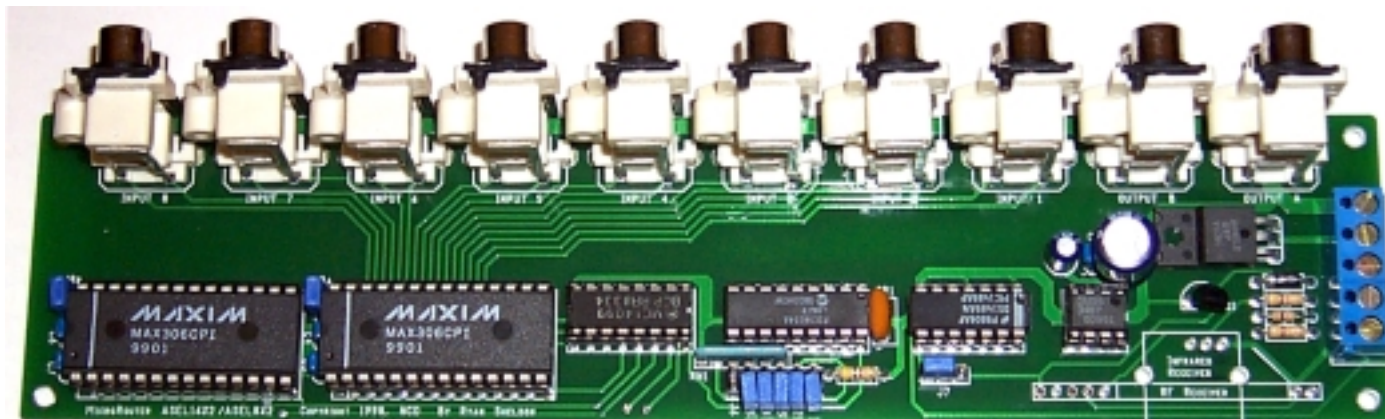


# ASEL/8S2S

## V2.0 Firmware Upgrade



### Device Variations

ASEL 16x2 Video Switcher  
8S2S 8 Stereo Input, 2 Stereo Output Switcher

### Device Features

Control 256 Devices Simultaneously or Individually from a Single Serial Port  
User-Selectable Communication Rates of 2400 or 9600 Baud  
E3C Compliant Command Set  
ASEL: 16 Inputs, 2 Outputs, Route Any Input to Any Output  
8S2S: 8 Stereo Inputs, 2 Stereo Outputs, Route Any Input to Any Output  
Use for Audio, Video, or Analog Signal Sources  
30 MHz Bandwidth  
NTSC/PAL Compatible

### Firmware 2.0 Upgrade Notice

This publication is intended to supplement the original ASEL.PDF manual available from [www.controleverything.com](http://www.controleverything.com). This publication provides an updated command set for controlling the ASEL video switcher with Version 2.0 firmware installed. If you currently own an ASEL video switcher, the main controller may be upgraded for \$10. Request part number UPGASEL. If you purchased your ASEL video switcher prior to April 14, 2000, version 1.0 firmware is installed. If you purchased your ASEL video switcher on or after April 14, 2000, version 2.0 firmware is installed. If you are unsure which version is installed in your ASEL video switcher, look for the chip that begins PIC16C. If this part number is PIC16C54, then your ASEL video switcher has version 1.0 firmware installed. If this part number is PIC16C554, then your ASEL video switcher has version 2.0 firmware installed and does not require an upgrade. Version 2.0 firmware supports the E3C command set, allowing up to 256 devices to be controlled from a single serial port. In addition, writing software to control the ASEL is much easier with the new command-based instruction set.

### Installing Firmware 2.0 Upgrade

- 1) Touch a grounded metal surface to drain all static from your body.
- 2) Locate the chip labeled PIC16C54 and take note of the direction of the notch and letters written on this chip.
- 3) Use a small flat head screwdriver and carefully pry each end of the chip until it is out of the socket.
- 4) Carefully seat the new chip, PIC16C554 in the same direction as the original firmware.
- 5) Make sure all leads are within the socket before seating.
- 6) Make sure force is evenly applied and carefully but firmly press the new chip into place.

**STOP AND CAREFULLY REMOVE IF LEADS BEGIN TO SHOW SIGNS OF BENDING.**

**Warranty**

NCD Warrants its products against defects in materials and workmanship for a period of 90 days. If you discover a defect, NCD will, at its option, repair, replace, or refund the purchase price. Simply return the product with a description of the problem and a copy of your invoice (if you do not have your invoice, please include your name and telephone number). We will return your product, or its replacement, using the same shipping method used to ship the product to NCD.

This warranty does not apply if the product has been modified or damaged by accident, abuse, or misuse.

**30-Day Money-Back Guarantee**

If, within 30 days of having received your product, you find that it does not suit your needs, you may return it for a refund. NCD will refund the purchase price of the product, excluding shipping/handling costs. This guarantee does not apply if the product has been altered or damaged.

**Copyrights and Trademarks**

Copyright 2000 by NCD. All rights reserved. Other brand and product names are trademarks of registered trademarks of their respective holders.

**Disclaimer of Liability**

NCD is not responsible for special, incidental, or consequential damages resulting from any breach of warranty, or under any legal theory, including lost profits, downtime, goodwill, damage to or replacement of equipment or property, and any costs or recovering, reprogramming, or reproducing any data stored in or used with NCD products.

**Technical Assistance**

Technical questions should be e-mailed to Ryan Sheldon at [ncdryan@aol.com](mailto:ncdryan@aol.com). Technical questions submitted via e-mail are answered up to 20 times daily. Technical support is also available by calling (417) 646-5644.

**NCD Contact Information****Mailing Address:**

National Control Devices  
P.O. Box 455  
Osceola, MO 64776

**Telephone:**

(417) 646-5644

**FAX:**

(417) 646-8302

**Internet:**

[ncdryan@aol.com](mailto:ncdryan@aol.com)  
[www.controlanything.com](http://www.controlanything.com)  
[www.controleverything.com](http://www.controleverything.com)  
207.198.72.52

## Sending Commands to the ASEL Video Switcher

The ASEL is capable of receiving data via RS-232 serial communications. The ASEL is compatible with just about any computer or microcontroller ever produced, including the Macintosh, Amiga, Basic Stamp, and of course, Windows & DOS based machines.

Regardless of the system you are using, you will need access to a programming language that supports program control of the serial port on your system.

A terminal program is not suitable for controlling the ASEL. Commands should be sent using ASCII character codes 0-255 rather than ASCII characters (A, B, C etc.). See "ASCII Codes vs. Characters" on this page.

Most systems require you to open the appropriate serial port (COM port) prior to sending or receiving data.

Because there are so many different ways to send and receive data from various languages on various platforms, we will provide generic instructions that can be easily converted to your favorite language.

For example, if this manual says "Send ASCII 254", the user will need to translate this instruction into a command that is capable of sending ASCII character code 254.

To Send ASCII 254 from Visual Basic, you will use the following line:

```
MSComm1.Output = Chr$(254)
```

In Qbasic, you can send ASCII 254 using the following line of code:

```
Print #1, Chr$(254);
```

Note that sending ASCII character code 254 is NOT the same as sending ASCII characters 2, 5, and 4 from a terminal program. Typing 2, 5, and 4 on the keyboard will transmit three ASCII character codes.

For your convenience, we have provided several programming examples in various popular programming languages for controlling the ASEL. These examples should greatly speed development time. You may want to visit [www.controleverything.com](http://www.controleverything.com) for the latest software and programming examples.

Programming examples for the ASEL are much more extensive for Visual Basic 6 users than for any other programming language. If you are not a VB programmer, you may consider looking at the VB6 source code, as it is easily translated into other popular languages.

**Regardless of your programming background, the provided Visual Basic 6 source code is very easy to understand and will likely resolve any communication questions you may have. VB6 programming examples may be viewed in any text editor.**

## ASCII Codes vs. Characters

The differences between ASCII codes and ASCII characters tend to generate a lot of confusion among first-time RS-232 programmers. It is important to understand that a computer only works with numbers. With regard to RS-232 data, the computer is only capable of sending and receiving numbers from 0 to 255.

What confuses people is the simple idea that the numbers 0 to 255 are assigned letters. For instance, the number 65 represents the letter A. The number 66 represents the letter B. Every character (including numbers and punctuation) is assigned a numeric value. This standard of assignments is called ASCII, and is a universal standard adopted by all computers with an RS-232 serial port.

ASCII characters codes can be clearly defined as numbers from 0 to 255.

ASCII characters however are best defined as letters, A, B, C, D, as well as punctuation, !@#%\$, and even the numbers 0-9.

Virtually all programming languages permit you to send ASCII in the form of letters or numbers. If you wanted to send the word "Hello" out the serial port, it is much easier to send the letters H, e, l, l, and o than it is to send the ASCII character codes that represent each letter.

For the purposes of controlling NCD devices however, it is much easier to build a numeric command set. Especially when communicating to devices where you want to speak to lots of outputs (which are numbered), inputs (which are also numbered), or control specific devices using their device number (from 0 to 255).

Put simply, it is easier to control NCD devices using ASCII character codes 0 to 255 than it is to use ASCII characters A, B, C, D, etc.

Because terminal programs are ASCII character based, it may be difficult to generate the proper series of keystrokes that would be necessary to activate a particular function. Therefore, they are not suitable for controlling NCD devices. In a real world control application, a terminal program would not likely be used to control NCD devices anyway. Therefore, a programming language that supports the transmission and reception of ASCII character codes 0 to 255 is highly recommended.

## The E3C Command Set: Software Control of Multiple NCD Devices

The E3C command set allows you to control up to 256 NCD devices from a single serial port. It is OK to mix different types of devices, as long as the devices are E3C compliant. The ASEL supports most E3C commands.

### **How does E3C Work?**

First of all, each device must be assigned a device number from 0 to 255. The ASEL can be set to device numbers 0-15 using jumpers J3-J6. Additional device base numbers can be special ordered for controlling the ASEL at any device number.

E3C stands for Enabled 3-Wire Communication. Put simply, when you first power up your computer and all the devices attached to the serial port, all devices will respond to your commands.

Using the E3C command set, you can specify which devices will listen and which devices will ignore your commands. Note that E3C commands are never ignored by any device, regardless of the commands you send to the controller.

The number to the left of each command indicates the ASCII character code that must be sent to issue the command. All commands must be preceded with ASCII character code 254 to place the device in command mode. See examples at right.

## The E3C Command Set

### **248 Enable All Devices:**

Tells all devices to respond to your commands.

### **249 Disable All Devices:**

Tells all devices to ignore your commands.

### **250 Enable a Selected Device:**

Tells a specific device to listen to your commands.

### **251 Disable Selected Device:**

Tells a specific device to ignore your commands.

### **252 Enable Selected Device Only:**

Tells a specific device to listen to your commands, all other devices will ignore your commands.

### **253 Disable a Selected Device Only:**

Tells a specific device to ignore your commands, all others will listen. THIS COMMAND IS NOT SUPPORTED BY THE ASEL.

## E3C Visual Basic Programming Examples

The E3C command set is easily used from any programming language that supports serial communication. The following Visual Basic 6 Example source code demonstrates subroutines that can be used to control which devices will listen and which devices will ignore your commands.

## Sample Code: The E3C Command Set

```
Public Sub EnableAllDevices()  
    'Enable All E3C Devices  
    MSComml.Output = Chr$(254) 'Enter Command Mode  
    MSComml.Output = Chr$(248) 'E3C Enable All Device Command  
End Sub  
  
Public Sub DisableAllDevices()  
    'Disable All E3C Devices  
    MSComml.Output = Chr$(254) 'Enter Command Mode  
    MSComml.Output = Chr$(249) 'E3C Disable All Device Command  
End Sub  
  
Public Sub EnableSpecificDevice(Device)  
    'Enable A Specific E3C Devices, Other Devices will be unchanged  
    MSComml.Output = Chr$(254) 'Enter Command Mode  
    MSComml.Output = Chr$(250) 'E3C Disable Specific Device Command  
    MSComml.Output = Chr$(Device) 'Device Number that will be Disabled  
End Sub  
  
Public Sub DisableSpecificDevice(Device)  
    'Disable A Specific E3C Devices, Other Devices will be unchanged  
    MSComml.Output = Chr$(254) 'Enter Command Mode  
    MSComml.Output = Chr$(251) 'E3C Disable Specific Device Command  
    MSComml.Output = Chr$(Device) 'Device Number that will be Disabled  
End Sub  
  
Public Sub DisableAllDevicesExcept(Device)  
    'Disable All E3C Devices Except (Device)  
    MSComml.Output = Chr$(254) 'Enter Command Mode  
    MSComml.Output = Chr$(252) 'E3C Disable All Device Except Command  
    MSComml.Output = Chr$(Device) 'Device Number that will be Active  
End Sub
```



# Interfacing to the Serial Port of Your Computer

Before we begin, we need to build a cable to transmit data from the serial port of your computer. Since most problems can be attributed to improper connection to the serial port, extra time should be spent verifying these simple connections. Follow the diagrams to build a cable for the system you intend to use. Take care not to accidentally short out pins on the connectors. Though it is theoretically OK to short all the lines together from an RS-232 port, we strongly recommend you take every precaution to avoid this. Once the cable is built, connect an LED with a 2.2K resistor to the RS-232 ground and RS-232 data output lines. The ground should connect to the short lead of the LED. The resistor should connect to the long lead. The remaining end of the resistor should connect to the RS-232 data line. Enter and run the program for your computer found to the right of each box. You should see the LED flash so fast that it appears to be on. Do not proceed until your cable is working. *Please note that all illustrations show the solder side of the connector.*

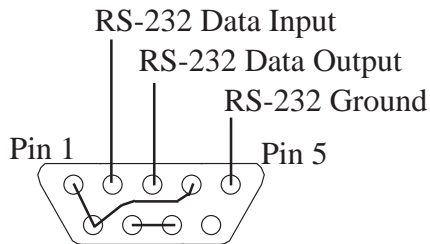
## PC Specific:

If you are using a PC, identify the type of serial connector you will need. There are only two common choices available. Both connectors are female; the first diagram is for a DB-25 connection, and the second is for a DB-9 connection. PC users will need to find out if the serial port is accessed as "COM1:" or "COM2:". The best way to do this is to try both software settings and see which one works. Software from here on will assume you are using "COM2:".

## Macintosh Specific:

If you are using a Macintosh system, make sure the connections are not accidentally touching each other. The lines on the mini 8-pin din are very close together and very difficult to work with.

## PC Users

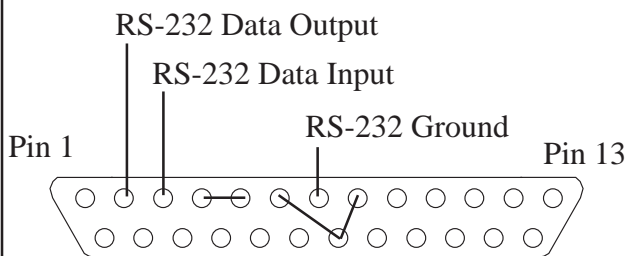


Example Shown Using a DB-9 Female Connector

In QBasic for PC compatible systems, enter and run the following program to test the serial port of your computer:

```
OPEN "com2:9600,N,8,1,CD0,CS0,DS0,OP0" FOR OUTPUT AS #1
DO
  FOR N = 0 TO 255
    PRINT #1, CHR$(N);
  NEXT N
LOOP
```

## PC & Amiga Users

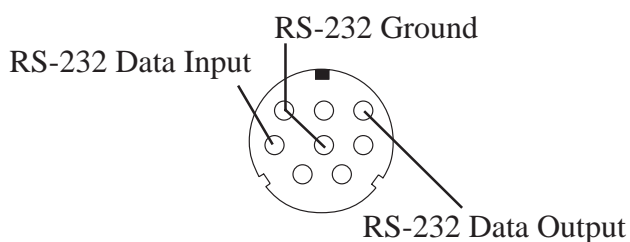


Example Shown Using a DB-25 Female Connector

In Amos for Commodore Amiga systems, enter and run the following program to test the serial port of your computer. Press the mouse button to exit the program.

```
Serial Open 0,0
Serial Speed 0,9600
Serial Buf 0,64
Serial Bits 0,8,1
While Mousekey=0
  For N = 0 To 255
    Serial Send 0,Chr$(N)
  Next N
Wend
```

## Macintosh Users



Example Shown Using a MINI DIN8 Male Connector

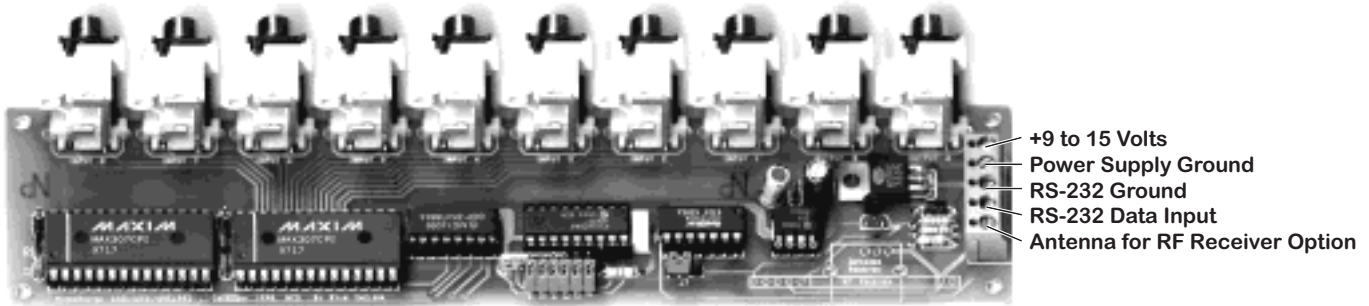
In Chipmunk Basic for Apple Macintosh systems, enter and run the following program to test the serial port of your computer. Be sure to select the serial port your cable is connected to (modem or printer). Press the mouse button to exit the program.

```
10 open "COM1:" for output as #1
20 while not mouse(0)
30   for n = 0 to 255
40     fputbyte n,1
50   next n
60 wend
```

- J1 - Baud Rate Select
- J2 - Reserved
- J3 - Address Select 1 Weight 1
- J4 - Address Select 2 Weight 2
- J5 - Address Select 3 Weight 4
- J6 - Address Select 4 Weight 8
- J7 - Left RS-232/Right Remote

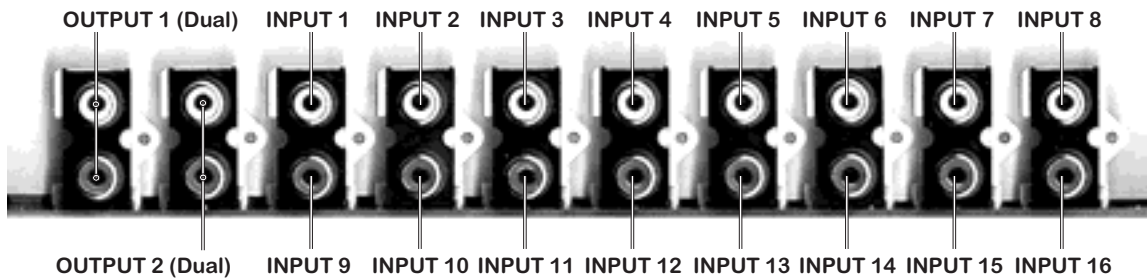
Baud rate is normally set to 9600 baud (J1 removed). Install J1 to set the baud rate to 1200 baud.

The "Weight" column provides a simplified way of setting the address. Removing all jumpers sets the weight to 0, setting the address to 0. Installing all the jumpers has a weight of 1+2+4+8 setting the address to 15. Always change the software device number to match the hardware device number.



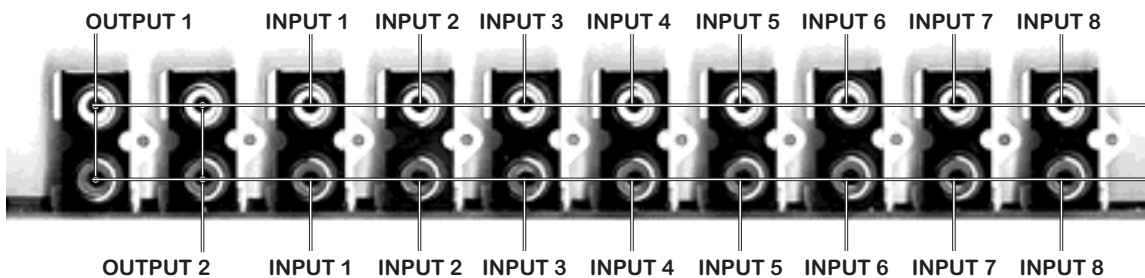
The MAX306 or MAX307 pair may be used in the NCD-ASEL board. When the MAX306 pair is installed, the ASEL provides 16 inputs and 2 outputs. Any of the 16 inputs may be routed to any of the 2 outputs. When using the MAX307 pair, the ASEL provides 8 stereo inputs and 2 stereo outputs. Any stereo input may be routed to any stereo output, both analog channels are switched simultaneously and channels cannot cross (e.g. a white input may only be routed to a white output).

The MAX306 pair configures the connectors of the ASEL (rear view) as follows.



Jumpers J8A and J8B should be installed when using the MAX306 pair.

The MAX307 pair configures the connectors of the ASEL (rear view) as follows.



Jumpers J8A and J8B should be removed when using the MAX307 pair.

Channels A&B always switch simultaneously.

Channel A Analog I/O

Channel B Analog I/O

Channels A&B may both be video, audio, or 1 audio and 1 video. A&B Channels are always isolated from each other.