Parallax Continuous Rotation Servo (#900-00008)

The Parallax Standard Servo is ideal for adding bidirectional continuous rotation to your robotics projects.

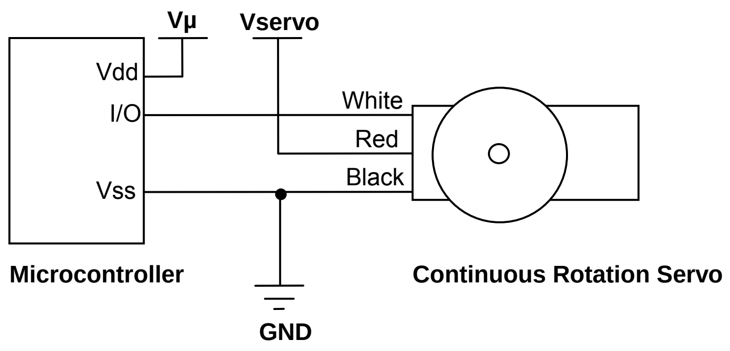
Features

- Bidirectional continuous rotation
- 0 to 50 RPM, with a linear response to PWM for easy ramping
- Accepts four mounting screws
- Easy to interface with any Parallax microcontroller or PWM-capable device
- Very easy to control with the PULSOUT command in PBASIC or SX/B
- Weighs only 1.50 oz (42.5 g)
- 38 oz-in torque @ 6 V

Key Specifications

- Power requirements: 4 to 6 VDC; Maximum current draw 140 +/- 50 mA at 6 VDC when operating in no load conditions, 15 mA when in static state
- Communication: pulse-width modulation
- Dimensions: approx 2.2 x 0.8 x 1.6 in (5.58x 1.9 x 4.06 cm) excluding servo horn
- Operating temperature range: 14 to 122 °F (-10 to +50 °C)

Quick-Start Circuit



Vμ = microcontroller voltage supply

Vservo = 4 to 6 VDC, regulated or battery

I/O = PWM TTL or CMOS output signal, 3.3 to 5 V; < Vservo + 0.2 V

Device Information

The Parallax continuous rotation servo relies on pulse width modulation to control the rotation speed and direction of the serv o shaft. Before u sing the servo in a project, it is important to c alibrate the center position of the servo in order to defi ne the pul se width value at which the servo holds still (see the section Calibration – "Center" the Servo on page 4).

Specifications

Pin	Name	Description	Minimum	Typical	Maximum	Units
1 (White)	Signal	Input; TTL or CMOS	3.3	5.0	Vservo + 0.2	V
2 (Red)	Vservo	Power Supply	4.0	5.0	6.0*	V
3 (Black)	Vss	Ground		0		V

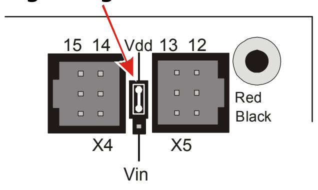
*See Board of Education Servo Header Connection Diagram.

Power Precautions

- Do not use this servo with an unregulated wall-mount supply. Such powe supplies may deliver variable voltage far above the stated voltage.
- Do not power this servo through the BASIC Stamp® Module's Vin pin, this can deliver voltages above the stated voltage. See the Board of Education Connection Diagram below for jumper settings.
- Servo current draw can spike while under peak load; be sure your application's regulator is prepared to supply adequate current for all servos used in combination.

Board of Education Servo Header Connection Diagram

When connecting the servo to the Board of Education® servo header, be sure the jumper is set t o Vdd (regulated 5 VDC for this board) as shown in the figure below. Failure to place the jumper at this setting can cause damage your servo! (Note: see the Board of Education product documentation for instructions regarding earlier board revisions that do not have a servo header with a jumper.)



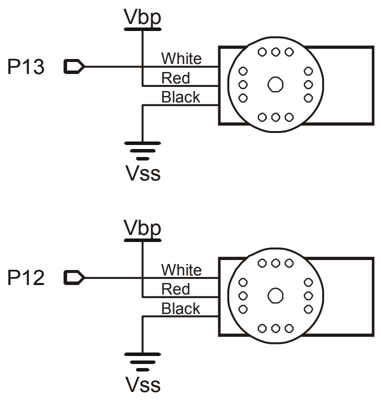
Using a Separate Power Supply on a HomeWork Board

The BASIC Stamp HomeWork Board uses a 9 V battery for a power supply. A servo can drain a fresh 9 V battery in un der 20 minutes! Foll ow these directions to build two servo p orts on the breadboard, and power them with a separate battery pack.

Hardware Required

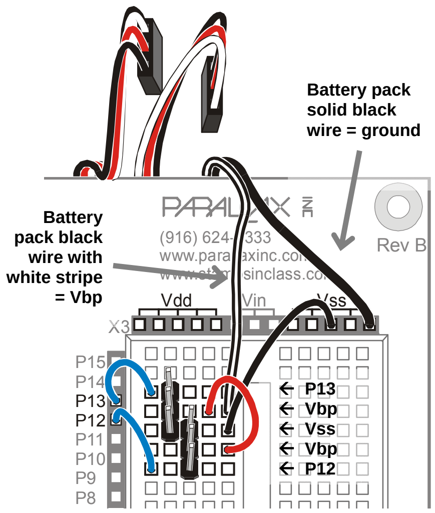
(1) BASIC Stamp HomeWork Board (serial #555-28158 or USB #555-28188)
(1) Battery pack with tinned leads (Parallax #753-00001)
(2) Parallax continuous rotation servos
(2) 3-pin male-male headers (Parallax #451-00303)
(4) Jumper wires (10-pack: Parallax #800-00016
(4) 1.5 V AA batteries

- √ Disconnect the 9 V battery from the board, and do not put the AA batteries in their holder yet.
- √ Build the servo ports shown by the schematic and wiring diagram below.
- √ Double-check to make sure the black wire with the white stripe is connected to Vbp, the solid black wire is connected to Vss, and that all the connections for P13, Vbp, Vss, Vbp (another one), and P12 all exactly match the wiring diagram.
- √ Connect the servo plugs to the male headers on the right side of the wiring diagram.
- √ Connect the 9 V battery, and insert the AA batteries into their holder.

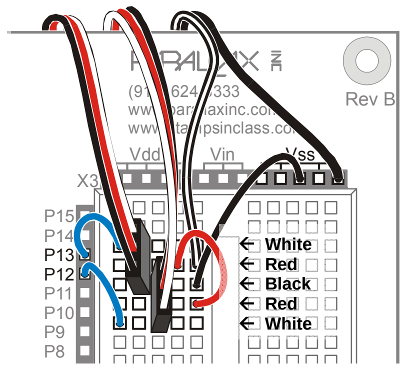


Vbp stands for Voltage battery pack.

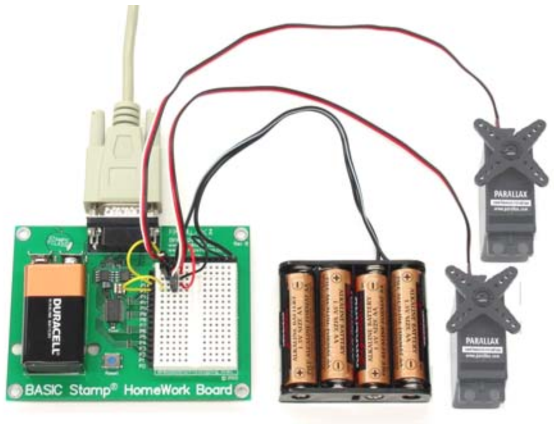
It refers to the 6 VDC supplied by the four 1.5 V batteries. This is brought directly to the breadboard to power the servos for Boe-Bots built with the HomeWork Board. Your BASIC Stamp is still powered by the 9 V battery.



Port connections



Servo connections by wire color



HomeWork Board with servo ports built on the breadboard, with a separate battery pack power supply for the servos.

Calibration – "Center" the Servo

The servo has a potentiometer access port, right above the place where the cable attaches to the case. The port allows the user to adjust the servo to hold completely still when receiving a 1.5 ms pulse width. This is the value in the "center" of the range of control pulses the servo will accept.

To center the servo, program your host device to deliver a 1.5 ms pulse, continually refreshed every 20 ms. Sample calibration code is given below for all BASIC Stamp models, Spin for the Propeller™ P8X32A microcontroller, and SX/B for the SX chip. All are available for download from the 900-00008 product page at www.parallax.com.

Connect the servo to your microcontroller's I/O pin. The example programs below specify an I/O pin.

BASIC Stamp Calibration Code - for all BS2 models

- √ Connect the servo to BASIC Stamp 1/O pin P12, or update the ToServo PIN declaration.
- √ Run the program, and gently twist the potentiometer adjustment screw until the servo does not turn or vibrate. *NOTE: Calibrating the servo may take some patience. The potentiometer is very sensitive so a very light touch will be required.*

```
'  CenterParallaxCrServo.bs2
' {$STAMP BS2}
' {$PBASIC 2.5}

#SELECT $Stamp
  #CASE BS2, BS2E, BS2PE          ' PULSOUT Duration units are 2 us for these models
    Center CON 750
  #CASE BS2SX, BS2P, BS2PX        ' PULSOUT Duration units are 0.8 us for these models
    Center CON 1875
#ENDSELECT

ToServo PIN 12                    ' connect servo to I/O pin P12, or change it here

DO
  PULSOUT ToServo, Center         ' ToServo pin outputs 1.5 ms pulse
  PAUSE 20                        ' refresh pulse every 20 milliseconds
LOOP
```

Propeller Chip Calibration Code – for P8X32A

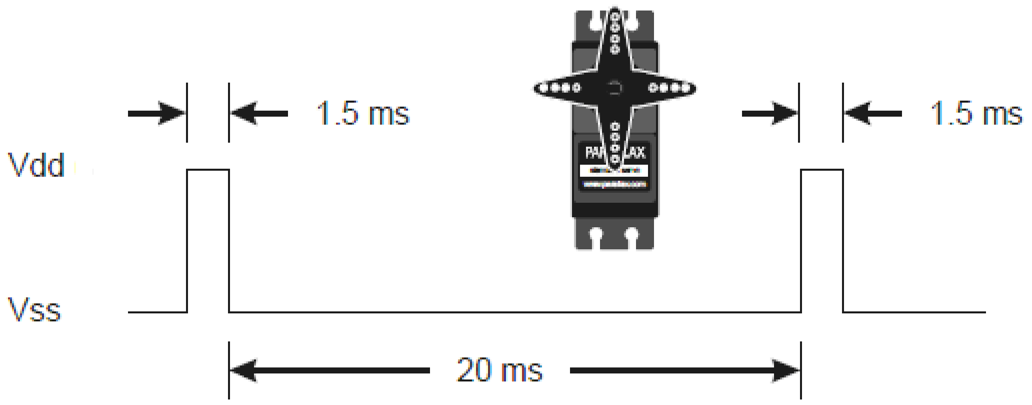
- √ Download and unzip the Propeller code file from the 900-00008 product page.
- √ Connect the servo signal pin to Propeller I/O pin P0.
- √ Run the program CenterParallaxServo.spin, and gently twist the potentiometer adjustment screw until the servo does not turn or vibrate. *NOTE: Calibrating the servo may take some patience. The potentiometer is very sensitive so a very light touch will be required.*

```
{{ CenterParallaxServo.spin
For centering Parallax Continuous Rotation Servo
or holding Parallax Standard Servo at 90° position.
Sends a 1.5 ms pulse approx every 20 ms }}

CON
_clkmode = xtal1 + pll16x              ' System clock → 80 MHz
_xinfreq = 5_000_000                   ' Using 5 MHz external crystal oscillator
servoPin = 0                           ' Servo signal to this I/O pin-change if needed

PUB CenterServo | tInc, tc, tHa, t
```

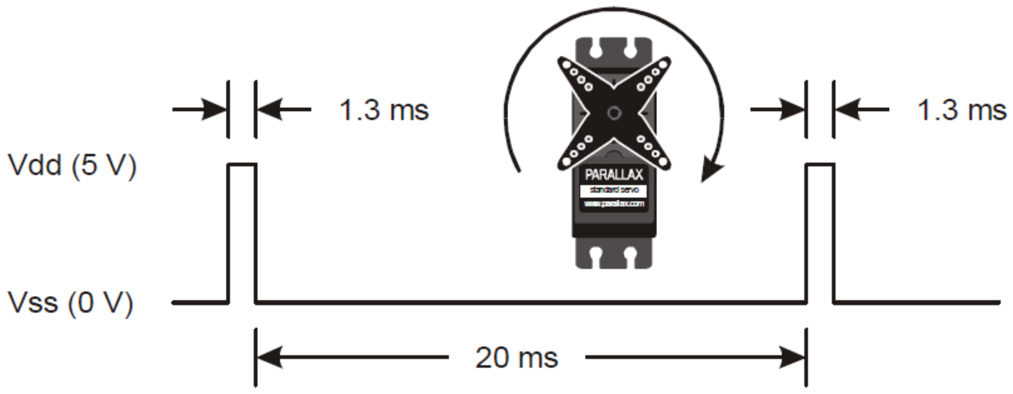
```
ctra[30..26] := %00100                      ' Configure Counter A to NCO
ctra[8..0]   := servoPin

frqa := 1
dira[servoPin]~~

' Set up cycle and high times
tInc := clkfreq/1_000_000
tC   := tInc * 21_500
tHa  := tInc * 1500
t    := cnt                                 ' Mark counter time

repeat                                      ' Repeat PWM signal
  phsa := -tHa                              ' Set up the pulse
  t += tC                                   ' Calculate next cycle repeat
  waitcnt(t)                                ' Wait for next cycle
```

Communication Protocol

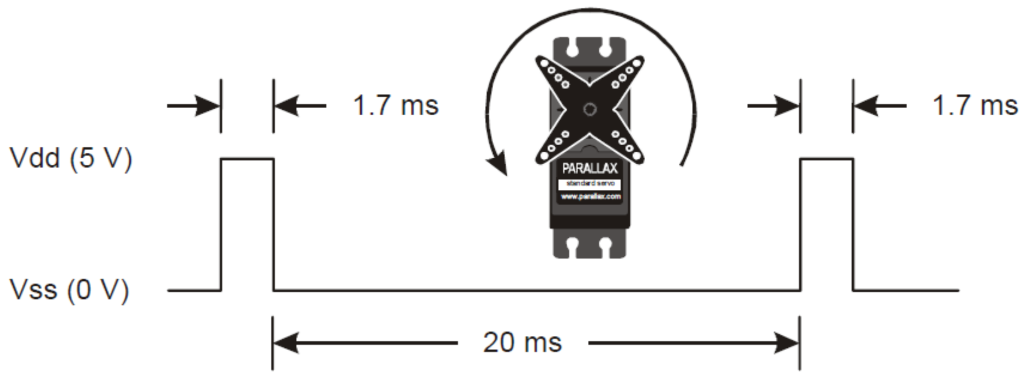
The Parallax Continuous Rotation Servo is controlled through pulse width modulation. Rotational speed and direction are determi ned by the d uration of a high pulse, in the 1.3–-1.7 ms range. In order for smooth rotation, the servo needs a 20 ms pause between pulses. Below is a sample timing diagram for a centered servo:



As the length of the pulse decreases from 1.5 ms, th e servo will gradually rotate faster in th e clockwise direction, as can be seen in the figure below:



Likewise, as the length of the pulse increases from 1.5 ms, the servo will gradually rotate faster in the counter-clockwise direction, as can be seen in the figure below:



Voltage and RPM: Maximum RPM will vary with input voltage; 50 RP M @ 5 V is typical. Using regulated Vdd as the sup ply source wi ll reduce fluct uations in RPM for a given pulse width that might otherwise occur with unregulated battery supplies.

BASIC Stamp® Programming Examples

PBASIC has a PULSO UT command that sets the I/O *Pin* to an output and sends a pulse of the specified *Duration*. Since the servo needs this pulse refreshed every 20 ms for continuous operation, the PULSOUT command is put in a coun ted FOR…NEXT loop to su stain continuous operation for the speci fied number of cycles.

PULSOUT *Pin*, *Duration*

Different BASIC Stamp m odules use different units for the P ULSOUT command's *Duration* argument. When adapting BS2 code to another BASIC Stamp model, you may need to make adjustments. The table below lists the PULSOUT ranges for each BASIC Stamp microcontroller. See the BASIC Stamp Manual or BASIC Stamp Editor Help for more information.

BASIC Stamp Model	1.3 ms (Full speed clockwise)	1.5 ms (Center, no rotation)	1.7 ms (Full speed counterclockwise)
BS1	130	150	170
BS2, BS2e, BS2pe	650	750	850
BS2sx, BS2p, BS2px	1625	1875	2125

The example shown below for a BASIC Stamp 2 causes a servo connected to BASIC Stamp 1/0 pin 12 to first rotate full-speed counterclockwise for about 3 seconds, hold still f or about 3 seconds, and then rotate counterclockwise for about 3 seconds.

```
' RotateParallaxCrServo.bs2
' {$STAMP BS2}
' {$PBASIC 2.5}

counter VAR Word
servoPin PIN 12             ' change I/O pin for servo signal here


FOR counter = 1 TO 100      ' Rotate counterclockwise for ~3 seconds

  PULSOUT servoPin, 850
  PAUSE 20

NEXT
```

```
FOR counter = 1 TO 100      ' Hold still for ~3 seconds

  PULSOUT servoPin, 750
  PAUSE 20

NEXT

FOR counter = 1 TO 100      ' Rotate clockwise for ~3 seconds

  PULSOUT servoPin, 650
  PAUSE 20

NEXT

END
```

For more examples with the BASIC Stamp 2, incl uding 2-wheeled robot maneuvers and ramping, see *Robotics with the Boe-Bot* Chapter 4, available for free downl oad from the 28132 product page at www.parallax.com.

Propeller™ P8X32A Application

The program below uses counter modules to rotate the servo first clockwise at full speed for 2 seconds, then rests for 2 seconds, and rotates counterclockwise at full speed for another 2 seconds. This code can also be downloaded from the 900-00008 product page.

```
{{ ServoContinuousRotation.spin
Turn Parallax Continuous Rotation Servo clockwise full speed for 21 sec.
hold still 2 sec, and then counterclockwise full speed for 2 sec. }}

CON
_clkmode = xtal1 + pll16x                 ' System clock → 80 MHz
_xinfreq = 5_000_000                      ' Using 5 MHz external crystal oscillator
servoPin = 0                              ' Servo signal to this I/O pin-change if needed

PUB CenterServo | tInc, tc, tCtr, tCw, tCcw, t

ctra[30..26] := %00100                    ' Configure Counter A to NCO
ctra[8..0] := servoPin

frqa := 1
dira[servoPin]~~


tInc := clkfreq/1_000_000                 ' 1 µs increment
tC   := tInc * 21_500                     ' Low pulse
tCtr := tInc * 1500                       ' Center pulse = 1.5 ms
tCw  := tInc * 1300                       ' Clockwise fast = 1.3 ms
tCcw := tInc * 1700                       ' Counter-Clockwise fast = 1.7 ms
t    := cnt                               ' Mark counter time

repeat 100                                ' Repeat PWM signal 100x
  phsa := -tCw                            ' Set up clockwise fast pulse
  t += tC                                 ' Calculate next cycle repeat
  waitcnt(t)                              ' Wait for next cycle (20 ms)

repeat 100                                ' Repeat PWM signal 100x
  phsa := -tCtr                           ' Set up the center pulse
  t += (tC + 200)                         ' Calculate next cycle repeat
  waitcnt(t)                                ' Wait for next cycle (20 ms)

repeat 100                                  ' Repeat PWM signal 100x
  phsa := -tCcw                             ' Set up counter-clockwise fast pulse
  t += (tC - 200)                           ' Calculate next cycle repeat
  waitcnt(t)                                ' Wait for next cycle (20 ms)
```

Revision History

Version 2.1: corrected values in BASIC Stamp Model PULSOUT table; updated example programs to use a constant for the servo pin.

Version 2.2: added **Voltage and RPM** note on p age 6. Added Usin g a Separate Power Supply on a HomeWork Board section beginning on page 2. Updated specifications.