Web Site: www.parallax.com
Forums: forums.parallax.com
Sales: sales@parallax.com
Technical: support@parallax.com

Office: (916) 624-8333
Fax: (916) 624-8003
Sales: (888) 512-1024
Tech Support: (888) 997-8267

Gimbal Joystick with Adapter (#27808)

This quad-bearing gimbal joystick provides a two-axis resistive output for controlling position, velocity, tilt, etc., as an input device to various microcontrollers, including Parallax's P8X32A Propeller chip and BASIC Stamp family of modules. It may be used with microcontrollers that have analog inputs, with an external ADC, or with resistor-capacitor circuits for a purely digital solution.

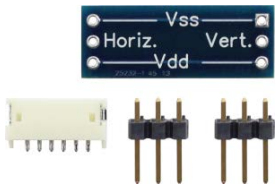
The Joystick comes pre-wired with a 6-wire cable and 1.5 mm plug. An adapter kit with a matching 1.5 mm socket is included, for convenient prototyping on breadboards or thru-hole boards. Simple soldering is required to use the optional but highly recommended adapter.

Features

- 5 kΩ linear-taper potentiometer on each axis
- Smooth ball-bearing action on both axes
- Selectable spring return-to-center on vertical axis
- Selectable detents and/or friction on vertical axis
- Adapter kit for 0.1" spacing is included for easy prototyping (simple soldering required)

Specifications

- Power Requirements: Passive device; will work with any low-voltage supply
- Interface: Resistive or voltage divider output
- Dimensions: 2.70 x 2.32 x 2.42 in (68.6 x 58.9 x 61.5 mm)



Application Ideas

- Game controller
- Remote control for robotics
- Manual control for interactive displays

Packing List

- Quad-Bearing Gimbal Joystick (#27806)
- Adapter Kit (#27809) (including PCB, two 3-pin headers, and 1.5 mm 6-connector socket)

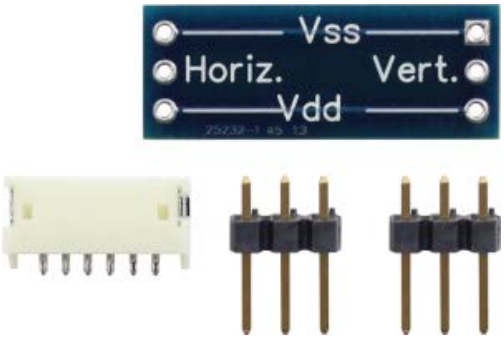
Resources and Downloads

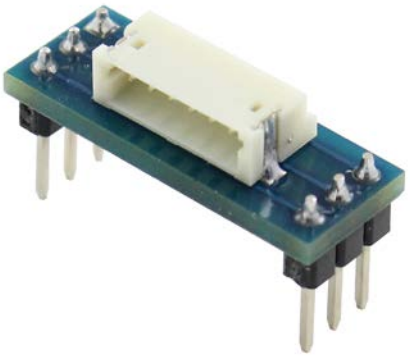
Find example code, the latest version of this document, and additional resources from the Gimbal Joystick product page. Go to www.parallax.com and search "27808".

Adapter Assembly

Using the adapter kit is optional but highly recommended. The example wiring photos in this document assume you are using it. The underside of the adapter board PCB is labeled as shown at left. The top of the adapter board has pads to solder the 1.5 mm socket in place, as shown at right.

(1) Put on your safety glasses!
(2) Begin by inserting the short ends of the 3-pin headers into the PCB from the underside.
(3) Solder the 3-pin headers in place from the top of the board.
(4) Solder the 1.5 mm socket in place on top of the board.

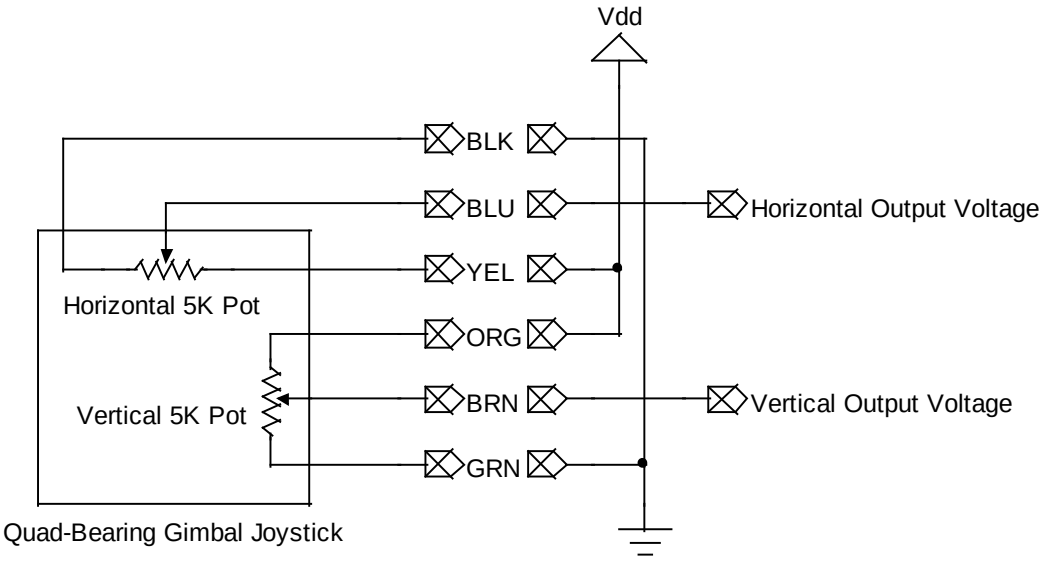




Quick-Start Guide

Voltage Divider Circuit

The voltage divider is most basic circuit using the joystick:



The horizontal output voltage will increase as the joystick moves from left to right, from about 0.38 * Vdd (full left) to 0.63 * Vdd (full right). The center voltage is approximately Vdd / 2. The vertical output voltage covers the same range, increasing from bottom to top, with the center voltage also being Vdd / 2.

The horizontal and vertical output voltages can feed to an analog-to-digital converter (ADC) directly for use with either a P8X32A Propeller chip or a BASIC Stamp. Parallax modules that supply two or more channels of on-board ADC include the following:

- Propeller Activity Board (#32910)
- Propeller Board of Education (#32900)
- BASIC Stamp 2pe Motherboard (#28300)

It is also possible to read the voltage outputs via the Propeller's sigma-delta ADC. (See Parallax Appnote #AN008.) The following products provide two sigma-delta ADC channels each:

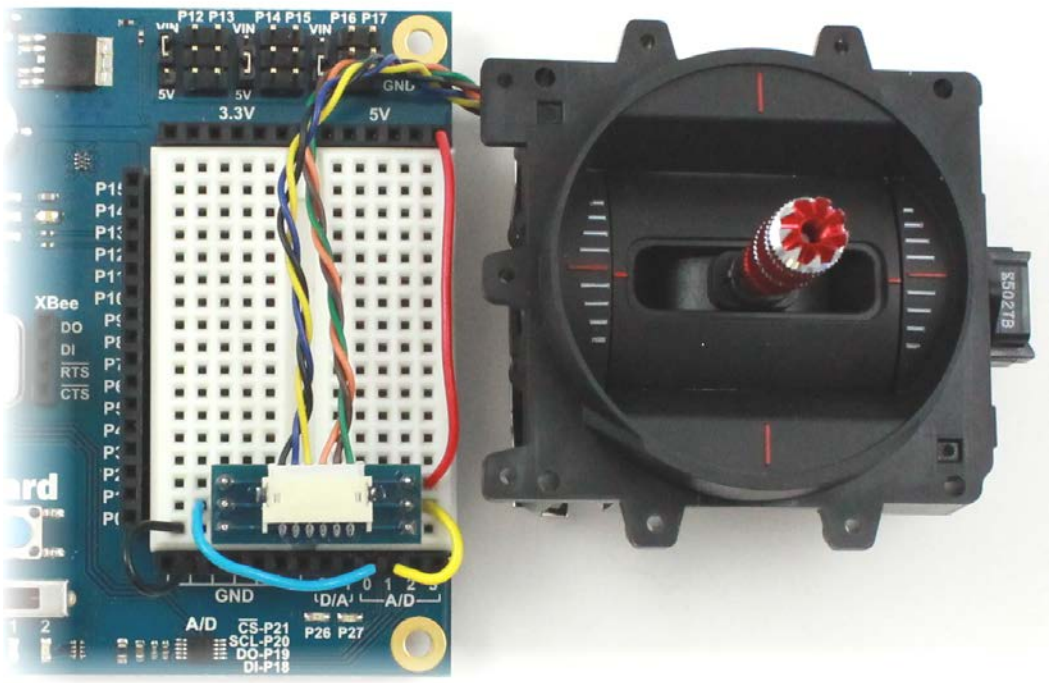
- Propeller Backpack (#28327)
- S2 Robot "hacker port" (#28136, #28336)

In addition, one can deploy an external ADC, such as the MCP3202 (#604-00060) in conjunction with a BASIC Stamp or any other microcontroller.

Propeller Microcontroller Example

The wiring diagram below is compatible with the Propeller Activity Board and the Propeller Board of Education. Both of those boards have an ADC that uses +5V for full-scale reading, rather than Vdd. However, the two boards use different ADC models, so example code may not be compatible for both boards. Propeller C code is provided for the Activity Board, and Spin code is provided for the Propeller Board of Education.

Connect the joystick to the breadboard as shown below.



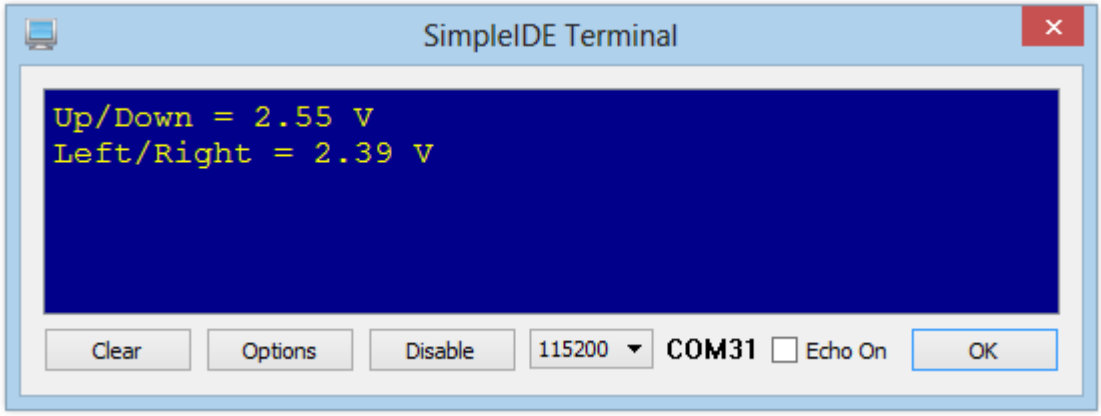
Propeller C Example Program for Activity Board

The example code from the Propeller C Tutorials series, written for a smaller joystick, works fine with the Gimbal Joystick wiring example shown above by simply updating the ADC channels used. This example program and the libraries it uses are included with the SimpleIDE software. Download for Windows or Mac from http://learn.parallax.com/propeller-c.

In the SimpleIDE software, browse to Learn>Examples>Devices>Joystick, and open the project. Update the first two lines in the while(1) loop to use ADC channels 1 and 0 as shown in bold below, instead of 2 and 3.

```
/*
  Joystick.c
  http://learn.parallax.com/propeller-c-simple-devices/joystick
*/

#include "adcDCpropab.h"                              // Include adcDCpropab
#include "simpletools.h"                              // Include simpletools


int main()                                            // Main function
{
  pause(1000);                                        // Wait 1 s for Terminal app
  adc_init(21, 20, 19, 18);                           // CS=21, SCL=20, DO=19, DI=18

  float lrV, udV;                                     // Voltage variables

  while(1)                                            // Loop repeats indefinitely
  {
    udV = adc_volts(0);                               // Update to Check A/D 1
    lrV = adc_volts(1);                               // Update to Check A/D 0

    putChar(HOME);                                    // Cursor -> top-left "home"
    print("Up/Down = %.2f V %c\n", udV, CLREOL);      // Display voltage
    print("Left/Right = %.2f V %c\n", lrV, CLREOL);   // Display voltage

    pause(100);                                       // Wait 1/10 s
  }
}
```

Click the Run with Terminal button. The SimpleIDE Terminal will show the voltage values returned for each axis. As you move the joystick around, you will see the values change. These values may be used as-is, or scaled to appropriate input values for the device you would like to control with the joystick.

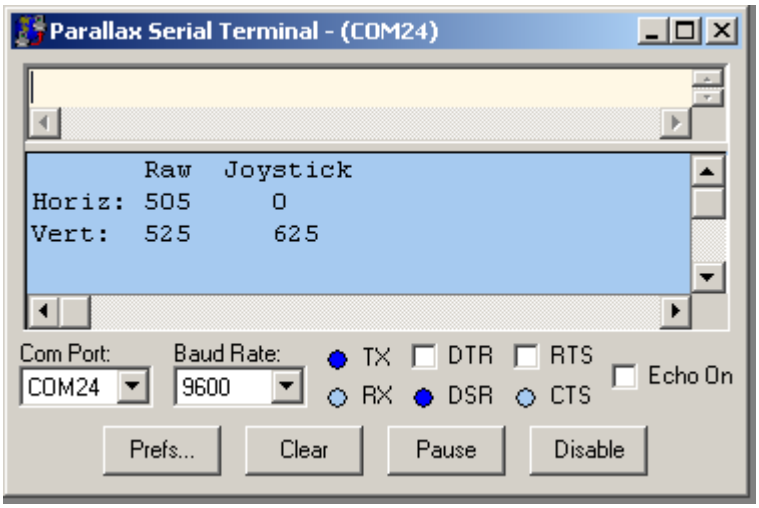


Spin Example Program for Propeller Board of Education

Open the program **PropBOE_joystick_demo.spin** in the Propeller Tool (a free download from www.parallax.com/propellertool):

```
CON

  _clkmode      = xtal1 + pll16x
  _xinfreq      = 5_000_000

OBJ

  pst   : "Parallax Serial Terminal"
  adc   : "PropBOE ADC"
  joy   : "joystick"

PUB  start | rawh, rawv

  pst.start(9600)
  joy.setup(0, 400, 500, 600, 50, -500, 500)
  joy.setup(1, 400, 500, 600, 0, 0, 1000)
  repeat
    rawh := adc.in(0)
    rawv := adc.in(1)
    pst.str(string(1, "       Raw  Joystick", 11, 13, "Horiz:", 14, 7))
    pst.dec(rawh)
    pst.str(string("     ", 14, 15))
    pst.dec(joy.value(0, rawh))
    pst.str(string(11, 13, "Vert:", 14, 7))
    pst.dec(rawv)
    pst.str(string("     ", 14, 15))
    pst.dec(joy.value(1, rawv))
    pst.char(11)
```

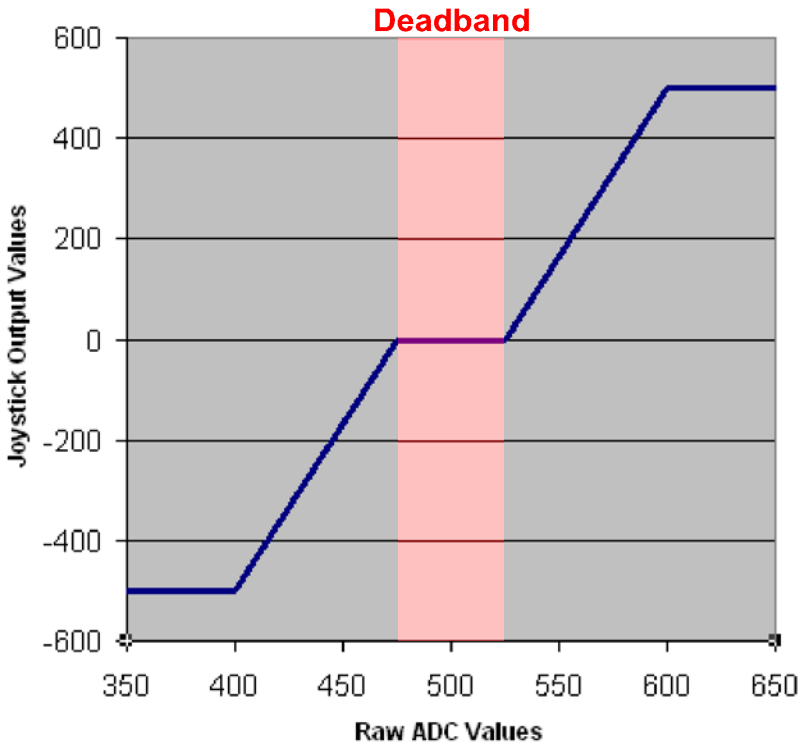
Load the program using the **F10** key, and open the Parallax Serial Terminal via the **F12** key. You should see a display that looks like this:



As you move the joystick around, the values in the display will change to show its position. The **Raw** values are the numbers read from the ADC. The joystick.spin object scales the raw ADC values to produce the **Joystick** values. In this example, the horizontal values range from –500 to +500 and include a deadband in the center, so that the spring-return value is guaranteed to be zero. The vertical axis ranges from 0 to 1000 with no deadband.

Propeller Spin Object

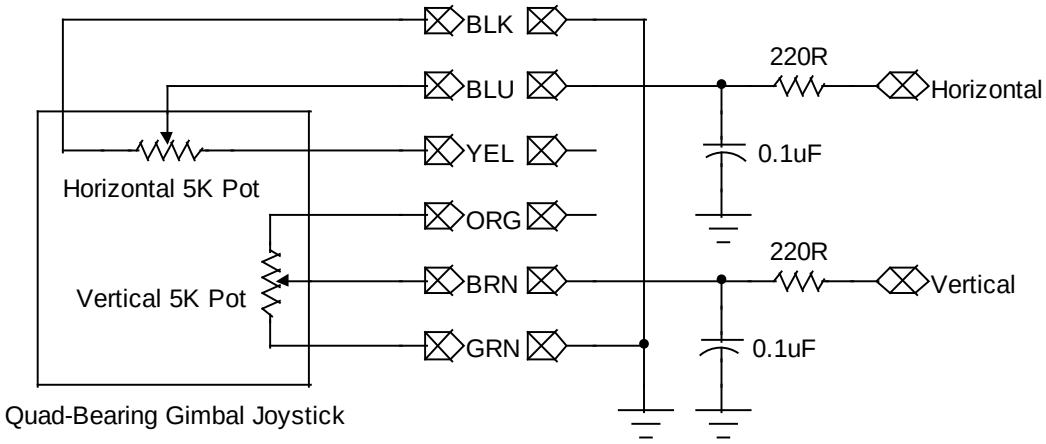
The Spin object, joystick.spin, handles the scaling of the raw joystick data, transforming the measurements into values that are more useful to a given application. The following graph illustrates this process:



In this example of a center-return joystick, the input values can range from about 380 to 630. These are rescaled to a –500 to +500 range, with a 50-unit-wide deadband in the center. See the object's comments for further details.

RCTIME Circuit

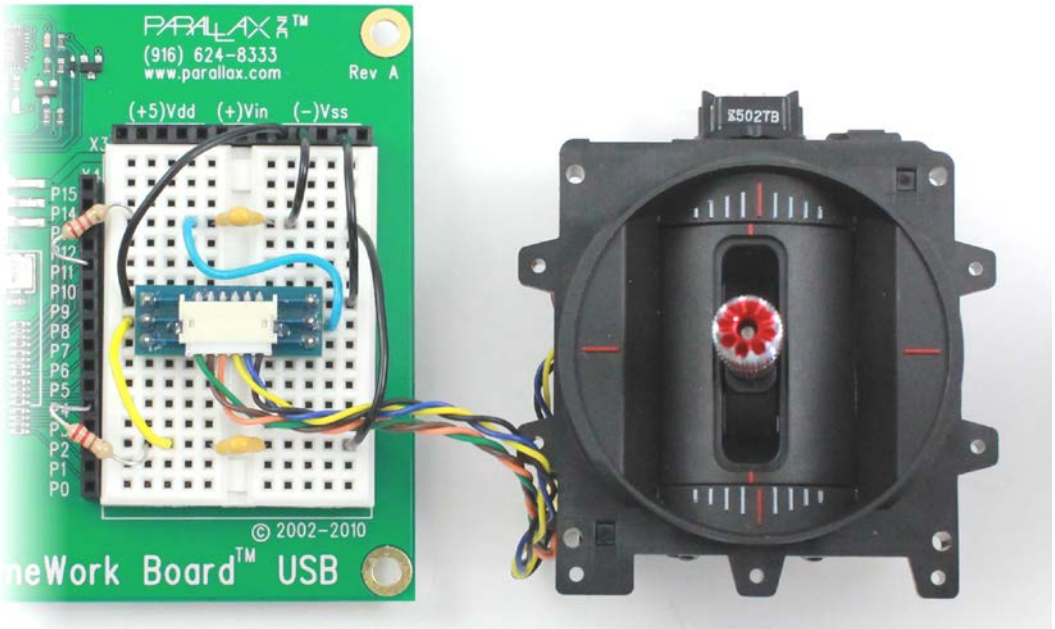
Lacking an analog-to-digital converter, it is still possible to measure the joystick position by timing the discharge of a capacitor. Here's the circuit, which is applicable to the BASIC Stamp family of products:



To measure each wiper position, the attached microcontroller will charge that wiper's capacitor by outputting a high on the associated pin. Then it will tri-state the pin and let the cap discharge through the potentiometer to ground and time how long it takes for the voltage to go below the micro's input logic threshold. The higher the resistance being measured, the longer the discharge time will be.

BASIC Stamp Example

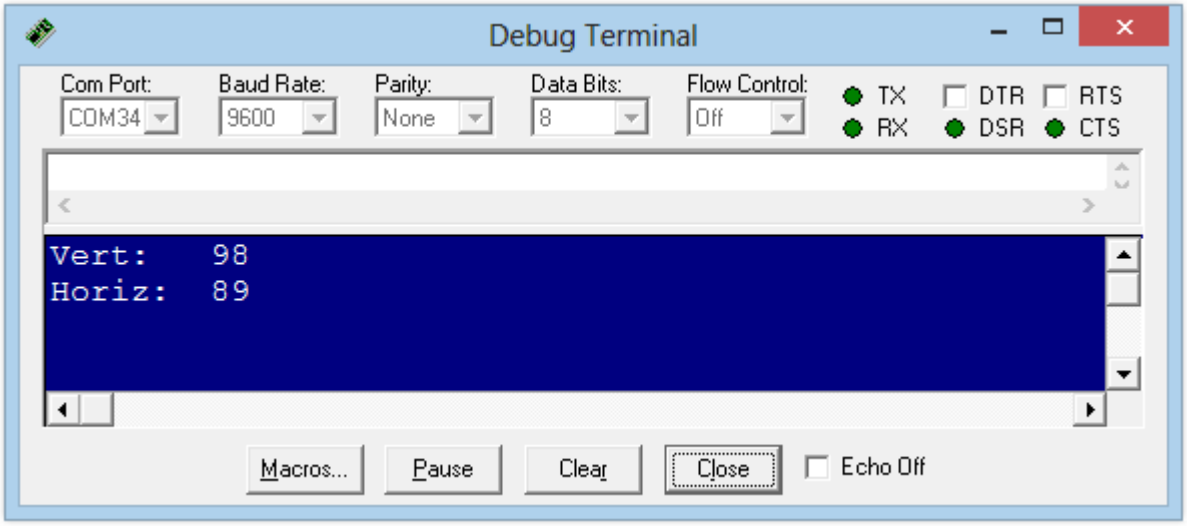
The wiring diagram below is compatible with the BASIC Stamp Board of Education or HomeWork Board. Connect the joystick, the two resistors, and the two capacitors to the breadboard as shown below.



Next, enter the following demo program into the BASIC Stamp Editor software (a free download from http://www.parallax.com/basicstampsoftware):

```
' {$STAMP BS2}
' {$PBASIC 2.5}

LR VAR Word
UD VAR Word

DO

  HIGH 4
  PAUSE 10
  RCTIME 4, 1, UD

  HIGH 11
  PAUSE 10
  RCTIME 11, 1, LR

  DEBUG HOME, "Vert:   ", DEC UD, CLREOL, CR,
              "Horiz:  ", DEC LR, CLREOL
  PAUSE 50

LOOP
```

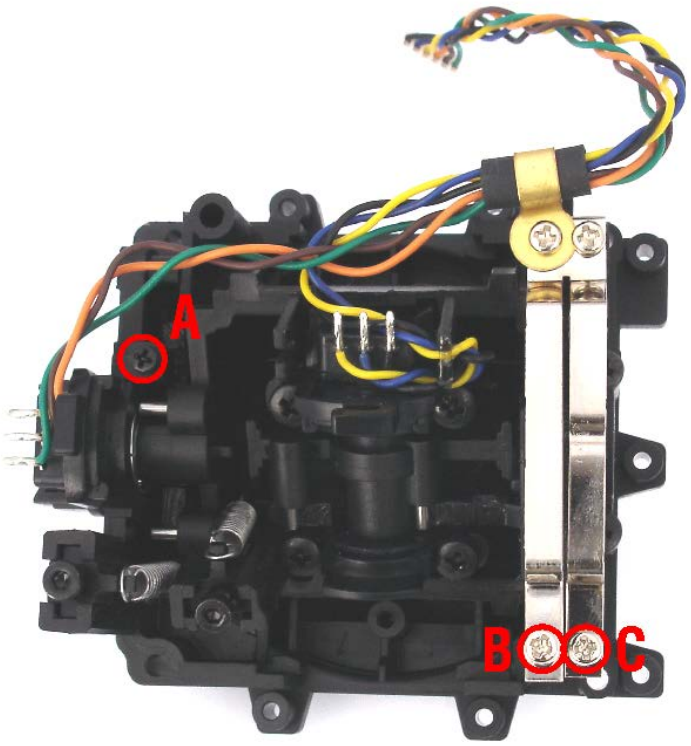
Press the **F9** key to load and run the program. You should see output that looks like this:



As you move the joystick around, the numbers will change with the positions of the horizontal and vertical axes. These values may be used as-is, or scaled to appropriate input values for the device you would like to control with the joystick.

Mechanical Adjustment

There are three mechanical adjustments possible which affect the operation and feel of the joystick. Loosening or tightening the screws on the underside of the joystick labeled **A**, **B**, and **C** in the photo below, will effect these adjustments:



These are the possible adjustments:

- **Screw A:** Loosen for a spring return-to-center on the vertical axis. Tighten to disable the return-to-center action.
- **Screw B:** Loosen to remove friction from the vertical axis. Tighten to add friction.
- **Screw C:** Loosen to remove detents from the vertical axis. Tighten to add detents.

Resources and Downloads

Check for the latest version of this document and example programs, including the **joystick.spin** object, from the **Quad-Bearing Gimbal Joystick** product page. Go to www.parallax.com and search "**27808"**.

Revision History

Version 1.0 – original release.