}
µ = micro = 10^{-6}

Therefore:
Pulsout 14, 1000 'counter clockwise extreme
Pulsout 14, 500 ' clockwise extreme

However: A series of pulses need to be sent at 20 ms intervals

Therefore, the programming block looks like the following:

```
For counter = 1 to 200
    Pulsout 14, 1000
    Pause 20
Next
```

Nov 5 - 5:34 PM

```
' What's a Microcontroller - ServoControlWithDebug.bs2
' Send messages to the BASIC Stamp to control a servo using
' the Debug Terminal.

' {$STAMP BS2}
' {$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
```

Nov 8 - 5:48 AM

Practical Test:

A button at a railroad crossing triggers the change of the light from green to red and also movement of a traffic-blocking arm.

Start with Green On, Red Off, Servo at 500
Wait for button to be pressed then
Turn Red ON turn Green OFF, Move Servo to 1000
After 5 seconds, assuming the button is not pressed, reset to the starting state and wait.

Nov 20 - 5:13 PM

Nov 20 - 5:05 PM