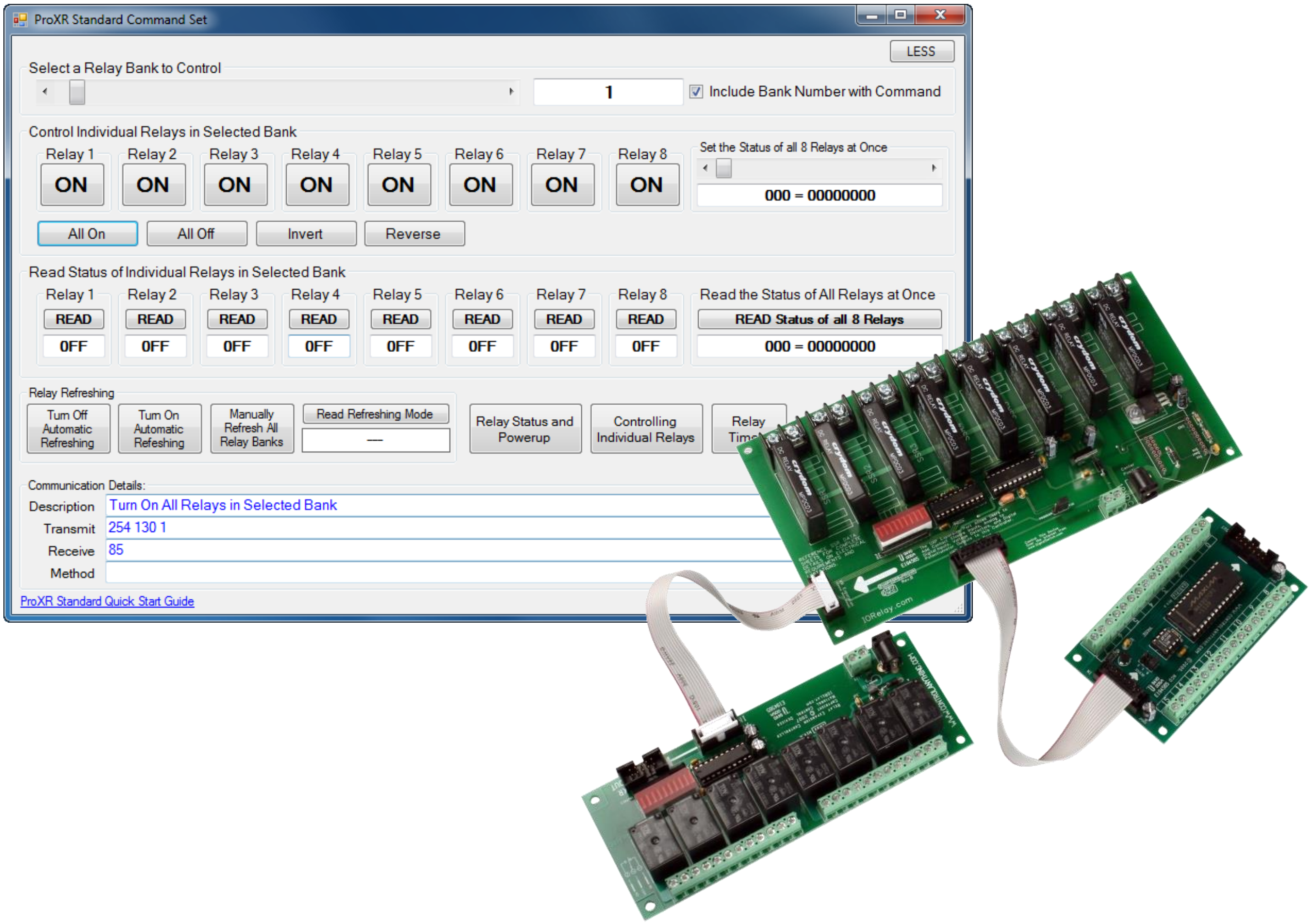
Control Anything...Anywhere.

National Control Devices

ProXR Standard Quick Start Guide



ProXR Standard Command Set

LESS

Select a Relay Bank to Control

1 ☑ Include Bank Number with Command

Control Individual Relays in Selected Bank

Relay 1	Relay 2	Relay 3	Relay 4	Relay 5	Relay 6	Relay 7	Relay 8
ON	ON	ON	ON	ON	ON	ON	ON

Set the Status of all 8 Relays at Once

000 = 00000000

All On | All Off | Invert | Reverse

Read Status of Individual Relays in Selected Bank

Relay 1	Relay 2	Relay 3	Relay 4	Relay 5	Relay 6	Relay 7	Relay 8
READ	READ	READ	READ	READ	READ	READ	READ
OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF

Read the Status of All Relays at Once

READ Status of all 8 Relays

000 = 00000000

Relay Refreshing

Turn Off Automatic Refreshing | Turn On Automatic Refeshing | Manually Refresh All Relay Banks | Read Refreshing Mode —

Relay Status and Powerup | Controlling Individual Relays | Relay Tim

Communication Details:

Description: Turn On All Relays in Selected Bank

Transmit: 254 130 1

Receive: 85

Method:

ProXR Standard Quick Start Guide

Relay Control Command Set

NATIONAL CONTROL DEVICES

ProXR Standard

Relay Control Command Set

National Control Devices, LLC
PO Box 455
Osceola, MO 64776
Phone 417.646.5644 • Fax (866) 562-0406

Notice: Portions of this manual require internet access.

NOTICE:

All customers are STRONGLY ADVISED to purchase at least ONE USB Communication Module. This communication module may be used to recover a controller or to reconfigure a controller should there be an accidental loss of communications. NCD Tech support may be unable to assist customers who do not have a USB Communications Module available for troubleshooting purposes.

Purchase USB Communications Module from our website at this link:

http://www.controlanything.com/Relay/Device/ZUSB

Device Variations

This manual covers all NCD products with ProXR in the Part Number as well as other part numbers that may reference this document.

Table of Contents

Chapter

1

Introduction

NCD ProXR Standard Series Controllers were developed in 2005 and released into mass production for 2006. These controllers place a heavy emphasis on communication speed, relay bank expansion, and relay timing control.

The ProXR Series Controllers include an XR Expansion port, allow you to add additional banks of relays (any kind of relays you need), directly to the main controller. With a few minor programming modifications, you will be controlling the new relay banks in seconds. Up to 256 relays (32 Banks of 8 Relays) are supported. You can now control individual relays, relay banks, or all relays at one time.

The ProXR Series Controllers also offer full-speed RS-232 communications, up to 115.2K Baud, and are E3C Network Compliant up to 38.4K Baud. Also available are 16 background timers ideal for watchdog, keep alive, server reboot, and duration timing applications.

- Control 256 Devices from a Single Serial Port
- Watchdog/Server Reboot/Keep Alive Timing Functions
- Supports Duration Timing Commands (turn a light on for 8 hours)
- Control 256 Relays (32 Relay Banks) From One Controller
- User-Selectable Communication Rates up to 115.2K Baud
- Change Parameters in Configuration Mode ONLY to Prevent Accidental Changes
- E3C Compliant Command Set
- Diode Clamped Relay Driver Stage
- Relay Status LEDs
- Device Enabled/Power LED
- Data Receive LED
- 12 Volt DC Operation
- User-Programmable Startup Status
- Simultaneously Set the Status All Relays
- Ask the Status of Individual or All Relays
- Protected E3C Device Numbering
- O.C. RS-232 Communication for Networking Multiple Devices
- Powerful ASCII Character Code Based Command Set
- Compatible with ANY Computer or Microcontroller

Chapter

2

Getting Started

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 in 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. ProXR series controllers can be used in 12 volt automotive electrical systems.

Notice:

Never install NCD Relay Controllers near High Power RF Transmitters, such as CB Radio and Emergency Vehicle Voice/Data Transmitters. These devices may cause all relays to turn off.

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.

Hardware diagrams

Hardware diagrams may be found in the NCD Hardware Reference Guide on our website. It will assist you with connecting your device.

Mechanical Drawings

Complete mechanical drawings for each device in the ProXR Series Line can be found on the product description page of each controller at www.controlanything.com.

Chapter

3

Base Station Software

The ProXR Standard Series Controllers are capable of sending and receiving data via RS-232 serial communications. NCD Base Station Software is the program used to identify, configure, and test devices. You can download the Base Station Software from our website: Download Base Station Software. You can also read more information on this program at the following link from our website: Base Station Quick Start Guide

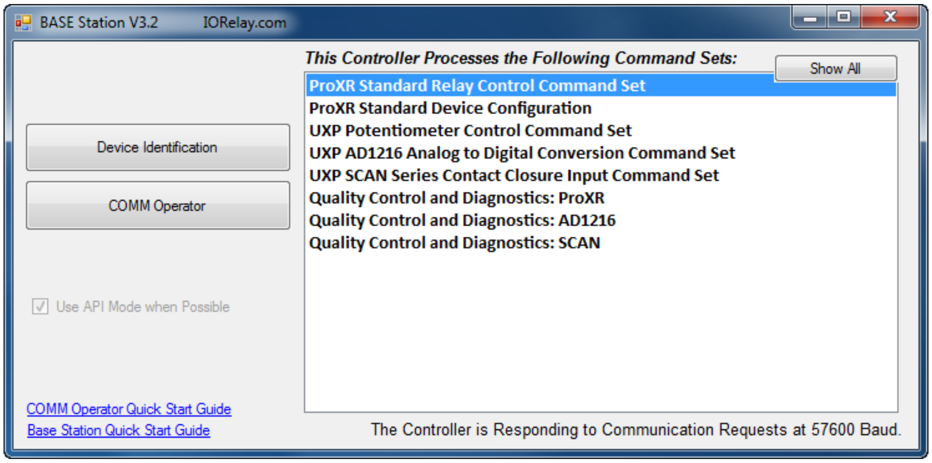
Base Station software is in constant development, always evolving to fit the needs of users. The software is constantly being updated; therefore, you may need to download the software frequently.

Base Station software provides you with a list of command sets processed by you device. Some devices will have more command sets and some will have less. What is displayed is specific to the controller you are using.

In order to communication with the ProXR Standard Relay Controller, run the NCD Base Station software.

- Connect the device to your computer using your favorite interface technology.
- Run the NCD Base Station software
- Select the appropriate COM port or IP Address
- Click OK.

When the dialog box appears, choose *'ProXR Standard Relay Control Command Set'* as shown in the screen shot below.


BASE Station V3.2 IORelay.com
Device Identification
COMM Operator
Show All
COMM Operator Quick Start Guide
Base Station Quick Start Guide

☑ Use API Mode when Possible

This Controller Processes the Following Command Sets:

ProXR Standard Relay Control Command Set
ProXR Standard Device Configuration
UXP Potentiometer Control Command Set
UXP AD1216 Analog to Digital Conversion Command Set
UXP SCAN Series Contact Closure Input Command Set
Quality Control and Diagnostics: ProXR
Quality Control and Diagnostics: AD1216
Quality Control and Diagnostics: SCAN

The Controller is Responding to Communication Requests at 57600 Baud.

Configuring your controller:

A. Select Relay Bank commands are directed to. (Note: Selecting 0 directs commands to all available relay banks).

B. Simple On/Off control of individual relay in selected bank.

C. Set status of all relays in the selected relay bank.

D. Read the status of individual relays in selected bank.

E. Turn Off Automatic Relay Refreshing, see page 11.

F. Section is shown when the MORE option is chosen in section L in diagram. Provides communication details.

G. Turn on Automatic Relay Refreshing, see Page 11.

H. Manually Refresh All Relay Banks, see Page 12.

I. Store current relay status as default power up status. See page 16.

J. Bring up window for controlling individual relay.

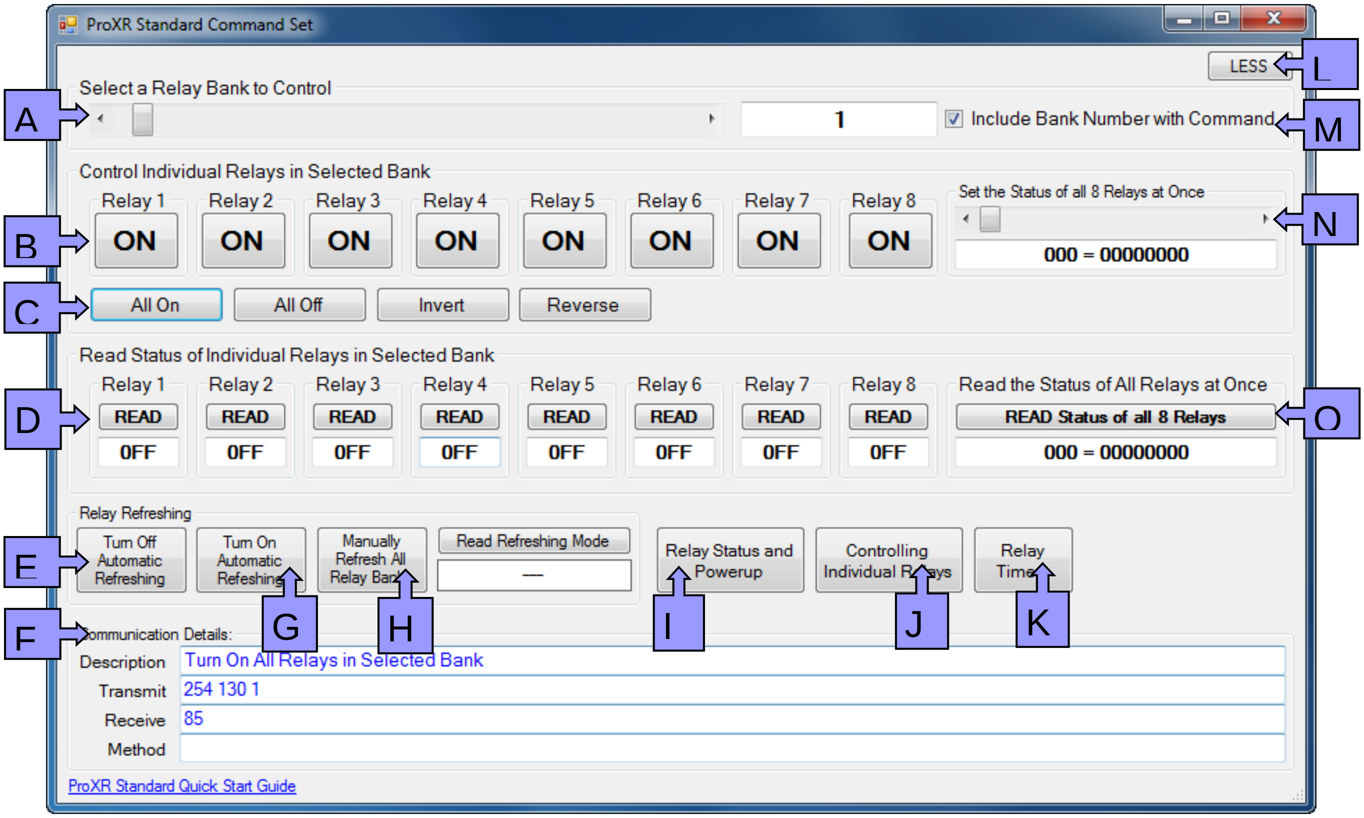
K. Test built-in ProXR Timers.

L. MORE or LESS. More expands the window to include communication details provided in section F in diagram. LESS shrinks the window to exclude section F.

M. This box is recommended to stay checked so commands are directed to banks correctly.

N. Set status of all relays in bank, see page 16.

O. Read status of all relays in selected bank, see page 13.


ProXR Standard Command Set
LESS
L
A
M
B
N
C
D
O
E
F
G
H
I
J
K
Control Individual Relays in Selected Bank
Relay 1
Relay 2
Relay 3
Relay 4
Relay 5
Relay 6
Relay 7
Relay 8
ON
All On
All Off
Invert
Reverse
Set the Status of all 8 Relays at Once
Read Status of Individual Relays in Selected Bank
READ
Read the Status of All Relays at Once
READ Status of all 8 Relays
Relay Refreshing
Turn Off Automatic Refreshing
Turn On Automatic Refeshing
Manually Refresh All Relay Banks
Read Refreshing Mode
Relay Status and Powerup
Controlling Individual Relays
Relay Timers
ProXR Standard Quick Start Guide

Select a Relay Bank to Control: 1

☑ Include Bank Number with Command

Set the Status of all 8 Relays at Once: 000 = 00000000

Relay 1	Relay 2	Relay 3	Relay 4	Relay 5	Relay 6	Relay 7	Relay 8	Read the Status of All Relays at Once
0FF	0FF	0FF	0FF	0FF	0FF	0FF	0FF	000 = 00000000

Read Refreshing Mode: —

Communication Details:

Description: Turn On All Relays in Selected Bank

Transmit: 254 130 1

Receive: 85

Method:

Chapter

4

Relay Banking Commands

Understanding Relay Banks and Basic Control Concepts

A Relay Bank is simply a group of 8 relays. The ProXR Series Controllers allow you to control up to 256 relays (32 relay banks). You control which bank of relays you are speaking to at all times. It is VERY IMPORTANT that you understand that there are two ways to specify which relay bank you are talking to. These topics will be discussed in greater length later in this manual, providing you with specific instructions and program examples. This page will help prepare you for seeing two different commands that do the same thing. In this manual, you will see the work "bank". This word should be equated to a number from 0 to 32. A value of 1 speaks to relay bank 1 (the first 8 relays on the board). A value of 2 speaks to relay bank 2 (the second group of 8 relays on the board). A value of 32 speaks to the last group of 8 relays (which is connected to the main controller using the XR expansion ports). A value of 0 speaks to all banks of relays at one time. When bank 0 is selected, you can then specify a command to turn on relay 1, and relay 1 on all relay banks will be activated. There will be more examples of this and detailed information on each command.

Understanding the concept of controlling multiple relays across multiple banks is very important to your understanding how our controller command set is organized.

Bank Directed Commands:

a) Specify a relay bank (there is a command you will send just for this purpose).
All Subsequent Commands will be directed to the previously specified relay bank.

b) Specify a different relay bank.
All Subsequent Commands will be directed to the new relay bank you have specified.

NOTE: VERY IMPORTANT READING ON THIS PAGE

Understanding Relay Bank Refreshing: Controlling Multiple Banks

Under normal operation, you will send a command to the relay controller, and the relay controller will respond to your command by activating or deactivating a relay. This system works well if you only need to control 1-8 relays, but it does not necessarily work very well if you are taking advantage of the XR Expansion ports, allowing you to control up to 256 relays (32 relay banks). In these cases, you may want to set the status of all relays at the exact same time. The easiest way to do this is to turn off automatic relay refreshing. Once turned off, you can use the relay control command set to activate relays, but the commands will not appear to have any effect. The effects will not be seen until you manually refresh the relay bank. Rest assured, when auto refreshing is off, your relay control commands are working, the processor memory is copied to the physical relay bank memory when you manually refresh the relays.

Follow this methodology to set the status of lots of relays at one time across multiple banks:

a) Turn Off Relay Bank Auto Refreshing.

b) Use Relay Control Command to activate different relays on different banks. ***These commands will not appear to work; they will only modify internal memory.***

c) Send the Manual Refresh Command to update all relays at one time.

NOTE: VERY IMPORTANT READING ON THIS PAGE

The ProXR Series Controllers allow you to control up to 256 relays. Relays are divided into groups of 8 called banks, and are addressed by their bank number. For instance, a ProXR series controller with 32 on-board relays has four on-board banks, the on-board relays respond to bank values of 1-4. If you use the XR Expansion port to add another bank of 24 relays, then you will need to specify bank values of 5-7 to control the extra relays. The firmware doesn't actually know how many relays are attached to the relay controller, but it assumes there are 256 relays (bank values 1-32, which is the maximum supported number of relays).

In this manual, you will see two commands that appear to do the same thing, for example:

254 0-7	Turn Off Individual Relays
254 100-107 Bank	Turn Off Individual Relays in Bank
254 8-15	Turn On Individual Relays
254 108-115	Turn On Individual Relays in Bank
254 16-23	Get the Status of an Individual Relay
254 116-123 Bank	Get the Status of an Individual Relay in Bank

While the outcome is the same, these commands function in slightly different ways.

For instance:

254 8	Turn On Relay 1

To make this command work, you will send a 254, then an 8 to activate a relay. But default, relay bank 1 will be affected by this command. However, you can redirect this command to a different relay bank using the following command:

254 49 2	Direct Commands to Relay Bank 2

Then you can send:

254 8	Turn On Relay 1 in Bank 2

Here are more examples:

254 49 1	Direct Commands to Relay Bank 1
254 8	Turn On Relay 1 in Bank 1
254 49 2	Direct Commands to Relay Bank 2
254 8	Turn On Relay 1 in Bank 2
254 9	Turn On Relay 2 in Bank 2
254 10	Turn On Relay 3 in Bank 2
254 49 3	Direct Commands to Relay Bank 3
254 8	Turn On Relay 1 in Bank 3
254 11	Turn On Relay 4 in Bank 3
254 12	Turn On Relay 5 in Bank 3
254 13	Turn On Relay 6 in Bank f3
254 14	Turn On Relay 7 in Bank 3
254 49 0	Direct Commands to All Relay Banks
254 8	Turn On Relay 1 in All Relay Banks

This command structure has the advantage of being very fast and efficient. However, if power to the controller is ever lost, commands will automatically be directed to bank 1 when power to the controller has been restored.

Chapter 5

Relay Banking Command Set

Activate Several Relays on Several Relay Banks

We have also included a set of commands that require you to specify the bank as part of the command structure. This ensures commands are always directed to the correct relay bank. Here are a few examples: (note: relays will not be updated, they will not appear to change).

Send Bytes:	**Byte 1:**	**Byte 2:**	**Byte 3:**
Function:	Command	Turn On Relay	Bank
Decimal Values:	254	108-112	0-255
Hex Values:	0xFE	0x6C-0x70	0x00-0xFF

Receive Byte:	Decimal:	85
	Hex:	0x55

COMM Operator Examples:

254 108 1	Turn On Relay 1 in Bank 1
254 108 2	Turn On Relay 1 in Bank 2
254 109 2	Turn On Relay 2 in Bank 2
254 110 2	Turn On Relay 3 in Bank 2
254 108 3	Turn On Relay 1 in Bank 3
254 111 3	Turn On Relay 4 in Bank 3
254 112 3	Turn On Relay 5 in Bank 3
254 112 0	Turn On Relay 5 in All Relay Banks (Bank 0)

NCD Component Library Command Method

Not Yet Implemented

Note:

You can keep adding relay control commands to activate different relays on different banks, but keep in mind; these commands will not appear to function. These commands are changing the memory pattern for the relays inside the controller. You will not see the effects of your changes until you send a Manual Refresh command. You can return to automatic refreshing at any time. Turning on automatic refreshing does NOT refresh the relays until the NEXT relay control command is issued.

Turn On/Off Relay Refreshing

The ProXR Series Controllers allow you to manually or automatically refresh relay banks. When you first receive your ProXR Series Controller, Relay Refreshing is turned on. Meaning, every time you send a relay control command, the relays will respond to your commands.

Turn off relay refreshing allows you to control when the relays actually switch. When refreshing is turned off, you can send relay control commands to the ProXR controller, and the controller will work just like normal, but the relays will never change state.

You can then use the Manual Refresh command to set the status of all relays at one time. Whether you are controlling 1 or 235 relays, you will be able to refresh all relays at one time to any given state by taking manual control of the relay refreshing capabilities.

Send Bytes:	**Byte 1:**	**Byte 2:**
Function:	Command	Turn On/Off Relay Refreshing
Decimal Values:	254	25-26
Hex Values:	0xFE	0x19-0x1A

Receive Byte:	Decimal:	85
	Hex:	0x55

COMM Operator Examples:

254 25	Turn On Automatic Relay Refreshing
254 26	Turn Off Automatic Relay Refreshing

NCD Component Library Command Method:

Not Yet Implemented

Store Relay Refreshing Mode

Send Bytes:	**Byte 1:**	**Byte 2:**
Function:	Command	Store Relay Refreshing Mode
Decimal Values:	254	35
Hex Values:	0xFE	0x23

Receive Byte:	Decimal:	85
	Hex:	0x55

COMM Operator Examples:

254 35	Store Relay Refreshing Mode

NCD Component Library Command Method:

Not Yet Implemented

Report Back Stored Refresh Settings

This command reports back a 0 or 1, indicating whether automatic refreshing is Off or On when power is first applied to the controller.

Send Bytes:	**Byte 1:**	**Byte 2:**
Function:	Command	Report back refresh setting
Decimal Values:	254	36
Hex Values:	0xFE	0x24

Receive Byte:	Decimal:	0-1
	Hex:	0x00-0x01

COMM Operator Examples:

254 36	Report Back Stored Refresh Settings

NCD Component Library Command Method

Not Yet Implemented

Manual Refreshing Relay Banks

This command stores the current status of relay refreshing in non-volatile memory. The next time the relay controller is powered up, refreshing will be set to the stored state.

Send Bytes:	**Byte 1:**	**Byte 2:**
Function:	Command	Turn On/Off Relay Refreshing
Decimal Values:	254	37
Hex Values:	0xFE	0x25

Receive Byte:	Decimal:	85
	Hex:	0x55

COMM Operator Examples:

254 37	Manually Refresh Relay Bank

NCD Component Library Command Method:

Not Yet Implemented

Chapter

6

ProXR Standard Command Set

The ProXR Series Controllers support an extensive command set used to control relays, set operation modes, and store and recall relay status. Most users will only use a few of the functions built into this controller. The best way to learn the capabilities of the ProXR series is to carefully read through the command set. The "plain English" examples provide a quick, easy-to-understand definition of each command.

The number under 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 below.

Some commands require that you specify a bank value from 0-32, selecting the relay bank you would like to control. A value of 0 selects all banks. Bank values of 1-4 control up to 32 on-board relays. When powered up, the default bank value in the controller is always 1.

Controlling Individual Relays

These commands are used to turn a relay on or off under computer control.

Send Bytes:	**Byte 1:**	**Byte 2:**
Function:	Command	Turn Off Relay
Decimal Values:	254	0-7
Hex Values:	0xFE	0x00 – 0x0F

Send Bytes:	**Byte 1:**	**Byte 2:**	**Byte 3:**
Function:	Command	Turn Off Relay in Bank	Bank
Decimal Values:	254	100-107	0-32
Hex Values:	0xFE	0x00 – 0x0F	0x00 – 20

Send Bytes:	**Byte 1:**	**Byte 2:**
Function:	Command	Turn On Relay
Decimal Values:	254	8-15
Hex Values:	0xFE	0x00 – 0x0F

Send Bytes:	**Byte 1:**	**Byte 2:**	**Byte 3:**
Function:	Command	Turn On Relay in Bank	Bank
Decimal Values:	254	108-115	0-32
Hex Values:	0xFE	0x00 – 0x0F	0x00 – 0x20

Receive Byte:	Decimal:	85
	Hex:	0x55

COMM Operator Examples:

254 0	Turn Off Relay 0 in Selected Bank
254 100 1	Turn Off Relay 0 in Bank 1
254 8	Turn On Relay 0 in Selected Bank
254 108 2	Turn On Relay 0 in Bank 2

NCD Component Library Command Method:

Not Yet Implemented

Please review pages 9-11 to better understand how to direct commands to different relay banks.

Reading Status of Individual Relays

This command allows you to read the on/off status of an individual relay. 16 (or 116) corresponds to relay 1, 23 (or 123) corresponds to relay 8. This command will return a 1 indicating the relay is ON or a 0 indicating the relay is OFF.

Send Bytes:	**Byte 1:**	**Byte 2:**
Function:	Command	Get Relay Status
Decimal Values:	254	16-23
Hex Values:	0xFE	0x10 – 0x17

Send Bytes:	**Byte 1:**	**Byte 2:**	**Byte 3:**
Function:	Command	Get Relay Status	Bank
Decimal Values:	254	116 – 123	0-32
Hex Values:	0xFE	0x74 – 0x7B	0x00 – 0x20

Received Byte:	Decimal:	0 or 1
	Hex:	0x00 or 0x01

NOTE: A bank value of 0 is invalid for this command. Returned result may be unpredictable.

COMM Operator Examples:

254 16	Read the Status of Relay 0 in Selected Bank
254 116 2	Read the Status of Relay 0 in Bank 2
254 117 5	Read the Status of Relay 1 in Bank 5

NCD Component Library Command Method

Not Yet Implemented

Please review pages 9-11 to better understand how to direct commands to different relay banks.

Reading the Status of Relay Banks

This command allows you to read the status of a single bank of relays. A value of 0-255 is returned indicating the status of all 8 relays. The binary pattern of the value returned directly corresponds to the on/off status of each of the 8 relays in the selected relay bank. If the currently selected relay bank is 0, or if you specify relay bank 0, then the status of all 32 relay banks will be sent. In this condition, your program should be written the read 32 bytes of data from the serial port. The controller will report back 26 ASCII values from 0 to 255, indicating the status of all 26 relay banks, beginning with Bank 1. Convert returned values to binary to see actual relay pattern.

Send Bytes:	**Byte 1:**	**Byte 2:**
Function:	Command	Get the status of all relays
Decimal Values:	254	24
Hex Values:	0xFE	0x18

Send Bytes:	**Byte 1:**	**Byte 2:**	**Byte 3:**
Function:	Command	Get status all relays in bank	Bank
Decimal Values:	254	124	0-32
Hex Values:	0xFE	0x7C	0x00 – 0x20

Receive Byte:	Decimal:	26 ASCII values from 0-255
	Hex:	0x00 – 0xFF

COMM Operator Examples:

254 24	Read the Status of All 8 Relays in a Selected Bank
254 124 2	Read the Status of All 8 Relays in Bank 2
254 124 0	Read the Status of All 8 Relays in All 26 Banks

Please review pages 9-11 to better understand how to direct commands to different relay banks.

Reporting Mode

This command sets and stores (in non-volatile EEPROM while in configuration mode only) the reporting mode status Reporting mode, by default, is ON, meaning every time a command is send to the controller, the controller will send an 85 back to the computer, indicating that the command has finished executing your instructions. We recommend leaving it on, but doing so requires 2-way communication with the controller. You should turn it off it you intend to use 1-way communication only. A delay between some commands may be required when using 1-way communications. For optimum reliability, leave reporting mode on and use 2-way communications with the ProXR Series controllers.

Send Bytes:	**Byte 1:**	**Byte 2:**
Function:	Command	Get the status of all relays
Decimal Values:	254	27-28
Hex Values:	0xFE	0x1B – 0x1C

Receive Byte:	Decimal:	85
	Hex:	0x55

NOTE: Reporting Mode may be turned ON or OFF at any time. The default power-up of reporting mode is ONLY stored when the device is in configuration mode (all DIP switches of when the controller is powered up).

COMM Operator Examples:

254 27	Turn On and Store Reporting Mode in EEPROM
254 28	Turn Off and Store Reporting Mode in EEPROM

NOTE: Use other relay control commands to set relays to the desired power-up state before issuing this command. For instance, if you want all relays to come on at power-up, use other relay control commands to activate all relays, then issue this command. The next time power is applied to the controller, all relays will automatically activate.

NCD Component Library Command Method:

Not Yet Implemented

Please review pages 9-11 to better understand how to direct commands to different relay banks.

All Relays On/Off

This command is used to activate all relays or to turn all relays off in a selected relay bank.

Send Bytes:	**Byte 1:**	**Byte 2:**
Function:	Command	Turn relays On/Off
Decimal Values:	254	29-30
Hex Values	0xFE	0x1D – 0x1E

Send Bytes:	**Byte 1:**	**Byte 2:**	**Byte 3:**
Function:	Command	Turn relays On/Off in bank	Bank
Decimal Values:	254	129-130	0-32
Hex Values:	0xFE	0x81 – 0x82	0x20

Receive Byte:	Decimal:	85
	Hex:	0x55

COMM Operator Examples:

254 29	Activate All Relays in the Selected Relay Bank
254 129 2	Activate All Relays In Relay Bank 2
254 30	Deactivate All Relays in the Selected Relay Bank
254 130 2	Deactivate All Relays In Relay Bank 2

NOTE: A Bank Values of 0 applies this command to all relay banks.

NCD Component Library Command Method:

Not Yet Implemented

Please review pages 9-11 to better understand how to direct commands to different relay banks.

Inverting Relays

This command is used to Invert the status of all relays. Relays that were on, turn off, relays that were off, turn on.

Original Relay Pattern	0 ON	1 OFF	2 ON	3 ON	4 OFF	5 OFF	6 OFF	7 OFF
Inverted Relay Pattern	0 OFF	1 ON	2 OFF	3 OFF	4 ON	5 ON	6 ON	7 ON

Send Bytes:	**Byte 1:**	**Byte 2:**
Function:	Command	Invert all relays
Decimal Values:	254	31
Hex Values:	0xFE	0x1F

Send Bytes:	**Byte 1:**	**Byte 2:**	**Byte 3:**
Function:	Command	Invert all relays in bank	Bank
Decimal Values:	254	131	0-32
Hex Values:	0xFE	0x83	0x00 – 0x20

Receive Byte:	Decimal:	85
	Hex:	0x55

COMM Operator Examples:

254 31	Invert All Relays in the Selected Relay Bank
254 131 3	Invert All Relays In Relay Bank 3

NCD Component Library Command Method:

Not Yet Implemented

Please review pages 9-11 to better understand how to direct commands to different relay banks.

Reversing Relays

This command is used to reverse the current pattern of relays in a given relay bank. For example:

Original Relay Pattern	0 ON	1 OFF	2 ON	3 ON	4 OFF	5 OFF	6 OFF	7 OFF
Reversed Relay Pattern	0 OFF	1 OFF	2 OFF	3 OFF	4 ON	5 ON	6 OFF	7 ON

Send Bytes:	**Byte 1:**	**Byte 2:**
Function:	Command	Invert all relays
Decimal Values:	254	32
Hex Values:	0xFE	0x20

Send Bytes:	**Byte 1:**	**Byte 2:**	**Byte 3:**
Function:	Command	Invert all relays in bank	Bank
Decimal Values:	254	132	0-32
Hex Values:	0xFE	0x84	0x00 – 0x20

Receive Byte:	Decimal:	85
	Hex:	0x55

COMM Operator Examples:

254 32	Reverse All Relays in the Selected Relay Bank
254 132 3	Reverse All Relays in Relay Bank

NCD Component Library Command Method:

Not Yet Implemented

Please review pages 9-11 to better understand how to direct commands to different relay banks.

Test 2-Way Communication

This command is used to test 2-way communications between the PC and the relay controller. This command does nothing except report back an ASCII character code 85 when executed.

Send Bytes:	**Byte 1:**	**Byte 2:**
Function:	Command	Test 2-Way Communications with Relay Controller Command
Decimal Values:	254	33
Hex Values:	0xFE	0x21

Receive Byte:	Decimal:	85
	Hex:	0x55

NCD Component Library Command Method:

Not Yet Implemented

Reading the Currently Selected Relay Bank

This command queries the relay controller and reports back the relay bank all commands are being sent to. The importance of relay bank selection is discussed heavily on pages 9-11 of this manual. You can redirect relay control commands to a different relay bank by send '254 49 Bank' discussed later in this manual. Controller will report back ASCII Character Codes 0-26, indicating which relay bank commands will be directed to.

Send Bytes:	**Byte 1:**	**Byte 2:**
Function:	Command	Report Selected Relay Bank to User
Decimal Values:	254	34
Hex Values:	0xFE	0x22

Receive Byte:	Decimal:	0-26
	Hex:	0x00 – 0x1A

NCD Component Library Command Method:

Not Yet Implemented

Please review pages 9-11 to better understand how to direct commands to different relay banks.

Set the Status of a Relay Bank

This command writes a byte of data directly to a relay bank. This allows you to easily set the status of 8 relays at one time. RelayData is a parameter value from 0-255. A value of 0 turns off all the relays. A value of 255 turns on all the relays. Other values set the status of the relays in the equivalent binary pattern of the RelayData parameter value.

Send Bytes:	**Byte 1:**	**Byte 2:**	**Byte 3:**
Function:	Command	Set Status	Relay
Decimal Values:	254	40	0-255
Hex Values:	0xFE	0x28	0x00 – 0xFF

Send Bytes:	**Byte 1:**	**Byte 2:**	**Byte 3:**	**Byte 4:**
Function:	Command	Set Status	Relay	Bank
Decimal Values:	254	140	0-255	0-32
Hex Values:	0xFE	0x8C	0x00 – 0xFF	0x00 – 0x20

NOTE: A bank value of 0 applies this command to all relay banks.

Receive Byte:	Decimal:	85
	Hex:	0x55

COMM Operator Examples:

254 40 170	Set the Status of All Relays in the Selected Relay Bank
254 140 170 1	Set the Status of All Relays in Relay Bank 1

NCD Component Library Command Method:

Not Yet Implemented

Please review pages 9-11 to better understand how to direct commands to different relay banks.

Power Up Relay Status Configuration

This command stores the current status of the relays in a given bank into memory. The next time power is applied to the controller, relays will return to the stored on/off state. A bank value of 0 stores the pattern of all relays in all 26 banks.

Send Bytes:	**Byte 1:**	**Byte 2:**
Function:	Command	Store Power Up Default Status
Decimal Values:	254	42
Hex Values:	0xFE	0x2A

Send Bytes:	**Byte 1:**	**Byte 2:**	**Byte 3:**
Function:	Command	Store Power Up Default Status in Bank	Bank
Decimal Values:	254	142	0-32
Hex Values:	0xFE	0x8E	0x00 – 0x20

Receive Byte:	Decimal:	85
	Hex:	0x55

COMM Operator Examples:

254 42	Store Relay Settings in the Selected Relay Bank
254 142 2	Store Relay Settings in Relay Bank 2

NCD Component Library Command Method:

Not Yet Implemented

Please review pages 9-11 to better understand how to direct commands to different relay banks.

Read Power Up Relay Status Configuration

This command reads the stored power-up default status of the relays in a given bank. A bank value of 0 reports back 26 bytes of data, indicating the stored pattern of all 26 relay banks.

Send Bytes:	**Byte 1:**	**Byte 2:**
Function:	Command	Read Power Up Default Status
Decimal Values:	254	43
Hex Values:	0xFE	0x2B

Send Bytes:	**Byte 1:**	**Byte 2:**	**Byte 3:**
Function	Command	Read Power Up Default Status in Bank	Bank
Decimal Values:	254	143	0-32
Hex Values:	0xFE	0x8F	0x00 – 0x20

Receive Byte:	Decimal:	0-255
	Hex:	0x00 0xFF

COMM Operator Examples:

254 43	Read Relay Settings in the Selected Relay Bank
254 143 1	Read Relay Settings in Relay Bank

NCD Component Library Command Method:

Not Yet Implemented

Please review pages 9-11 to better understand how to direct commands to different relay banks.

Changing Relay Banks

This command is used to direct commands to a selected relay bank. All subsequent commands will be sent to the selected relay bank. This command only applies to command values less than 100. Commands in the 100+ value range allow you to specify a relay bank as part of the command.

Send Bytes:	**Byte 1:**	**Byte 2:**	**Byte 3:**
Function:	Command	Direct Commands to Selected Relay	Bank
Decimal Values:	254	49	0-32
Hex Values:	0xFE	0x31	0x00 – 0x20

Receive Byte:		
	Decimal:	85
	Hex:	0x55

COMM Operator Examples:

254 49 2	Direct Relay Control Commands To Relay Bank 2
254 8	Activate Relay 0 on Bank 2 (Bank is Selected in Above Command)

NCD Component Library Command Method:

Not Yet Implemented

Please review pages 9-11 to better understand how to direct commands to different relay banks.

Chapter 7

Helpful Routines

Helpful Routines and Safe Programming Practices

There are two helpful routines you may want to structure into your program that help handle serial communications. First, the *ClearBuffer* routine can be called prior to sending each command to the relay controller, helping ensure the serial buffers is empty and the new command is not returning old data to your program. This has been a problem for some customers, and it is easily avoided. While this is not always necessary, it should be a troubleshooting starting point if the controller seems to be sending you invalid data.

Second, the *GetData* routine should be used to read data from the controller. The simple routine waits for data to arrive in the serial buffer. Once it arrives, it grabs the data and stores the value in the *GetData* variable.

Sample Code:

Clearing the Serial Buffer

```
Public Function ClearBuffer()          ‘Read a Byte of Data from the Controller in VB
   While MSComm1.InBufferCount > 0     ‘If there is data in the serial port
      DUMP = Asc(MSComm1.Input)        ‘Read Data Byte from the Serial Port
      DoEvents                         ‘Service Other Windows Tasks (Very Important!!!)
   Wend                                ‘Repeat the Check Again, Loop will Repeat until Empty
End Function
```

Reading a Byte of Data from the PC

```
Public Function GetData()              ‘Read a Byte of Data from the Controller in VB
   Do                                  ‘Start a Continuous Loop
      DoEvents                         ‘Service Other Windows Tasks (Very Important!!!)
   Until MSComm1.InBufferCount > 0     ‘Continue Loop Until a Byte of Data is Received
   GetData = ASC(MSComm1.Input)        ‘Read Data Byte from the Serial Port
End Function
```

Chapter

8

Relay Timers

Introduction

Relay Timer Commands

The ProXR Series controllers have 16 user-programmable timers. Each independent timer can be assigned to any relay, and can be programmed to hold the relay in the On state, or to pulse the relay at the end of the timer. The ProXR timing features are ideally suited for Watchdog, Keep Alive, and Server Reboot applications, as well as sprinkler systems, gate openers, and day/night lighting applications.

Relay Timing Features support two modes of operation:

Duration and *Pulse*. Duration timing is ideally suited for keeping a light on overnight, watering the lawn for a given period of time, or other applications where a device should be activated for a period of time. Pulse timing mode is designed specifically for server reboot applications, whereby, if the timer is not reset periodically by your software, the timer will run out and reboot your computer.

Interactive Timing Commands

The ProXR timing commands can be used by themselves, or in conjunction with other commands as building blocks to create some very sophisticated timing applications. The timing command set covers many aspects of relay activation/deactivation; make the ProXR series ideally suited for a broad range of timing tasks.

Limitations

Unless otherwise noted, the ProXR series controllers do NOT have an integrated real-time clock. NCD Devices are not typically stand-alone. They require computer interaction with the controller. Time scheduling is possible, but it would require a program to be written on the PC to handle the schedule. The timing features are suitable for applications where you may want a light to go on for 5 minutes, or you may want to keep a relay alive to prevent a server from automatically rebooting. The ProXR series controllers are capable of processing timing commands as long as 255 hours, 255 minutes, and 255 seconds. (4days, 19 Hours, 19 Minutes, and 15 Seconds) + Deviation.

Timing Accuracy

The Accuracy of the relay timers is dependent on many factors, but ProXR Enhanced controllers use timer interrupts to help improve timing accuracy. However, the possibility does exist that timers may drift slightly as it is not possible to generate an exact second without additional electronics.

When a timer is already active, and you engage another timer, the duration of the previously set times may be increased by as much as one full second. You can enable all timers simultaneously if you need more accurate timing.

Best timing accuracy is achieved by setting up your timing commands and leaving the controller alone during the timing operations. Each time you communicate with the controller, you will slow down the timer (lengthening the time period the timer is set for). The more you communicate with the controller, the more you will slow down all timers. Timing accuracy tends to drift over time. The timing functions built into this controller should NOT be used if timing accuracy is critical. The timing features are, however, very useful in applications where a little timing drift is not a big concern.

Timing Calibration

Timing is generically recalibrated for 60 seconds using 8 timers. Our test controller calibration value was 26,576. In other words, a calibration value of 26,576 equals 1 second when the controller is only processing timing tasks.

The calibration value was established on our prototype and may be off by as much as 3% based on individual resonator, processor, and temperature characteristics. Baud rate was set at 115.2K when this number was established. The calibration value may need to be changed for other baud rates, but 115.2K baud is the best choice for calibration.

You can adjust the calibration value at any time, but the calibration value can ONLY be stored while in setup mode. If you need to communicate frequently with the controller while the timing functions are active, you will need to decrease the calibration value. Reasonably accurate timing can be achieved with some experimentation.

It should also be stated that you can spend a week achieving perfect calibration for your controller, and if you were to plug the calibration number into a different controller, it will likely not be accurate.

In addition, the timing routines built into the firmware are huge. There are so many factors that affect timing, that even a well calibrated controller will not post consistent timing scores at all timing intervals. So you should NEVER expect to find a calibration value that works under every circumstance. It is not possible to achieve this level accuracy without a real time clock. So before you waste hours finding a timing score for your controller that works perfectly at 10 seconds or 24 hours, then you should be warned that this is not possible.

Chapter

9

ProXR Relay Timer Command Set

Getting Started with Simple Timers

While timing commands are pretty easy to use, simple timers are the easiest. Once you have sent a simple timer command, the timer automatically starts counting down.

There are two types of simple timers: Duration and Pulse.

Simple Duration Timers

These timers activate a relay for a user specified period of time. When the timer expires, the relay turns off. Duration Timer 50-65 controls timers 0-15. The relay is active during the duration of the timer and turns off when timer counts down to 0 Hours, 0 Minutes, 0 Seconds. Relay is a value from 0-255, as timers may be applied only to the first 256 relays of the controller. Here is an example sending a simple duration timer command:

Send Bytes:	**Byte 1:**	**Byte 2:**	**Byte 3:**	**Byte 4:**	**Byte 5:**	**Byte 6:**	**Byte 7:**
Function:	Command	Timer Setup	Timer	Hours	Minutes	Seconds	Relay
Decimal Values:	254	50	50-65	0-255	0-255	0-255	0-255
Hex Values:	0xFE	0x32	0x32–0x41	0x00–0xFF	0x00–0xFF	0x00–0xFF	0x00–0xFF

Receive Byte:	Decimal:	85
	Hex:	0x55

COMM Operator Examples:

254 50 50 8 10 15 0	Hold Relay 0 On for 8 Hours, 10 Minutes, 15 Seconds using Simple Duration Timer 0
254 50 51 0 0 10 1	Hold Relay 1 On for 10 Seconds using Simple Duration Timer 1

When the above two commands have been sent, both relays 0 and 1 will turn on. Relay 0 will turn off after 8 hours, 10 minutes, and 15 seconds. Relay 1 will turn off after only 10 seconds. While the timers are running, you may send other relay control commands. It is also possible to manually turn off the relays while the timers are still running. In these cases, the timers will not appear to have any effect. You can also pause the timers using other commands.

Keep in mind; you have 16 timers to work with. If you ever need this many timers, it would be prudent to assign a different relay to each timer. Assigning the same relay to 2 timers will cause the relay to turn off when the first timer expires. The second timer will appear to have no effect.

Also note that relays are assigned in numeric order of 0-255 when using the timing commands. Relay 0 is located on Bank 1, Relay 0. Relay 8 is located on Bank 2, Relay 0. Relay 255 is located on bank 32, Relay 7 (*you will have to make use of the XR Expansion port to access this relay*).

NCD Component Library Command Method:

Not Yet Implemented

Simple Pulse Timers

Pulse Times are slightly different than duration timers. When a pulse timer is activated, the relay will not do anything until the timer has expired. Once expired, the relay will pulse for a short duration. This pulse is designed specifically to reboot a computer by connection a relay directly to the RESET lines of a motherboard. While this may be used for other applications, the intent of the pulse timer is the reboot a computer should there be a lack of communication between the computer and the relay controller (indicative of a system crash).

Pulse Timer 70-80 controls times 0-15. The timer counts and when it expires, the relay is pulsed. Set Hours, Minutes, and Seconds to determine how long the timer will hold the relay on. Relay is a value from 0-255, as timers may be applied only to the first 256 relays of the controller. Below is a simple example of setting up a pulse timer.

Send Bytes:	**Byte 1:**	**Byte 2:**	**Byte 3:**	**Byte 4:**	**Byte 5:**	**Byte 6:**	**Byte 7:**
Function:	Command	Timer Setup	Timer	Hours	Minutes	Seconds	Relay
Decimal Values:	254	50	70-85	0-255	0-255	0-255	0-255
Hex Values:	0xFE	0x32	0x46-0x55	0x00-0xFF	0x00-0xFF	0x00-0xFF	0x00-0xFF

Receive Byte:	Decimal:	85
	Hex:	0x55

COMM Operator Examples:

254 50 70 0 0 15 0	Pulse Relay 0 after 15 Seconds using Simple Pulse Timer 0
254 50 71 0 0 45 1	Pulse Relay 1 after 45 Seconds using Simple Pulse Timer 1

In the examples above, Relay 0 will pulse after 15 seconds and Relay 1 will pulse after 45 seconds.

NCD Component Library Command Method:

Not Yet Implemented

Mixing Duration and Pulse Timers

You can mix duration and pulse timers are your application requires, in any combination. Care should be taken not to mix timers. For example, in our first sample, we utilized timer 0 and 1 using the commands 50 and 51. In our second sample, we utilized timers 0 and 1 using the command 70 and 71. The commands 50 and70 both use timer 0. Likewise, the commands 51 and 71 use timer 1. Here is a simple overlap map that will help you keep track of what commands address specific timers. The table below shows the beginning command bytes:

Timer Number	Duration Timer	Pulse Timer	Duration Timer	Pulse Timer
0	254, 50, 50	254, 50, 70	254, 50, 90	254, 50, 110
1	254, 50, 51	254, 50, 71	254, 50, 91	254, 50, 111
2	254, 50, 52	254, 50, 72	254, 50, 92	254, 50, 112
3	254, 50, 53	254, 50, 73	254, 50, 93	254, 50, 113
4	254, 50, 54	254, 50, 74	254, 50, 94	254, 50, 114
5	254, 50, 55	254, 50, 75	254, 50, 95	254, 50, 115
6	254, 50, 56	254, 50, 76	254, 50, 96	254, 50, 116
7	254, 50, 57	254, 50, 77	254, 50, 97	254, 50, 117
8	254, 50, 58	254, 50, 78	254, 50, 98	254, 50, 118
9	254, 50, 59	254, 50, 79	254, 50, 99	254, 50, 119
10	254, 50, 60	254, 50, 80	254, 50, 100	254, 50, 120
11	254, 50, 61	254, 50, 81	254, 50, 101	254, 50, 121
12	254, 50, 62	254, 50, 82	254, 50, 102	254, 50, 122
13	254, 50, 63	254, 50, 83	254, 50, 103	254, 50, 123
14	254, 50, 64	254, 50, 84	254, 50, 104	254, 50, 124
15	254, 50, 65	254, 50, 85	254, 50, 105	254, 50, 125

**These timers do not automatically activate, they must be started using a different command.*

Server Reboot Methodology

You can call it a watchdog timer, a keep-alive timer, or a server reboot timer. They can all mean about the same thing, as their goals are basically the same. The idea is simple: If the computer crashes, the computer cannot reset the timer built into the ProXR controller, so the controller reboots the computer reboots the computer. Implementation is not too difficult.

Implementing a Server Reboot strategy for a Single Computer

A Server reboot system can work many ways. One possible strategy is a system whereby a server would boot up with a ProXR relay controller attached to the serial port. The relay controller would also be connected to the reset lines of server motherboard. As part of the startup items, a program would be launched to activate the pule timer function for a period of 10 minutes (*for example*). The relay would do nothing since a pulse timer is used. Using this strategy, the relay controller would reboot the computer if communications is lost between the server and the relay controller. Once the timer in the relay controller has expired, it can only be restarted when the computer boots up normally. The monitoring program could be exited at any time. In which case, all timers would be cleared to prevent rebooting the computer.

Implementing a Server Reboot Strategy in a Network

The strategy above could be implemented on a single computer with an enhanced version of the software. The relay controller could be tied into the reset lines on the other computers as well. The program could be enhanced to "ping" other computers on the network. If one of them should fail to respond to your "ping", a command could be sent to reboot the computer that failed to respond. In this case, one computer (*a main server*) is protected, as well as all other computers on the network. The main server is acting as the watchdog for all the other computers on the network. The relay controller itself is acting as the watchdog for the main server computer.

While there are many other strategies that could be easily implemented, these strategies could perhaps serve as building blocks to greater, more powerful and sophisticated watchdog monitoring applications.

Duration Timers

Duration Timer 90-105 controls timers 0-15. The Relay is active during the duration of the timer. This command sets up the timer only, it does NOT begin to start. Use a separate command to control when this timer starts. The relay turns off when timer counts downs to 0 Hours, 0 Minutes, 0 Seconds. Hours, Minutes, and Seconds sets the number of hours, minutes, and seconds the timer will hold the relay on. Relay is a value from 0-255, as timers may be applied only to the first 256 relays of the controller.

Send Bytes:	**Byte 1:**	**Byte 2:**	**Byte 3:**	**Byte 4:**	**Byte 5:**	**Byte 6:**	**Byte 7:**
Function:	Command	Timer Setup	Timer	Hours	Minutes	Seconds	Relay
Decimal Values:	254	50	90-105	0-255	0-255	0-255	0-255
Hex Values:	0xFE	0x32	0x5A-0x69	0x00-0xFF	0x00-0xFF	0x00-0xFF	00x0-0xFF

Receive Byte:	Decimal:	85
	Hex:	0x55

COMM Operator Examples:

254 50 91 0 0 35 1 Setup a Duration Timer on Relay 1 for 35 Seconds using Timer 1

NCD Component Library Command Method:

Not Yet Implemented

Pulse Timers

Pulse Timers 110-125 controls timers 0-15. The timer counts and when it expires, the relay is pulsed. This command sets up the timer only, it does NOT begin to start. Use a separate command to control when this timer starts. Set Hours, Minutes, and Seconds to determine how long the timer will hold the relay on. Relay is a value from 0-255, as timers may be applied only to the first 256 relays of the controller. Below is a simple example of setting up a pulse timer.

Send Bytes:	**Byte 1:**	**Byte 2:**	**Byte 3:**	**Byte 4:**	**Byte 5:**	**Byte 6**	**Byte 7:**
Function:	Command	Timer Setup	Timer	Hours	Minutes	Seconds	Relays
Decimal Values:	254	50	110-125	0-255	0-255	0-255	0-255
Hex Values:	0xFE	0x32	0x6E-0x7D	0x00-0xFF	0x00-0xFF	0x00-0xFF	0x00-0xFF

Receive Byte:	Decimal:	85
	Hex:	0x55

COMM Operator Examples:

254 50 110 0 0 15 0 Setup a Pulse Timer on Relay 0 for 15 Seconds using Timer 0

NCD Component Library Command Method:

Not Yet Implemented

Query Remaining Time

This command will query the time remaining for the selected timer 1 through 16. This command reports 4 bytes back to the user, indicating Hours remain (0-255), Minutes remaining (0-255), Seconds remaining (0-255), and the Relay Number the timer is assigned to (0-255).

Send Bytes:	**Byte 1:**	**Byte 2:**	**Byte 3:**	**Byte 4:**
Function:	Command	Timer setup		Timer
Decimal Values:	254	50	130	1-16
Hex Values:	0xFE	0x32	0x82	0x01-0x10

Receive Byte:	Decimal:	0-255 (hours)	0-255 (minutes)	0-255 (seconds)	0-255 (relay)
	Hex:	0xFF	0xFF	0xFF	0xFF

NCD Component Library Command Method:

Not Yet Implemented

Halt or Resume Timers

This command is used to manually halt or resume all 16 timers. This command works with all forms of timers. The LSB and MSB are the least significant and most significant bytes in the 16-bit in a 16-bit word. The status of each bit within the 16-bit word is used to control which timers are running. Every bit that is high in the word indicates the timer is active. Every bit that is low in the word indicates the timer is not running.

Send Bytes:	**Byte 1:**	**Byte 2:**	**Byte 3:**	**Byte 4:**	**Byte 5:**
Function:	Command	Timer Setup		LSB	MSB
Decimal Values:	254	50	131	0-255	0-255
Hex Values:	0xFE	0x32	0x83	0x00-0xFF	0x00-0xFF

Receive Byte:	Decimal:	85
	Hex:	0x55

COMM Operator Examples:

254 50 131 3 0 — Begin Countdown of Timers 0 and 1
LSB = 3 (1 + 2 = 3 1 is for Timer 0, 2 is for Timer 1, See Table on next page)
MSB = 0 (Keep all other timers off)

254 50 131 1 0 — Indicates timer 1 is running, all others are off

254 50 131 3 128 — Indicates timers 1, 2, and 16 are running (1 + 2 = and 128 in the MSB indicates Timer 16 is running)

NCD Component Library Command Method:

Not Yet Implemented

LSB/MSB 16-Bit Value

If you understand how binary works, this is a pretty simple command. A 16 bit value is used to control which timers are active and which timers are halted. Each of the 16 bits identifies with each of the 16 timers. A binary 0 in any bit location indicates the timer is off while a binary 1 in any bit location indicates the timer is on. If you are not familiar with binary, here is a crash course:

16 Timers have 16 Bits, but we have to divide these into two 8-Bit values to communicate these data via a serial port. We call these two different bytes LSB for Least Significant Byte and MSB for Most Significant Byte.

Follow the Table below to figure LSB and MSB Values:

LSB Values	**MSB Values**
Timer 0 has a value of 1 on the LSB	Timer 8 has a value of 1 on the MSB
Timer 1 has a value of 2 on the LSB	Timer 9 has a value of 2 on the MSB
Timer 2 has a value of 4 on the LSB	Timer 10 has a value of 4 on the MSB
Timer 3 has a value of 8 on the LSB	Timer 11 has a value of 8 on the MSB
Timer 4 has a value of 16 on the LSB	Timer 12 has a value of 16 on the MSB
Timer 5 has a value of 32 on the LSB	Timer 13 has a value of 32 on the MSB
Timer 6 has a value of 64 on the LSB	Timer 14 has a value of 64 on the MSB
Timer 7 has a value of 128 on the LSB	Timer 15 has a value of 128 on the MSB

To Turn On timers, add up the LSB and MSB Values.

For example:

To turn on timers 0, 1, 2, and 3 we add up 1, 2, 4, and 8.
So the LSB = 15.

To turn on timers 10, 12, 14, and 15, we add up 4, 16, 64, and 128.
So the MSB = 212.

After you send the LSB and MSB timer data to the controller, the selected timers will be activated. All other timers will be halted.

Calibrate the Relay Timer

The command will calibrate the relay timer. The LSB and MSB make up a word, indicating the speed of the timer. Lower values indicate a faster timer while higher values indicate a slower timer. This command no-longer stores the calibration data into the controller. Use device configuration to store a new calibration value. This command is valuable for experimenting with other calibration values.

Send Bytes:	**Byte 1:**	**Byte 2:**	**Byte 3:**	**Byte 4:**	**Byte 5:**
Function:	Command	Timer Setup		LSB	MSB
Decimal Values:	254	50	132	0-255	0-255
Hex Values:	0xFE	0x32	0x84	0x00-0xFF	0x00-0xFF

Receive Byte:		
	Decimal:	85
	Hex:	0x55

NCD Component Library Command Method:

Not Yet Implemented

Retrieve the Timer Calibration Values

This command retrieves the Timer Calibration Value. The Timer Calibration Value controls the actual length of a second. Two bytes will be returned by this command, LSB and MSB. The Word value of the timer = LSB + (MSB*256). Decreasing this value will speed up the timer, increasing this value will slow it down. Use device configuration commands 254, 53 and 254, 54 to store and retrieve these data out of EEPROM.

Send Bytes:	**Byte 1:**	**Byte 2:**	**Byte 3:**
Function:	Command		
Decimal Values:	254	50	133
Hex Values:	0xFE	0x32	0x85

Receive Byte: Timer Calibrator LSB Value
Timer Calibrator MSB Value

NCD Component Library Command Method:

Not Yet Implemented

Turn On Timing Calibration Markers

This command turns on the timing calibration markers. This command is not generally used by software developers, but is used by Base application software to help the user measure the elapsed time of the timer values.

Send Bytes:	**Byte 1:**	**Byte 2:**	**Byte 3:**
Function:	Command	Timer Setup	
Decimal Values:	254	50	134
Hex Values:	0xFE	0x32	0x86

Receive Byte:	Decimal:	85	90 (Start)	91(Stop)
	Hex:	0x55	0x5A	0x5B

NCD Component Library Command Method:

Not Yet Implemented

Turn Off Timing Calibration Markers

This command turns off the timing calibration markers. This command is not generally used by software user by software developers, but is used by Base application software to help the user measure the elapsed time of the timer values.

Send Bytes:	**Byte 1:**	**Byte 2:**	**Byte 3:**
Function:	Command	Timer Setup	
Decimal Values:	254	50	135
Hex Values:	0xFE	0x32	0x87

Receive Bytes:	Decimal:	85
	Hex:	0x55

NCD Component Library Command Method:

Not Yet Implemented

Chapter

10

Advanced Configuration Commands

Fighting Electromagnetic Interference (EMI)

Any relay controller has to consider EMI as a possible mechanism that can cause serious interference with normal operation. In extreme cases, EMI can cause the relays in a given relay bank to turn off, or even reboot the ProXR processor. /we have taken several steps to help reduce EMI absorption on the logic side of the controller. However, there are some conditions that MAY still exist whereby EMI will cause all relays on a given bank to turn off.

When relay status information is sent from the controller to the relay bank, data is normally sent only one time. But the possibility does exist that the relay is turning on a device that generates a tremendous amount of EMI. Most of these kinds of devices, such as large DC motors, generate most of their EMI when power is first applied. During this time (the initial startup of the motor), huge amounts of EMI must be absorbed by the relays and inducted back onto the logic of the controller. It is during this time the controller becomes the most vulnerable to EMI absorption.

It is for this reason we have introduced the Repetitions (Reps) command. The Reps command is used to tell the controller to send relay control data repeatedly to the relay banks. The theory is that motor will be fighting to turn on, generating EMI that can clear the relay logic circuits, but the controller can now fight back by telling the relay bank to go back on again.

Reps has been introduced as an experimental command, in hopes that we would have a solution for this problem should it ever arise for our customers. We would greatly appreciate any feedback our customers would be willing to provide on the usefulness of the Reps command. The value specified by the Reps command controls how many times relay status data is sent to the Relay control chips. Normally, Reps is set to a value of 1. But if you experience relays that turn themselves off when activated, try changing the Reps value to any number from 2 to 255 and see what happens. It takes more than a second to process each relay control command when Reps is set to 255.

NOTE: If the controller is in configuration mode, the Reps value will be stored in Non-Volatile EEPROM. Otherwise, the Reps value will be lost any time the controller is power cycles.

Set the Reps Value

Send Bytes:	**Byte 1:**	**Byte 2:**	**Byte 3:**	**Byte 4:**
Function:	Command	Timer Setup	Set Reps	Reps
Decimal Values:	254	50	137	1-255
Hex Values:	0xFE	0x32	0x89	0x01-0xFF

Receive Byte:	Decimal:	85
	Hex:	0x55

Read the Reps Value

This command reports back to the user the current Reps value.

Send Bytes:	**Byte 1:**	**Byte 2:**	**Byte 3:**
Function:	Command	Timer Setup	Read Reps
Decimal Values:	254	50	136
Hex Values:	0xFE	0x32	0x88

Receive Byte: Current Reps Value

Safe Parameters

Another "Just-in-case" feature we have included in the ProXR firmware is a command that sets all the key variables inside the controller to Safe Parameters. If for some reason the controller does not seem to be functioning properly, or it appears you have lost communication, try sending this command and see if it recovers. We have never needed this command, but just in case it proves to be useful under highly unusual circumstances…we thought we would throw it in anyway.

Send Bytes:	**Byte 1:**	**Byte 2:**	**Byte 3:**
Function	Command	Timer Setup	Setup Safe Parameters
Decimal Values:	254	50	147
Hex Values:	0xFE	0x32	0x93

Receive Byte:	Decimal:	85
	Hex:	0x55

NCD Component Library Command Method:

Not Yet Implemented

Setting the Character Delay Value

The Character Delay value determines the spacing between data bytes sent from the controller back to your PC. By default, CDEL is set to 35, which is known to be a conservatively safe value. The minimum allowed CDEL value is 3 (we have not seen a PC that can handle this setting). If you want to boost performance, set the CDEL lower. A value of 7-10 should provide greatly improved communication speed.

NOTE: This command stores the CDEL value into NON-Volatile EEPROM while in configuration mode only. The CDEL value will have no effect in configuration mode (CDEL is ALWAYS 35 in configuration mode to ensure safe communications). This command is only processed in configuration mode; it is ignored in runtime mode. In runtime mode, you will receive ASCII character code 85 if this command is issued.

Send Bytes:	**Byte 1:**	**Byte 2:**	**Byte 3:**	**Byte 4:**
Function:	Command	Timer Setup	Set the CDEL	Value
Decimal Values:	254	50	139	7-10
Hex Value:	0xFE	0x32	0x8B	0x07-0x0A

Receive Byte:	Decimal:	85
	Hex:	0x55

NCD Component Library Command Method:

Not Yet Implemented

Reading the Character Delay Value

This command reads the stored character delay value from the controller. This command returns a value from 3 to 255 indicating the Character delay value.

Send Bytes:	**Byte 1:**	**Byte 2:**	**Byte 3:**
Function:	Command	Timer Setup	Read the CDEL Value
Decimal Values:	254	50	138
Hex Values:	0xFE	0x32	0x8A

Receive Byte:	Decimal:	3-255
	Hex:	0x03-0xFF

NCD Component Library Command Method:

Not Yet Implemented

Setting the Attached Banks Value

The ProXR series controllers are shipped as though the customer has 254 relays attached. This allows the controller to be instantly compatible with several XR expansion boards, right out of the box. However, you can gain a performance increase by setting the Attached Banks value to the actual number of relays you are using. If you are using a 16 Relay ProXR controller, and you reduce the Attached Banks value to 2, relay control operation will be processed 16 times faster. If you have a 32 relay controller and you reduce the Attached Banks value to 4, relay control operations will be processed 8 times faster. To determine the appropriate Attached Banks value, divide the number of total relays you will be using by 8. Valid attached bank values are 1 to 32.

WARNING: It is not possible to control relays beyond the Attached Value. Extra relays will appear to mirror previous banks and will not be directly controllable. If you experience any problems communicating to extra relay banks, the Attached Banks value should be checked. This Command ONLY Works in Configuration Mode.

Send Bytes:	**Byte 1:**	**Byte 2:**	**Byte 3:**	**Byte 4:**
Function:	Command	Timer Setup	Set Attached Banks	Banks
Decimal Values:	254	50	141	1-32
Hex Values:	0xFE	0x32	0x8D	0x01-0x20

Receive Byte:	Decimal:	85
	Hex:	0x55

COMM Operator Example:

254 50 141 4 Controller processes 4 banks of 8 relays (32 relays)

NCD Component Library Command Method:

Not Yet Implement

Reading the Attached Banks Value

This command reports back a value from 1 to 32, indicating how many relays are attached to the relay controller. Note that this number is always set by the user. By default, this command returns a value of 32. You can gain a performance increase by reducing the Attached Banks value. *See Above:* **Seeing the Attached Banks Value**

Send Bytes:	**Byte 1:**	**Byte 2:**	**Byte 3:**
Function:	Command	Timer Setup	Read Attached Banks Value
Decimal Values:	254	50	140
Hex Values	0xFE	0x32	0x8C

Receive Byte:	Decimal:	1-32(by default reports 32)
	Hex:	0x01-0x20

NCD Component Library Command Method:

Not Yet Implemented

Setting the Test Cycle Value

When the ProXR series controllers are powered up in Configuration mode (all DIP switches in the off position), the first instruction the controller processes is a function that tests all the onboard relays. By default, the ProXR controller will test 4 banks of relays, which is the maximum number of relays that we currently produce on a single circuit board. If you have a controller with fewer relays, you will speed up the test cycle by setting the controller to the actual number of relay banks on your ProXR controller. Valid Test Cycle values range from 0 to 32, allowing the controller to test all 256 relays (32 banks of 8). A Test Cycle value of 0 turns off automatic relay testing in configuration mode. Since this is a relatively harmless setting, this command works in both configuration and run modes.

Warning: We strongly encourage customers NOT to use configuration mode for daily use. As a reminder that the controller is in configuration mode, we recommend leaving the Test Cycle value to 1 or more. When the controller is in configuration mode, internal memory is not protected, and is fully changeable at any time. Runtime mode (and DIP switch setting other than all switches in the off position) protects internal memory from accidental modification.

Warning: The ProXR controller will appear to lock up if the Test Cycle value is set greater than the total number of available relay banks. Rest assured, the controller has NOT locked up, it is testing extended banks. Leave the controller on for several seconds and the heartbeat LED will begin flashing.

Send Bytes:	**Byte 1:**	**Byte 2:**	**Byte 3:**	**Byte 4:**
Function:	Command	Timer Setup	Set Test Cycle	Banks
Decimal Values:	254	50	146	0-32
Hex Values:	0xFE	0x32	0x92	0x00-0x20

Receive Byte:		
	Decimal:	85
	Hex:	0x55

COMM Operator Example:

254 50 146 4   Controller Test 4 Banks of 8 Relays (32 Relays)

NCD Component Library Command Method:

Not Yet Implemented

Reading the Test Cycle Value

This command reports back a value from 1 to 32, indicating how many relays are tested when the controller is powered up in configuration mode (all DIP switches in the off position). Note that this number is always set by the user. By default, this command returns a value of 4. You can gain a performance increase by reducing the Test Cycle Banks value. *See Page Above:* ***Setting the Test Cycle Banks Value.***

Send bytes:	**Byte 1:**	**Byte 2:**	**Byte 3:**
Function:	Command Timer Setup		‘Read the Attached Banks Value
Decimal Values:	254	50	145
Hex Values:	0xFE	0x32	0x91

Receive Byte:	Decimal:	0-32 (by default will report back 4)
	Hex:	0x00-0x20

NCD Component Library Command Method:

Not Yet Implemented

Restoring Factory Default settings

This command operates in configuration mode only (all DIP switches in the off position). This command restores many internal parameters to the factory default settings. See below for a detailed list:

E3C Device Number is Set to 0
Auto Refresh is Activated
Default Timing is Set to 254, 576
Relay Command repetitions is Set to 1
Character Delay is Set to 35
Reporting Mode is Turned On
Attached Relay Banks is Set to 32

Send Bytes:	**Byte 1:**	**Byte 2:**	**Byte 3:**
Function:	Command	Timer Setup	Restore Factory Default Settings
Decimal Values:	254	50	144
Hex Values:	0xFE	0x32	0x90

Receive Byte:	Decimal:	85
	Hex:	0x55

NCD Component Library Command Method:

Not Yet Implemented

Chapter

11

ProXR Standard Command Set Summary

The ProXR Standard for Data Collection and Control

By now, you have seen ProXR all over our website. Just about every controller we currently make is a ProXR controller. You will find ProXR in relays, inputs, outputs, potentiometers, analog to digital conversion, and much more. Put simply, ProXR is a standardized set of commands. When you choose a ProXR controller, you can upgrade that controller without having to re-write your software. One of the greatest benefits to ProXR is that you can develop your software to work with (for instance) and RS-232 device. Later on, you may find you don't want to use RS-232 anymore. Maybe you want to talk to the device wirelessly, or maybe use an Ethernet, Wi-Wi, USB, or Bluetooth interface. No Problem. You don't have to re-write your program. You can simply use the communication technology that most benefits your application WITHOUT having to redevelop your software.

ProXR is rooted in simplicity just as much as it is standardization. Most people who visit our website are programmers who need to integrate a custom hardware application with a minimal learning curve. If you are a programmer, then you are ready to dive right in and start communicating with our devices. However, we also work with several customers who have NO PROGRAMMING EXPERIENCE. We take pride in making sure they are brought up to speed with a simple set of tutorials. So whether you are a programmer or an electrician, we can help get you started. We move at a pace you can easily follow, so you don't have to worry too much about steep learning curves. We put a lot of effort into making our devices easy to use and standardized, while supporting all of today's popular communication technologies at a pace anyone can understand.

Before diving into the complete ProXR Standard Commend Set, there are three reading prerequisites:

1. Our articles will introduce you to various technologies, and you should read the article that pertains to your favored communication technology first. Click here to see our articles
2. Review our Tutorials and Quick Start Guides; they will give you everything you need for first time users to configure a programming language to speak to our devices.
3. Our ProXR Manual gives you a lot of background details on our robust and growing ProXR command set. Use it as a reference, even if you are an experienced programmer.

The purpose of the following section is to give you convenient location to review a summary of ProXR commands. The list will only grow as our products evolve. This list divided into 4 simple categories: Input, Parameters, Descriptions, and Output.

Reading the Table	
Command	These are the bytes that you send to the controller. These bytes are shown in decimal format, and can be converted to HEX if you prefer. We use the comma character to separate our bytes, but you should NOT send the comma character to the device. Also, you do NOT need to send a enter or return to complete the command. The controller knows when the command is complete.
Parameters	Some commands need parameters, such as hours, minutes, and seconds. Other parameters include a Bank value, which indicates which group of relays you will be speaking to.
Command Description	While there is no substitute for reading the ProXR manual, these descriptions give you a basic guideline of what to expect from the command.
Response	ProXR controllers will respond to most commands that you send. Under normal operation (when the controller is in runtime mode) the controller will respond with an 85 for most commands. If the controller happens to be in configuration mode (a jumper setting on the controller), the controller will respond with 86. Some commands, such as Analog to Digital Conversion, report 8-bit values from 0-255. In some cases, 2 or more bytes are sent back to the user. These will be noted in the Response column.

Relay Control Commands

These commands are used to activate relays.

Command	Parameters	Command Description	Response
What You Send	Values	What Controller Does	What Controller Send Back to You
254,0	None	Turns Off Relay 0 in the Currently Selected Relay Bank	85
254,1	None	Turns Off Relay 1 in the Currently Selected Relay Bank	85
254,2	None	Turns Off Relay 2 in the Currently Selected Relay Bank	85
254,3	None	Turns Off Relay 3 in the Currently Selected Relay Bank	85
254,4	None	Turns Off Relay 4 in the Currently Selected Relay Bank	85
254,5	None	Turns Off Relay 5 in the Currently Selected Relay Bank	85
254,6	None	Turns Off Relay 6 in the Currently Selected Relay Bank	85
254,7	None	Turns Off Relay 7 in the Currently Selected Relay Bank	85
254,8	None	Turns On Relay 0 in the Currently Selected Relay Bank	85
254,9	None	Turns On Relay 1 in the Currently Selected Relay Bank	85
254,10	None	Turns On Relay 2 in the Currently Selected Relay Bank	85
254,11	None	Turns On Relay 3 in the Currently Selected Relay Bank	85
254,12	None	Turns On Relay 4 in the Currently Selected Relay Bank	85
254,13	None	Turns On Relay 5 in the Currently Selected Relay Bank	85
254,14	None	Turns On Relay 6 in the Currently Selected Relay Bank	85
254,15	None	Turns On Relay 7 in the Currently Selected Relay Bank	85
254,16	None	Report the Current On/Off Status of Relay 0 in the Currently Selected Relay Bank	0 or 1
254,17	None	Report the Current On/Off Status of Relay 1 in the Currently Selected Relay Bank	0 or 1
254,18	None	Report the Current On/Off Status of Relay 2 in the Currently Selected Relay Bank	0 or 1
254,19	None	Report the Current On/Off Status of Relay 3 in the Currently Selected Relay Bank	0 or 1
254,20	None	Report the Current On/Off Status of Relay 4 in the Currently Selected Relay Bank	0 or 1
254,21	None	Report the Current On/Off Status of Relay 5 in the Currently Selected Relay Bank	0 or 1
254,22	None	Report the Current On/Off Status of Relay 6 in the Currently Selected Relay Bank	0 or 1
254,23	None	Report the Current On/Off Status of Relay 7 in the Currently Selected Relay Bank	0 or 1
254,24	None	Report the Current On/Off Status of Relay 8 in the Currently Selected Relay Bank	0-255 (Note 2)
254,25	None	Turns On Automatic Relay Refreshing	85
254,26	None	Turns Off Automatic Relay Refreshing	85
254,27	None	Turns On Reporting Mode (Setting Stored)	85

Command	Parameters	Command Description	Response
What You Send	Values	What Controller Does	What Controller Send Back to You
254,28	None	Turns Off Reporting Mode (Setting Stored)	85
254,29	None	Turns Off All Relays	85
254,30	None	Turn On All Relays	85
254,31	None	Inverts All Relays in Bank	85
254,32	None	Reversed Order of Relays in Bank	85
254,33	None	Test 2-Way Communication	85
254,34	None	Sends Currently Selected Relay Bank	0-32
254,35	None	Store Relay Automatic Refresh Setting	85
254,36	None	Reports the Automatic Refresh Setting	0 or 1
254,37	None	Manually Refresh All Relay Boards	85
254,40	Relay (0-255)	Set Status of All Relays at One Time	85
254,41	None	No Function	85
254,42	Bank (0-32)	Store Startup Status of Relays in Selected Bank	85
254,43	Bank (0-32)	Recall Startup Status of Relays in Selected Bank	0-255 (Note 2)
254,46	Relay (0-255)	Turn On One Relay ONLY, Safe Break Before Make	85
254,47	Relay (0-255)	Turn Off a Relay Specified by its Relay Number	85
254,48	Relay (0-255)	Turn On a Relay Specified by its Relay Number	85
254,49	Bank (0-32)	Sets the current Relay Banks Commands are Directed To	85
254,50	Timer (50-66) Hour (0-255) Minute (0-255) Second (0-255) Relay(0-255)	Choose a Duration Timer (16 Timer Available) Relay Will Stay Active for Selected Hours Relay Will Stay Active for Selected Minutes Relay Will Stay Active for Selected Seconds	85
254,50	Timer (50-66) Hour (0-255) Minute (0-255) Second (0-255) Relay (0-255)	Choose a Duration Timer (Timer Activated Using Different Command) Relay Will Stay Active for Selected Hours Relay Will Stay Active for Selected Minutes Relay Will Stay Active for Selected Seconds Assign a Relay to This Timer	85
254,50	Timer (50-66)	Choose a Duration Timer (Timer Activated Using Different Command) Relay Will Stay Active for Selected Hours Relay Will Stay Active for Selected Minutes Relay Will Stay Active for Selected Seconds Assign a Relay to This Timer	85
254,50	Timer (50-66) Hour (0-255) Minute (0-255) Second (0-255) Relay (0-255)	Choose a Pulse Timer (Timer Activate Using Different Command) Relay Will Stay Active for Selected Hours Relay Will Stay Active for Selected Minutes Relay Will Stay Active for Selected Seconds Assign a Relay to This Timer	85
254, 50, 130	Timer (1-16)	Query Selected Timer Device Returns Current Hour of Timer Device Returns Current Minute of Timer Device Returns Current Second of Timer Device Returns Relay Number that is Assigned to Selected Timer	Hour (0-255) Minute (0-255) Second (0-255) Relay (0-255)

Command	Parameters	Command Description	Response
What You Send	Values	What Controller Does	What Controller Send Back to You
254, 50, 131	TimeLSB (0-255) TimeMSB (0-255)	Manually Activate/Halt Selected Timers (Least Sig. Byte)	85
254, 50, 132	CalibrateLSB (0-255) CalibrateMSB (0-255)	Sets the LSB Calibration of the Timer (Stores in Config. Mode Only). Sets the MSB Calibration of the Timer (Stores in Config. Mode Only).	85
254, 50, 133	None	Retrieves the Calibration Value of the Timer Least Significant Bytes will be Sent First (LSB) Most Significant Bytes will be Sent Least (MSB)	CalibrateLSB (0-255) CalibrateMSB (0-255)
254, 50, 134	None	Calibrators ON, Measure Time Between Data to Help Set Calibration Patches In an ASCII Character Output Signifying the Start of a Timer Patches In an ASCII Character Output Signifying the End of a Timer	85 Start (90) Stop(91)
254, 50, 135	None	Calibrators OFF, Timers Will Not Signify Start/Stop	85
254, 50, 136	None	Retrieves the Repetitions Value (Fights EMI by Holding Relays On)	Reps (1-255)
254, 50, 137	REPS (1-255)	Stores the Repetitions Value (Config. Mode Only)	85
254, 50, 138	None	Retrieves the Character Delay Value (Delay Between Bytes Sent to PC)	CDEL (3-255)
254, 50, 139	CDEL (3-255_	Stores the Character Delay (Config Mode Only)	85
254, 50, 140	None	Retrieves the Number of Relay Banks Attached to Controller	ATBanks (1-255)
254, 50, 141	ATBanks (1-255)	Stores the Number of Relay Banks Attached (Config Mode Only)	85
254, 50, 144	None	Return to Safe Factory Default Values (Config Mode Only)	85
254, 50, 145	None	Get TestCycle Value (How Many Banks Tested in Config. Mode Only)	TCycle (0-32)
254, 50 146	TCycle (0-32)	Set TestCycle Value (works in any mode)	85
254, 50, 147	None	Attempt to Recover from a Controller that has Lost Communications	85
254, 100	Bank (0-32)	Turn Off Relay 0 in Specified Relay Bank	85
254, 101	Bank (0-32)	Turn Off Relay 1 in Specified Relay Bank	85
254, 102	Bank (0-32)	Turn Off Relay 2 in Specified Relay Bank	85
254, 103	Bank (0-32)	Turn Off Relay 3 in Specified Relay Bank	85
254, 104	Bank (0-32)	Turn Off Relay 4 in Specified Relay Bank	85
254, 105	Bank (0-32)	Turn Off Relay 5 in Specified Relay Bank	85
254, 106	Bank (0-32)	Turn Off Relay 6 in Specified Relay Bank	85
254, 107	Bank (0-32)	Turn Off Relay 7 in Specified Relay Bank	85
254, 108	Bank (0-32)	Turn On Relay 0 in Specified Relay Bank	85
254, 109	Bank (0-32)	Turn On Relay 1 in Specified Relay Bank	85
254, 110	Bank (0-32)	Turn On Relay 2 in Specified Relay Bank	85
254, 111	Bank (0-32)	Turn On Relay 3 in Specified Relay Bank	85
254, 112	Bank (0-32)	Turn On Relay 4 in Specified Relay Bank	85
254, 113	Bank (0-32)	Turn On Relay 5 in Specified Relay Bank	85
254, 114	Bank (0-32)	Turn On Relay 6 in Specified Relay Bank	85
254, 115	Bank (0-32)	Turn On Relay 7 in Specified Relay Bank	85

Command	Parameters	Command Description	Response
What You Send	Values	What Controller Does	What Controller Send Back to You
254, 116	Bank(0-32)	Report the Current On/Off Status of Relay 0 in the Specified Relay Bank	0 or 1
254,117	Bank(0-32)	Report the Current On/Off Status of Relay 1 in the Specified Relay Bank	0 or 1
254,118	Bank(0-32)	Report the Current On/Off Status of Relay 2 in the Specified Relay Bank	0 or 1
254,119	Bank(0-32)	Report the Current On/Off Status of Relay 3 in the Specified Relay Bank	0 or 1
254,120	Bank(0-32)	Report the Current On/Off Status of Relay 4 in the Specified Relay Bank	0 or 1
254,121	Bank(0-32)	Report the Current On/Off Status of Relay 5 in the Specified Relay Bank	0 or 1
254,122	Bank(0-32)	Report the Current On/Off Status of Relay 6 in the Specified Relay Bank	0 or 1
254,123	Bank(0-32)	Report the Current On/Off Status of Relay 7 in the Specified Relay Bank	0 or 1
254,124	Bank(0-32)	Reports Status of Relay Bank	0-255**
254,129	Bank(0-32)	Turn Off All Relays	85
254,130	Bank(0-32)	Turn On All Relays	85
254,131	Bank(0-32)	Inverts All Relays in Bank	85
254,132	Bank(0-32)	Reversed Order of Relays in Bank	85

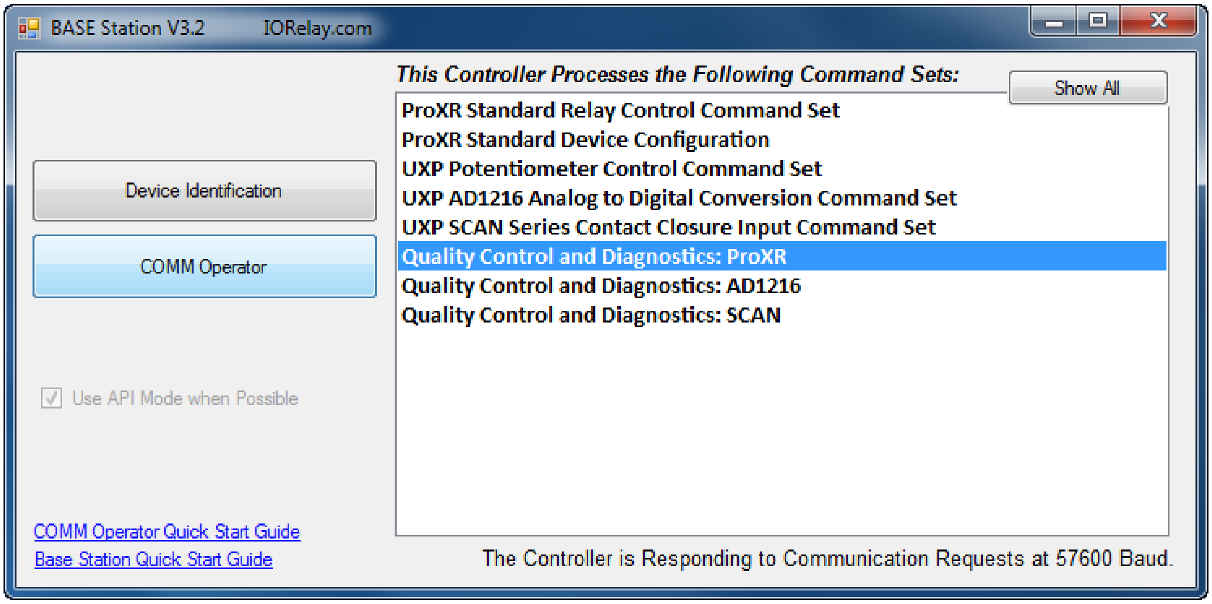
254,140	Relay(0-255)
	Bank(0-32)
		Set Status of All Relays at One Time	85

254,142	Bank(0-32)	Store Startup Status of Relays in Selected Bank	85
254,143	Bank(0-32)	Recall Startup Status of Relays in Selected Bank	0-255**
254,246	None	Get Device Description Data from Controller: Outputs Below Device ID Part 1 Device ID Part 2 Device Year of Design Device Firmware Version E3C Device Number	 1 0 205*** 17*** Default is 0

Chapter 12

Troubleshooting

Use the Base station Software to diagnose any problems with your device. Choose the option **'ProXR Quality Control and Diagnostics'** as shown below.


BASE Station V3.2 IORelay.com
Show All
Device Identification
COMM Operator
☑ Use API Mode when Possible
COMM Operator Quick Start Guide
Base Station Quick Start Guide

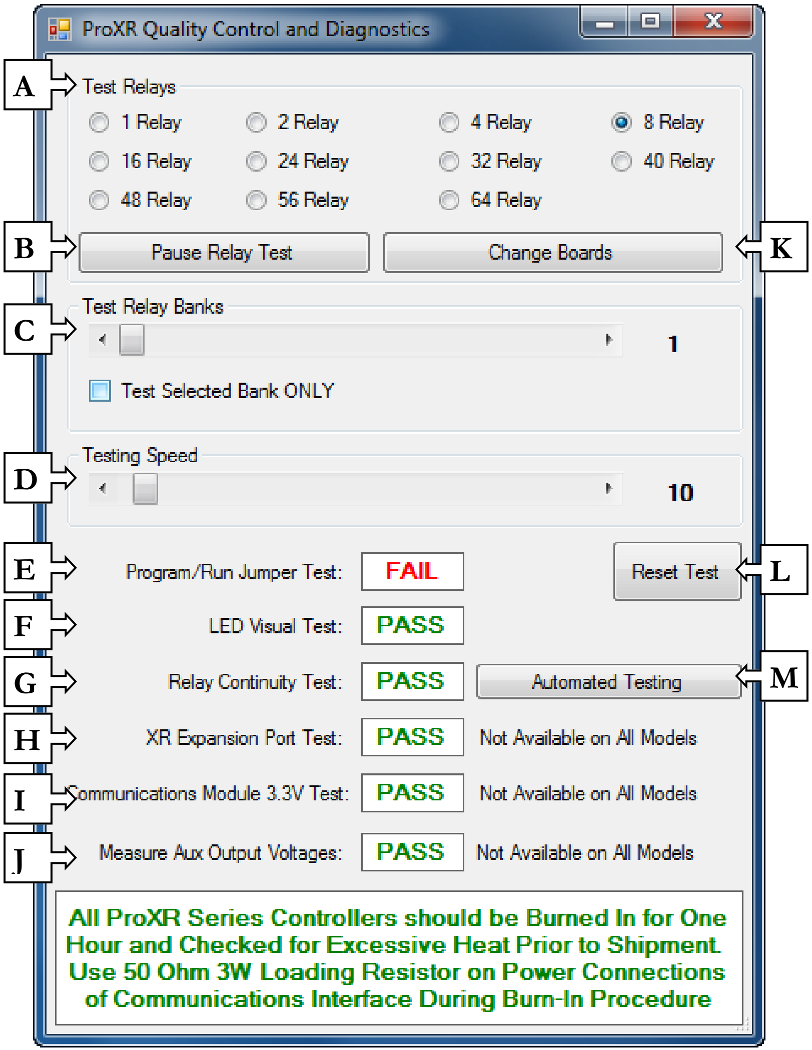
This Controller Processes the Following Command Sets:

- ProXR Standard Relay Control Command Set
- ProXR Standard Device Configuration
- UXP Potentiometer Control Command Set
- UXP AD1216 Analog to Digital Conversion Command Set
- UXP SCAN Series Contact Closure Input Command Set
- Quality Control and Diagnostics: ProXR
- Quality Control and Diagnostics: AD1216
- Quality Control and Diagnostics: SCAN

The Controller is Responding to Communication Requests at 57600 Baud.

To diagnose any problems with the device:

A. Set number of Relays to test.
B. Start Relay Test sequence.
C. Selected Individual Bank to test.
D. Set Relay Test sequence speed.
E. Test functionality of PGM/RUN Jumper. To “Pass” move jumper to PGM position, then to RUN position.
F. Used as a reminder to testing staff to check functionality of all on board LEDs, click to “Pass”.
G. Used by testing staff to check continuality of relays. Click to “Pass”.
H. Used by testing staff to check XR Expansion Ports. Click to “Pass”.
I. Used by testing staff to check 3.3 vdc circuit voltage. Click to “Pass”.
J. Used by testing staff to check 5 vdc circuit voltage. Click to “Pass”.
K. Pause Testing sequence.
L. Reset all status boxes.
M. Used by testing staff for continuity outputs.



ProXR Quality Control and Diagnostics

Test Relays

○ 1 Relay ○ 2 Relay ○ 4 Relay ● 8 Relay
○ 16 Relay ○ 24 Relay ○ 32 Relay ○ 40 Relay
○ 48 Relay ○ 56 Relay ○ 64 Relay

Test Relay Banks: 1

□ Test Selected Bank ONLY

Testing Speed: 10

Program/Run Jumper Test: FAIL
LED Visual Test: PASS
Relay Continuity Test: PASS
XR Expansion Port Test: PASS Not Available on All Models
Communications Module 3.3V Test: PASS Not Available on All Models
Measure Aux Output Voltages: PASS Not Available on All Models

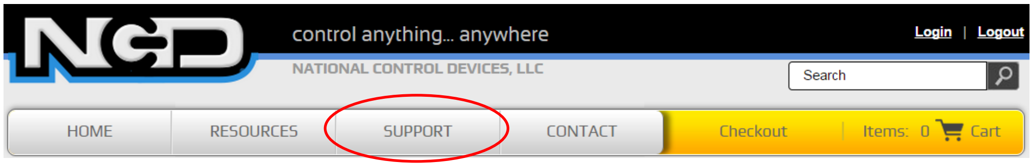
All ProXR Series Controllers should be Burned In for One Hour and Checked for Excessive Heat Prior to Shipment. Use 50 Ohm 3W Loading Resistor on Power Connections of Communications Interface During Burn-In Procedure

Chapter

13

Technical Support

Technical support is available through our website, controlanything.com. **Support** is the way we connect NCD engineers to our customers.



*Click on the **Support** tab at the top of any page on our website to be taken to the **Forum** page. Here you can publicly post or review problems that customers have had, and learn about our recommended solutions.*

Our engineers monitor questions and respond continually throughout the day. Before requesting telephone technical support, we ask that customers please try to resolve their problems through **Support** first. However, for persistent problems, NCD technical support engineers will schedule a phone consultation.

Contact Information

National Control Devices, LLC
PO Box 455
Osceola, MO 64776
417-646-5644 phone
866-562-0406 fax
Open 9 a.m. - 4 p.m. CST

Like "National Control Devices" on Facebook, and follow us on Twitter @ControlAnything.

All orders *must* be placed online at our website, www.controlanything.com

Notice:

The only authorized resellers of NCD products are

- www.controlanything.com
- www.relaycontrollers.com
- www.relaypros.com

All other websites are not authorized dealers; we have noticed some retailers offering our products fraudulently.