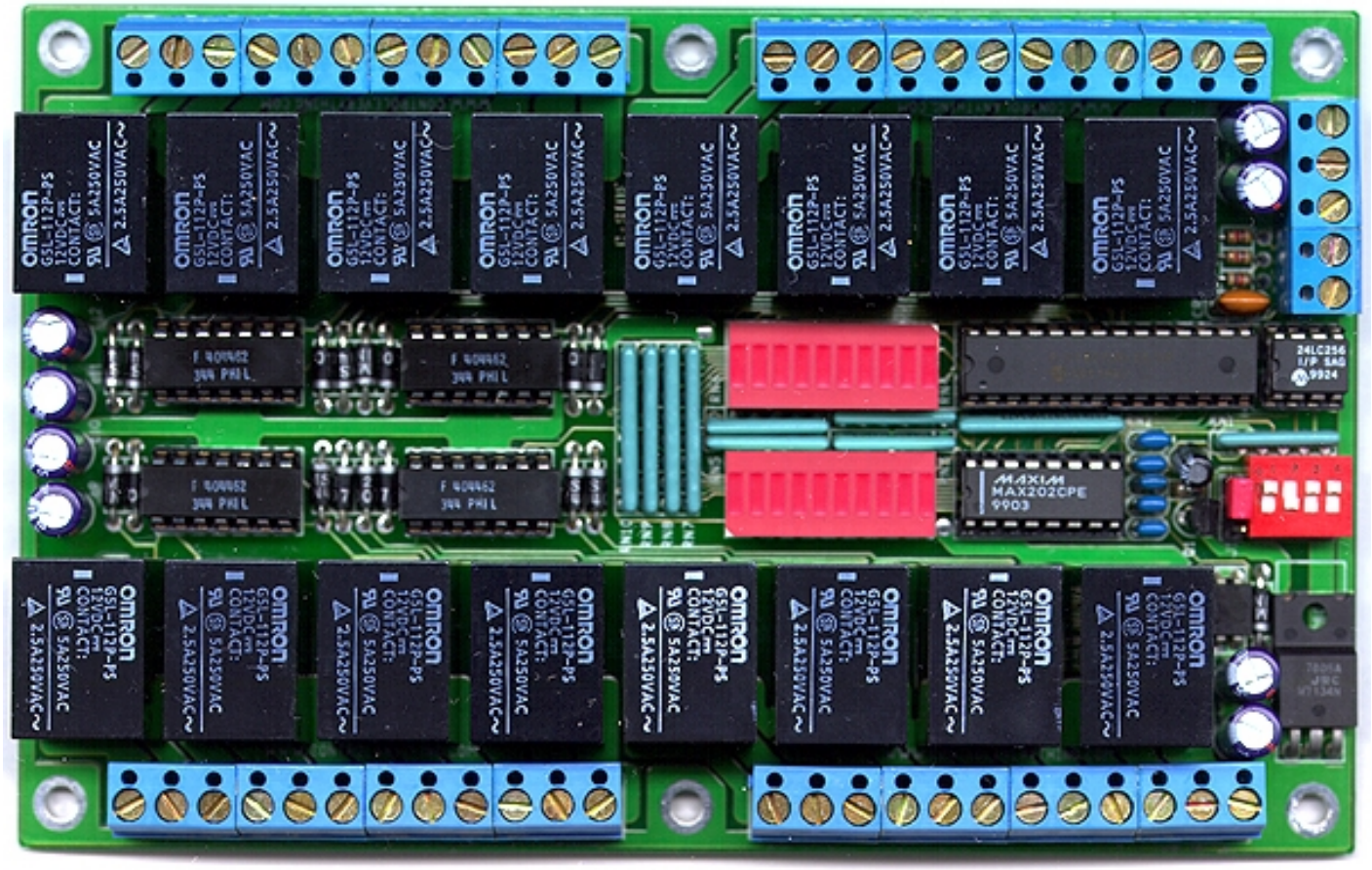
R165/R1610

RS-232 Networkable 16-Relay Controller



Device Variations

R165 16-Relay controller with sixteen 5-amp relays
R16510 16-Relay controller with eight 5-amp relays and eight 10-amp relays
R1610 16-Relay controller with sixteen 10-amp relays

Device Features

Control 256 Devices Simultaneously or Individually from a Single Serial Port
User-Selectable Communication Rates from 2400 to 115.2K Baud
E3C Compliant Command Set
16 Relay Status LEDs
Bank Enable LED
Heartbeat LED
Power LED
1 Data Receive LED
12 Volt DC Operation
6 Mounting Holes
256 Memory Banks for Storing/Recalling the Status of ALL Relays
User-Programmable Startup Status
Simultaneously Set the Status of 1, 8, or 16 Relays
Ask the Status of 1, 8, or 16 Relays
Low Power Mode

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

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

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

Copyrights and Trademarks

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:
ncdryan@aol.com
www.controlanything.com
www.controleverything.com
207.198.72.52

IMPORTANT POWER SUPPLY REQUIREMENTS

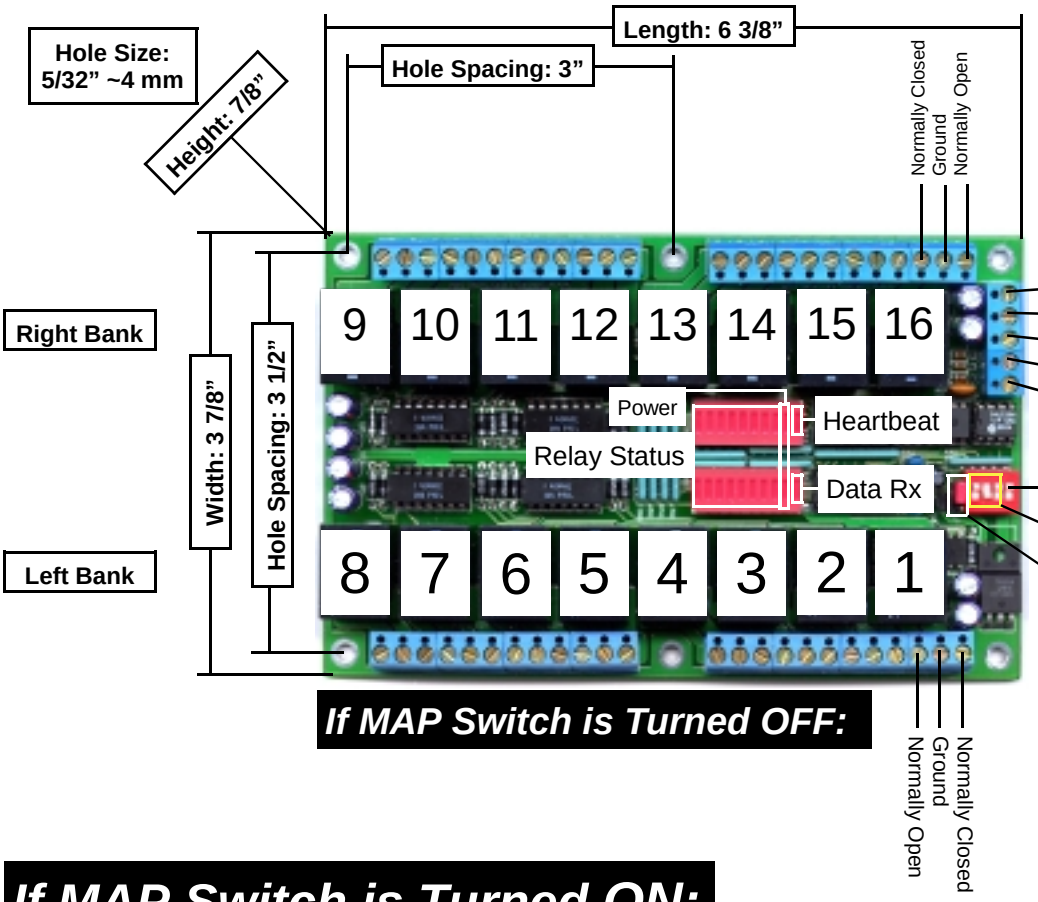
1) DO NOT USE A WALL WART TYPE UNREGULATED POWER SUPPLY.
2) USE ONLY A COMPUTER GRADE REGULATED SWITCHER SUPPLY RATED AT 12 VOLTS DC, 2.5 AMPS OR GREATER.
3) USE A SUPPLY RATED FOR MORE AMPERAGE WHEN POWERING MULTIPLE BOARDS.
4) OPERATING THIS DEVICE FROM AN AUTOMOTIVE ELECTRICAL SYSTEM MAY SHORTEN THE LIFE OF THE RELAYS. AVOID ACTIVATION OF RELAYS FOR EXTENDED PERIODS OF TIME.
5) 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.
6) 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.

Status LEDs:

Heartbeat LED:
When power is applied to the R16, this LED will slowly flash, indicating the CPU has properly booted and is ready to accept commands. If this LED stops flashing, the CPU may have stalled. It is possible to stall the CPU by using an inadequate power supply. See notes at left. This LED will be turned off in low power standby mode.

Receive LED:
Indicates a signal has been detected by the RS-232 receiver. This LED may be VERY DIM at high baud rates.

Power LEDs:
Two LEDs indicate power. The top LED indicates the voltage regulator is functioning. The lower LED indicates the relay bank is enabled. Both LEDs should always be on for normal operation. These LEDs will turn off in low power standby mode.



If MAP Switch is Turned OFF:

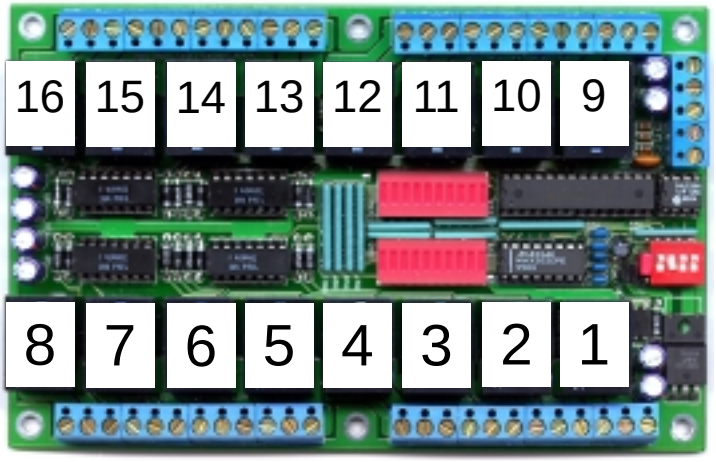
NOTE:

Baud and MAP settings are read when power is first applied to the R16 controller. Changing these jumpers while in use will have no effect.

MAP

The MAP Switch adjusts the internal map of Relays 9-16. Please see the photos on this page to see how these relays are internally numbered.

If MAP Switch is Turned ON:



RS-232 Output Select

When jumper J1 is installed between the upper two posts (as shown), the R16 sends and receives normal RS-232 data.

When installed between the lower two terminals, the RS-232 data output becomes an open collector signal. This output is ONLY compatible with the RSB serial booster and is used for 2-way communication to multiple R16 relay controllers.

Baud Select

1	2	3	Baud
OFF	OFF	OFF	TEST
ON	OFF	OFF	2400
OFF	ON	OFF	4800
ON	ON	OFF	9600
OFF	OFF	ON	19.2K
ON	OFF	ON	38.4K
OFF	ON	ON	57.6K
ON	ON	ON	115.2K

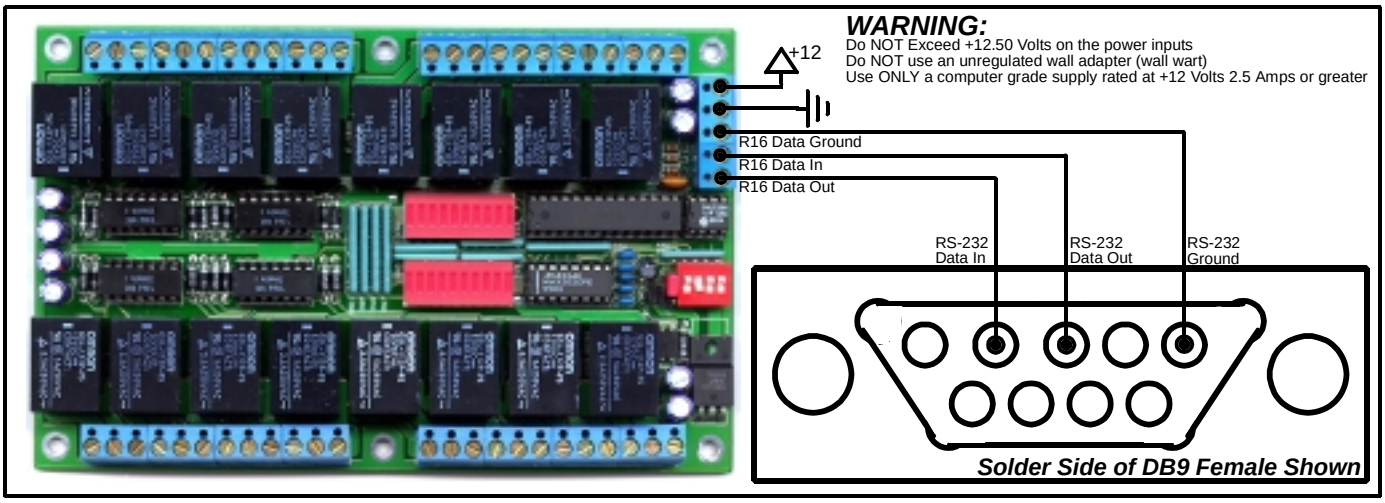
In TEST mode, each relay is automatically activated and deactivated. The R16 resumes normal operation at 9600 baud.

R16 Two-Way Communication:

The R16 supports two-way communication for confirming the receipt of commands and for reporting the status of the relays back to the host computer.

The R16 should be connected as shown below when using this device for the first time. Even if you plan to connect several R16 controllers to a single serial port, this wiring diagram must first be used to program the device number into the controller using our R16 Setup Utility.

The R16 Setup Utility expects this wiring configuration. Without it, some commands may respond incorrectly. By default, the R16 acknowledges all commands (except E3C commands) by sending ASCII character code 85 after the command has finished execution. This simply allows the host computer to verify the receipt of commands, and halts further instructions until the last command could be processed by the R16 relay controller.



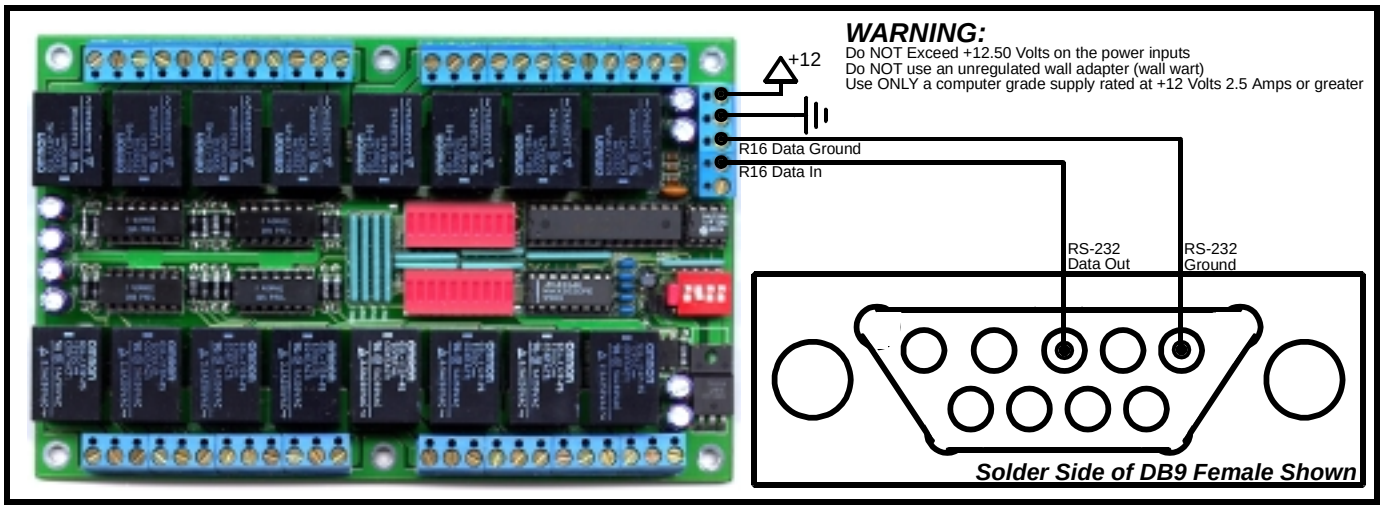
R16 One-Way Communication:

The R16 can be connected to a computer or microcontroller using as little as two wires. Some 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 some commands (particularly the memory storage functions).

When in this mode, reporting should be turned off for optimal one-way communication. When reporting is on, the R16 sends ASCII character code 85 back to the host computer to acknowledge the receipt of commands. Turning off reporting will allow you to send commands to the R16 much faster, but it is impossible to ask the controller for the status of relays when wired as shown below.

Reporting can be deactivated using the following ASCII characters:

ASCII 254	'Enter Command Mode
ASCII 48	'Disable Reporting
ASCII 254	'Enter Command Mode
ASCII 50	'Store Reporting Mode as Default.



Multiple R16 Controllers: Two-Way & One-Way Hybrid Communication

The R16 can be connected for two-way and one-way communication in applications where you may need to only monitor the status of one board, but control several others. Before using this wiring configuration, each R16 must be programmed with a device number (use our R16 setup utility to program the device numbers without writing a single line of code). Once a device number has been stored into each controller this wiring configuration may be used to control up to 256 different relay boards.

When all boards are first powered up, all devices will respond to incoming commands. You can tell each device to respond individually by controlling which devices are active and inactive.

The E3C command set is used to enable and disable multiple devices, allowing you to control one or more relay controllers simultaneously or individually. The E3C command set is more thoroughly described later in this manual.

Multiple Device Control: Quick Example

Step 1: Store a Device Number from 0 to 255 into Each Controller using the R16 Setup Utility.

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

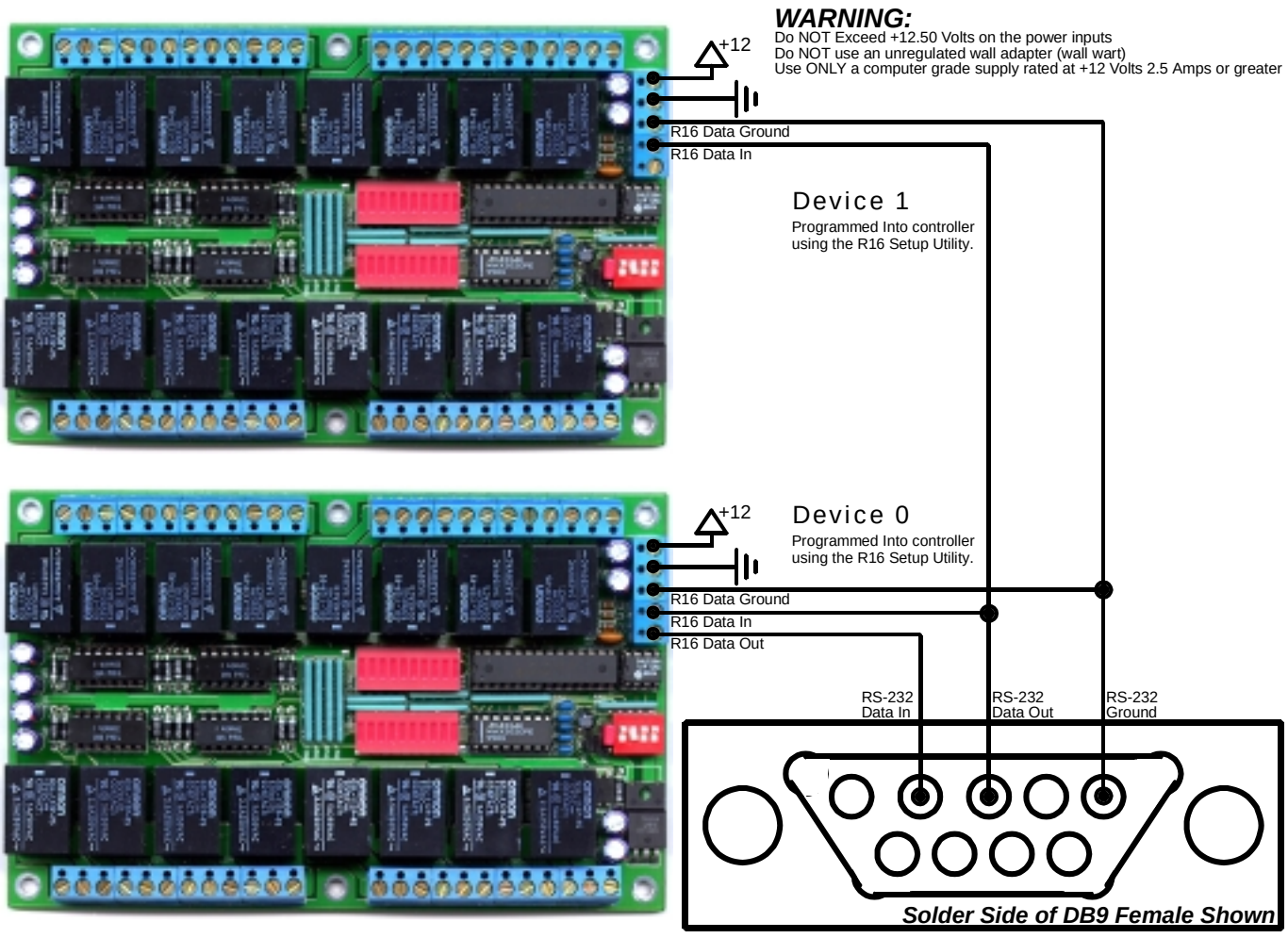
ASCII 254	'Enter Command Mode
ASCII 16	'Turn Relay 0 On

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

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



Multiple R16 Controllers: Two-Way Communication

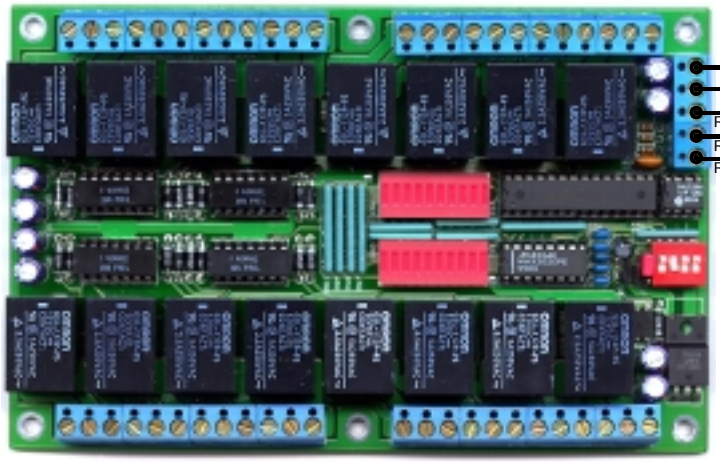
The R16 supports two-way communication to multiple devices using the RSB serial booster. Jumper J1 must be set for Open Collector data transmission. Set jumper J1 between the lower two terminals. This is ONLY required when using the RSB serial booster. The RSB serial booster should be used when controlling 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.

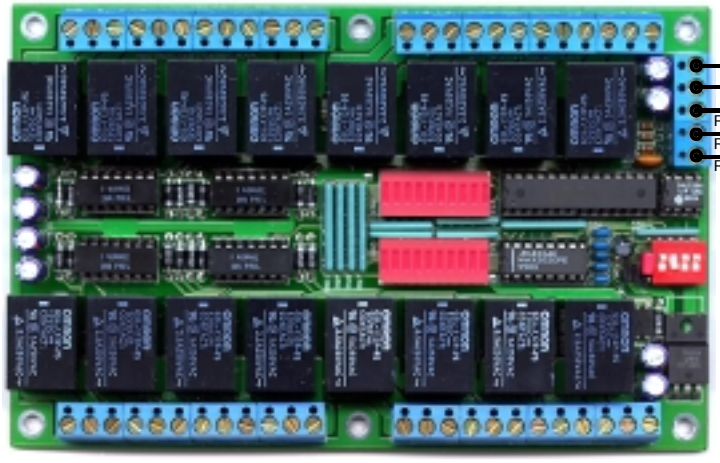
If you have not programmed device numbers into your R16 relay controllers, you may do so at this time. Simply apply power to the device you wish to program. All other devices must not be powered.

Use the R16 Setup Utility to read and change the device number of each board. By default, each R16 is shipped as device 0.

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, 2.5 AMPS OR GREATER.*
3) *USE A SUPPLY RATED FOR MORE AMPERAGE WHEN POWERING MULTIPLE BOARDS.*
4) *OPERATING THIS DEVICE FROM AN AUTOMOTIVE ELECTRICAL SYSTEM MAY SHORTEN THE LIFE OF THE RELAYS. AVOID ACTIVATION OF RELAYS FOR EXTENDED PERIODS OF TIME.*
5) *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.*
6) *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.*





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
R16 Data Out

Jumper J1 MUST be Set between the Lower Two Terminals. Set jumpers opposite of what is shown in these photos.

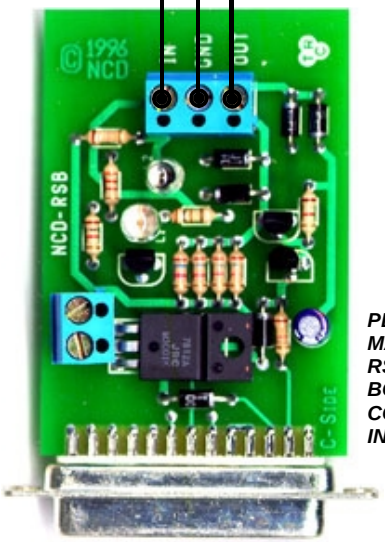
RS-232 Data In
RS-232 Ground
RS-232 Data Out

Device 1
The Device Number is Programmed Into Controller using the R16 Setup Utility.

Connect up to 256 R16s on a Single Serial Port. Each R16 MUST be Programmed with a Device Number using the R16 Setup Utility.

+12
R16 Data Ground
R16 Data In
R16 Data Out

Device 0
The Device Number is Programmed Into Controller using the R16 Setup Utility.



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 R16 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 R16 Relay Controller

The R16 is capable of sending and receiving data via RS-232 serial communications. The R16 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 R16. 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 R16 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 various popular programming languages for controlling the R16. 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 R16 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 R16 relay controller supports 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 R16 must be programmed with a device number, which is accomplished using the R16 Setup Utility.

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 R16 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 R16 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 R16 Command Set

The R16 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 of R16 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.

Controlling Relays

0-15 Turing Off Individual Relays
All other relays are unaffected.

16-31 Turing On Individual Relays
All other relays are unaffected.

32 Set the Status of Left Bank Relays
This command allows you to set the status of all left bank relays at one time. Relays will be activated in the equivalent binary pattern of the parameter byte.

33 Set the Status of Right Bank Relays
This command allows you to set the status of all right bank relays at one time. Relays will be activated in the equivalent binary pattern of the parameter byte.

34 Set the Status of All Relays
This command allows you to set the status of all relays at one time. Left bank relays will be activated in the equivalent binary pattern of the first parameter byte. Right bank relays will be activated in the equivalent binary pattern of the second parameter byte.

35 Turn Off Left Bank Relays

36 Turn On Left Bank Relays

37 Turn Off Right Bank Relays

38 Turn On Right Bank Relays

39 Turn Off All Relays

40 Turn On All Relays

Low Power Mode

41 Bank Disable: Low Power Mode
Low power mode allows you to minimize power consumption and may be used as an emergency deactivation of relays. While in low power mode, all relays and LEDs are turned off. The CPU stays running while in low power mode. You will be able to send and receive commands to the relay board as if the board were active. Any change in relay status will not be visible until normal operation is resumed.

42 Bank Enable: Normal Operation Mode
Resumes normal operation. The status of relays may change when in low power mode. The status will not be reflected on the LED and relays until this command is executed.

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 R16 relay controller. Additional source code re- can be found on the CD-Rom included with the R16, as well as on our web site at www.controleverything.com.

Sample Code: Controlling Relays

```
Public Sub SetRelayStatus(Relay,Stat)
    If Stat = 0                        'Turn Off Relay
       MSComm1.Output = Chr$(254)      'Enter Command Mode
       MSComm1.Output = Chr$(Relay-1)  'Turn Off Relay
    Else
       MSComm1.Output = Chr$(254)         'Enter Command Mode
       MSComm1.Output = Chr$((Relay-1)+16) 'Turn On Relay
    Endif
    WaitForReply                'Wait for R16 to Acknowledge Command
End Sub

Public Sub SetLeftBankStatus(Stat)
    MSComm1.Output = Chr$(254)  'Enter Command Mode
    MSComm1.Output = Chr$(32)   'Set Left Bank Status Command
    MSComm1.Output = Chr$(Stat) 'Parameter Byte Sets Status
    WaitForReply                'Wait for R16 to Acknowledge Command
End Sub

Public Sub SetRightBankStatus(Stat)
    MSComm1.Output = Chr$(254)  'Enter Command Mode
    MSComm1.Output = Chr$(33)   'Set Right Bank Status Command
    MSComm1.Output = Chr$(Stat) 'Parameter Byte Sets Status
    WaitForReply                'Wait for R16 to Acknowledge Command
End Sub

Public Sub SetAllRelayStatus(LStat,RStat)
    MSComm1.Output = Chr$(254)  'Enter Command Mode
    MSComm1.Output = Chr$(34)   'Set Right Bank Status Command
    MSComm1.Output = Chr$(LStat) 'LStat Byte Sets Left Bank Status
    MSComm1.Output = Chr$(RStat) 'RStat Byte Sets Right Bank Status
    WaitForReply                'Wait for R16 to Acknowledge Command
End Sub

Public Sub LeftBankOff()
    MSComm1.Output = Chr$(254)  'Enter Command Mode
    MSComm1.Output = Chr$(35)   'Turn Off Left Bank Relays
    WaitForReply                'Wait for R16 to Acknowledge Command
End Sub

Public Sub LeftBankOn()
    MSComm1.Output = Chr$(254)  'Enter Command Mode
    MSComm1.Output = Chr$(36)   'Turn On Left Bank Relays
    WaitForReply                'Wait for R16 to Acknowledge Command
End Sub

Public Sub RightBankOff()
    MSComm1.Output = Chr$(254)  'Enter Command Mode
    MSComm1.Output = Chr$(37)   'Turn Off Right Bank Relays
    WaitForReply                'Wait for R16 to Acknowledge Command
End Sub

Public Sub RightBankOn()
    MSComm1.Output = Chr$(254)  'Enter Command Mode
    MSComm1.Output = Chr$(38)   'Turn On Right Bank Relays
    WaitForReply                'Wait for R16 to Acknowledge Command
End Sub

Public Sub AllRelaysOff()
    MSComm1.Output = Chr$(254)  'Enter Command Mode
    MSComm1.Output = Chr$(39)   'Turn Off All Relays
    WaitForReply                'Wait for R16 to Acknowledge Command
End Sub

Public Sub AllRelaysOn()
    MSComm1.Output = Chr$(254)  'Enter Command Mode
    MSComm1.Output = Chr$(40)   'Turn On All Relays
    WaitForReply                'Wait for R16 to Acknowledge Command
End Sub
```

Sample Code: Low Power Mode

```
Public Sub BankDisable()
    MSComm1.Output = Chr$(254)  'Enter Command Mode
    MSComm1.Output = Chr$(41)   'Turn Off All Relays
    WaitForReply                'Wait for R16 to Acknowledge Command
End Sub

Public Sub BankEnable()
    MSComm1.Output = Chr$(254)  'Enter Command Mode
    MSComm1.Output = Chr$(42)   'Turn On All Relays
    WaitForReply                'Wait for R16 to Acknowledge Command
End Sub
```

The R16 Command Set

Reading the Status of Relays

43, 0-15 Get Status of an Individual Relay
Command 43 with a parameter of 0-15 will ask the R16 for the status of relay 1-16. The R16 will report back a status for the selected relay. A status reply of 0 indicates the relay is off. A status reply of 1 indicates the relay is on.

43, 16 Get Status of Left Bank Relays
Command 43 with a parameter of 16 will ask the R16 for the status of the left relay bank. A byte will be reported back to the computer indicating the status of all 8 left-bank relays.

43, 17 Get Status of Right Bank Relays
Command 43 with a parameter of 17 will ask the R16 for the status of the right relay bank. A byte will be reported back to the computer indicating the status of all 8 right-bank relays.

43, 18 Get Status of All Relays
Command 43 with a parameter of 18 will ask the R16 for the status of all relays. Two bytes will be reported back to the computer from the R16. The first byte indicates the status of the left bank relays. The second byte indicates the status of the right bank relays.

Memory Storage Functions

44 Store Relay Settings in Memory Bank
The R16 allows you to store and recall the status of all 16 relays in 256 different memory banks. This is useful for creating hardware macros that electronically prevent some relays from coming on simultaneously. To use this command, use other relay control commands to set the status of all 16 relays. Once relays are set, the status can be stored. A parameter is required for this command, indicating the memory bank to store the relay setting in.

45 Recall Settings from Memory Bank
Recalls the on/off status of all 16 relays from 1 of 256 different memory banks. A parameter is required for this command indicating the memory bank to recall.

46 Store Default Relay Power-up Status
It is possible to control the power-up status of relays. By default, all relays are turned off when power is first applied to the board. To use this command, use other relay control commands to set the status of the relays in the desired default state. Then execute this command to record the relay settings. The next time power is applied to the board, the relays will automatically return to the stored power-up status.

47 Clear Default Relay Power-up Status
This command erases the stored power-up status. All relays will be off when power is first applied.

Sample Code: Reading the Status of Relays

```
Public Function GetRelayStatus(Relay)
    MSComm1.Output = Chr$(254)          'Enter Command Mode
    MSComm1.Output = Chr$(43)           'Get Status Command
    MSComm1.Output = Chr$(Relay-1)      'of an Individual 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 GetLeftBankStatus()
    MSComm1.Output = Chr$(254)          'Enter Command Mode
    MSComm1.Output = Chr$(43)           'Get Status Command
    MSComm1.Output = Chr$(16)           'of Left Bank Relays
    Do                                  'Wait for Device to Reply
        DoEvents                        'Allow Windows to MultiTask
    Until MSComm1.InBufferCount > 0     'If the Device Replies
    GetLeftBankStatus = Asc(MSComm1.Input) 'Get Status from R16
    Debug.Print GetLeftBankStatus        'Display in Immediate Window
End Sub

Public Function GetRightBankStatus()
    MSComm1.Output = Chr$(254)          'Enter Command Mode
    MSComm1.Output = Chr$(43)           'Get Status Command
    MSComm1.Output = Chr$(17)           'of Right Bank Relays
    Do                                  'Wait for Device to Reply
        DoEvents                        'Allow Windows to MultiTask
    Until MSComm1.InBufferCount > 0     'If the Device Replies
    GetRightBankStatus = Asc(MSComm1.Input) 'Get Status from R16
    Debug.Print GetRightBankStatus      'Display in Immediate Window
End Sub

Public Sug GetAllRelayStatus()
    MSComm1.Output = Chr$(254)          'Enter Command Mode
    MSComm1.Output = Chr$(43)           'Get Status Command
    MSComm1.Output = Chr$(18)           'of Right Bank Relays
    Do                                  'Wait for Left Bank Status
        DoEvents                        'Allow Windows to MultiTask
    Until MSComm1.InBufferCount > 0     'If the Device Replies
    LeftBankStatus = Asc(MSComm1.Input) 'Get Left Status from R16
    Debug.Print LeftBankStatus          'Display in Immediate Window
    Do                                  'Wait for Right Bank Status
        DoEvents                        'Allow Windows to MultiTask
    Until MSComm1.InBufferCount > 0     'If the Device Replies
    RightBankStatus = Asc(MSComm1.Input)'Get Left Status from R16
    Debug.Print RightBankStatus         'Display in Immediate Window
End Sub
```

Sample Code: Memory Storage Functions

```
Public Sub StoreRelaySettings(MemBank)
    MSComm1.Output = Chr$(254)   'Enter Command Mode
    MSComm1.Output = Chr$(44)    'Store Current Relay Settings Command
    MSComm1.Output = Chr$(MemBank)'Memory Bank to Store Settings In
    WaitForReply                 'Wait for R16 to Acknowledge Command
End Sub

Public Sub RecallRelaySettings(MemBank)
    MSComm1.Output = Chr$(254)   'Enter Command Mode
    MSComm1.Output = Chr$(45)    'Recall Relay Settings Command
    MSComm1.Output = Chr$(MemBank)'Memory Bank to Recall Settings From
    WaitForReply                 'Wait for R16 to Acknowledge Command
End Sub

Public Sub StoreDefaultRelaySettings()
    MSComm1.Output = Chr$(254)   'Enter Command Mode
    MSComm1.Output = Chr$(46)    'Store Relay Settings as Default Cmd
    WaitForReply                 'Wait for R16 to Acknowledge Command
End Sub

Public Sub ClearDefaultRelaySettings()
    MSComm1.Output = Chr$(254)   'Enter Command Mode
    MSComm1.Output = Chr$(47)    'Clear Default Relay Settings Command
    WaitForReply                 'Wait for R16 to Acknowledge Command
End Sub
```

Memory Notice

The R16 Relay Controller uses nonvolatile EEPROM memory, which is rated to store data for 100+ years without the need for power. EEPROM memory is self-destructive in that it can only be written to ~100,000 times. EEPROM memory may be read an infinite number of times without any degradation of life. When developing software to control the R16, avoid using the memory storage functions on a daily basis to preserve memory life. Use the memory recall functions as frequently as needed.

The R16 Command Set

Reporting Mode

When reporting mode is turned on, the R16 acknowledges most incoming commands by sending ASCII character code 85 back to the computer once the command has been executed by the board. When properly used, this can prevent you from sending commands to the R16 faster than they can be processed. And because incoming commands are acknowledged, you know the board is properly functioning from your desktop computer.

In some installations, you may not want to acknowledge incoming commands. When reporting mode is turned off, data can be communicated faster because it only has to travel one direction. So if you experience unreliable or erratic operation, try turning the reporting mode off.

48 Turn Off Reporting Mode
The R16 does NOT send ASCII 85 to acknowledge incoming commands. Provides faster, but blind operation.

49 Turn On Reporting Mode
The R16 sends ASCII 85 back to the computer after most commands to acknowledge execution. Slower, but the receipt of commands is confirmed by the R16.

50 Store Reporting Mode as Power-up Default
Once you have set the reporting mode, you can store this mode as the default. The next time power is applied to the board, the reporting mode will be recalled from stored memory. The R16 is shipped with the reporting mode turned on. The R16 setup utility turns on the reporting mode, but does not change the stored settings.

Sample Code: Reporting Mode

```
Public Sub ReportingOff()
    MSComm1.Output = Chr$(254)   'Enter Command Mode
    MSComm1.Output = Chr$(48)    'Turn Off Reporting
    WaitForReply                 'Wait for R16 to Acknowledge Command
End Sub

Public Sub ReportingOn()
    MSComm1.Output = Chr$(254)   'Enter Command Mode
    MSComm1.Output = Chr$(49)    'Turn Off Reporting
    WaitForReply                 'Wait for R16 to Acknowledge Command
End Sub

Public Sub StoreReportingStatus()
    MSComm1.Output = Chr$(254)   'Enter Command Mode
    MSComm1.Output = Chr$(50)    'Store Reporting Status Command
    WaitForReply                 'Wait for R16 to Acknowledge Command
End Sub
```

If Reporting is ON

If Reporting is turned on, a routine should be used to handle incoming data from the R16. The WaitForReply subroutine simply waits for an acknowledgment from the R16 and displays the acknowledgment in the immediate window in Visual Basic. If reporting is turned off, this routine is not necessary. Note that all programming examples that generate an acknowledgment call this subroutine before continuing. E3C commands do not report back to the computer. Commands that ask the board for information are not concluded with ASCII code 85.

```
Public Sub WaitForReply()
    Do                                  'Wait for Replay
        DoEvents                        'Allow Windows to MultiTask
    Until MSComm1.InBufferCount > 0     'If the Device Replies
    Confirm = Asc(MSComm1.Input)        'Confirm should equal 85
    Debug.Print Confirm                 'Display Confirmation
End Sub
```

Troubleshooting

Unreliable or Unpredictable Relay Activation
There are two possible causes for erratic relay activation.
1) Check your wiring. When in doubt, run our test program with a short RS-232 cable to test board operation. Improper wiring can cause unpredictable behavior.

2) If your program is written to send commands to the R16, but is not written to accept acknowledgements, then you are probably sending data to the R16 at the same time the R16 is trying to send data to you. Turn off the reporting mode and try again. Please read the section above for more detailed information.

Erratic Heartbeat
If the heartbeat LED blinks at varying speeds, this may be normal operation of the R16 trying to parse incoming RS-232 data. When no data is received by the R16, the heartbeat LED will blink steadily. If it does not, please return it to us for immediate repair.

All LEDs are Off and Relays will NOT Actuate
The R16 may have been inadvertently placed in low power mode. Remove power and try again, or send the Normal Power Mode command. The power LEDs should light and the heartbeat LED should flash.

Test Mode
The R16 can be placed in test mode by turning off dip switches 1, 2, and 3. The next time power is applied, each relay will activate slowly. The bank enabler will click all relays off and on a couple of times, then all relays will slowly deactivate. Once test mode has completed, the R16 is ready to accept incoming commands at 9600 baud. Use the test mode to verify the activation of all relays and LEDs (except the Rx LED).

Warm Components
Some components on the R16 may become warm after being on for a long period of time. This is normal operation and should not affect operation and performance. As a general rule of thumb, all components should never be too hot to touch. If you suspect a component has overheated, please contact us as soon as possible.

Additional Problems
Please contact National Control Devices if you experience any irresolvable problems with the R16 relay controller. Visit **www.controleverything.com** for the latest contact information.