Peter Norberg Consulting, Inc.
Professional Solutions to Professional Problems
P.O. Box 10987
Ferguson, MO 63135-0987
(314) 521-8808

Application Note

Parallax Quick Start Guide

for use of the BS0710 Series of Stepper Motor Controllers with Parallax Basic Stamp™ Products

Version 1.01
By
Peter Norberg Consulting, Inc.

Table of Contents

Overview

Thank you for your interest in Peter Norberg Consulting, Inc.'s line of stepper motor control products! This application note has been written to let you get your new stepper motor control board up and running as quickly as possible with your Parallax Basic or Javalin Stamp. A much more comprehensive manual, called "First Use", is also available which is designed to give far more details of the process, and to explain how to test any of our boards using your computer and a compatible serial connection.

This document provides a full example of connecting one of our BS0710 boards to a Parallax "Board of Education" with their "BS2" basic stamp installed. The technique for connecting any of our boards to any of the Parallax "stamp" products is equivalent to this example.

This document assumes familiarity with basic electronics, access to standard electronics tools (such as digital volt/ohm meters, jeweler's screw drivers, clip leads, etc.), familiarity with those tools, and access to a computer that can run a terminal emulator. (Most versions of Microsoft Windows™ have this ability.)

The revision history of this manual is as follows:

Version	Date	Description
1.00	June 14, 2006	First internal release
1.01	June 21, 2006	First public release

If you have any problems that our documentation cannot resolve for you, please contact us for customer support, via email or telephone.

Our customer support email address is:

(support@stepperboard.com)

Our phone numbers are:

U.S. Toll Free: (877)-230-5270

International: (314)-521-8808

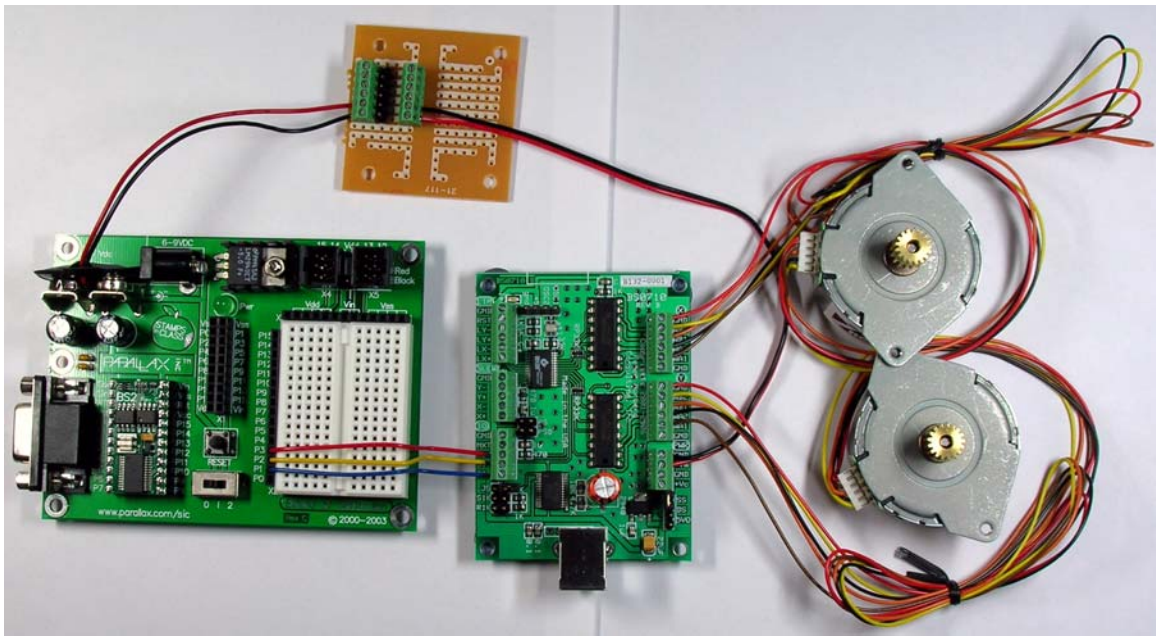
Summary

This application note provides a quick summary of operating one of our BS0710 dual axis stepper motor controllers using a Parallax Basic Stamp BS2 mounted on a Parallax Board Of Education (BOE), driving two small Parallax educational stepper motors. For more detailed theory and instructions, please refer to our "First Use" manual (starter's guide to first use of any of our stepper motor controller products), and to our "BS0710 Hardware and Firmware Reference Manual" (complete description of our controller).

A short summary of the procedure is as follows:

1. Verify that you have the correct parts from our parts list (below)
2. Configure the BS0710 jumpers to match the requirements of this connection technique
3. Connect the BS0710 to the two motors
4. Connect the BS0710 to the power supply (supply off!)
5. Do a stand-alone test to verify motor operation with the BS0710
6. Configure the BOE (insert the stamp, set the power switch, make sure jumpers are correct)
7. Wire the power supply to the BOE (power supply off!)
8. Program the BS2 per Parallax's documentation, using the sample program provided, then turn the supply back off.
9. Connect the BOE to the BS0710
10. Power everything back on. Both motors should start spinning per the sample program.

A picture of the complete assembly follows.



1. Parts List

This example makes use of the following components:

1. 1 A-BS0710USB-TTT-GS dual axis stepper motor controller
2. 2 Parallax 12V 100 ohm unipolar stepper motors (for educational use only)
3. Parallax Board Of Education (BOE)
4. Parallax Basic Stamp, the BS2
5. A single 12 volt 1 amp (or more) regulated DC power supply. This may be either switching or linear; it should also be grounded.
6. A 9 volt battery adapter
7. In our case, we also added a small breadboard from Radio Shack to simplify our wiring of the power from the power supply to both the BOE and the BS0710.

Items 6 (the '9 volt battery adapter) and 7 (small breadboard from Radio Shack) simplify the connection of the power supply to the BOE and the BS0710.

For tools, you will need (at least)

1. A wire cutter/stripper
2. A 2.0 mm jeweler's flat-blade screwdriver (to connect wires to the BS0710 screw terminals)

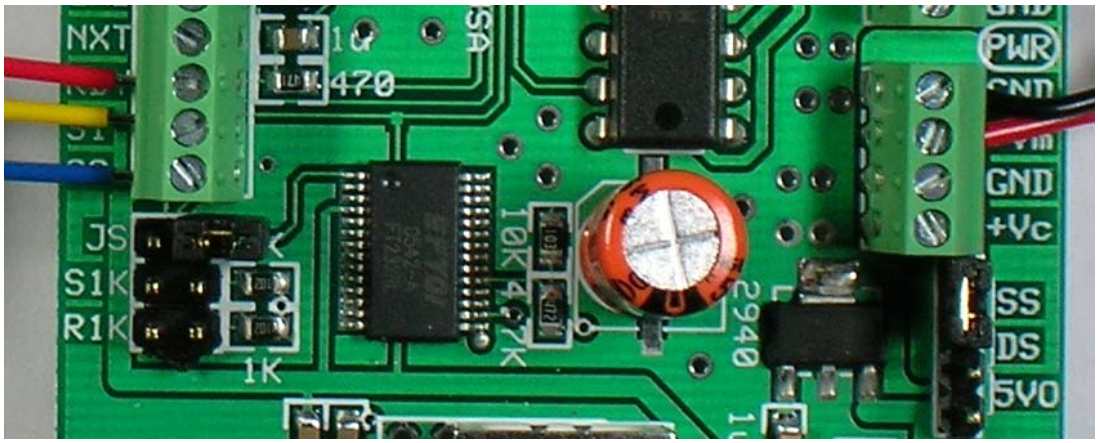
2. Configure the BS0710 Jumpers

The BS0710 has several jumpers that are used to control how the board is powered, how it communicates with the outside world, and how it operates the motors.

For the purposes of this application, the jumpers should be:

- R1K – Not installed
- S1K – Not installed
- JS – Not installed
- PWR – Set to the 'SS' position

A picture of the portion of the board shown with the jumpers correctly positioned follows:



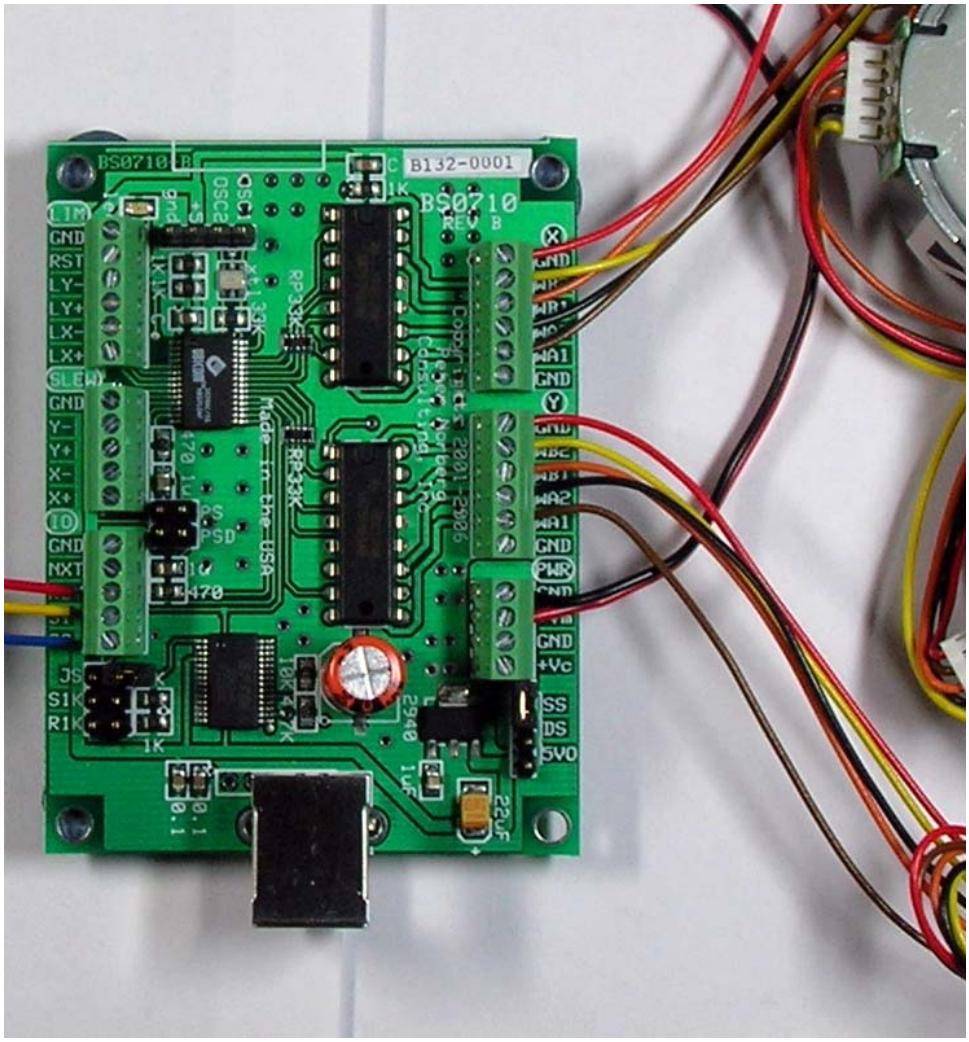
(note that the JS jumper is positioned so that only one pin is connected to the jumper; this allows you to not lose the jumper….)

3. Connect the BS0710 to the two motors

Connect the two motors to the X and Y connectors of the BS0710. You will need to cut the connector off of the end of the wire from the motors, and strip the wires back about 1/8th inch (so that they will fit into the screw-terminal connectors on the BS0710, without leaving any bare wire visible after being inserted). Be very careful to exactly match the wiring shown, and to not short any of the wires together**. If you short wires, you can damage the BS0710 controller!** If you do not match the wiring order, then there is a good chance that the motors will not work correctly.

Each motor gets attached to one connector set, either the 'X' or the 'Y' connector as shown below. The wiring is based on the color of the wires (if you cannot see the colors, please get someone to help you who can see them):

Color	*X or Y connector pin*
Red	GND
Yellow	WB2
Orange	WB1
Black	WA2
Brown	WA1



4. Connect the BS0710 to the power supply (supply off)

Using the PWR connector on the BS0710, connect the 12 volt supply output "-" to one of the GND connections, and the supply '+' to the +Vm input.

In this picture, the black wire is connected to the supply '-', while the red wire is connected to the supply '+'. Please note that the power jumper must be in the "SS" position for this to work.

5. Do a stand-alone test to verify motor operation with the BS0710

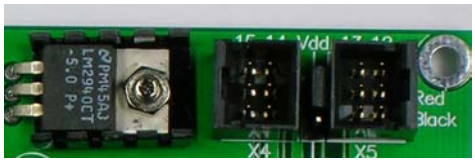
For this test you will want to temporarily connect a 2 to 5 inch wire to the GND pin on the SLEW connector of the BS0710, and leave the other side of the wire bare (but not touching anything**). In each step below, if the expected observation does not occur, immediately power off the supply, and trace your wiring problem.**

1. Power on the power supply. You should see the LED on the BS0710 illuminate.
2. Temporarily touch the bare end of your shorting wire to the '-X' pin of the SLEW connector. The X motor should spin in one direction while the wire is touching that pin.
3. Temporarily touch the bare end of your shorting wire to the '+X' pin of the SLEW connector. The X motor should spin in the reverse direction of that from step 2 while the wire is touching that pin.
4. Temporarily touch the bare end of your shorting wire to the '-Y' pin of the SLEW connector. The Y motor should spin in one direction while the wire is touching that pin.
5. Temporarily touch the bare end of your shorting wire to the '+Y' pin of the SLEW connector. The Y motor should spin in the reverse direction of that from step 2 while the wire is touching that pin.
6. Turn off the power supply

6. Configure the BOE

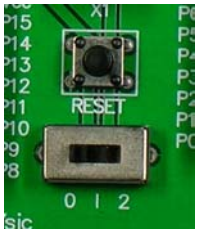
The current version of the Parallax Board Of Education has one slide switch and one jumper that need to be correctly configured. Additionally, the BS2 needs to be correctly inserted into the BOE.

The jumper is positioned in between the two black rectangular connectors near the top right corner of the board, and needs to be in the top position (Vdd).

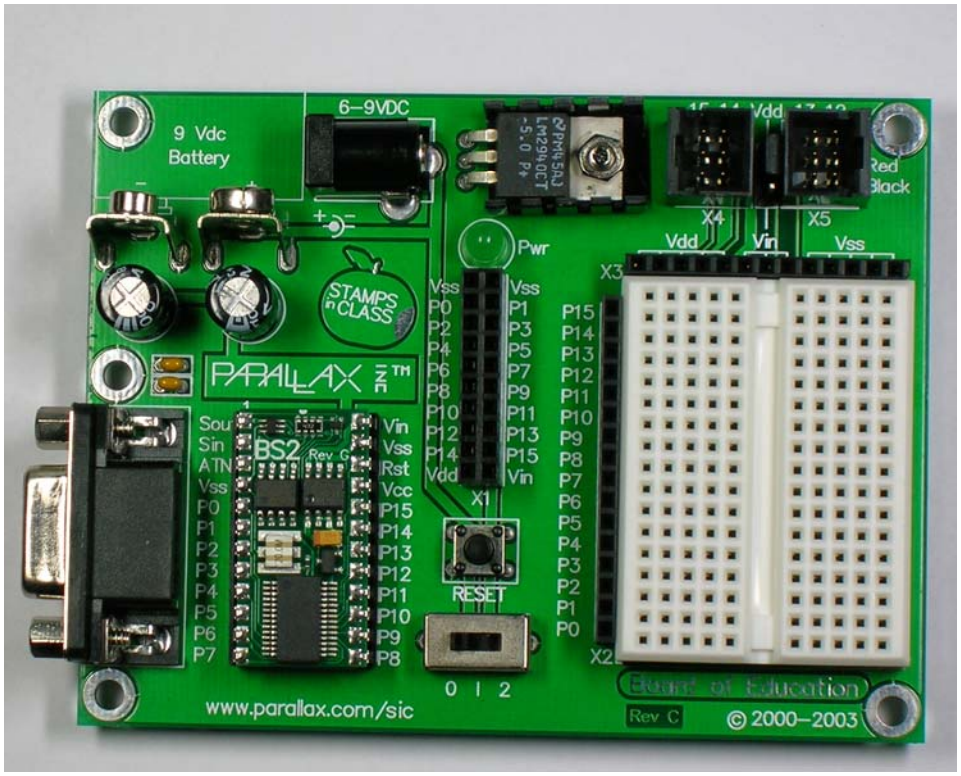


The slide switch is used to power the board. We only use two positions – 0 (which is 'Off') and 1 (which is 'On', as far as our use is concerned).

Leave the switch at '0' in between tests: turn it to '1' to actually program and run the stamp.



The Parallax BS2 stamp must be installed into the BOE as shown in order to operate.

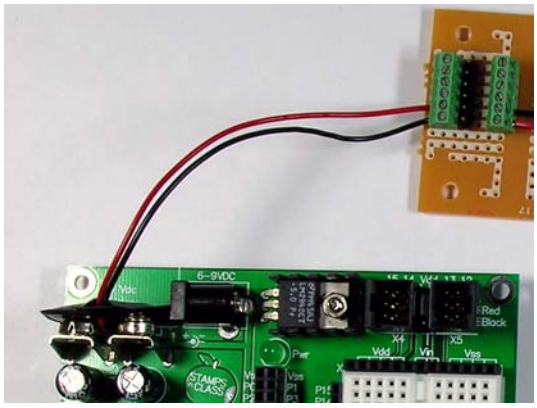


7. Wire the power supply to the BOE (power supply off)

Despite the board labels showing use of a 6-9 volt supply, the BOE will accept use of a 12 volt supply without harm for this particular application. In order to simplify our wiring to the board, we elected to use a standard 9 volt battery adapter to easily connect to the board. When this is done, the colors of the leads coming from the adapter is reversed from the normal assignments: the black lead needs to be attached to the '+' of the 12 volt power supply, while the red lead is to be connected to the '-' side of the 12 volt power supply.

You should use a DVM to confirm that the red lead from the adapter is connected to the BOE ground; the easiest test point is the shell of the DB9 connector on the BOE – you should see a resistance of less than 2 ohms when you do the measurement.

We also routed this power to a small perf board from Radio Shack, to simplify connecting the BOE and the BS0710 to the same power supply.



8. Program the BS2 per Parallax's documentation

Program the stamp using an appropriate sample program. For our testing, we used our trivial application "GenDemo.bs2", as found in the "C:\UniversalStepper" subdirectory (on a normal install from our CD).

You will probably need to edit two source lines in the file which define which stamp and what baud rate is to be used.

1. Look for the line which has the text "{$STAMP BS2SX}" on it, and change the stamp type to match your stamp (in this example, we changed that line to read "{$STAMP BS2}").
2. A few lines later, you will find the definition for a constant called "PortStepperBaud", as well as comments describing the values which should be used with that constant. On the BS2, this constant should be set to 84. On the BS2SX, this needs to be set to 240 (i.e., set it to allow for 9600 baud communication)

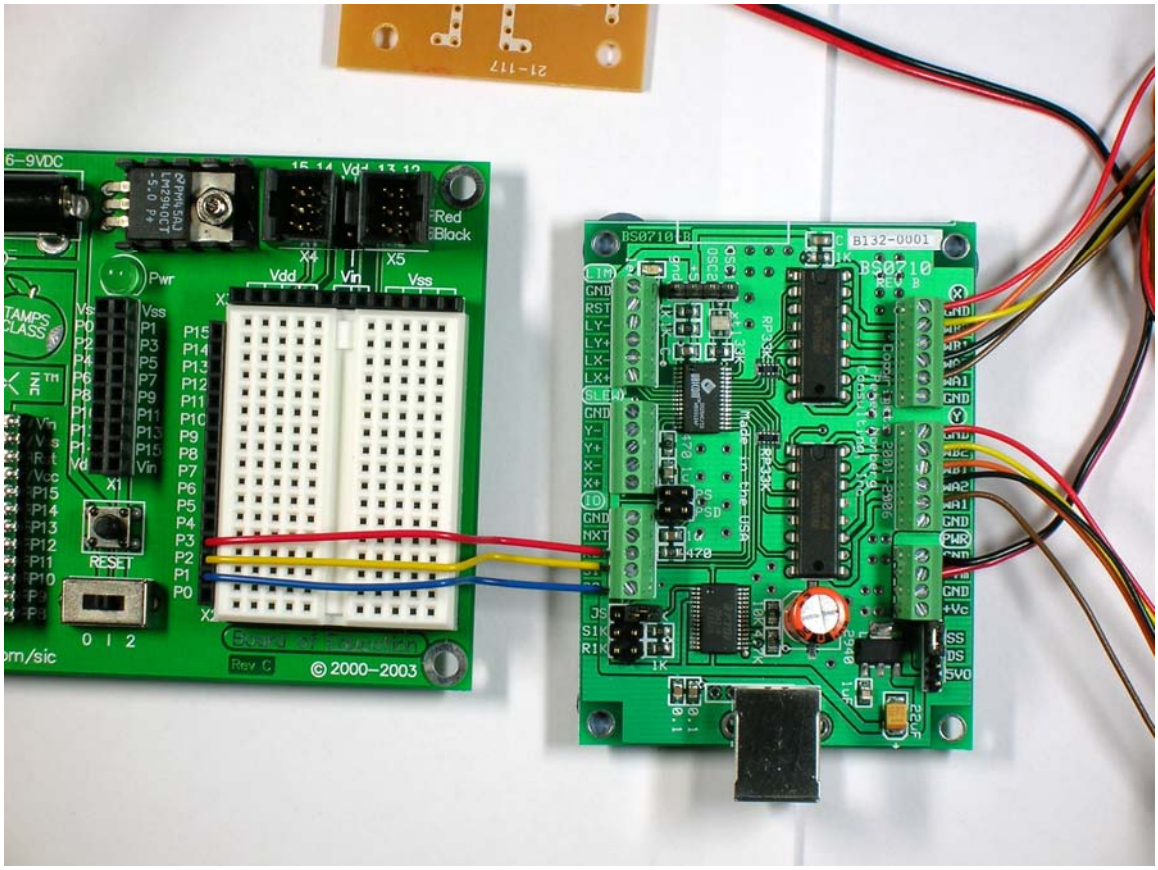
A listing of the sample application appears later in this document.

9. Connect the BOE to the BS0710

In addition to both products sharing a common ground (since they are powered off of the same supply), three wires need to be run between the BOE and the BS0710.

Stamp Signal	Stepper Board Signal	Description
P1	SO	Serial Out from stepper board to stamp
P2	SI	Serial In to stepper board from stamp
P3	RDY	TTL signal from stepper board, used to state if stepper motion has completed

You will observe that these signals "run straight across" when you position the boards as shown below. In this example, the P1 wire is blue, P2 is yellow, and P3 is red.



10. Power everything back on. Both motors should start spinning per the sample program.

You will also have to move the switch on the BOE to the "1" position, in order to power the stamp.

Assuming that you are using the "GenDemo.BS2" sample, the two motors will endless spin back and forth, using different stepping modes (which will therefore 'sound' different).

Sample program listing – GenDemo.bs2

```
'
'{$STAMP BS2}
'{$PBASIC 2.5}
'
'*****************************************************************************
' $modname$
' $nokeywords$
' Demonstates some of the serial commands using goto and TTL Busy line to the SimStep and BiStep
' set of controllers from Peter Norberg Consulting, Inc.
'
' The tool first initializes the stepper to operate at 16 microsteps/full step,
' with the start/stop rate being 80 uSteps/second, and the ramp rate at 1000 uSteps/sec/sec.
' The target ramp rate is 1000 uSteps/second;
' The auto-power switch mode (the 'A' command) is left at its default of 3072, which is equivalent to 192 full
' steps/second.
'
' Note that both motors are selected for the actions by default.
' It then enters the speed test loop.
'
' The code first waits for the stepper unit to report idle.
' and it is instructed to move to logical location 2000 (in) 1/16th steps.
' (Note that this is full step location 62.5).
' This is then followed by a move to location 0, and then a new stepping mode
' is selected.  A 1/5th second pause is inserted to make it easy to identify
' when the cycle is occuring.  All three modes of stepping are cycled:
'   Mode    Use
'    0      Single Winding mode (1/2 power full steps)
'    1      Half step mode (alternate single/double windings on)
'    2      Full step mode (double windings on)
'    3      Microstep mode (full microstep processing; DEFAULT MODE)
'
'
' SPECIAL TIMING NOTE: It can take the SimStep/BiStep up to 100 uSeconds to respond to
' a new serial "go" command (goto or slew); therefore, you must always wait
' a small amount of time (at least a few milliseconds uSecs) before testing the "busy" line, since
' you may get a "false idle" response.
'
' Additional note: The SimStep/BiStep products operate at 9600 baud.  Although
' the Basic Stamp series can send this rate reliably,  many of them cannot receive
' at this rate without data loss; therefore, no attempt is made in this
' sample to receive serial data from the controller.
```

```
'
'
*****************************************************************************
'
' Be Sure to select the correct stamp and baud rate
' You might have the BS2, BS2SX, etc.
' {$STAMP BS2SX}

' Select the baud rate control:
'   Stamp         9600   2400
'   ----------  ----    ----
'   BS2/BS2e       84    396
'   BS2sx/BS2p    240   1021

PortStepperBaud     con 240      ' Baud rate to generate 9600 baud on BS2sx

'
*****************************************************************************

' SimStep or BiStep controller connected as follows.  Please remember to
remove the "JS" jumper (or the MAX232
'   or equivalent serial buffer chip) from the controller, so that the SI
line will not be overloaded.

'   Serial Input P1 to controller SO/B7 Serial output
'   Serial Output p2 to controller SI/B6 Serial Input
'   busy          p3 to controller RDY/B5 Status Output (HIGH = idle, LOW =
motion in progress)

PortStepperSerFrom  con  1      ' Serial from stepper port
PortStepperSerTo    con  2      ' Serial to stepper port
PortStepperBusy     con  3      ' Busy line

PortStepperBusyTest var in3    ' Same as PortStepperBusy, used for input
test

idMicroStep var byte            ' Gets microstep mode; cycles 0 to 3



' Code restarts here if RESET button pressed

        input           PortStepperBusy                          ' BUSY
from stepper

        pause 250                                               ' Wait for
stepper power on cycle
        serout PortStepperSerTo,PortStepperBaud,["4!"]      ' Reset the
stepper, set 4/64 full-step step size
```

```
        pause 1000                                              ' Wait for
stepper to send its wake-up copyright text
        serout PortStepperSerTo,PortStepperBaud,["80K"]     ' Set Stop OK to
'can start/stop at 80 microsteps/sec'
        serout PortStepperSerTo,PortStepperBaud,["1000p"]   ' For demo
purposes, set a slow ramp of 1000 microsteps/sec
        serout PortStepperSerTo,PortStepperBaud,["1000R"]   ' For demo
purposes, set a target rate of 1000 microsteps/sec
        idMicroStep = 0                                         ' Start at
microstep 0



loop:
        serout PortStepperSerTo,PortStepperBaud,[dec idMicroStep,"o"]      '
Set microstep mode


        serout PortStepperSerTo,PortStepperBaud,["2000g"]   ' Go to location
2000


        gosub WaitReady                                         ' Wait until
ready


        serout PortStepperSerTo,PortStepperBaud,["0g"]      ' Go back to 0


        idMicroStep = (idMicroStep + 1) & 3 ' Cycle step type
        gosub WaitReady                                         ' Wait until
ready
        pause 200                           ' wait 0.2 seconds before we
cycle
        goto        loop                    ' Cycle forever


WaitReady:
        pause 100                           ' Wait 0.1 seconds for prior
character to be processed
        if PortStepperBusyTest = 0 then WaitReady       'Wait till not busy
        return
```

Common Parallax Stamp Issue Summary

Once the board has passed the above set of tests, you can connect it to your Stamp. The Stamp products are usually connected to our board via the "TTL-Serial" technique, described earlier in this manual. In summary:

Make certain both the Stamp and the board's logic are run off of the same power supply.

If this is not possible, make certain that the "GND" line of the stepper controller is connected to a GND (also called Vss by Parallax) on the Parallax board. Using a common power supply for the boards avoids ground referencing problems. ***This is the single most common cause of issues when using the stamp***. If separate supplies must be used, then you must be certain that you connect their grounds together with a reasonable size wire, in order to make certain that both boards "float" at the same potential. Otherwise, TTL signals will not be correctly interpreted, and serial communications will fail.

When you actually try to run your Stamp application, disconnect the "download" programming RS232 cable from between the computer and the stamp.

We have often had issues with ground-distortion induced by leaving the serial cable (used to program the stamp) connected while trying to have the stamp "talk" to a stepper controller board. Merely unplugging the cable has always been enough to "fix" the problem.

Disable RS232 communications to the board.

Remove the JS jumper from our board (Jumper Serial) if that jumper is available (it is present on all of our newer artworks which have RS232 or USB serial built in). If you have the BiStepA04, BiStepA05, BiStep2A04 or SimStepA04, remove (and save) the 16-pin Max232 serial controller chip located right by the 9 pin serial connector. Removal of the JS jumper or the Max232 chip is needed to avoid connecting the outputs of two chips to each other. This generates what is called a "wired or", and can cause eventual failure of the chips involved.

Make certain that you have adjusted our sample code to match your stamp product.

Different stamps require different settings for the "PortStepperBaud" parameter used in our samples. When you edit the sample code for your system, you need to look for the line which contains the "{$STAMP" command, and adjust it to match your stamp (such as BS2, BS2sx, etc.). This may also be done by pressing the button in the Basic Stamp Editor that selects the type of stamp that you are using. You then need to edit the line that defines the "PortStepperBaud" parameter to contain the correct value for your stamp. Please refer to your stamp manual for the values. For some of the stamps, the values are:

Stamp Unit	*9600 Baud*	*2400 Baud*
BS2/BS2e	84	396
BS2sx/BS2p	240	1021

For example, to generate 9600 baud on a BS2sx, you need to have the PortStepperBaud line contain:

PortStepperBaud con 240 ' Baud rate to generate 9600 baud on BS2sx

Use a linear power supply, not a switching power supply.

Use of a linear power supply (as opposed to a switching power supply) is a very strong recommendation. Switchers generally do not have the short-term power reserves needed to handle the extreme variances in power demanded on motor startups, and they are usually much "noisier" from an electrical point of view. We have had no reported power-supply related problems when our customers use linear supplies of adequate size; we have had frequent issues when switchers are used. For example, switchers will often have 10-20% of "ripple" in their voltage (i.e., there will be a periodic variance of several volts in their supplied power, usually at the power line frequency of 50 or 60 hz). This means that the motors are "feeling" a 10-20% variance in their drive power, which means that microstepping will be unreliable in position.

Make sure your power supply is large enough to meet the demands of your application.

We have frequently had customers complain that their motor stalls or does not appear to have enough power. In most cases, this has been traced to the customer having selected a power supply that can only deliver about 1/2 of the current required by the motor. Remember that a stepper motor can have 2 windings on at a given time; this means that it can draw twice as much current as you would expect, referring only to the current rating of the motor. Please refer to our "Universal Stepper" manual section "Calculating Motor Current Requirements" for a more complete summary of how to calculate the current requirements for your application.

Keep your leads between the two units short.

Even though we have we have successfully operated via TTL serial at distances of up to 26 feet, TTL signals are only "clean" (that is to say, little "bounce" or induced voltages) at up to 3 feet. They are quite susceptible to induced voltages from antenna effects, impedance matching issues, and from neighboring signals. In general, when the signals are kept to under 3 feet, the amount of induced signals are too small to cause a logic circuit to trigger (unless they are routed beside a motor driver line). They also require a good ground reference between the boards which are transmitting the signals.

When operating using TTL-Serial mode (as opposed to RS232 serial, which uses much larger voltages) on a 26 foot cable, we have found that there can be about a 0.1 microsecond wide 5 volt spike induced onto one signal line when a neighboring signal line changes state (i.e., when a serial bit is sent to our board, a narrow spike can be induced on the serial output line from our board to your stamp). Due to the nature of the serial timing between the products, this spike is normally ignored under serial I/O (our firmware rejects it via a digital filter, and the timing of serial I/O to and from the stamp is such that the stamp would not see it). Our suggestion of 3 feet or less avoids this potential issue.

We suggest running communications at 9600 baud.

The most recent versions of the firmware (GenStepper versions 1.65 and above) have enough idle time built into it when operating in the suggested modes for Stamp communications that you should have no problems. Operate at 2400 baud only if you cannot get reliable operation at 9600 baud (and please contact us for further assistance in this case).

Re-issue the 'H' command on current regulating boards after reset or power-cycling

If you are using any of the current regulating series of boards (the BC2D15, BC4D15, or the UCC30xx series), remember to re-issue the appropriate "H" command at any time that the board is reset or power-cycled. Otherwise, the board will operate at its default power-on level, which is usually full-power; ***this may damage your motor*** if it is more than 1.5 to 2 times the recommended current for the particular motor! Please see the manual for your specific board and firmware for documentation about the "H" command.

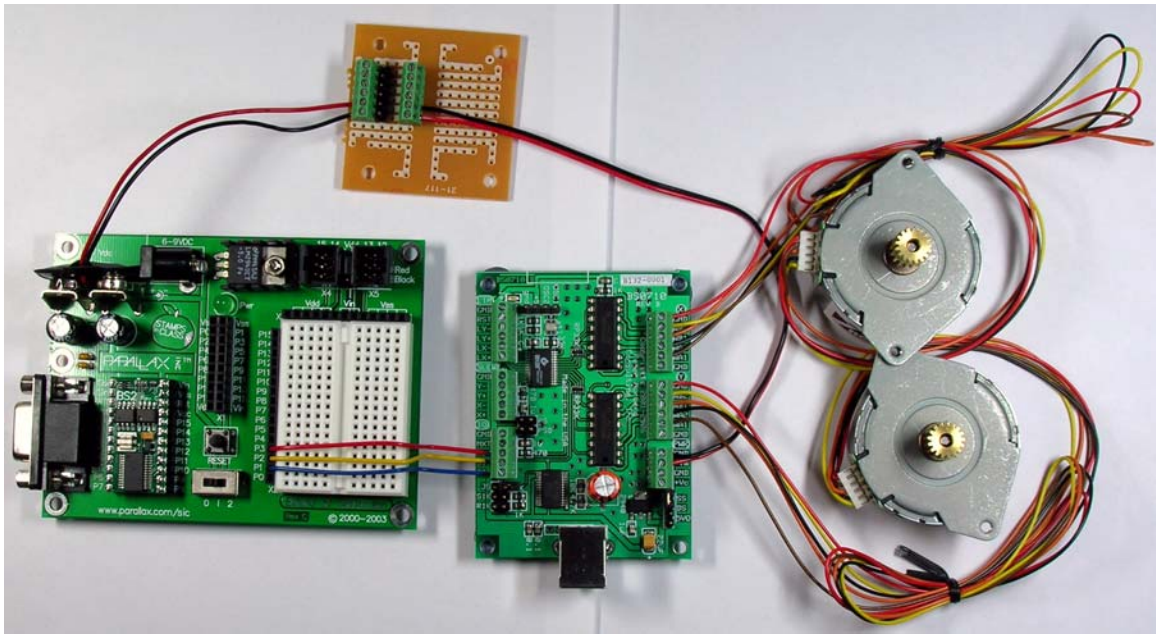
Some Stamp Connection Examples

This section contains several examples of connecting Parallax Stamp-based products to our stepper controller boards.

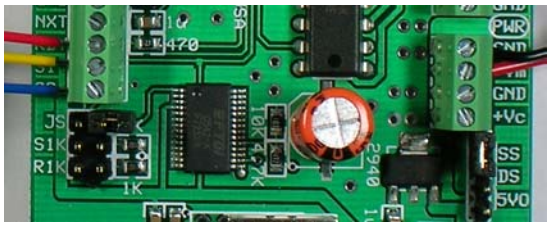
In each case, the connections shown are a combination of those required to use the sample programs that we provide with the stepper controller boards. This means that the following signals are connected:

Stamp Signal	Stepper Board Signal	Description
P1	SO	Serial Out from stepper board to stamp
P2	SI	Serial In to stepper board from stamp
P3	RDY	TTL signal from stepper board, used to state if stepper motion has completed

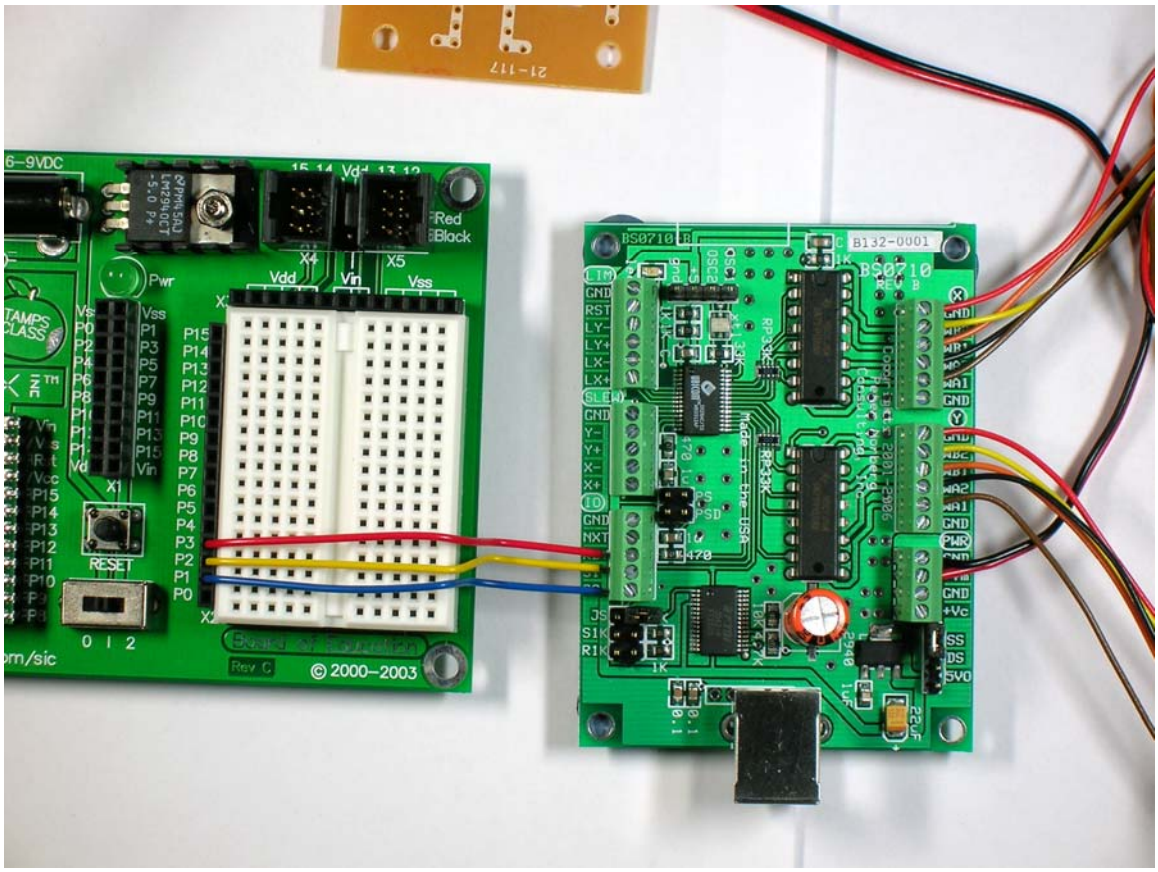
The following photo shows a complete setup of a Parallax "Board Of Education", wired to a BS0710 board.



This 'zoomed in' image of the BS0710 shows the jumpering used in this example (power option SS, with the JS jumper disabled)

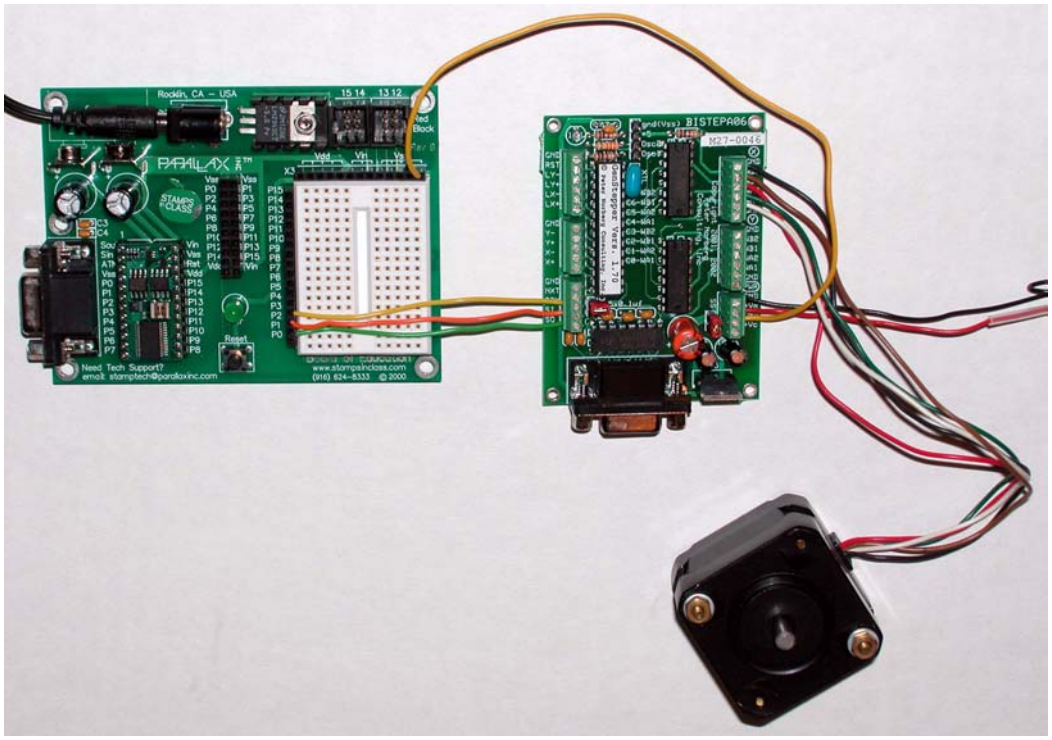


Similarly, this a "zoomed in" image shows the connections between the BOE and the BS0701 in greater detail:

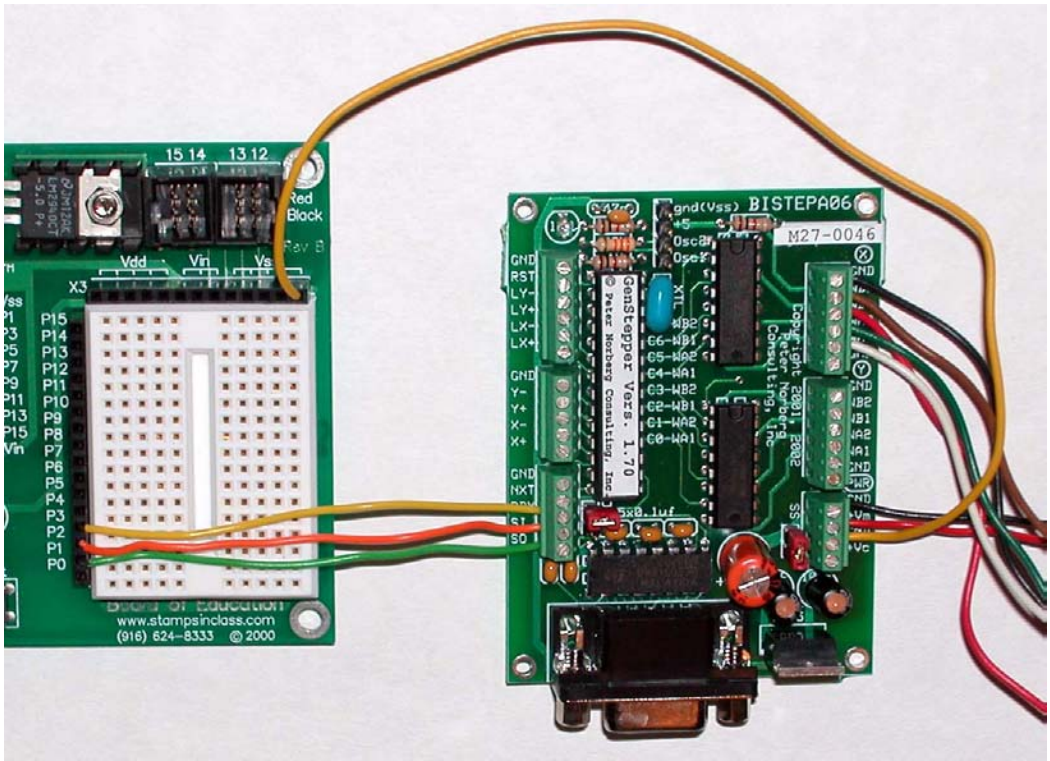


In this case, both boards were powered off of the same 12 volt power source (the BOE can support this without trouble, even though they show 9 volts as the maximum). This allows one power supply to be used to operate both the board logic and the motors themselves.

The following photo shows a complete setup of a Parallax "Board Of Education", wired to a BiStepA06 board.



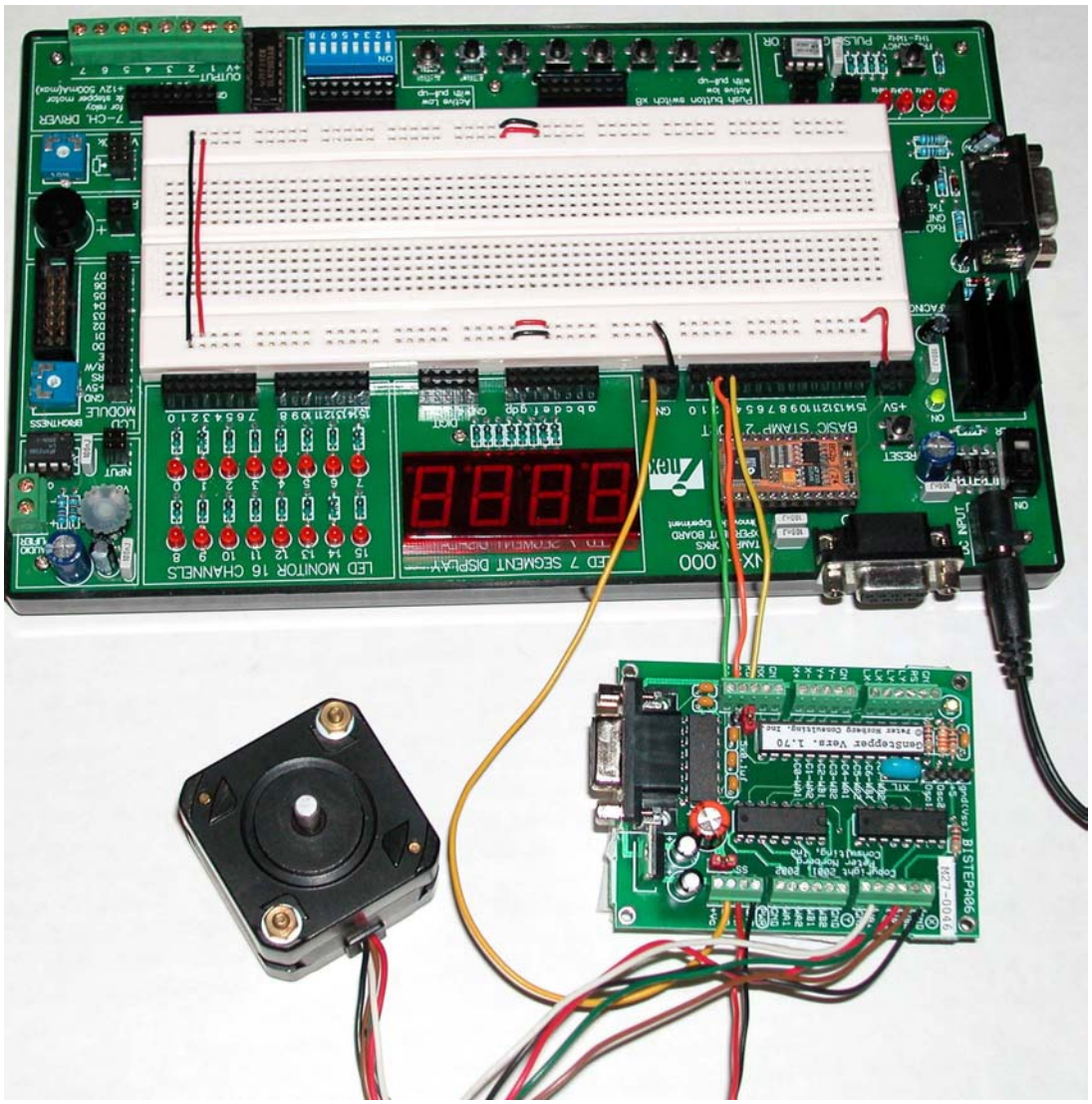
A "zoomed in" image shows the signals a little bit more clearly:

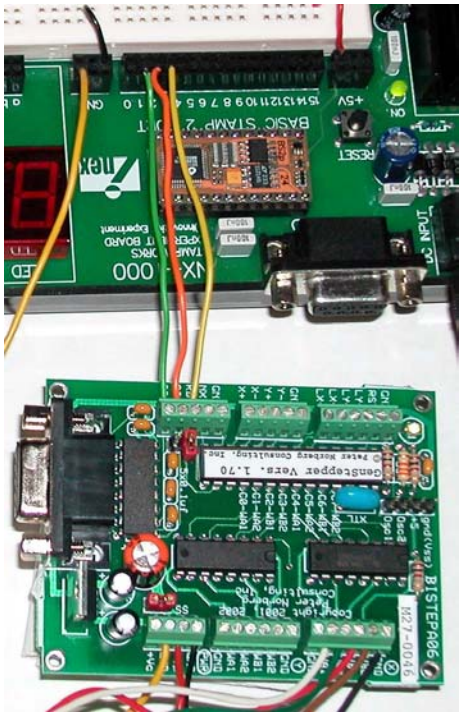


Observe that the ground between the boards is accomplished by a wire from the extra "GND" connector on the BiStepA06 power connector (lower right of board as shown) to one of the "Vss" signals on the BOE.

Also observe that the P1→SO, P2→SI, and P3→RDY signals actually go "straight across": the signals line up in a set of rows.

An identical setup using the NX-1000 board looks as follows:





Similarly, a "zoomed in" image just showing the key signals appears to the left.

In this case, the extra "GND" signal on the BiStepA06 is run to the "GND" connector which is beside the P0-P15 signals on the NX-1000 board. As with the BOE, the P1→SO, P2→SI, and P3→RDY signals are all consecutively aligned, and operate without any signal wire crossings.

In both of the above examples, power was supplied by switching power supplies in the form of "power bricks". While we strongly urge that such items not be used, they can work if they are wired as described in the above lists. A photo of the supplies correctly plugged into a grounded power strip follows. Please observe that they are both plugged in "facing" the same way, so that the neutral wire is on the same side for both plugs.

Trademarks Used in This Document

Microsoft and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.