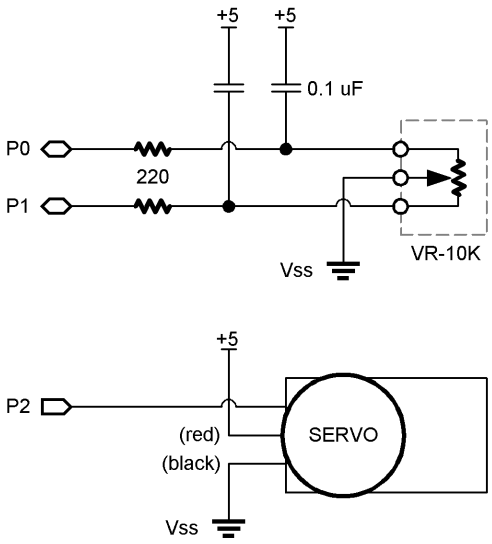
Experiment #25: Hobby Servo Control

This experiment demonstrates the control of a standard hobby servo. Hobby servos frequently are used in amateur robotics.

New PBASIC elements/commands to know:

- SDEC, SDEC1 – SDEC16 (DEBUG modifier)

Building The Circuit




```
' ===============================================================================
'
'   File..... Ex25 - Servo.BS2
'   Purpose... Hobby Servo Control
'   Author.... Parallax
'   E-mail.... stamptech@parallaxinc.com
'   Started...
'   Updated... 01 MAY 2002
'
'   {$STAMP BS2}
```

Experiment #25: Hobby Servo Control

```
'
' ===============================================================================


' -------------------------------------------------------------------------------
' Program Description
' -------------------------------------------------------------------------------

' This program shows how to control a standard servo with the BASIC Stamp.


' -------------------------------------------------------------------------------
' I/O Definitions
' -------------------------------------------------------------------------------

PotCW           CON     0                       ' clockwise pot input
PotCCW          CON     1                       ' counter-clockwise pot input
Servo           CON     2                       ' servo control pin


' -------------------------------------------------------------------------------
' Constants
' -------------------------------------------------------------------------------

Scale           CON     $0068                   ' scale RCTIME to 0 - 250, BS2
' Scale         CON     $002C                   ' BS2sx
' Scale         CON     $002A                   ' BS2p


' -------------------------------------------------------------------------------
' Variables
' -------------------------------------------------------------------------------

rcRt            VAR     Word                    ' rc reading - right
rcLf            VAR     Word                    ' rc reading - left
diff            VAR     Word                    ' difference between readings
sPos            VAR     Word                    ' servo position


' -------------------------------------------------------------------------------
' Program Code
' -------------------------------------------------------------------------------

Main:
  HIGH PotCW                                    ' discharge caps
  HIGH PotCCW
  PAUSE 1
```

```
RCTIME PotCW, 1, rcRt                          ' read clockwise
RCTIME PotCCW, 1, rcLf                         ' read counter-clockwise

rcRt = (rcRt */ Scale) MAX 250                 ' scale RCTIME to 0-250
rcLf = (rcLf */ Scale) MAX 250
sPos = rcRt - rcLf                             ' calculate position (-250 to 250)

PULSOUT Servo, (750 + sPos)                    ' move the servo
PAUSE 20

GOTO Main
```

Behind The Scenes

Hobby servos are specialized electromechanical devices used most frequently to position the control surfaces of model aircraft. The position of the servo output shaft is determined by the width of an incoming control pulse. The control pulse is typically between one and two milliseconds wide. The servo will center when the control signal is 1.5 milliseconds. In order to maintain its position, the servo must constantly be updated. The typical update frequency is about 50 times per second.
The BASIC Stamp's **PULSOUT** command is ideal command for controlling hobby servos. In this experiment, two **RCTIME** circuits are constructed around the 10K potentiometer. This circuit and the project code can be used to determine the relative position of the potentiometer. The readings from each side of the potentiometer are scaled between 0 and 250 with the ***/** and **MAX** operators. By subtracting one side from the other, a servo position value between –250 and +250 is returned.

This value is added to the centering position of 750. Remember that **PULSOUT** works in two-microsecond units, so a **PULSOUT** value of 750 will create a pulse that is 1.5 milliseconds wide, causing the servo to center. When the servo position is –250, the **PULSOUT** value is 500, creating a 1.0-millisecond pulse. At an **sPos** value of +250, the **PULSOUT** value is 1000, creating a 2.0 millisecond control pulse.

This code demonstrates that the BASIC Stamp does, indeed, work with negative numbers. You can see the value of **sPos** by inserting this line after the calculation:

```
DEBUG Home, "Position: ", SDEC sPos, "   "
```

Negative numbers are stored in two's compliment format. The **SDEC** (signed decimal) modifier prints standard decimal with the appropriate sign.

Experiment #25: Hobby Servo Control

Challenge

Replace the potentiometer with two photocells and update the code to cause the servo to center at the brightest light source.