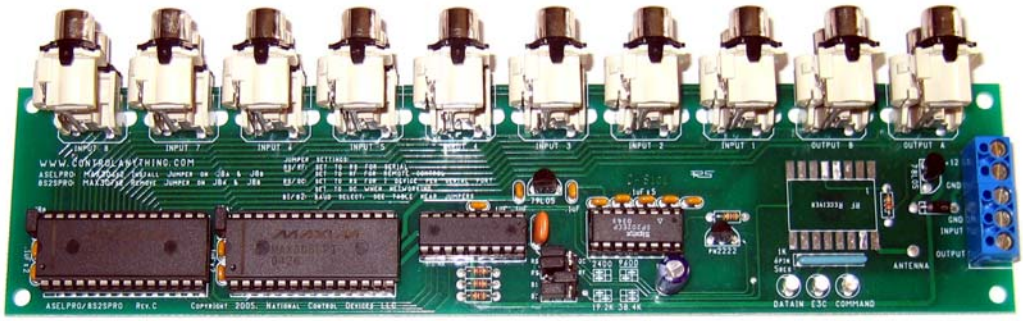
ASELPRO/8S2SPRO

RS-232 E3C Networkable Analog Matrix Switchers

5-Year Repair or Replace Warranty!!!



The ASELPRO and 8S2SPRO represent the latest version of our popular analog signal switchers. Ideally suited for audio and video switching, as well as many analog switching applications, these new devices now support 2-Way communications, 3 Status LEDs, and communications rates up to 38.4K baud. The ASELPRO has 16 analog inputs and 2 analog outputs. Any analog input can be routed to any analog output under computer control. The 8S2SPRO is often used in switching of stereo audio signals. The 8S2SPRO has 8 stereo inputs and 2 stereo outputs. Any stereo input can be routed to any stereo output under computer control. Left and Right audio channels are switched simultaneously. Alternatively, the inputs of the 8S2S can be used for one audio channel and one video channel or, the inputs can be used for video or other analog signals. Input to output resistance is about 120 Ohms.

Control 256 Devices from a Single Serial Port
User-Selectable Communication Rates from 2400, 9600, 19.2K, and 38.4K Baud
E3C Compliant Command Set
Data Receive LED Shows Incoming Signals
E3C Led Shows with the Device Enabled or Disabled
Command LED Shows the Processor is Accepting and Interpreting Commands
12 Volt DC Operation
Direct Form and Function Upgrade to the 8S2S and ASEL
User-Programmable Startup Status
Simultaneously Set Input and Output Channels
Capable of Reporting Current Channel Selection
High Speed RS-232 Data Input
O.C. RS-232 Communication for Networking Multiple Devices
Powerful ASCII Character Code Based Command Set
Compatible with ANY Computer or Microcontroller

Device Variations

ASELPRO 16 Input 2 Output Analog Switcher, Route Any Input to Any Output under Computer Control
8S2SPRO 8 Stereo Inputs 2 Stereo Outputs, Route Any Stereo to Any Stereo Output under Computer Control

ASEL/8S2S vs. ASELPRO/8S2S Pro: *Feature Differences*	Original ASEL/8S2S Analog Switchers	NEW ASEL/8S2S Pro Analog Controllers
RS-232 Interface	MC1489	MAX202
ASEL/8S2S Pro Command Set Compatible with Original ASEL/8S2S Design	-	Yes
ASEL/8S2S Pro Form Factor Compatible with Original ASEL/8S2S Design	-	Yes
Set Status of Individual Channels	Yes	Yes
Set Status of Multiple Channels Simultaneously	Yes	Yes
Read Status of Individual Channels	No	Yes
Maximum Supported Baud Rates	9600	38.4K
2-Way Communications Support	Limited	Yes
E3C 256 Device Network Compliance	Limited	Yes
Programmable Device Number	Jumper Configured	Software Configured
Programmable Power-up Input/Output Status	No	Yes
Integrated User-Programmable Memory	No	Yes
20 MHz CPU Operation	No	Yes

IMPORTANT NOTE:

A DETAILED MECHANICAL DRAWING CAN BE FOUND ON THE FOLLOWING PAGE OF OUR WEB SITE:
http://www.controlanything.com/photos/ASELDIM.gif

5-Year Repair or Replace Warranty

Warranty
NCD Warrants its products against defects in materials and workmanship for a period of 5 years. 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

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 ryan@controlanything.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 LLC
P.O. Box 455
Osceola, MO 64776

Telephone:
(417) 646-5644

FAX:
(417) 646-8302

Internet:
ryan@controlanything.com
www.controlanything.com
www.controleverything.com

POWER SUPPLY REQUIREMENTS

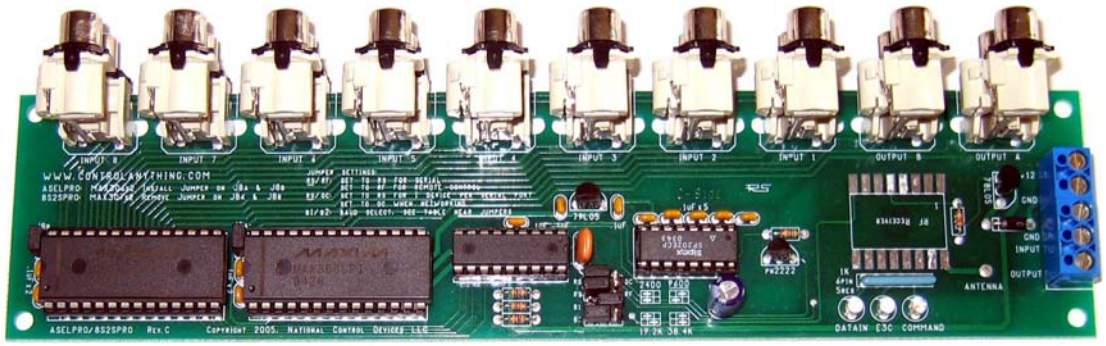
1) DO NOT USE A WALL WART TYPE UNREGULATED POWER SUPPLY. THESE SUPPLIES MAY INTRODUCE NOISE INTO THE ANALOG SIGNAL THAT IS BEING SWITCHED.
2) USE ONLY A COMPUTER GRADE REGULATED SWITCHER SUPPLY RATED AT 12 VOLTS DC, 100 ma OR GREATER.
3) USE A SUPPLY RATED FOR MORE AMPERAGE WHEN POWERING MULTIPLE BOARDS.
4) DC POWER SHOULD NEVER TRAVEL GREATER THAN 20 FEET. A SEPARATE POWER SUPPLY SHOULD BE USED FOR EACH CONTROLLER IF CONTROLLERS ARE NOT LOCATED WITHIN 20 FEET OF EACH OTHER.

Status LEDs:

Data In LED:
This LED is connected to the RS-232 input path. This LED flashes any time a data voltage appears on the RS-232 input line. If this LED does not flash, the controller is not receiving data.

E3C LED:
This LED is used to display the E3C status of the board. By default, this LED is lit, meaning it is ready to process RS-232 serial commands. This LED turns off when an E3C command is issued to direct data to a different E3C device. This LED must be lit for proper processing of switching commands. Use the E3C command set to enable/disable this device. This LED will light when the device is enabled.

Command LED:
This LED flashes with the Data In LED when commands are being processed. Both Data In and Command LEDs appear to do the exact same thing under normal operation However, this LED is driven by the microprocessor and ONLY lights when a valid command sequence is executed. If the Data In LED lights, and the Command LED does not light at the same time, the device may not be enabled (see E3C LED above). If this LED does not light in sync with the DATA IN LED, check to make sure the baud rate of the controller matches the baud rate of your software.



NOTE:
The RS-232 Data Input (above) connects to the RS-232 Output on your computer. The RS-232 Output shown above connects to the RS-232 Input on your computer. The RS-232 Data Output line is optional, and is only required for 2-way communications.

J8A and J8B Jumpers

The ASELPRO has two large chips installed labeled MAX306CPI. Both jumpers J8A and J8B should be installed when using this chipset.

The 8S2SPRO has two large chips installed labeled MAX307CPI. Both jumpers J8A and J8B should be removed when using this chipset.

Default Jumper Settings

The jumper settings shown in the photo above represent the default jumper settings. This sets communication to 9600 baud, 8 data bits, 1 stop bit, no parity.

RS/OC & RS/RF Jumpers

RS/OC Jumper:
Install on the left side as shown in the photo above for normal RS-232 communications. Install to the right side when using multiple controllers attached to a single serial port using 2-Way communications to each device (the RSB Serial Booster will be required for proper operation).

RS/RF Jumper:
Install on the left side as shown in the photo above for normal RS-232 communications. Install to the right side when using the RF receiver option for remote communications. The RF receiver option is scheduled for release in 2005. A firmware upgrade will be required for remote control applications. This jumper is read only when power is first applied to the controller. Changing this jumper while power is applied will have no effect until the next power-up cycle.

B1/B2 Jumpers

The B1/B2 jumpers set the baud rate. These jumpers are read only when power is first applied to the controller. Changing this jumper while power is applied will have no effect until the next power-up cycle. For your convenience, a small map is printed on the circuit board that shows the jumper setting for the various baud rates. Set the jumpers in accordance with the map printed on the controller. The default setting of 9600 baud is shown in the photo above. This controller supports 2400, 9600, 19.2K, and 38.4K baud.

QS12 Quick Start Kit

The QS12 is attached to the ASELPRO/8S2SPRO as shown in the photo above. The QS12 has 5 prongs that extend out the side. These prongs are secured using the terminal block screws. This configuration should be considered temporary for the purposes of initial testing, wires should be connected between the QS12 and the ASELPRO/8S2SPRO for a more permanent installation.

IMPORTANT NOTE:

A DETAILED MECHANICAL DRAWING CAN BE FOUND ON THE FOLLOWING PAGE OF OUR WEB SITE:
http://www.controlanything.com/photos/ASELDIM.gif

ASELPRO Operational Description

The ASELPRO has 16 inputs and 2 outputs. Any input can be routed to any output under computer control. Two output ports, Output A and Output B, each have two connectors. The red and white connectors on output A are electrically tied together. Output B is configured the same way as output port A. This is the most popular of the two controller varieties.

8S2SPRO Operational Description

The 8S2SPRO has 8 stereo inputs and 2 stereo outputs. Any stereo input can be routed to any stereo output under computer control. Two output ports, Output A and Output B, each have two connectors. Signal cannot cross the colors boundaries of these connectors. Meaning, signals applied to the red connectors are switched to red connector outputs ONLY. Signals applied to the white connectors are switched to white connector outputs ONLY. Red and White signals cannot be crossed under software control. This devices is ideally suited for stereo audio switching operations.

Switching Characteristics

Approximately 30 MHz Signal Bandwidth with True Bipolar Signal Switching +/-5VDC. Capable of switching most analog signals. Ideally suited for many audio, video, and sensor signal switching. This controller introduces a 120 Ohm loss in signal between input and output. Signals can be reversed, inputs may be used as outputs and outputs may be used as inputs for true bi-directional signal switching. Maximum theoretical switching speed at 38.4K baud is 1,280 switches per second.

RCA Phono Connector Varieties

The photo below shows an ASELPRO/8S2SPRO with JALCO RCA phono connectors installed. We may be required to use connectors from two other suppliers, which are similar in form and function, to the JALCO brand shown below. The mechanical characteristics may vary slightly between manufacturers. We purchase our connectors in large quantities to help stabilize this product. JALCO brand connectors are slightly different than other varieties of connectors. Depending on the brand of connectors used, you may need to make provisions for future software changes. The JALCO brand connectors, by default, MAY place inputs 1-8 on the lower row of an ASELPRO, inputs 9-16 will be on the upper row. This is easily compensated for in software. We have also made some provisions to swap the connector map to the correct position using the MAP command in version 4.0 of the ASELPRO firmware. The 8S2SPRO has the same side effect, but it is not noticeable because there are only 8 inputs switched in pairs. For these controllers, the MAP setting will not appear to have no effect and the connector map will never appear incorrect. Please review the MAP function later in this manual. Note that the MAP command is ONLY supported by Version 4.0 firmware and later, which began shipping November 8th, 2004.

Electrical Notice:

The ring on the outside of each connector is grounded. All grounds are tied together, including RS-232 ground, power supply ground, and signal ground. The center of each connector is switched under software control.

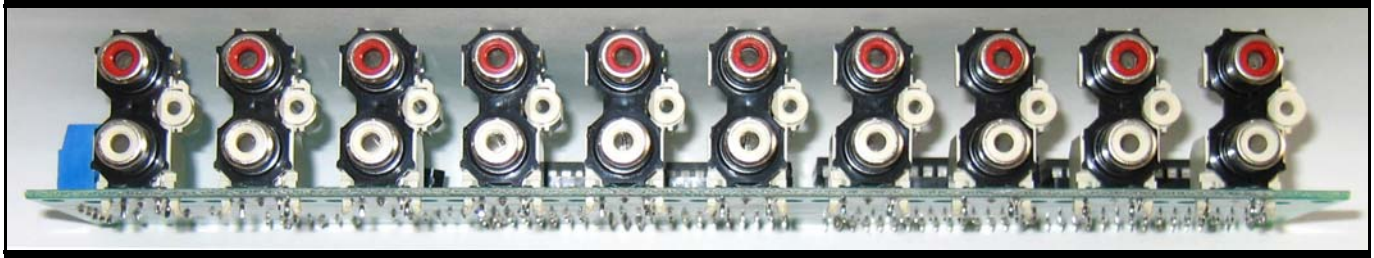
Communication Parameters:

User-Selected Baud Rate, 8 Data Bits, 1 Stop Bit, and No Parity.

ASELPRO Connector Layout:

Output A Connectors (Upper and Lower) are Electrically Tied Together, Same with Output B

Output A	Output B	Input 1	Input 2	Input 3	Input 4	Input 5	Input 6	Input 7	Input 8

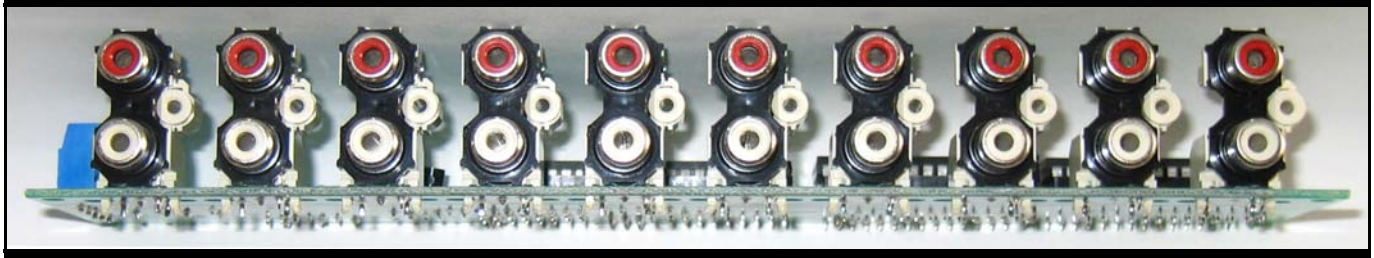


Output A	Output B	Input 9	Input 10	Input 11	Input 12	Input 13	Input 14	Input 15	Input 16

8S2SPRO Connector Layout:

Signals on the Red Side are Isolated from Signals on the White Side and Switched Simultaneously

Output A	Output B	Input 1	Input 2	Input 3	Input 4	Input 5	Input 6	Input 7	Input 8



Output A	Output B	Input 1	Input 2	Input 3	Input 4	Input 5	Input 6	Input 7	Input 8

IMPORTANT NOTE:

A DETAILED MECHANICAL DRAWING CAN BE FOUND ON THE FOLLOWING PAGE OF OUR WEB SITE:
http://www.controlanything.com/photos/ASELDIM.gif

Two-Way Communication:

The ASELPRO/8S2SPRO support two-way communication for reporting the signal routing status back to the host computer.

The ASELPRO/8S2SPRO should be connected as shown below when using this device for the first time. Even if you plan to connect several ASELPRO/8S2SPRO controllers to a single serial port, this wiring diagram must first be used to program the device number into the controller.

The ASELPRO/8S2SPRO Visual Basic Example Program expects this wiring configuration.

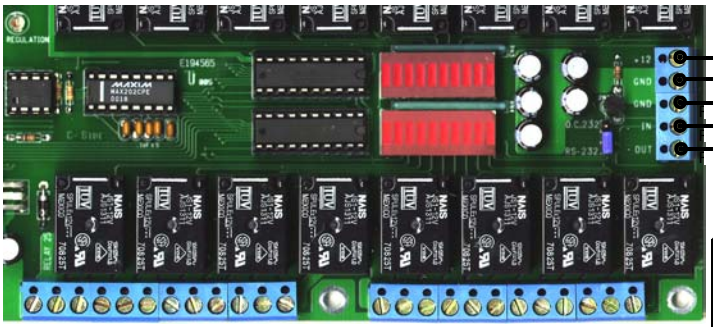
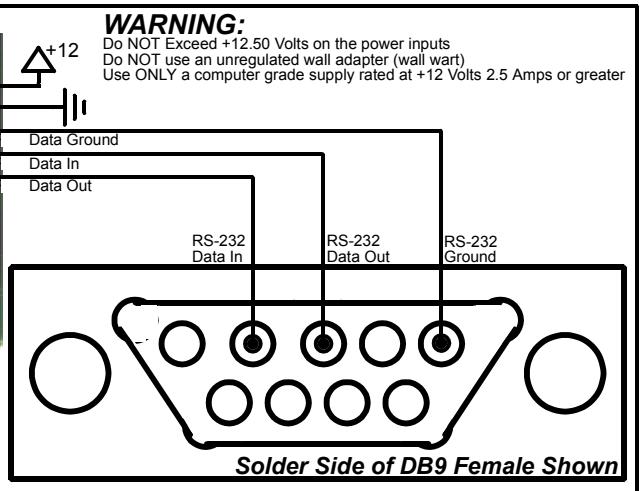


Illustration shows the R32 relay board, this wiring configuration is the same for all NCD controllers with a 5-position terminal block.



ASELPRO/8S2SPRO One-Way Communication:

The ASELPRO/8S2SPRO can be connected to a computer or microcontroller using as little as two wires. Memory Storage commands may take a little longer to process than others, so it may be necessary to add short delays in your program to allow time for execution of these commands. It is not possible to request data from the ASELPRO/8S2SPRO using this wiring configuration.

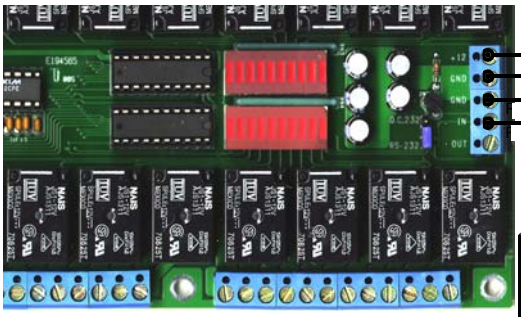
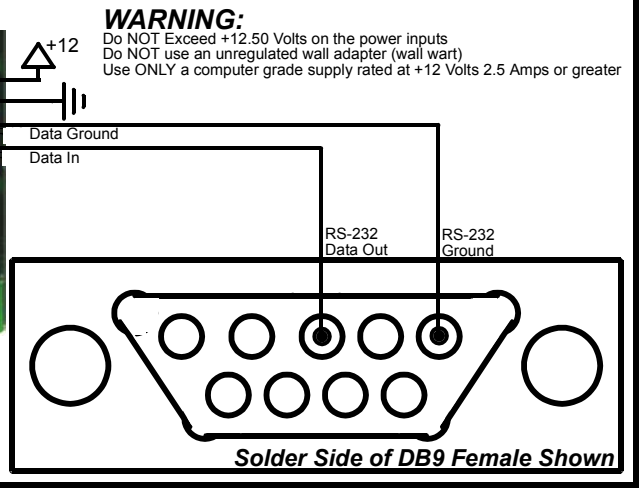


Illustration shows the R32 relay board, this wiring configuration is the same for all NCD controllers with a 5-position terminal block.



Multiple Controllers, One Serial Port: Two-Way & One-Way Hybrid Communication

Multiple NCD Devices can be connected to a single serial port and controlled individually. This example shows an R16 and an R32 connected to a single serial port. This example can be applied to any NCD device with a 5-Position terminal block connector, including the ASELPRO/8S2SPRO . The NCD E3C network allows you to mix different NCD devices on a single serial port.

Before using this wiring configuration, each device must be programmed with a unique device number (See E3C Commands for each individual controller for programming details). Once a device number has been stored into each controller this wiring configuration may be used to control up to 256 different relay boards or other NCD devices in any combination. This wiring configuration only allows 2-way communication with the R32. Relay status information cannot be read from the R16.

When all boards are first powered up, all devices will respond to incoming commands. Use E3C Command 252 to speak to one device at a time. Send 252, 0, any subsequent commands should be for the R32, Device 0. Send 252, 1, any subsequent commands should be for the R16, Device 1.
This E3C Command 252 is useful when mixing different types of controllers on a single serial port.

Multiple Device Control: Quick Example

Step 1: Store a Device Number from 0 to 255 into Each Controller. Example Shows Device 0 and 1.

Step 2: Route Commands to Device 0 Only by Sending the Following Commands:

ASCII 254	'Enter Command Mode
ASCII 252	'Select a Device to Control Command
ASCII 0	'Set Device to Control to 0

Step 3: Activate Relay 1 on Device 0 (R32)

ASCII 254	'Enter Command Mode
ASCII 1	'Relay On Command
ASCII 0	'Turn On Relay 1

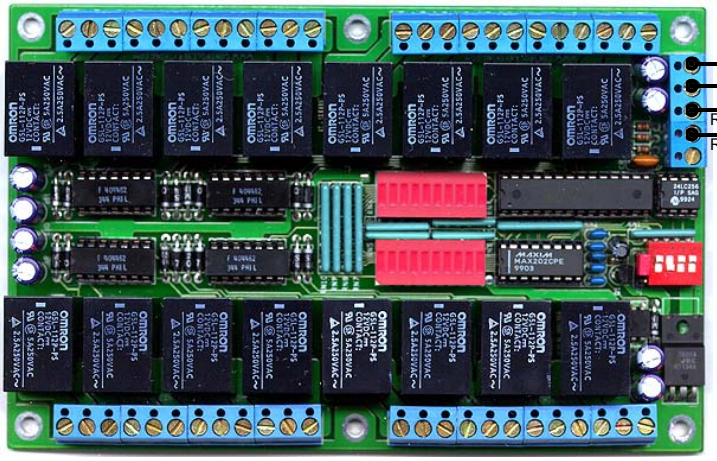
Step 4: Route Commands to Device 1 Only by Sending the Following Commands:

ASCII 254	'Enter Command Mode
ASCII 252	'Select a Device to Control Command
ASCII 1	'Set Device to Control to 1

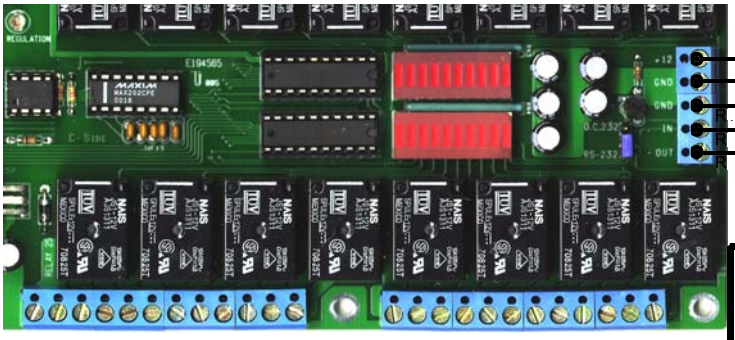
Step 3: Activate Relay 1 on Device 1 (R16)

ASCII 254	'Enter Command Mode
ASCII 16	'Turn Relay 0 On

R16 Relay Controller:



R32 Relay Controller:



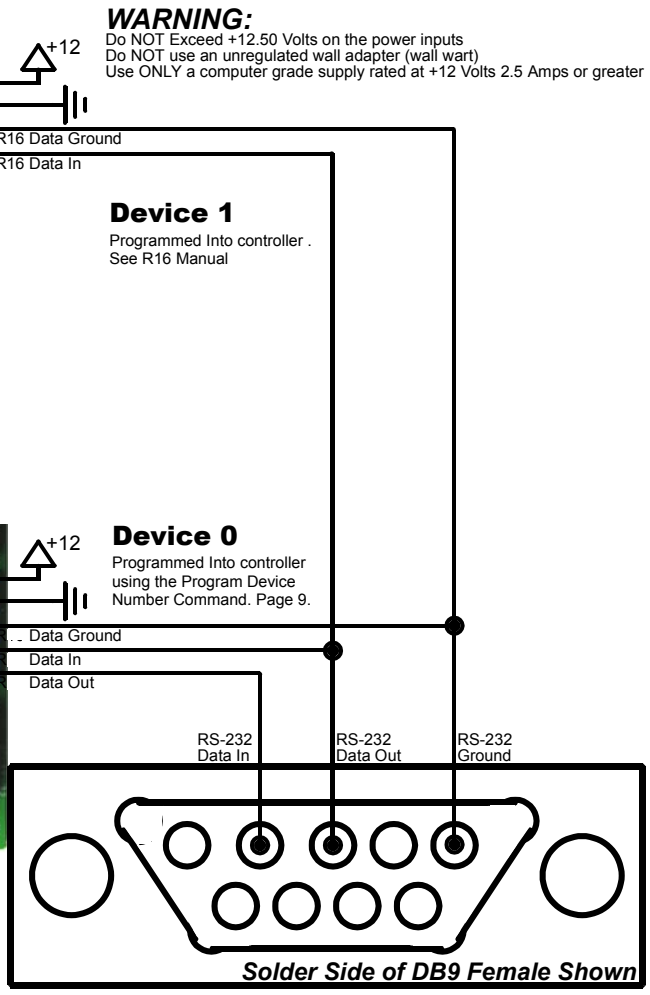


Illustration shows the R16 and R32 relay boards, this wiring configuration is the same for all NCD devices with a 5-position terminal block.

Multiple Controllers: Two-Way Communication

Many NCD Devices support two-way communication using the RSB serial booster. Jumpers must be set on all networked devices for Open Collector data transmission. See the appropriate manual for each controller. This is ONLY required when using the RSB serial booster. The RSB serial booster should be used when controlling several devices over long distances (has been tested in excess of 500 feet). Actual reliability over long distances depends greatly on baud rate and type of wire used. Experimentation will be required. Always start at the lowest baud rate and work your way up.

A unique device number should be programmed into each device prior to using this example.

Selecting a Power Supply

1) *DO NOT USE A WALL WART TYPE UNREGULATED POWER SUPPLY.*
2) *USE ONLY A COMPUTER GRADE REGULATED SWITCHER SUPPLY RATED AT 12 VOLTS DC, 1.25 AMPS OR GREATER.*
3) *USE A SUPPLY RATED FOR MORE AMPERAGE WHEN POWERING MULTIPLE BOARDS.*
4) *DC POWER SHOULD NEVER TRAVEL GREATER THAN 20 FEET. A SEPARATE POWER SUPPLY SHOULD BE USED FOR EACH CONTROLLER IF CONTROLLERS ARE NOT LOCATED WITHIN 20 FEET OF EACH OTHER.*
5) *RELAY COILS ARE RATED AT 12 VOLTS DC. HIGHER VOLTAGES MAY SHORTEN THE COIL LIFE. LOWER VOLTAGES MAY CAUSE UNRELIABLE OPERATION, BUT WILL NOT DAMAGE THE CONTROLLER.*
6) *IT IS SAFE TO CONTROL ANY +12 VOLT RELAY CONTROLLER FROM AN AUTOMOTIVE POWER SYSTEM. YOU MAY DISREGARD THE +12.5 VOLT WARNING ON THIS PAGE AND THE PREVIOUS PAGE IN THIS APPLICATION.*

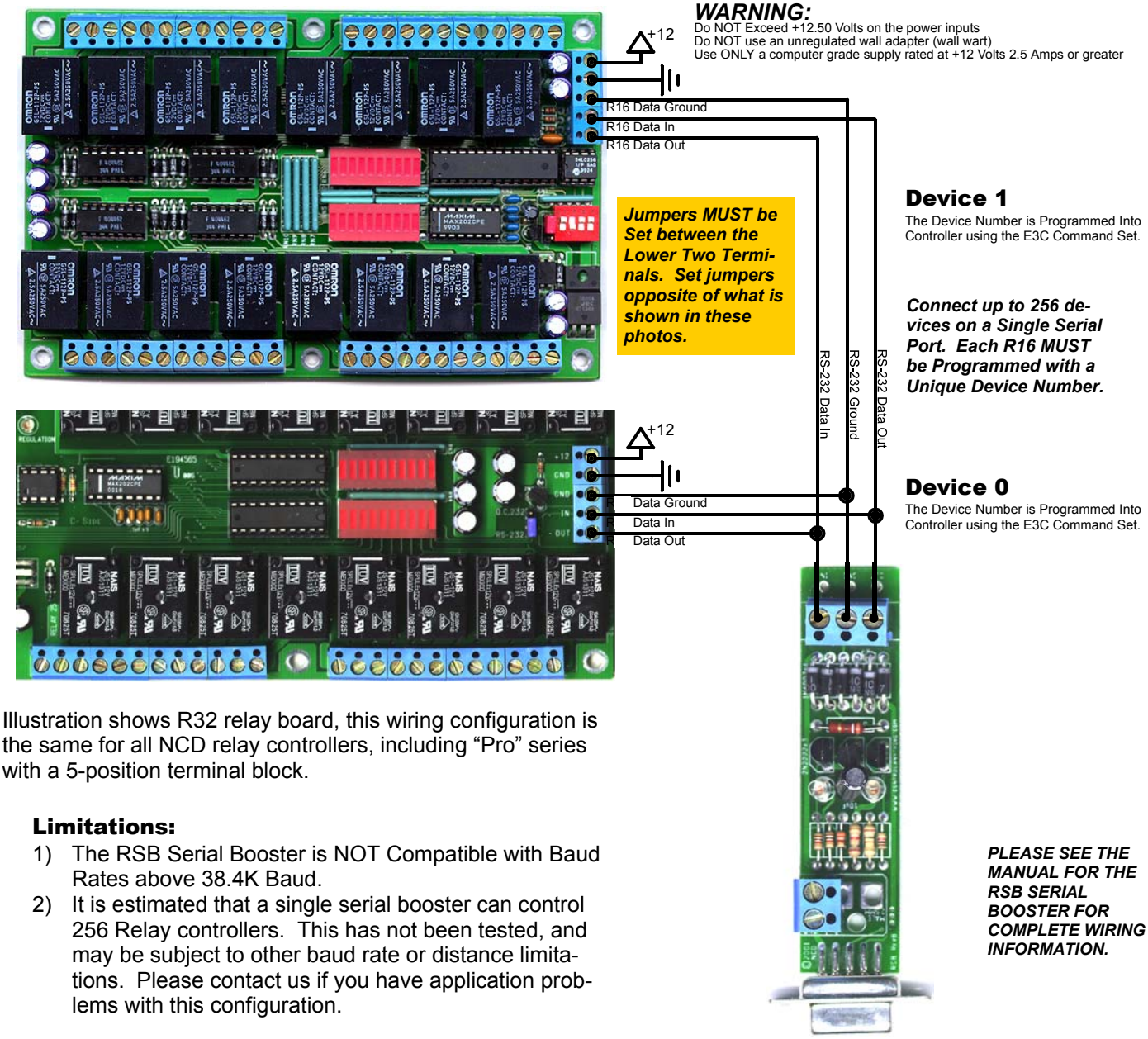


Illustration shows R32 relay board, this wiring configuration is the same for all NCD relay controllers, including “Pro” series with a 5-position terminal block.

Limitations:

1) The RSB Serial Booster is NOT Compatible with Baud Rates above 38.4K Baud.
2) It is estimated that a single serial booster can control 256 Relay controllers. This has not been tested, and may be subject to other baud rate or distance limitations. Please contact us if you have application problems with this configuration.

PLEASE SEE THE MANUAL FOR THE RSB SERIAL BOOSTER FOR COMPLETE WIRING INFORMATION.

Sending Commands to NCD Controllers

Most NCD devices are capable of sending and receiving data via RS-232 serial communications. NCD products are 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 NCD devices. 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.

In your program, you may want to ask an NCD device for the status informaion, just to confirm a command was received and processed. If so, your programming language will support commands for reading data from the serial port.

For your convenience, we have provided several programming examples in Visual Basic 6. These examples should greatly speed development time. You may want to visit **www.controleverything.com** for the latest software and programming examples.

Programming examples 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. Most NCD devices support the full set of E3C commands, plus a set of extended commands for storing and recalling the device number.

How does E3C Work?
First of all, each device must be assigned a device number from 0 to 255. This device number must be programmed using the "Store Device Number" command shown below.

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.

Extended E3C Commands

This device supports two additional E3C commands which should only be used when a single device is attached to your serial port. Extended commands will report back to the computer.

255 Store Device Number:
Stores the device number into the controller. The device number takes effect immediately. The enabled/disabled status of the device is unchanged.

247 Recall Device Number:
Allows you to read the stored device number from the controller.

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
    MSComm1.Output = Chr$(254)    'Enter Command Mode
    MSComm1.Output = Chr$(248)    'E3C Enable All Device Command
End Sub

Public Sub DisableAllDevices()
    'Disable All E3C Devices
    MSComm1.Output = Chr$(254)    'Enter Command Mode
    MSComm1.Output = Chr$(249)    'E3C Disable All Device Command
End Sub

Public Sub EnableSpecificDevice(Device)
    'Enable A Specific E3C Devices, Other Devices will be unchanged
    MSComm1.Output = Chr$(254)    'Enter Command Mode
    MSComm1.Output = Chr$(250)    'E3C Disable Specific Device Command
    MSComm1.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
    MSComm1.Output = Chr$(254)    'Enter Command Mode
    MSComm1.Output = Chr$(251)    'E3C Disable Specific Device Command
    MSComm1.Output = Chr$(Device) 'Device Number that will be Disabled
End Sub

Public Sub DisableAllDevicesExcept(Device)
    'Disable All E3C Devices Except (Device)
    MSComm1.Output = Chr$(254)    'Enter Command Mode
    MSComm1.Output = Chr$(252)    'E3C Disable All Device Except Command
    MSComm1.Output = Chr$(Device) 'Device Number that will be Active
End Sub

Public Sub EnableAllDevicesExcept(Device)
    'Enable All E3C Devices Except (Device)
    MSComm1.Output = Chr$(254)    'Enter Command Mode
    MSComm1.Output = Chr$(253)    'E3C Enable All Device Except Command
    MSComm1.Output = Chr$(Device) 'Device Number that will be Inactive
End Sub
```

Sample Code: Extended E3C Commands

```
Public Sub StoreDeviceNumber(Device)
    'Store an E3C Device Number into the Controller
    MSComm1.Output = Chr$(254)    'Enter Command Mode
    MSComm1.Output = Chr$(255)    'E3C Store Device Number Command
    MSComm1.Output = Chr$(Device) 'Device Number that will be Stored
    WaitForReply                  'Wait for R16 to Acknowledge Command
End Sub

Public Function GetDeviceNumber()
    'Read the E3C Device Number from the Controller
    MSComm1.Output = Chr$(254)    'Enter Command Mode
    MSComm1.Output = Chr$(247)    'E3C Get Device Number Command
    Do                            'Wait for Device to Reply
        DoEvents                  'Allow Windows to MultiTask
    Until MSComm1.InBufferCount > 0 'If the Device Replies
    GetDeviceNumber = Asc(MSComm1.Input)'Get Device Number from Buffer
End Sub
```

The ASELPRO/8S2SPRO Command Set

The ASELPRO/8S2SPRO supports a comprehensive command set, used to input/output channel selection, set operation modes, and store and recall device information. Most users will not use many of the functions built into this controller. The best way to familiarize yourself with the capabilities of this devices is to carefully read through the command set in this section. The "plain English" examples provide a quick, easy to understand definition of what each command does.

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.

Switching Commands

1 Set Output Channel A to Desired Input
This command connects an input to output channel A. This command has one parameter, indicating the desired input channel. Valid input channels are 1-16 for the ASELPRO or 1-8 for the 8S2SPRO.

2 Set Output Channel B to Desired Input
This command connects an input to output channel B. This command has one parameter, indicating the desired input channel. Valid input channels are 1-16 for the ASELPRO or 1-8 for the 8S2SPRO.

3 Set Output Channel A and B Simultaneously (Friendly)
This command routes two inputs to each of the two outputs. This command has two parameters, indicating the desired inputs for output channels A and B. Valid input channel parameters are 1-16 for the ASELPRO or 1-8 for the 8S2SPRO.

4 Set Output Channel 1 and 2 Simultaneously (Fast)
Version 5.0+ Firmware ONLY
This command routes two inputs to each of the two outputs. This command has one 8-bit parameter. The lower four bits of this parameter set the input channel to be routed to Output A, the upper four bits of this parameter set the input channel to be routed to Output B. Experience with the binary numbering system is required to make use of this command. This allows for fastest possible switching routes with minimal serial communications.

10 Get E3C Device Number
This command reports back the currently stored E3C device number. Identical to E3C command 247.

11 Report the Input Currently Routed to Output A
This command reports a Value of 1-16 indicating the current input channel that is currently routed to Output A. Only the lower 8 bits of the returned value are valid for 8S2SPRO users.

12 Report the Input Currently Routed to Output B
This command reports a Value of 1-16 indicating the current input channel that is currently routed to Output B. Only the lower 8 bits of the returned value are valid for 8S2SPRO users.

13 Store Current Route Pattern as Powerup Default
Stores Current Input Channel Selection for Output Channels A and B in Non-Volatile Memory. Every time power is applied, inputs and outputs will automatically switch to the user-defined states of each output channel. Simply use other switching commands to setup the input route. Then issue this command to store the route pattern.

14 Report Default Powerup Route Pattern
This command reports back two values, indicating the input route for Channel A and Channel B (respectively) when power is first applied to the controller.

Visual Basic Programming Examples

Many Visual Basic 6 programming examples are provided in the following pages to assist in the development of software for controlling the ASELPRO/8S2SPRO. Additional source code can be found on our web site at www.controleverything.com.

Sample Code: Switching Commands

```
Public Sub ChannelA(In)                 'Channel Select for Output A
    MSComm1.Output = Chr$(254)          'Enter Command Mode
    MSComm1.Output = Chr$(1)            'Set Channel A Command
    MSComm1.Output = Chr$(In)           'to Input (1-16)
End Sub

Public Sub ChannelB(In)                 'Channel Select for Output B
    MSComm1.Output = Chr$(254)          'Enter Command Mode
    MSComm1.Output = Chr$(2)            'Set Channel B Command
    MSComm1.Output = Chr$(In)           'to Input (1-16)
End Sub

Public Sub ChannelAB(InA,InB)           'Channel Select for Output B
    MSComm1.Output = Chr$(254)          'Enter Command Mode
    MSComm1.Output = Chr$(3)            'Set Channels A & B Command
    MSComm1.Output = Chr$(InA)          'ChannelA Out to Input (1-16)
    MSComm1.Output = Chr$(InB)          'ChannelB Out to Input (1-16)
End Sub

Public Sub ChannelABFriendly(InAB)      'Channel Select for Output B
    MSComm1.Output = Chr$(254)          'Enter Command Mode
    MSComm1.Output = Chr$(4)            'Set Channels A & B Command
    MSComm1.Output = Chr$(InAB)         'ChannelA & B FAST Mode
End Sub

Public Sub ChannelABFast(InAB)          'Channel Select for Output B
    MSComm1.Output = Chr$(254)          'Enter Command Mode
    MSComm1.Output = Chr$(4)            'Set Channels A & B Command
    MSComm1.Output = Chr$(InAB)         'ChannelA & B FAST Mode
End Sub

Public Sub GetE3CDevice                 'Read Stored E3C Device Number
    MSComm1.Output = Chr$(254)          'Enter Command Mode
    MSComm1.Output = Chr$(10)           'Get E3C Device Number Command
    GetI                        'Call GetI Function to Read Device Data
End Sub

Public Sub GetChannelAInput             'Read Input Selection Chan. A
    MSComm1.Output = Chr$(254)          'Enter Command Mode
    MSComm1.Output = Chr$(11)           'Read Channel A Input Command
    GetI                        'Call GetI Function to Read Device Data
End Sub

Public Sub GetChannelBInput             'Read Input Selection Chan. B
    MSComm1.Output = Chr$(254)          'Enter Command Mode
    MSComm1.Output = Chr$(12)           'Read Channel B Input Command
    GetI                        'Call GetI Function to Read Device Data
End Sub

Public Sub SetPowerupInputSelections    'Store as Powerup Default
    MSComm1.Output = Chr$(254)          'Enter Command Mode
    MSComm1.Output = Chr$(13)           'Store Inputs as Power Default
End Sub

Public Sub GetPowerupInputSelections    'Read Powerup Input Selection
    MSComm1.Output = Chr$(254)          'Enter Command Mode
    MSComm1.Output = Chr$(14)           'Read Input Settings
    GetI                                'Report Back Channel A Input
    GetI                                'Report Back Channel B Input
End Sub
```

Sample Code: Reading Data from Controller

```
Public Function GetI
    Do                                  'Wait for Device to Reply
        DoEvents                        'Allow Windows to Multitask
    Until MSComm1.InBufferCount > 0     'If the Device Replies
    GetI = Asc(MSComm1.Input)           'Get Status from Serial Buffer
    Debug.Print GetI                    'Display in Immediate Window
End Sub
```

The ASELPRO/8S2SPRO Command Set

Mapping Commands

The Mapping commands only apply to the ASELPRO, the 8S2S is not affected by the use of these commands. Because we have three different sources for RCA phono connectors, two of which have slightly different wiring configurations, it may become necessary to re-map the inputs. When using connectors manufactured by JALCO (white), the Mapping value should be set to 1. All other connectors (black ones as well as some varieties of white connectors), the mapping value should be set to 0. The purpose of the Mapping commands is to swap the inputs. When using JALCO phono connectors, inputs 1-8 are on the lower row instead of the intended top row. The Mapping commands will move inputs 1-8 to the top row and the bottom row will be reserved for inputs 9-16 (which is our intended operation). If at any time, you experience input one on the lower row, change the Map setting to move the inputs back to the top row of connectors. Mapping commands are non-volatile and will be kept in memory until changed, even if power is removed from the controller. Mapping commands only work with Version 4 firmware or later. If you do not have Version 4 firmware and you need to remap the inputs, use the VB code below titled “Mapping Without V4 Firmware”.

Mapping Commands

15 Set Mapping Value
Version 4.0+ firmware only
This command stores a MAP Value used to swap the Upper and Lower Rows of Inputs on the ASELPRO. This command has one Parameter with a Valid Value of 0 or 1. A value of 0 Sets the inputs to the standard map. A value of 1 swaps upper and lower eight rows of inputs most commonly used with JALCO connectors. This command works with Version 4.0 firmware and later ONLY.

16 Get Mapping Value
Version 4.0+ firmware only
This command reads the current Map setting and returns a 0 or a 1.

Version Command

99 Get Version
Version 4.0+ firmware only
This command reads the current version of firmware and returns a value of 4 or greater depending on the current version of firmware installed. Version 5 firmware supports improved switching efficiency by reducing the number of bytes required for switching. Version 5 (upon release) will support RF wireless control of this device.

Future Upgrade Plans

One issue that remains to be addressed in the ASELPRO/8S2SPRO is a minor bug whereby this device may not report ASCII character 0. We are researching the cause of this problem and hope to offer an upgrade for this in version 5 firmware. This problem only applies to reading E3C device number 0 or reading the MAP setting when set to 0, the device may not reply with any data. This does not affect the reading of input channels, these functions report back values from 1 to 16.

We also hope to have RF communications implemented in Version 5 firmware. At the time of writing, we are developing our first wireless communication processors that allow you to control up to 1.7 Million NCD devices under wireless control. Full wireless implementation protocols will likely be introduced in version 5 firmware.

Sample Code: Mapping Commands

```
Public Sub MapNormal                    'Map Connectors for Most RCAs
    MSComm1.Output = Chr$(254)          'Enter Command Mode
    MSComm1.Output = Chr$(15)           'Map Command
    MSComm1.Output = Chr$(0)            'Set to Standard Map
End Sub

Public Sub MapJALCOSwap                 'Map Connectors for JALCO RCAs
    MSComm1.Output = Chr$(254)          'Enter Command Mode
    MSComm1.Output = Chr$(15)           'Map Command
    MSComm1.Output = Chr$(1)            'Swap Upper and Lower Inputs
End Sub

Public Sub MapNormal                    'Get Map Settings
    MSComm1.Output = Chr$(254)          'Enter Command Mode
    MSComm1.Output = Chr$(16)           'Get Map Settings Command
    GetI                        'Call GetI Function to Read Device Data
End Sub

Public Sub GetVersion                   'Get Version of Firmware
    MSComm1.Output = Chr$(254)          'Enter Command Mode
    MSComm1.Output = Chr$(99)           'Get Firmware Version Command
    GetI                        'Call GetI Function to Read Device Data
End Sub
```

Sample Code: Reading Data from Controller

```
Public Function GetI
    Do                                  'Wait for Device to Reply
        DoEvents                        'Allow Windows to Multitask
    Until MSComm1.InBufferCount > 0     'If the Device Replies
    GetI = Asc(MSComm1.Input)           'Get Status from Serial Buffer
    Debug.Print GetI                    'Display in Immediate Window
End Sub
```

Sample Code: Mapping Without V4 Firmware

```
Use the following routine to Swap the Upper and Lower set of Inputs
Without Upgrading the ASELPRO firmware.  We only shipped approximately
70 boards where this section of the manual will apply.

A Normal Switching routine looks like this:

Public Sub ChannelA(In)                 'Channel Select for Output A
    MSComm1.Output = Chr$(254)          'Enter Command Mode
    MSComm1.Output = Chr$(1)            'Set Channel A Command
    MSComm1.Output = Chr$(In)           'to Input (1-16)
End Sub

Public Sub ChannelB(In)                 'Channel Select for Output B
    MSComm1.Output = Chr$(254)          'Enter Command Mode
    MSComm1.Output = Chr$(2)            'Set Channel B Command
    MSComm1.Output = Chr$(In)           'to Input (1-16)
End Sub

It may be necessary to swap the Upper and Lower set of connectors to
ensure Input 1 is on the top row (this only applies to boards that
have shipped that use white JALCO RCA phono jacks).  In this case, use
these commands for switching:

Public Sub ChannelA(In)                 'Channel Select for Output A
    If In < 9 then
       MSComm1.Output = Chr$(254)       'Enter Command Mode
       MSComm1.Output = Chr$(1)         'Set Channel A Command
       MSComm1.Output = Chr$(In+8)      'Move Inputs to Upper Row
    else
       MSComm1.Output = Chr$(254)       'Enter Command Mode
       MSComm1.Output = Chr$(1)         'Set Channel A Command
       MSComm1.Output = Chr$(In-8)      'Move Inputs to Lower Row
    endif
End Sub

Public Sub ChannelA(In)                 'Channel Select for Output B
    If In < 9 then
       MSComm1.Output = Chr$(254)       'Enter Command Mode
       MSComm1.Output = Chr$(2)         'Set Channel B Command
       MSComm1.Output = Chr$(In+8)      'Move Inputs to Upper Row
    else
       MSComm1.Output = Chr$(254)       'Enter Command Mode
       MSComm1.Output = Chr$(2)         'Set Channel B Command
       MSComm1.Output = Chr$(In-8)      'Move Inputs to Lower Row
    endif
End Sub
```

Electrical Ratings and Characteristics

Characteristics Data	Minimum	Maximum
Voltage DC Semi-Clean Power Supply Recomended	8.0 Volts DC	18 Volts DC
Current Consumption	Aprox. 100 ma	Aprox. 200 ma
Input to Output On Resistance	110 Ohms	122 Ohms
Transition time	110 ns	400 ns
Break Before Make Interval	10 ns	40 ns
Off Isolation	-69 db	
Crosstalk	-92 db	
Analog Signal Range	-5VDC	+5VDC
Operating Temperature	0° C	70° C
Switches per Second at 2400 Baud (Min = 4 Byte Commands, Max = 3 Byte Commands)	60	80
Switches per Second at 9600 Baud (Min = 4 Byte Commands, Max = 3 Byte Commands)	240	320
Switches per Second at 19200 Baud (Min = 4 Byte Commands, Max = 3 Byte Commands)	480	640
Switches per Second at 38400 Baud (Min = 4 Byte Commands, Max = 3 Byte Commands)	960	1,280
Communication Distance from PC Without Boosting Signal 1200 Baud*	N/A	Aprox. 2000 Feet
Communication Distance from PC Without Boosting Signal 2400 Baud*	N/A	Aprox. 1200 Feet
Communication Distance from PC Without Boosting Signal 9600 Baud*	N/A	Aprox. 600 Feet
Communication Distance from PC Without Boosting Signal 19.2K Baud*	N/A	Aprox. 200 Feet
Communication Distance from PC Without Boosting Signal 38.4K Baud*	N/A	Aprox. 50 Feet

* assumes good quality low-capacitive wire, twisted pair preferred. Note that distances are estimated. Typically, longer distances are easily achieved.

IMPORTANT NOTE:

A DETAILED MECHANICAL DRAWING CAN BE FOUND ON THE FOLLOWING PAGE OF OUR WEB SITE:
http://www.controlanything.com/photos/ASELDIM.gif