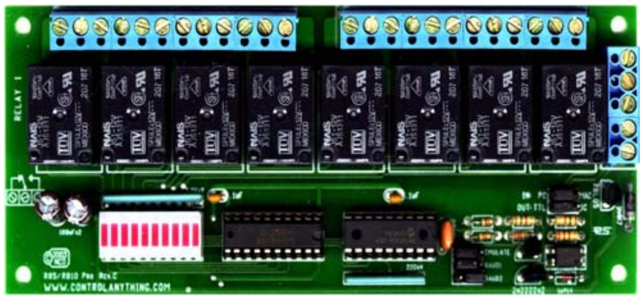
R4x/R8x Pro/DPDT

RS-232 E3C Networkable Relay Controllers

5-Year Repair or Replace Warranty!!!



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
Diode Clamped Relay Driver Stage
Relay Status LEDs
Device Enabled/Power LED
Data Receive LED
12 Volt DC Operation
Standardized Mounting Holes
16 Memory Banks for Storing/Recalling the Status of ALL Relays
User-Programmable Startup Status
Simultaneously Set the Status All Relays
Ask the Status of Individual or All Relays
Optoisolated 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

Our R4x or R8x series relay controllers have made their way into thousands of computer control applications all over the world. With the release of more sophisticated microcontrollers, it is now possible to expand the features of our original designs. The R8x and R4x Professional series relay controllers were designed to take advantage of recent advances in microcontroller technology. Our new Professional series relay controllers offer an enormously powerful new command set, E3C compliance, and the ability to emulate the original R4x and R8x commands. In addition, a variety of new products will take advantage of our new microprocessors, adding a large variety of choices to our 4 and 8-relay controller designs. This guide will show you how to take advantage of the extensive new features offered by our firmware. This guide is written to generically cover all R4x Pro and R8x Pro series relay controllers.

Device Variations

This manual covers all NCD products with part numbers that begin “R4” and end “Pro” or “DPDT”.
This manual covers all NCD products with part numbers that begin “R8” and end “Pro” or “DPDT”.

R4x/R8x vs. R4x/R8x Pro: *Feature Differences*

	Original R4x/R8x Relay Controllers	NEW R4x/R8x Pro Relay Controllers
Optoisolation on RS-232 Data Inputs	Yes	Yes
R4x Pro Form Factor Compatible with Original R4x Design	-	R45 Pro is Smaller
R8x Pro Form Factor Compatible with Original R8x Design	-	Yes
Set Status of Individual Relays	Yes	Yes
Set Status of Multiple Relays Simultaneously	No	Yes
Read Status of Individual Relays	No	Yes
Read Status of Multiple Relays Simultaneously	Yes	Yes
16 Device Network Compliance	Yes	Yes, Emulation Mode
E3C 256 Device Network Compliance	No	Yes
Programmable Relay Timer	No	Yes
Programmable Device Number	Jumper Configured	Software Configured
Programmable Power-up Relay Status	No	Yes
Programmable Relay Status Memory Banks	No	Yes
Multi-Parameter Command Structure	No	Yes
Relay Select/Deselect Command Set	No	Yes
Extended Relay Control Command Set	No	Yes
Integrated User-Programmable Memory	No	Yes
20 MHz CPU Operation	No	Yes

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

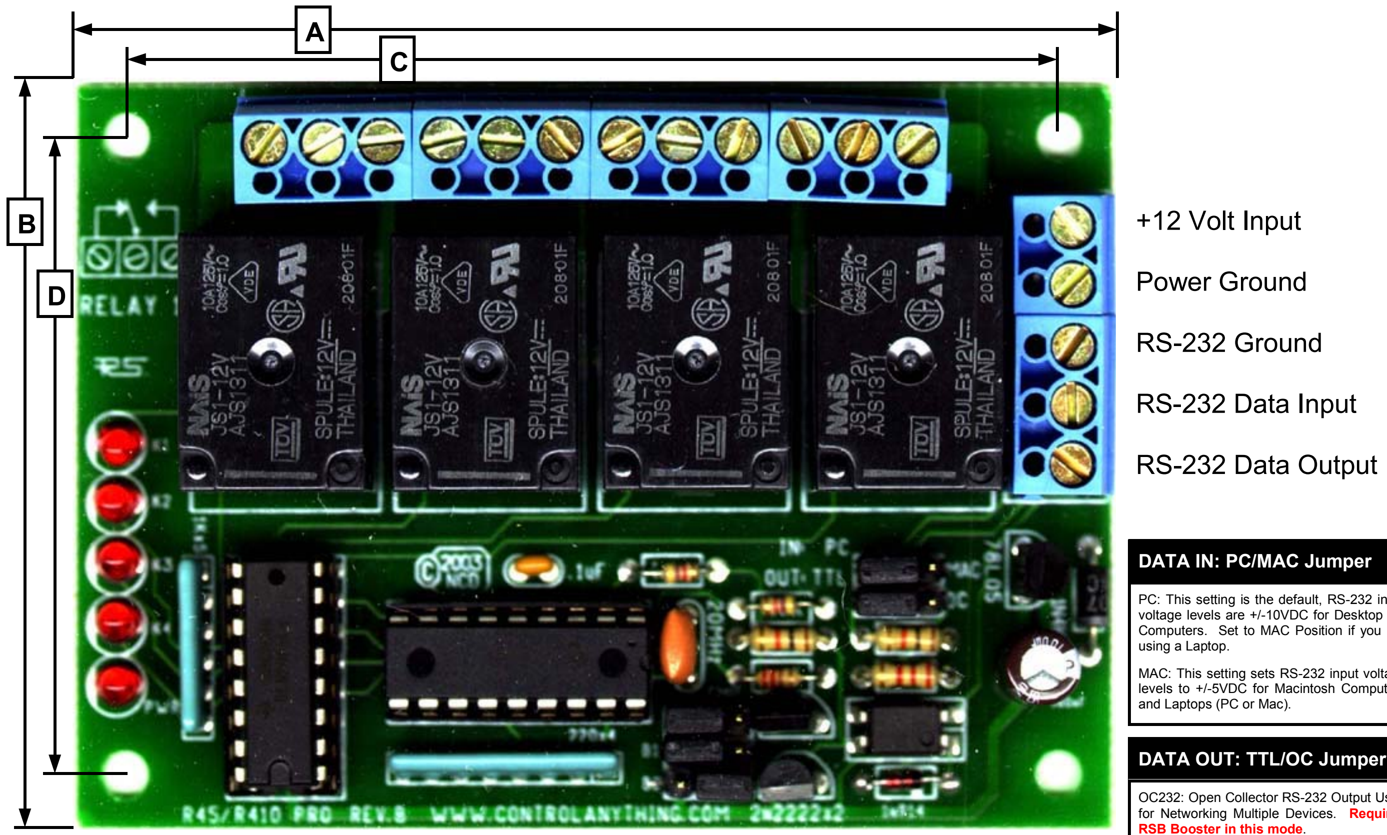
IMPORTANT POWER SUPPLY REQUIREMENTS

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 WILL SHORTEN THE COIL LIFE. LOWER VOLTAGES MAY CAUSE UNRELIABLE OPERATION, BUT WILL NOT DAMAGE THE CONTROLLER.
6) THE R32 CAN BE USED IN 12 VOLT AUTOMOTIVE ELECTICAL SYSTEMS.

Status LEDs:

Heartbeat LED:
The heartbeat LED has been omitted in the new “Pro” series R4x/R8x relay controllers.

Power LEDs:
The on-board Power LED is driven by the CPU and is software controlled using the E3C command set. When power is first applied to the board, the power LED will glow, indicating that the CPU has booted and it is ready to accept commands. If an E3C command is received to disable the device, the power LED will turn off. The LED will only come back on if the device is re-enabled using the E3C command set OR if power is removed and re-applied to the board.



DATA IN: PC/MAC Jumper

PC: This setting is the default, RS-232 input voltage levels are +/-10VDC for Desktop PC Computers. Set to MAC Position if you are using a Laptop.

MAC: This setting sets RS-232 input voltage levels to +/-5VDC for Macintosh Computers and Laptops (PC or Mac).

DATA OUT: TTL/OC Jumper

OC232: Open Collector RS-232 Output Used for Networking Multiple Devices. **Requires RSB Booster in this mode**.

TTL: Standard TTL Level RS-232 Data Output Mode. This is the default setting. This setting is NOT Compatible with Macintosh Computers unless the TR32 Quick Start Kit is Used.

Dimension Information

	A	B	C	D
R45/R410 Pro	3 9/16”	2 1/2”	3 1/16”	2 1/8”
R4xDPDT Pro	3 9/16”	2 5/8”	3 1/6”	2 1/4”
R420/R430 Pro	3 1/2”	3 3/8”	3 1/8”	3”
R85/R810 Pro	6 1/4”	2 1/2”	5 7/8”	2 1/8”
R8xDPDT Pro	5 5/8”	2 5/8”	5 1/4”	2 1/4”
R820/R830 Pro	6 1/8”	3 3/8”	5 3/4”	3”

Standard Hole Size is 1/8”.
Standard Inset of Mounting Holes from any edge of the board is 3/16” from edge to Center of Mounting Hole.
Standard Safe Mounting Hardware Area is 3/8” with Mounting Hole Located in Exact Center.
Board should be Mounted using Minimum 1/4” standoffs on the bottom.
The component side of the R45, R410, R85, R810 Pro requires 3/4” minimum clearance.
The component side of any Controller with DPDT in the part number requires 1 1/4” minimum clearance.
The component side of any Relay Controller with 20 or 30 Amp Relay requires 1 1/2” minimum clearance.

Emulation Jumper

Set between the LEFT two posts to take advantage of the advanced “Pro” command set. Set this jumper between the RIGHT two posts to “Emulate” the original R4x or R8x command set. Note: The device number defaults to 0 under emulation. To change the emulation device number, Set to “Pro” command set and issue the software command to program the emulation device number (explained under command 41 on Page 11).

Baud Jumpers B1 & B2

Please see the next page of this manual for a complete explanation of these jumper settings.

Did You Know?

The mechanical relays we have chosen for our controllers are rated to easily withstand ***several million continuous high-current switching cycles*** and have an operational life of more than 10 years. In many ways, today’s mechanical relays can outlast solid state relays because of their mechanical ability to handle high current surges. In addition, the mechanical relays we have chosen are hermetically sealed, greatly reducing wear on the contacts by eliminating internal sparking.

R81DPDT USERS: IMPORTANT NOTICE

The labeling on the PCB for the R8xDPDT Series Relay Controllers applies to the R83DPDT and R85DPDT ONLY. The relays used in the R81DPDT have a slightly different pinout. The Outside row of connectors (the row closest to the edge of the board) is the Common connection, the middle row is Normally Closed. The inside row is Normally Open as marked on the board.

Connecting Relays

NC:	Normally Closed—Connects to the COMMON Lead when relay is OFF.
NO:	Normally Open—Connects to the COMMON Lead only when the Relay is Turned ON.
COMMON:	Connects to the Normally Closed Lead when the relay is OFF. When relay is ON, Common Disconnects from the COMMON an Reconnects to the Normally Open Lead.

Relays are simple switches with 2, 3 or 6 leads. In some way, either the relay itself or the circuit board we manufacture has a diagram or label indicating how a relay is connected. Some relays, such as the 20 and 30 amp versions, are marked NC, COM, NO on the Top indicating Normally Closed, Common, and Normally Open. Relay controllers that require wiring to our boards are marked in the form of a small diagram. DPDT relays have two switches, completely isolated from each other, that switch at the same time.

It is important to understand that relays are just switches. Our relay controllers activate and deactivate these switches. Under no circumstances do our relay controllers provide power to any of the relay connectors. It is also important to understand that the switch side of the relay and the electronics portion of our controller is COMPLETELY isolated. As such, our relay controller requires a power supply to power the CPU and the driver circuit. A suitable power supply should provide 12 Volts DC, 400 ma for our quad relay controllers and 12 Volts DC 800 ma for our 8-relay controllers. Automotive power systems can be used to power our controllers with no additional circuitry.

Jumper Settings

Gray Indicates Default Setting

Jumper	Description	Nearest to Label on PCB	Furthest from Label on PCB
EMU or EMULATE	The EMULATE jumper is used to make the R4x/R8x Pro act as if it were there original R4x or R8x relay controller, responding to all the same commands as the original design, NEW COMMANDS ARE NOT AVAILABLE IN THIS MODE. THIS MODE DOES NOT SUPPORT E3C NETWORKING. Install a jumper closest to the EMU or EMULATE label to activate EMULATION.	Emulation Mode is Set ACTIVE, New Commands Not Available	▨ Standard E3C Mode with New Command Set
B1 or BAUD1	Baud Select Jumper 1	B1 is OFF when Set between the two posts closest to the PIC CPU.	▨ B1 is ON when Set between the two posts furthest from the PIC CPU.
B2 or BAUD2	Baud Select Jumper 2	▨ B2 is OFF when Set between the two posts closest to the PIC CPU.	B2 is ON when Set between the two posts furthest from the PIC CPU.
PC/MAC	Input Data Voltage Select, PC +/-12V RS232, MAC +/5V RS422 specification. Use MAC mode for Mac systems, PC Laptops, and TTL data Sources. It is OK to try both setting if unsure for your application. No damage will result due to improper setting of this jumper.	▨ Nearest to PC Label: Desktop PC Data Source	Nearest to MAC Label: Laptop PC, MAC, or TTL Data Source
TTL/OC	Output Data Mode, TTL 0/+5V TTL Data Output compatible with PCs and LAPTOP computers. Use TRUE32 converter (sold separately) for MAC systems. This jumper is ONLY necessary if using 2-way communication. O.C. Output Mode means Open Collector, use with RSB Serial Booster (sold separately) when using this mode. TLL mode is typically used for Laptop, Desktop PCs, and Embedded computers. No damage will result due to improper setting of this jumper.	▨ Nearest to TTL Label: Data Output compatible with Desktop PCs and Embedded Communications.	Nearest to OC Label: Data Output is electrically an Open Collector Signal. Use with NCD RSB Serial Booster ONLY.

Setting Jumpers

1) Jumpers should be set PRIOR to applying power to the board. Jumper setting will have no effect if changed while power is on.
2) Each jumper option has 3 pins. For example, the TTL/OC jumper has three pins. Installing a jumper between the Left two pins sets operation to TTL mode. Installing a jumper between the Right two pins sets operation to OC mode. A jumper MUST be installed between TWO of the THREE pins. NEVER remove the jumper entirely from the board. Right and Left are determined by setting the board in front of you in such a manner that all text is normally readable from the left to the right side. For the purposes of setting the Baud rate, OFF is on the LEFT side of the board, ON is on the RIGHT side of the board.

B1 B2: Baud Select

B1	B2	Baud
OFF	OFF	1200
ON	OFF	2400
▨ OFF	▨ ON	▨ 9600
ON	ON	19.2K

Factory Default Settings:

Default Setting are Highlighted in Gray Above.

Status LEDs:

Power/E3C Enabled LED:
This LED is normally lit when power is first applied to the board. This LED is controlled by the E3C command set. When lit, the R4x/R8x Pro is enabled and ready to accept relay control commands. When this LED is off, the only commands that will be processed are E3C commands. The R4x/R8x Pro will not process your relay control commands again until this LED is lit. Use the E3C command set to enable the R4x/R8x Pro.

Dat In:
This LED is normally off and flashes as data is received by the relay controller.

Communication Parameters:

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

Two-Way Communication:

The R4x/R8x Pro support two-way communication for confirming the receipt of commands and for reporting the status of the relays back to the host computer.

The R4x/R8x Pro should be connected as shown below when using this device for the first time. Even if you plan to connect several R4x/R8x Pro controllers to a single serial port, this wiring diagram must first be used to program the device number into the controller.

The R4x/R8x Pro Visual Basic Example Program expects this wiring configuration.

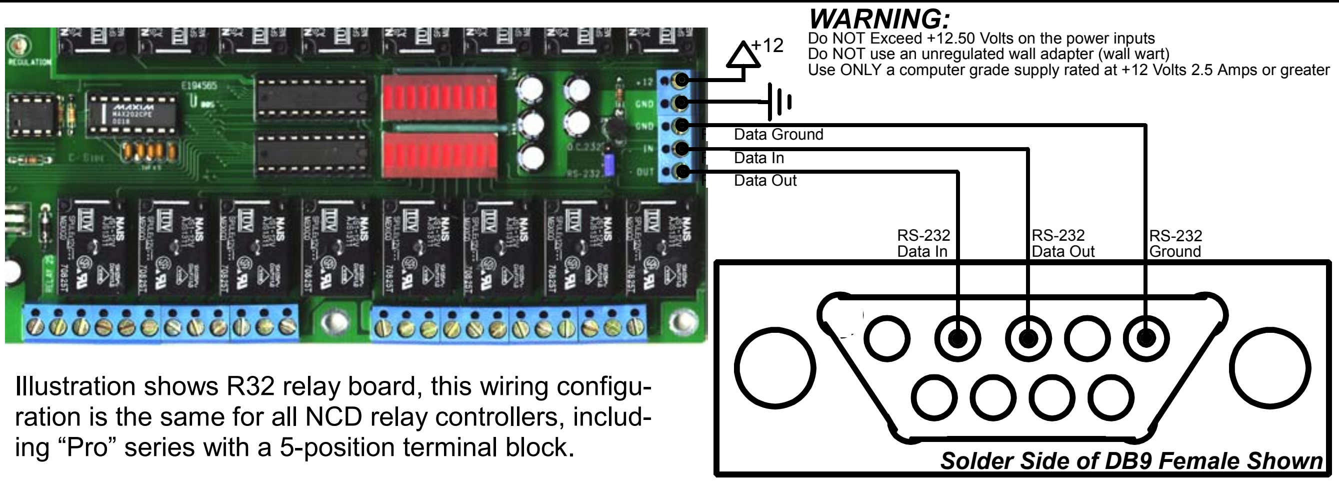

WARNING:
Do NOT Exceed +12.50 Volts on the power inputs
Do NOT use an unregulated wall adapter (wall wart)
Use ONLY a computer grade supply rated at +12 Volts 2.5 Amps or greater
+12
Data Ground
Data In
Data Out
RS-232 Data In
RS-232 Data Out
RS-232 Ground
Solder Side of DB9 Female Shown

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.

R4x/R8x Pro One-Way Communication:

The R4x/R8x Pro 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.

When used in 1-way mode, reporting should be turned off for highest communication speed. Turning off reporting will allow you to send commands to the R4x/R8x Pro much faster, but it is impossible to ask the controller for the status of relays when wired as shown below.

Reporting Mode is activated by sending ASCII character codes 254, 27.

Reporting Mode is deactivated by sending ASCII character codes 254, 28.

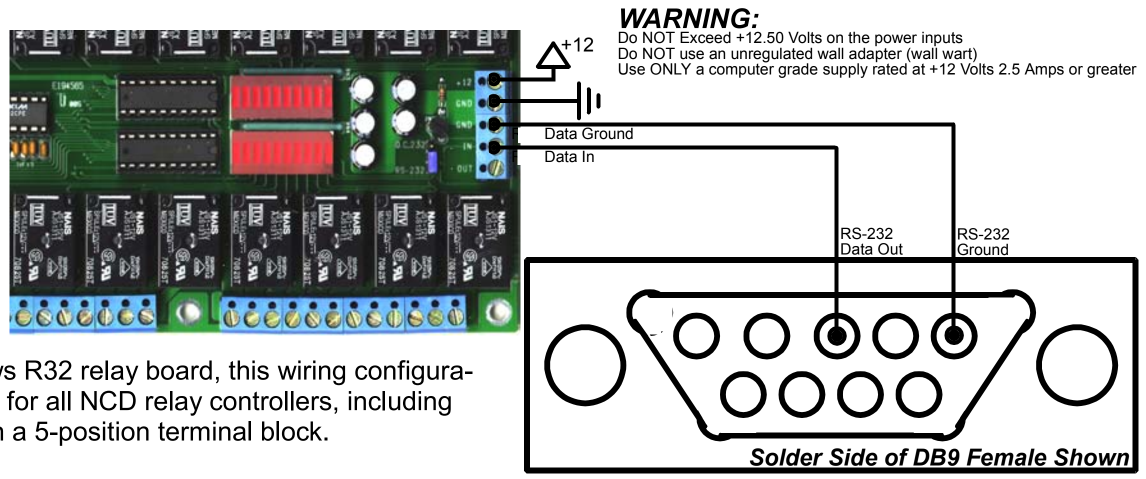

WARNING:
Do NOT Exceed +12.50 Volts on the power inputs
Do NOT use an unregulated wall adapter (wall wart)
Use ONLY a computer grade supply rated at +12 Volts 2.5 Amps or greater
+12
Data Ground
Data In
RS-232 Data Out
RS-232 Ground
Solder Side of DB9 Female Shown

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.

Multiple Relay Controllers: 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.

Before using this wiring configuration, each device must be programmed with a unique device number (See E3C Commands in Both Manuals for 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:

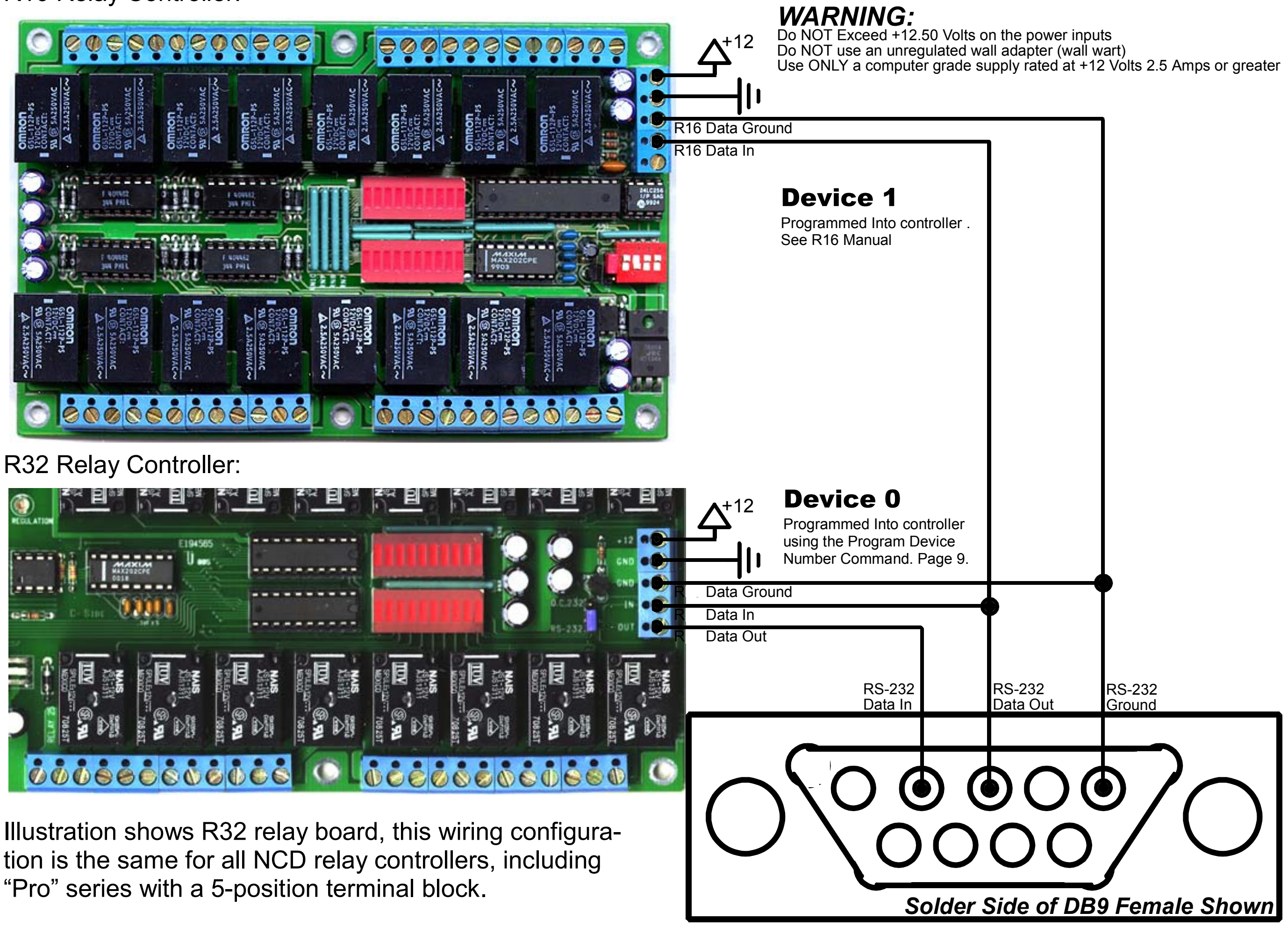

WARNING:
Do NOT Exceed +12.50 Volts on the power inputs
Do NOT use an unregulated wall adapter (wall wart)
Use ONLY a computer grade supply rated at +12 Volts 2.5 Amps or greater
+12
R16 Data Ground
R16 Data In
Device 1
Programmed Into controller . See R16 Manual
R32 Relay Controller:
+12
Device 0
Programmed Into controller using the Program Device Number Command. Page 9.
R Data Ground
R Data In
R Data Out
RS-232 Data In
RS-232 Data Out
RS-232 Ground
Solder Side of DB9 Female Shown

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.

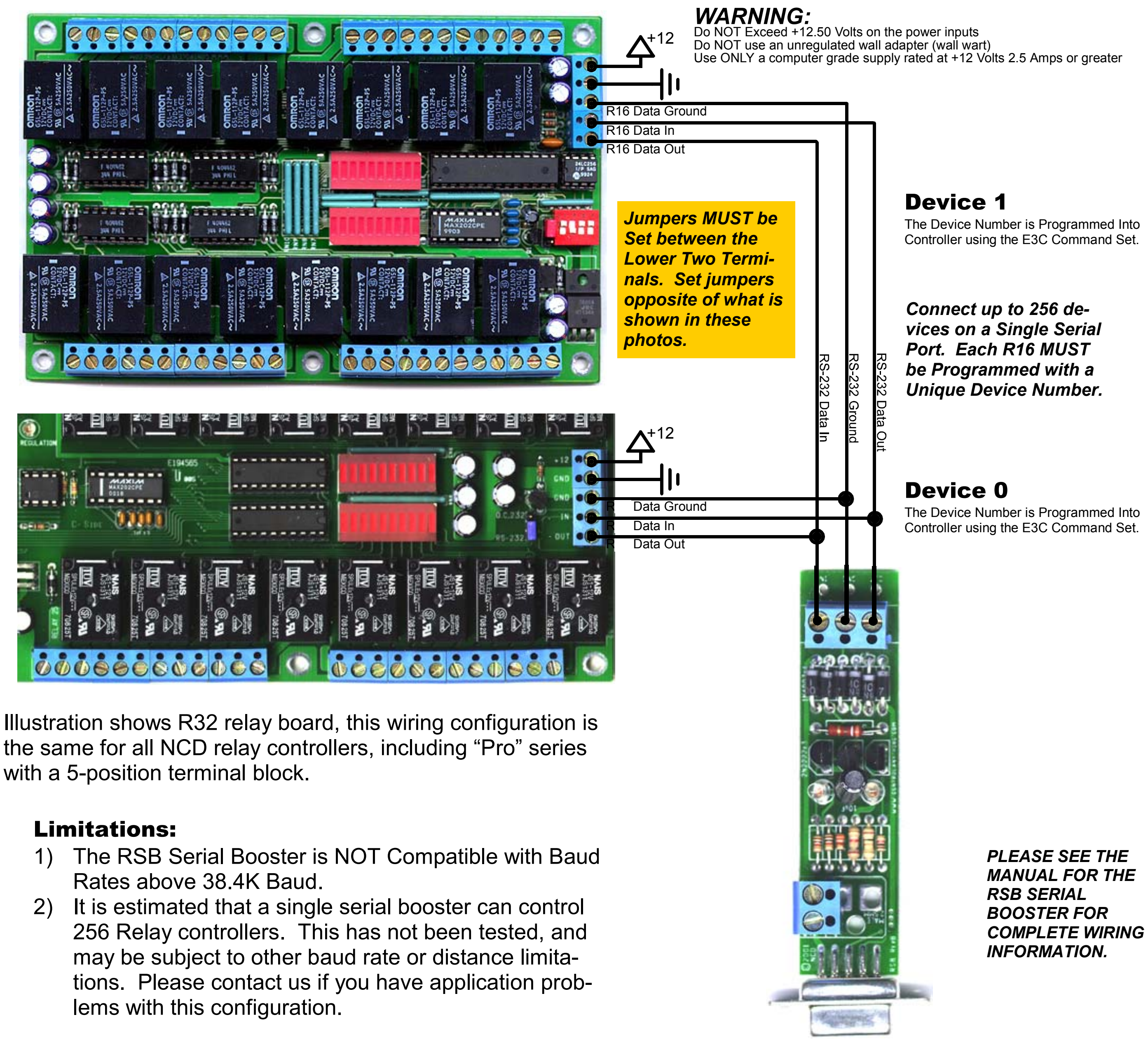
Multiple Relay Controllers: Two-Way Communication

Our relay controllers support two-way communication to multiple devices using the RSB serial booster. Jumpers must be set for Open Collector data transmission. See the appropriate manual for your relay 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.*



WARNING:
Do NOT Exceed +12.50 Volts on the power inputs
Do NOT use an unregulated wall adapter (wall wart)
Use ONLY a computer grade supply rated at +12 Volts 2.5 Amps or greater

Jumpers MUST be Set between the Lower Two Terminals. Set jumpers opposite of what is shown in these photos.

Device 1
The Device Number is Programmed Into Controller using the E3C Command Set.

Connect up to 256 devices on a Single Serial Port. Each R16 MUST be Programmed with a Unique Device Number.

Device 0
The Device Number is Programmed Into Controller using the E3C Command Set.

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.

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

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.

Sending Commands to the R4x/R8x Pro Relay Controllers

The R4x/R8x Pro is capable of sending and receiving data via RS-232 serial communications. The R4x/R8x Pro 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 R4x/R8x Pro. 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 the R4x/R8x Pro for the current status of the relays, just to confirm their activation. 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 for controlling the R4x/R8x Pro. These examples should greatly speed development time. You may want to visit **www.controleverything.com** for the latest software and programming examples.

Programming examples for the R4x/R8x Pro 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 R4x/R8x Pro relay controllers 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. The R4x/R8x Pro must be programmed with a device number, which is accomplished 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

The R4x/R8x Pro 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.

Most commands issued to the R4x/R8x Pro are acknowledged by sending ASCII character code 85 back to the host computer (when reporting is turned on). E3C commands are not acknowledged regardless of the reporting mode.

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 R4x/R8x Pro Command Set

The R4x/R8x supports an extensive command set, used to control relays, set operation modes, and store and recall relay status. Most users will not use many of the functions built into this controller. The best way to familiarize yourself with the capabilities 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.

Note the command set for the R4x Pro is very similar to the command set for the R8x Pro. R8x Pro commands issued to the R4x will be ignored, allowing for easy future upgrade to the R8x Pro.

Controlling Individual Relays

0-7 Turing Off Individual Relays

8-15 Turing On Individual Relays

Reading the Status of Relays

16-23 Get the Status of an Individual Relay
This command allows you to read the on/off status of an individual relay. 16 corresponds to relay 1, 23 corresponds to relay 8. This command will return a 1 indicating the relay is ON or a 0 indicating the relay is OFF.

24 Get the Status of All Relays
This command allows you to get the status all relays at one time. A value of 0-255 is returned indicating the status of all 8 relays from the R8x Pro. A value of 0-15 is returned from the R4x Pro. The binary pattern of the value returned directly corresponds to the on/off status of each relay.

Power-Up Default Relay Pattern

25 Store Relay Pattern as Power-Up Default
This command allows you to define the on/off status of all relays when power is first applied to the board. Use other commands to set the relays in the desired power-up state, then issue this command to store the current status of the relays as the power-up default.

26 Get the Power-Up Default Relay Pattern
This command allows you read the stored power-up default relay pattern. R4x Pro boards return a value of 0-15, R8x Pro board return a value of 0-255. The binary pattern of the value returned directly corresponds to the on/off status of each relay.

Reporting Mode

27 Turn Reporting Mode ON
By default, Reporting Mode is off. Reporting mode confirms the completion of most commands by sending an ASCII character code 85 back to the user. Reporting mode does NOT confirm the completion of E3C commands, or any command that sends data back to the user. Reporting mode is stored in non-volatile memory. Reporting mode slows communications and is only recommended in high-reliability installations.

28 Turn Reporting Mode OFF
This command turns off the reporting function. Reporting mode is stored in non-volatile memory. This is the default setting.

All On/Off

29 Turns All Relays OFF

30 Turns All Relays ON

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 R4x/R8x Pro relay controllers. Additional source code can be found on our web site at www.controleverything.com.

Sample Code: Controlling Individual Relays

```
Public Sub SetRelayStatus(Relay,Stat)   'Relay Parameter = 1 to 8
                                        'Stat Parameter = 1 or 9
    If Stat = 0                         'Turn Off Relay
       MSComm1.Output = Chr$(254)       'Enter Command Mode
       MSComm1.Output = Chr$(Relay-1)   'Relay to Turn Off
    Else                                'Turn On Relay
       MSComm1.Output = Chr$(254)       'Enter Command Mode
       MSComm1.Output = Chr$(Relay+7)   'Relay to Turn On
    Endif
End Sub
```

Sample Code: Reading Status of Relays

```
Public Function GetRelayStatus(Relay)   'Relay Parameter = 1 to 8
    MSComm1.Output = Chr$(254)          'Enter Command Mode
    MSComm1.Output = Chr$(Relay+15)     'Get Status of One Relay
    Do                                  'Wait for Device to Reply
        DoEvents                        'Allow Windows to Multitask
    Until MSComm1.InBufferCount > 0     'If the Device Replies
    GetRelayStatus = Asc(MSComm1.Input) 'Get Status from Serial Buffer
    Debug.Print GetRelayStatus          'Display in Immediate Window
End Sub

Public Function GetAllRelayStatus(Relay)
    MSComm1.Output = Chr$(254)          'Enter Command Mode
    MSComm1.Output = Chr$(24)           'Get Status of all Relays
    Do                                  'Wait for Device to Reply
        DoEvents                        'Allow Windows to Multitask
    Until MSComm1.InBufferCount > 0     'If the Device Replies
    GetAllRelayStat = Asc(MSComm1.Input)'Get Status from Serial Buffer
    Debug.Print GetAllRelayStat         'Display in Immediate Window
End Sub
```

Sample Code: Power-Up Relay Pattern

```
Public Sub StoreDefault
    MSComm1.Output = Chr$(254)          'Enter Command Mode
    MSComm1.Output = Chr$(25)           'Store Powerup Default Status
End Sub

Public Function GetDefaultStatus
    MSComm1.Output = Chr$(254)          'Enter Command Mode
    MSComm1.Output = Chr$(26)           'Get Status of all Relays
    Do                                  'Wait for Device to Reply
        DoEvents                        'Allow Windows to Multitask
    Until MSComm1.InBufferCount > 0     'If the Device Replies
    GetDefaultStatus=Asc(MSComm1.Input) 'Get Status from Serial Buffer
    Debug.Print GetDefaultStatus        'Display in Immediate Window
End Sub
```

Sample Code: Reporting Mode

```
Public Sub ReportingOn
    MSComm1.Output = Chr$(254)          'Enter Command Mode
    MSComm1.Output = Chr$(27)           'Turn On Reporting Mode
End Sub

Public Sub ReportingOff
    MSComm1.Output = Chr$(254)          'Enter Command Mode
    MSComm1.Output = Chr$(28)           'Turn Off Reporting Mode
End Sub
```

Sample Code: All On/Off

```
Public Sub AllRelaysOff
    MSComm1.Output = Chr$(254)          'Enter Command Mode
    MSComm1.Output = Chr$(29)           'Turn Off All Relays
End Sub

Public Sub ReportingOn
    MSComm1.Output = Chr$(254)          'Enter Command Mode
    MSComm1.Output = Chr$(30)           'Turn On All Relays
End Sub
```

The R4x/R8x Pro Command Set

Relay Pattern Inversion and Reversal

31 Invert All Relays
All relays that are currently off turn on, all relays that were on turn off.

32 Reverse Relay Order
The Status of Relays 12345678 are reversed to 87654321. This command does not permanently reassign relays, it only copies the status of the relays when executed.

Sample Code: Relay Inversion and Reversal

```
Public Sub InvertAllRelays
    MSComm1.Output = Chr$(254)          'Enter Command Mode
    MSComm1.Output = Chr$(31)           'Invert All Relays Command
End Sub

Public Sub ReverseOrder
    MSComm1.Output = Chr$(254)          'Enter Command Mode
    MSComm1.Output = Chr$(32)           'Reverse Relay Order Command
End Sub
```

Testing 2-Way Communication

33 Test 2-Way Communication
This command can be used to test 2-way communication between the host computer and the relay controller. When executed, the relay controller will send ASCII character code 85 back to the user. This command should be used for initial installations if 2-way communication is required. It can also be used to detect the presence of a relay controller on the serial port.

Sample Code: Test 2-Way Communication

```
Public Function Test2Way
    MSComm1.Output = Chr$(254)          'Enter Command Mode
    MSComm1.Output = Chr$(33)           'Request 2-way Comm. Test
    Do                                  'Wait for Device to Reply
        DoEvents                        'Allow Windows to Multitask
    Until MSComm1.InBufferCount > 0     'If the Device Replies
    Test2Way = Asc(MSComm1.Input)       'Get Status from Relay Board
    Debug.Print Test2Way                'Display in Immediate Window
End Sub
```

Commands with Parameters: Set Status of All Relays

40,0-15 Set Status of All Relays (R4x Pro)
40,0-255 Set Status of All Relays (R8x Pro)
This command is used to set the status of all relays at one time. A single parameter is required. The equivalent binary pattern of the parameter is copied directly to the relays, instantly setting the on/off status of all relays on the board.

Sample Code: Set Status of All Relays

```
Public Sub SetAllRelays(Relay)
    MSComm1.Output = Chr$(254)          'Enter Command Mode
    MSComm1.Output = Chr$(40)           'Set All Relay Status Command
    MSComm1.Output = Chr$(Relay)        'Pattern to Set Relays To
End Sub
```

Program Emulation Device Number

41,0-15 Program Emulation Device Number
When the R4x/R8x Pro is NOT in emulation mode, this command can be used, along with its parameter of 0-15, to define the device number for use in emulation mode. Once programmed, you must remove power from the board and set the board to Emulation mode.

Sample Code: Set Emulation Device Number

```
Public Sub ProgramEMUDevNum(Device)
    MSComm1.Output = Chr$(254)          'Enter Command Mode
    MSComm1.Output = Chr$(41)           'Program Emu Device Number
    MSComm1.Output = Chr$(Device)       'Device Number to Program
End Sub
```

Relay Pattern Banks

42,0-15 Store Relay Pattern in Memory Bank
This command stores the current on/off setting of all relays into a memory bank (0-15). This command is useful for creating macros or for making sure certain relays are never activated simultaneously.

43,0-15 Recall Relay Pattern from Memory Bank
This command recalls a stored relay pattern from the user selected memory bank (0-15) and update all relays on the board to the settings defined by command 42 above.

Sample Code: Memory Storage Functions

```
Public Sub StorePatterninBank(Bank)
    MSComm1.Output = Chr$(254)          'Enter Command Mode
    MSComm1.Output = Chr$(42)           'Store Pattern in Bank
    MSComm1.Output = Chr$(Bank)         'Mem. Bank to Store Pattern In
End Sub

Public Sub RecallPatterninBank(Bank)
    MSComm1.Output = Chr$(254)          'Enter Command Mode
    MSComm1.Output = Chr$(43)           'Recall Pattern from Bank
    MSComm1.Output = Chr$(Bank)         'Mem. Bank to Get Pattern From
End Sub
```

Relay Select and De-Select

44,0-3 Select a Relay for Activation (R4x Pro)
44,0-7 Select a Relay for Activation (R8x Pro)
This command turns off all relays and then turns on the selected relay only. This command performs a safe “Break Before Make”, ensuring that no two relays are ever activated at the same time.

45,0-3 Select a Relay for De-Activation (R4x Pro)
45,0-7 Select a Relay for De-Activation (R8x Pro)
This command turns on all relays and then turns off the selected relay only. This command performs a safe “Make Before Brake”, ensuring that no two relays are ever de-activated at the same time.

Sample Code: Relay Select/De-Select

```
Public Sub RelaySelect(Relay)
    MSComm1.Output = Chr$(254)          'Enter Command Mode
    MSComm1.Output = Chr$(44)           'Select Relay Command
    MSComm1.Output = Chr$(Relay)        'Relay to Select
End Sub

Public Sub RelayDeselect(Relay)
    MSComm1.Output = Chr$(254)          'Enter Command Mode
    MSComm1.Output = Chr$(44)           'Deselect Relay Command
    MSComm1.Output = Chr$(Relay)        'Relay to Deselect
End Sub
```

The R4x/R8x Pro Command Set

Toggle Relay

46,0-3 Toggle the Status of a Relay (R4x Pro)
46,0-7 Toggle the Status of a Relay (R8x Pro)
This command reverses the current on/off status of the selected relay.

Relay Timing Functions

The R4x/R8x Pro Series Relay controllers have a timer function used to activate one or more relays for a user specified period of time from 10 Milliseconds to 32 Seconds. Timing is accurate to within 5% of the user-specified period. The timer functions require a ***Time*** parameter in the commands shown below. Use the following guide to determine the appropriate value for the ***Time*** parameter:

The Time Variable sets 3 functions:

Feedback:	0=Off 128=On, Sends 85 to Host when Timer is Finished
Duration Interval:	0=(10 milliseconds x Duration) + 10 64=(.5 seconds x Duration) + .5
Duration:	0 to 63

To use the different modes of the timer, simply add together the values for each parameter. Feed the total into the TIME variable above. Then select the relay to apply the timer to.

Examples:

Time=0	10 Millisecond Timer with No Feedback
Time=4	50 Millisecond Timer with No Feedback
Time=132	50 Millisecond Timer with Feedback
Time=192	.5 Second Timer with Feedback
Time=73	5 Second Timer with No Feedback
Time=201	5 Second Timer with Feedback

Activate a Single Relay on a Timer

47, Time(0-255), Relay(0-3) Set Relay Timer (R4x Pro)
47, Time(0-255), Relay(0-7) Set Relay Timer (R8x Pro)
This command is used to activate a relay for a user-defined period of time. All other relays will remain unchanged. If the selected relay is already on, this function will have no effect, so make sure the relay is off before using this command. This command will send ASCII character code 85 back to the host computer if the timing function is enabled by the ***Time*** parameter.

Relay Pattern Select on a Timer

48, Time(0-255), RPOn(0-15), RPOff(0-15) - R4x Pro
48, Time(0-255), RPOn(0-255), RPOff(0-255) - R8x Pro
This command is used set the status of all relays (RPOn), apply a timer (Time 0-255), and then set all Relays to a new state once the timer has completed (RPOff). This command will send ASCII character code 85 back to the host computer if the timing function is enabled by the ***Time*** parameter.

NOTE: Timer Uses ALL CPU Resources

Commands cannot be sent to the relay controller while the timer is in operation. Any commands received during this period of time will be ignored. Use the feedback function (which is part of the <u>Time</u> parameter) to signal the host when the timer has completed its cycle or use the Test 2-Way command to query the relay controller.

Sample Code: Toggle Relay

```
Public Sub ToggleRelay(Relay)
    MSComm1.Output = Chr$(254)          'Enter Command Mode
    MSComm1.Output = Chr$(46)           'Toggle Relay Command
    MSComm1.Output = Chr$(Relay)        'Relay to Toggle
End Sub
```

Sample Code: Relay Timing Functions

```
Public Sub SetRelayTimer(Tymer,Relay)
    MSComm1.Output = Chr$(254)          'Enter Command Mode
    MSComm1.Output = Chr$(47)           'Set Timer for a Relay Command
    MSComm1.Output = Chr$(Tymer)        'Specify Time Period
    MSComm1.Output = Chr$(Relay)        'Relay to Activate
    'Debug.print GetFeedback            'Optional if Feedback is Used
End Sub

Public Sub SetMultiRelayTimer(Tymer,RPOn,RPOff)
    MSComm1.Output = Chr$(254)          'Enter Command Mode
    MSComm1.Output = Chr$(48)           'Set Timer for All Relays
    MSComm1.Output = Chr$(Tymer)        'Specify Time Period
    MSComm1.Output = Chr$(RPOn)         'Timer Start Relay Pattern
    MSComm1.Output = Chr$(RPOff)        'Timer Stop Relay Pattern
    'Debug.print GetFeedback            'Optional if Feedback is Used
End Sub
```

If feedback is enabled, the relay controller will send ASCII character code 85 back to the host computer to indicate the completion of the timer. Call the function below after you have issued command 47 or 48 if the feedback function is enabled. This routine will capture the 85 generated by the relay board.

```
Public Function GetFeedback
    Do                                  'Wait for Device to Reply
        DoEvents                        'Allow Windows to Multitask
    Until MSComm1.InBufferCount > 0     'If the Device Replies
    Feedback = Asc(MSComm1.Input)       'Get Status from Relay Board
End Function
```

Troubleshooting

Nothing Happens when Relay Control Commands are Sent
The most common cause is simply incorrect connection of the relay controller to the serial port of your computer or improper COM settings in software. Keep in mind, this manual indicates the RS-232 data INPUT of the relay controller. This means you must connect the RS-232 Data Output of your computer to the RS-232 Data INPUT of the relay controller. Note that the Data Receive LED on this device is intelligent. It will ONLY light up if it receives a valid ASCII character code 254 (enter command mode) preceding each command. Send 254 constantly to the board to make the LED flash.

Unreliable or Unpredictable Operation
In nearly 100% of the cases we have seen, unreliable or unpredictable operation is caused by improper setting of the PC/MAC jumper. Make sure this jumper is set in the MAC position if you are using a laptop computer of any kind, a microcontroller, or an Apple Macintosh product. Set it in the PC setting if using an Desktop PC.

No 2-Way Communication
Two-Way communication can be compromised by an incorrect jumper setting and/or improper wiring. Make sure the TTL/OC jumper is set in the TTL position for any kind of PC system (desktop or laptop). Also, note that the R4x/R8x Pro relay controllers use a hybrid optoisolation design. For two-way communication to work properly, you MUST connect the RS-232 ground to the Power Supply ground of the board.

R81DPDT Users:
The labeling on the PCB for the R8xDPDT Series Relay Controllers applies to the R83DPDT and R85DPDT ONLY. The relays used in the R81DPDT have a slightly different pinout. The Outside row of connectors (the row closest to the edge of the board) is the Common connection, the middle row is Normally Closed. The inside row is Normally Open as marked on the board.

Still Having Problems?
Please call us at (417) 646-5644 between 9AM and 5PM central standard time. Or, e-mail us at ryan@controlanything.com. We typically respond to e-mail requests as soon as they are received, even during the evenings and on weekends.

Electrical Ratings and Switching Characteristics

Model	Ratings	Relay Type	Notes
R41DPDT	3A 120VAC / 3A 20VDC	DPDT	DPDT: Two Switches per Relay Not for use with Inductive Loads
R43DPDT	3A 250VAC / 3A 30VDC	DPDT	DPDT: Two Switches per Relay
R45DPDT	5A 250VAC / 5A 30VDC	DPDT	DPDT: Two Switches per Relay
R45 Pro	10A 250VAC / 5A 100VDC	SPDT	Relay Ratings for this Model are Absolute Maximum, Not for Sustained Constant Use. Divide all ratings by 2 for Constant Operation
R410 Pro	10A 250VAC / 8A 30VDC	SPDT	
R420	20A 240VAC / 20A 28VDC	SPDT	Common to Normally Closed Terminal is rated at 10A 240VAC / 10A 28VDC Flange Type Connections to Relay
R430	30A 250VAC / 20A 28VDC	SPST	SPST: Common and Normally Open Terminals Only Flange Type Connections to Relay
R81DPDT	3A 120VAC / 3A 20VDC	DPDT	DPDT: Two Switches per Relay Not for use with Inductive Loads **NOTICE: RELAY LABELING ON PCB APPLIES TO R83DPDT AND R85DPDT ONLY** **R81DPDT Labeling:** **OUTSIDE ROW = COMMON, MIDDLE ROW = N.C., INSIDE ROW = N.O.**
R83DPDT	3A 250VAC / 3A 30VDC	DPDT	DPDT: Two Switches per Relay
R85DPDT	5A 250VAC / 5A 30VDC	DPDT	DPDT: Two Switches per Relay
R85 Pro	10A 250VAC / 5A 100VDC	SPDT	Relay Ratings for this Model are Absolute Maximum, Not for Sustained Constant Use. Divide all ratings by 2 for Constant Operation
R810 Pro	10A 250VAC / 8A 30VDC	SPDT	
R820	20A 240VAC / 20A 28VDC	SPDT	Common to Normally Closed Terminal is rated at 10A 240VAC / 10A 28VDC Flange Type Connections to Relay
R830	30A 250VAC / 20A 28VDC	SPST	SPST: Common and Normally Open Terminals Only Flange Type Connections to Relay

All ratings above are for Resistive Loads. Divide current switching capabilities by 2 for Inductive Loads. Inductive loads not recommended for 1A DPDT relay controller models.

Relay Types

SPST	Single Pole Single Throw: This relay is the most basic type of switch available. SPST relays have only two connections. Typically, an SPST relay shorts these two connections together when the relay is activated. When the relay is OFF, this connections return to their original OFF state (disconnected).
SPDT	Single Pole Double Throw: This is the most popular type of relay used in our relay controller designs. Each relay consists of a Normally Open (NO), Normally Closed (NC), and a Common (COM). When the relay is OFF, NC and COM are connected together. When the relay is turned ON, NC is disconnected from COM and NO and COM are connected together.
DPDT	Same as SPDT above, but there are two of these switches in a single relay that are activated/deactivated at the same time.

Characteristics Data	Minimum	Maximum
Relay Activation Time	>5 ms	<15 ms
Relay Deactivation Time	>5 ms	<20 ms
Activations per Second at 9600 Baud using Emulation Mode	N/A	960
Activations per Second at 9600 Baud using “Pro” Command Set	N/A	480
Activations per Second at 19.2K Baud using Emulation Mode	N/A	1,920
Activations per Second at 19.2K Baud using “Pro” Command Set	N/A	960
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
Maximum Allowed Activation Time per Relay (Relay Held in On State)	N/A	Unlimited
Expected Operational Life, Non-DPDT Models	>10,000,000 Cycles	N/A
Expected Operational Life, DPDT Models	>2,000,000 Cycles	N/A
Typical Operational Cycles Per Minute	N/A	1,800

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