

INSTRUCTION MANUAL

NXI NETLINX $^{\ensuremath{\mathbb{R}}}$ INTEGRATED CONTROLLER (NO MASTER)



AV FOR AN IT WORLD

IMPORTANT SAFETY INSTRUCTIONS

- 1. READ these instructions.
- 2. KEEP these instructions.
- 3. HEED all warnings.
- 4. FOLLOW all instructions.
- 5. DO NOT use this apparatus near water.
- 6. CLEAN ONLY with dry cloth.
- 7. DO NOT block any ventilation openings. Install in accordance with the manufacturer's instructions.
- 8. DO NOT install near any heat sources such as radiators, heat registers, stoves, or other apparatus (including amplifiers) that produce heat.
- 9. DO NOT defeat the safety purpose of the polarized or grounding type plug. A polarized plug has two blades with one wider than the other. A grounding type plug has two blades and a third grounding prong. The wider blade or the third prong are provided for your safety. If the provided plug does not fit into your outlet, consult an electrician for replacement of the obsolete outlet.
- 10. PROTECT the power cord from being walked on or pinched, particularly at plugs, convenience receptacles, and the point where they exit from the apparatus.
- 11. ONLY USE attachments/accessories specified by the manufacturer.



12. USE ONLY with a cart, stand, tripod, bracket, or table specified by the manufacturer, or sold with the apparatus. When a cart is used, use caution when moving the cart/apparatus combination to avoid injury from tip-over.

- 13. UNPLUG this apparatus during lightning storms or when unused for long periods of time.
- 14. REFER all servicing to qualified service personnel. Servicing is required when the apparatus has been damaged in any way, such as power-supply cord or plug is damaged, liquid has been spilled or objects have fallen into the apparatus, the apparatus has been exposed to rain or moisture, does not operate normally, or has been dropped.
- 15. DO NOT expose this apparatus to dripping or splashing and ensure that no objects filled with liquids, such as vases, are placed on the apparatus.
- 16. To completely disconnect this apparatus from the AC Mains, disconnect the power supply cord plug from the AC receptacle.
- 17. Where the mains plug or an appliance coupler is used as the disconnect device, the disconnect device shall remain readily operable.
- 18. DO NOT overload wall outlets or extension cords beyond their rated capacity as this can cause electric shock or fire.



The exclamation point, within an equilateral triangle, is intended to alert the user to the presence of important operating and maintenance (servicing) instructions in the literature accompanying the product.



The lightning flash with arrowhead symbol within an equilateral triangle is intended to alert the user to the presence of uninsulated "dangerous voltage" within the product's enclosure that may be of sufficient magnitude to constitute a risk of electrical shock to persons.



ESD Warning: The icon to the left indicates text regarding potential danger associated with the discharge of static electricity from an outside source (such as human hands) into an integrated circuit, often resulting in damage to the circuit.

WARNING:	То
WARNING:	No
WARNING:	Eq
WARNING:	То

To reduce the risk of fire or electrical shock, do not expose this apparatus to rain or moisture. No naked flame sources - such as lighted candles - should be placed on the product. Equipment shall be connected to a MAINS socket outlet with a protective earthing connection. To reduce the risk of electric shock, grounding of the center pin of this plug must be maintained.

COPYRIGHT NOTICE

AMX© 2016, all rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of AMX. Copyright protection claimed extends to AMX hardware and software and includes all forms and matters copyrightable material and information now allowed by statutory or judicial law or herein after granted, including without limitation, material generated from the software programs which are displayed on the screen such as icons, screen display looks, etc. Reproduction or disassembly of embodied computer programs or algorithms is expressly prohibited.

LIABILITY NOTICE

No patent liability is assumed with respect to the use of information contained herein. While every precaution has been taken in the preparation of this publication, AMX assumes no responsibility for error or omissions. No liability is assumed for damages resulting from the use of the information contained herein. Further, this publication and features described herein are subject to change without notice.

AMX WARRANTY AND RETURN POLICY

The AMX Warranty and Return Policy and related documents can be viewed/downloaded at www.amx.com.

Table of Contents

NXI NetLinx Integrated Controller	6
Overview	6
Front Panel	6
Rear Panel	6
Specifications	6
Connections and Wiring	8
Installing the Master or Hub Card	8
Preparing/connecting captive wires	8
RS-232/422/485 Wiring Specifications	8
Relay Connections and Wiring	8
IR/Serial Connections and Wiring	9
Input/Output (I/O) Connections and Wiring	9
Programming1	0
Overview 1	0
Using the ID Button 1	0
Device:Port:System (D:P:S) 1	0
Program Port Commands 1	0
DATE	10
DEVICE STATUS <d:p:s></d:p:s>	10
DOC FREE	10
ECHO OFF	11
	11
GET IP < D.P.S>	11
MSG OFF	11
MSG ON	11
OFF	11
PASS	11
	11
PROGRAM INFO	11 11
REBOOT <d:p:s></d:p:s>	11
RELEASE DHCP	11
SEND_COMMAND	11
SEND_STRANG	11
SET DNS <d:p:s></d:p:s>	11
SET IP <d:p:s></d:p:s>	11
SET TIME	11
SHOW DEVICE <d:p:s></d:p:s>	11
SHOW LOG	11
EJU Pass UUUES 1	2
+ + ESC ESC	12
+ + ESC A	12 12
+ + ESC H	12

Notes on Specific Telnet/Terminal Clients	12	2
Windows client programs	12	2
Linux Telnet client	. 1:	2
SHOW NOTTEY	1	2
SHOW REMOTE	1	2
SHOW ROUTE	1	2
SHOW SYSTEM	1	2
TCP LIST	1	2
TIME	1	2
URL LIST <d:p:s></d:p:s>	1	2
LED Disable/Enable SEND_COMMANDs	13	3
LED-DIS	1	3
	1	3
RS232/422/485 Ports Channels	13	3
255	1	3
RS-232/422/485 SEND_COMMANDs	13	3
B9M0FF	1	3
B9MON	1	3
CHARD	1	3
CHARDM	1	3
CTSPSH	1	3
	1	4
SET BAUD	1	4
	14	4 1
HSON	1. 1.	
RXCLR	1	4
RXOFF	1	4
RXON	14	4
TXCLR	14	4
RS-232/422/485 SEND_STRING Escape Sequences	15	5
27,17,	1	5
27,18,1	1	5
27,18,0	1	5
27,19,	1	5
27,20,0	1	5
27,20,1	1	5
ХUFF YON	··· 1: 1	5
ID / Carial Darta (Q. 15) Channels		~
IR / Serial Ports (8 - 15) Unannels	10	C
IR/Serial SEND_COMMANDs	16	5
CAROFF	1	6
CARON	1	6
СН	1	6
CP	1	6
CTOF	1	6
CTON	1	7
	1	/
	I 1'	/ -
POD	I 1'	, 7
PON	1	7
PTOF	1	7
PTON	1	8
SET IO LINK	1	8
SET MODE	1	8
SP	1	8
XCHM	1	8
Input/Output SEND_COMMANDs	19	9
GET INPUT	1	9

Table of Contents

SET INPUT	19
	19
XCH	19
ZAP НІGН	19
	19

NXI NetLinx Integrated Controller

Overview

The NXI NetLinx Integrated Controller represents the new generation of AMX multi-port central controllers. The NXI can be programmed to control RS-232/422/485, Relay, IR/Serial, and Input/Output devices using the NetLinx programming language and NetLinx Studio program. Depending on your specific control needs, the NXI can be equipped with either a Master or Hub Card. For use as a master controller, the NXI accepts the NXC-ME260 NetLinx Master Card.

Front Panel

The NXI is equipped with a removable faceplate that covers the front panel components (FIG. 1):



FIG. 1 NXI front panel with faceplate installed

Remove the faceplate to see the front panel containing groups of colored LED indicators that light when their corresponding control ports receive/ transmit data (FIG. 2):



FIG. 2 NXI front panel with faceplate removed

These LEDs are grouped by control type, and are numbered according to their corresponding port (connector) numbers on the rear panel.

Rear Panel

The rear panel contains all of the port connectors, plus the ID pushbutton and ICSP LED (FIG. 3):



FIG. 3 NXI rear panel

Specifications

NXI Specification	NXI Specifications	
Power requirements	1.09 A @ 12 VDC (NXI only/no card)	
Memory	64K of IR memory: • 32K IR memory for IR ports 8-11 • 32K IR memory for IR ports 12-15	
Enclosure	Metal with black matte finish	
Front faceplate	Plastic gray faceplate with translucent viewing window	
Weight	4.10 lbs (1.85 kg)	
Dimensions (HWD)	1.72" x 17.0" x 8.80" (43.68 mm x 431.80 mm x 223.52 mm)	
Heat Dissipation:	44.7 BTU/hr (Typical)	

NXI Specifications	s (Cont.)	
Ports		
RS-232/422/485 ports (Ports #1-6)	 Six RS-232/422/485 control ports with XON/XOFF, and CTS/RTS, 300-230,400 baud. Channel range = 1-255 Channels 1-254 provide feedback only. Channel 255 (CTS Push channel): Reflects the state of the CTS Input if a 'CTSPSH' command was sent to the port. 	
Relay port (Port #7)	12-channel relay port. Channel range = 1-12	
IR/Serial ports (Ports #8-15)	 8 IR/Serial control ports that support high-frequency carriers up to 1.14 MHz. Channel range = 1-32,767 Channels 1-253 (output): IR commands. Channel 254 (feedback): Power Fail (used with 'PON' and 'POF' commands). Channel 255 (feedback): Power status (when IOLink is set). 	
I/O port (Port #16)	8-channel I/O port for contact closure, 0-5 VDC voltage sensing, or interactive power sensing for IR ports. Channel range = 1-8	
Front Panel Compone	ints	
Card slot	Accepts NXC-ME260 NetLinx Master or Hub card. • NXC-NH - Hub Card • NXC-HS - Hub Server Card • NXC-HE - Hub Expander Card	
RS-232/422/485 LEDs	 6 sets of red and yellow LEDs light to indicate ports 1-6 are transmitting or receiving RS-232, 422, or 485 data: TX LEDs (red) blink when transmitting data. RX LEDs (yellow) blink when receiving data. 	
Relay LEDs	12 red LEDs light to indicate relay channels 1-12 are active (closed).	
IR/Serial LEDs	8 red LEDs light to indicate IR/Serial channels 1-8 are transmitting control data.	
I/O LEDs	8 yellow LEDs light when I/O channels 1-8 are active.	
Rear Panel Componer	nts	
ICSP LED (green)	Blinks in unison with the Master card's NetLinx LED indicating the ICSP bus is synchronized.	
ID pushbutton	Sets the NetLinx ID (D:P:S) assignment for the NXI.	
RS-232/422/485 ports (Ports #1-6)	 Six 10-pin (male) connectors that support bi-directional RS-232/422/485 communication (XON/XOFF, CTS/RTS, 300-230,400 baud). Channel range = 1-255 Channels 1-254 provide feedback only Channel 255 (CTS Push channel): Reflects the state of the CTS input if a 'CTSPSH' command was sent to the port. 	
Relay port (Port #7)	 Three 8-pin (male) relay connectors (normally open) that support up to 12 independent external relay devices. Each relay can switch up to 24 VDC or 28 VAC @ 1 A. Channel range = 1-12 	
IR/Serial ports (Ports #8-15)	 Two 8-pin (male) connectors that support IR or serial (wired) IR control. The 8 IR/Serial control ports support high-frequency carriers up to 1.14 MHz. Channel range = 1-32,767 Channels 1-253: = IR commands Channel 254: = PowerFail (used with 'PON' and 'POF' commands) Channel 255: = Power status (when IOLink is set) 	
I/O port (Port #16)	 8-channel I/O port for contact closure, 0-5 VDC voltage sensing, or interactive power sensing for IR ports. The 10-pin (male) connector has inputs that detect 0-1.5 VDC (low) as a Push, and 3.5-5 VDC (high) as a Release. When used as an input, each of the eight I/O ports act as a switch to ground and are rated at 200 mA @ 12 VDC. Channel range = 1-8 	
Certifications	FCC Part 15 Class B, CE, and IEC 60950	
Included accessories	 4 CC-NIRC emitters Metal tab strips for commoning adjacent relays Rack-mount brackets adapt for rack, wall, or shelf mounting NetLinx faceplate 	
Optional Accessories	 12 VDC power supply CC-N232 RS-232/422 cables CC-NIRC IR cables CC-NREL Relay cables CC-NSER IR/Serial cables 	

Connections and Wiring

Installing the Master or Hub Card

The NXC-ME260 NetLinx Master or any Hub Card can be installed in the NXI. The card mounts in a horizontal position, through the master card slot on the rear panel of the NXI enclosure (see FIG. 3 on page 1).

To install a Master or Hub Card in an NXI:

- 1. Discharge the static electricity from your body by touching a grounded metal object.
- 2. Unplug all the connectors from the NXI.
- 3. Remove the two screws that hold the front plate on the Master or Hub Card, and remove the front plate.
- 4. Align the edges of the card with the guide slots inside the Master Card slot on the NXI.
- 5. Slide the card about halfway into the slot.
- 6. Inside the Master Card slot on NXI, locate the 6-pin control cable connector.
- 7. Plug the connector from the NXI into the 6-pin terminal on the Master or Hub Card. This connector is keyed to ensure correct orientation.
- 8. Once the control cable is connected, gently slide the card all the way in until you feel the rear edge of the card lightly snap into place.
- 9. Re-apply power and other connections as necessary.

Preparing/connecting captive wires

- 1. Strip 0.25 inch of wire insulation off all wires.
- 2. Insert each wire into the appropriate opening on the connector according to the wiring diagrams and connector types described in this section. Do not tighten the screws excessively; doing so may strip the threads and damage the connector.

RS-232/422/485 Wiring Specifications

The following table lists the wiring specifications for the RS-232/422/485 connectors (ports 1-6).

RS-23	RS-232/422/485 Wiring Specifications				
Pin	Signal	Function	RS-232	RS-422	RS-485
1	GND	Signal ground	х	Х	
2	RXD	Receive data	х		
3	TXD	Transmit data	Х		
4	CTS	Clear to send	х		
5	RTS	Request to send	х		
6	TX +	Transmit data		Х	X (strap to pin 8)
7	ТХ -	Transmit data		Х	X (strap to pin 9)
8	RX +	Receive data		Х	X (strap to pin 6)
9	RX -	Receive data		Х	X (strap to pin 7)
10	12 VDC	Power	optional	optional	

Relay Connections and Wiring

You can connect up to 12 independent external relay devices to the Relay connectors on the NXI (port 7).

- Connectors labeled A are for common; B are for output.
- Each relay is isolated and normally open.
- A metal commoning strip is supplied with each NXI to connect multiple relays.

IR/Serial Connections and Wiring

You can connect up to eight IR- or serial-controllable devices to the IR/Serial connectors (ports 8-15). These connectors accept an IR emitter (CC-NIRC) that mounts on the device's IR window, or a mini-plug (CC-NSER) that connects to the device's control jack. The IR/Serial connector wiring specifications are listed in the following table.

IR/Serial Connector Wiring Specifications			
No.	Port	Signal	Function
1	8	GND (-) Signal 1 (+)	Signal GND IR/Serial data
2	9	GND (-) Signal 2 (+)	Signal GND IR/Serial data
3	10	GND (-) Signal 3 (+)	Signal GND IR/Serial data
4	11	GND (-) Signal 4 (+)	Signal GND IR/Serial data
5	12	GND (-) Signal 5 (+)	Signal GND IR/Serial data
6	13	GND (-) Signal 6 (+)	Signal GND IR/Serial data
7	14	GND (-) Signal 7 (+)	Signal GND IR/Serial data
8	15	GND (-) Signal 8 (+)	Signal GND IR/Serial data

Input/Output (I/O) Connections and Wiring

The I/O port responds to switch closures or voltage level (high/low) changes, or can be used for logic-level outputs. You can connect up to eight devices to the I/O connectors (port 16). A contact closure between GND and an I/O port is detected as a Push. When used for voltage inputs, the I/O port detects a low (0-1.5 VDC) as a Push, and a high (3.5-5 VDC) signal as a Release. When used for outputs, the I/O port acts as a switch to GND and is rated at 200 mA @ 12 VDC.

• The PWR pin (+12VDC @ 200 mA) is designed as a power output for the PCS2 or VSS2 (or equivalent).

• The GND connector is a common ground and is shared by all I/O ports.

The following table lists the wiring specifications for the I/O connectors.

I/O Port Wiring Specifications		
Pin	Signal	Function
1	GND	Signal GND
2	I/O 1	Input/output
3	I/0 2	Input/output
4	I/O 3	Input/output
5	I/0 4	Input/output
6	I/O 5	Input/output
7	I/O 6	Input/output
8	I/0 7	Input/output
9	I/O 8	Input/output
10	12 VDC	PWR

Programming

Overview

This section describes the SEND_COMMANDs, Send_Strings, and Channel commands you can use to program the NXI. The examples in this section require a declaration in the DEFINE_DEVICE section of your program to work correctly. Refer to the *NetLinx Programming Language* instruction manual for specifics about declarations and DEFINE_DEVICE information.

Using the ID Button

The ID Button on the rear panel of the NXI (FIG. 3 on page 1) is used in conjunction with the NetLinx Studio software program to allow you to assign new Device and System numbers for the NXI.

- 1. Using NetLinx Studio, place the system in Identity (ID) Mode. ID Mode means the entire system is put on hold while it waits for an event from any NetLinx device in the named system (for example, pushing the ID button on the NXI). The device that generates the first event is the identified device.
- 2. Press the ID Mode button to generate an event from the NXI and assign new device and system numbers in NetLinx Studio.

Device:Port:System (D:P:S)

A device is any hardware component that can be connected to an AXlink or ICSNet bus. Each device must be assigned a unique number to locate that device on the bus. The NetLinx programming language allows numbers in the range 0-32,767. Device 0 refers to the local master; numbers greater than 32,767 are reserved.

NetLinx requires a Device:Port:System (D:P:S) specification. This D:P:S triplet can be expressed as a series of constants, variables separated by colons, or a DEV structure. For example:

```
STRUCTURE DEV {
    INTEGER Number // Device number
    INTEGER Port // Port on device
    INTEGER System // System the device belongs to
  }
```

The D:P:S notation is used to explicitly represent a device number, port and system. For example, 128:1:0 represents the first port on device 128 on this system. If the system and Port specifications are omitted, (e.g. 128), system 0 (indicating this system) and port 1 (the first port) is assumed. Here's the syntax:

NUMBER:PORT:SYSTEM

where:

NUMBER:	16-bit integer represents the device number
PORT:	16-bit integer represents the port number (in the range 1 through the number of ports on the Controller or device)
SYSTEM:	16-bit integer represents the system number (0 = this system)

Program Port Commands

The Program port commands listed in the following table can be sent directly to the Master Card using a terminal program (i.e. Telnet). Be sure that your PC's COM port and terminal program's communication settings match those in the table below:

PC COM Port Communication Settings		
Baud	38400 (default)	
Parity	None	
Data Bits	8	
Stop Bits	1	
Flow Control	None	

In your terminal program, type "Help" or a question mark ("?") and <Enter> to display the Program port commands listed in the following table.

Program Port Commands		
Command Description		
DATE	Displays the current date and day of the week.	
DEVICE STATUS <d:p:s></d:p:s>	Displays a list of all active (on) channels for the specified D:P:S. Enter DEVICE STATUS without the D:P:S variable, the Master Card displays ports, channels, and version information.	
DNS LIST <d:p:s></d:p:s>	Displays: • Domain suffix • Configured DNS IP Information	
DOC FREE	Displays the total bytes of free space available on the Master Card's Disk on Chip.	

Program Port Commands (Cont.)		
Command	Description	
ECHO OFF	Disables terminal character's echo (display) function.	
ECHO ON	Enables terminal character's echo (display) function.	
GET IP <d:p:s></d:p:s>	Displays the Master Card's D:P:S, Host Name, Type (DHCP or Static), IP Address, Subnet Mask, Gateway IP, and MAC Address.	
MEM	Displays the largest free block of Master Card memory.	
MSG OFF	MSG OFF disables the MSG ON display (see below).	
MSG ON	MSG On sets the terminal program to display all messages generated by the Master Card.	
OFF	Turns off a channel on a device. The device can be on any system the master you are connected to can reach. You can specify the device number, port, and system, or the name of the device that is defined in the DEFINE_DEVICE section of the program.	
ON	Turns on a channel on a device. The device can be on any system the master you are connected to can reach. You can specify the device number, port, and system, or the name of the device that is defined in the DEFINE_DEVICE section of the program.	
PASS	Sets up a pass through mode to a device. In pass through mode, any string received by the device is displayed on the screen, and anything typed is sent as a string to the device. The device can be on any system the master you are connected to can reach. You can specify the device number, port, and system, or the name of the device that is defined in the DEFINE_DEVICE section of the program. See <i>ESC Pass Codes</i> on page 12 for descriptions of the escape codes available in pass mode.	
PING	Tests network connectivity to and confirms the presence of another networked device. It operates just like the PING application in Windows or Linux.	
PROGRAM INFO	Displays the NetLinx program's name residing in the Master Card.	
PULSE	Pulses a channel on a device on and off. The device can be on any system the master you are connected to can reach. You can specify the device number, port, and system, or the name of the device that is defined in the DEFINE_DEVICE section of the program.	
REBOOT <d:p:s></d:p:s>	Reboots the Master Card or specified device.	
RELEASE DHCP	Releases the DHCP setting for the Master Card.	
SEND_COMMAND	Sends a command to a device. The device can be on any system the master you are connected to can reach. You can specify the device number, port, and system, or the name of the device that is defined in the DEFINE_DEVICE section of the NetLinx Program. The data of the string is entered with NetLinx string syntax.	
SEND_STRING	Sends a string to a device. The device can be on any system the master you are connected to can reach. You can specify the device number, port, and system, or the name of the device defined in the DEFINE_DEVICE section of the NetLinx Program. The data of the string is entered with NetLinx string syntax.	
SET DATE	Prompts you to enter the new date for the Master Card. When the date is set on the Master Card, the new date will be reflected on all devices in the system that have clocks (i.e. touch panels). By the same token, if you set the date on any system device, the new date will be reflected on the system's Master, and all connected devices. This will not update clocks on devices connected to another Master (in Master-to-Master systems).	
SET DNS <d:p:s></d:p:s>	Prompts you to enter a Domain Name, DNS IP #1, DNS IP #2, and DNS IP #3. Then, you enter Y (yes) to approve/store the information in the Master Card. Entering N (no) cancels the operation.	
SET IP <d:p:s></d:p:s>	Prompts you to enter a Host Name, Type (DHCP or Fixed), IP address, Subnet Mask, and Gateway IP address. Enter Y (yes) to approve/store the information in the Master Card. Entering N (no) cancels the operation.	
SET TIME	Prompts you to enter the new time for the Master Card. When the time is set on the Master Card, the new time will be reflected on all devices in the system that have clocks (i.e. touch panels). By the same token, if you set the time on any system device, the new time will be reflected on the system's Master, and all connected devices. This will not update clocks on devices connected to another Master (in Master-to-Master systems)	
SET URL <d:p:s></d:p:s>	Prompts you to enter the URL address and port number. Enter Y (yes) to approve/store the new addresses in the Master Card. Entering N (no) cancels the operation.	
SHOW DEVICE <d:p:s></d:p:s>	Displays a list of all devices present on the bus.	
SHOW LOG	 Displays the log of messages stored in the Master's memory. The Master logs all internal messages and keeps the most recent messages. The log contains: Entries starting with first specified or most recent. Date, Day, and Time message was logged. Which object originated the message. The text of the message: SHOW LOG [start] [end] SHOW LOG [start] [end] SHOW LOG ALL If start is not entered, the most recent will be first. If end is not entered, the last 20 messages will be shown. 	
	 IT ALL is entered, all stored messages will be snown, starting with the most recent. 	

Program Port Commands (Cont.)	
Command	Description
SHOW NOTIFY	Displays a list of devices that other systems have requested input from and the types of information needed. Note that the local system number is 1061.
SHOW REMOTE	Displays a list of the devices this system requires input from and the types of information needed. When a NetLinx master connects to another NetLinx master, the newly connecting system has a device that the local system desires input from; the new system is told what information is desired from what device. Note the local system number is 1062.
SHOW ROUTE	Displays information about how this NetLinx master is connected to other NetLinx masters.
SHOW SYSTEM	Provides a list of all devices in all systems currently on-line. The system's lists are either directly connected to this master (i.e. 1 hop away), or are referenced in the DEFINE_DEVICE section of the NetLinx program. You may provide the desired system number as a parameter to display only that system's information (e.g. SHOW SYSTEM 2001). The systems listed are shown in numerical order.
TCP LIST	Lists all active TCP/IP connections.
TIME	Displays the current time on the Master Card.
URL LIST <d:p:s></d:p:s>	Displays the list of URL addresses programmed in the Master Card.

ESC Pass Codes

There are 'escape' codes in the pass mode. These codes can switch the display mode or exit pass mode. The following 'escape' codes are defined.

Escape Pass Codes		
Command	Description	
+ + ESC ESC	Exit Pass Mode: Typing a plus (shift =) followed by another plus followed by an ESC (the escape key) followed by another escape exits the pass mode. The Telnet session returns to "normal".	
+ + ESC A	ASCII Display Mode: Typing a plus (shift =) followed by another plus followed by an ESC (the escape key) followed by an 'A' sets the display to ASCII mode.	
	Any ASCII characters received by the device will be displayed by their ASCII symbol.	
	 Any non-ASCII characters will be displayed with a \ followed by two hex characters to indicate the characters hex value. 	
+ + ESC D	Decimal Display Mode: Typing a plus (shift =) followed by another plus followed by an ESC (the escape key) followed by a 'D' sets the display to decimal mode.	
	Any characters received by the device will be displayed with a $\$ followed by numeric characters to indicate the characters decimal value.	
+ + ESC H	Hex Display Mode: Typing a plus (shift =) followed by another plus followed by an ESC (the escape key) followed by an 'H' sets the display to hexadecimal mode.	
	Any characters received by the device will be displayed with a $\$ followed by two hex characters to indicate the characters hex value.	

Notes on Specific Telnet/Terminal Clients

Telnet and terminal clients will have different behaviors in some situations. This section states some of the known anomalies.

Windows client programs

Anomalies occur when using a Windows[™] client if you are not typing standard ASCII characters (i.e. using the keypad and the ALT key to enter decimal codes). Most programs will allow you to enter specific decimal codes by holding ALT and using keypad numbers. For example, hold ALT, hit the keypad 1, then hit keypad 0, then release ALT. The standard line feed code is entered (decimal 10). Windows will perform an AnsiToOem conversion on some codes entered this way because of the way Windows handles languages and code pages.

The following codes are known to be altered, but others may be affected depending on the computer's setup.

Characters 15, 21, 22, and any characters above 127.

This affects both Windows Telnet and Terminal programs.

Linux Telnet client

The Linux Telnet client has three anomalies that are known at this time:

- A null (\00) character is sent after a carriage return.
- If an ALT 255 is entered, two 255 characters are sent (per the telnet RAFT).
- If the code to go back to command mode is entered (ALT 29 which is ^]), the character is not sent, but telnet command mode is entered.

LED Disable/Enable SEND_COMMANDs

The following commands enable or disable the LEDs on the NXI.

LED SEND_COM	LED SEND_COMMANDs	
LED-DIS	Disables the LEDs. Issue this command to port 1 to disable all the LEDs on the NXI. When activity occurs on a port(s) or NXI, the LEDs will not light.	
	Syntax:	
	SEND_COMMAND <dev>,'LED-DIS'</dev>	
	Example:	
	SEND_COMMAND System_1,'LED-DIS'	
	Disables all the LEDs on the System_1 NXI.	
LED-EN	Enable LEDs (default). Issue the command to port 1 to enable the LEDs on the NXI (default setting). When activity occurs on a port(s) or NXI, the LEDs light.	
	Syntax:	
	SEND_COMMAND <dev>,'LED-EN'</dev>	
	Example:	
	SEND_COMMAND System_1,'LED-EN'	
	Enables the System_1 NXI's LEDs.	

RS232/422/485 Ports Channels

RS232/422/485 Ports Channels	
255	CTS push channel: Reflects the state of the CTS input if a 'CTSPSH' command was sent to the port.

RS-232/422/485 SEND_COMMANDs

RS-232/422	2/485 SEND_COMMANDs
B9MOFF	Sets the port's communication parameters for stop and data bits according to the software settings on the RS-232 port (default). This command works in conjunction with the B9MON command. Syntax: SEND_COMMAND <dev>, 'B9MOFF' Example: SEND_COMMAND RS232_1, 'B9MOFF' Sets the RS-232 port settings to match the port's configuration settings.</dev>
B9MON	Overrides and sets the communication settings on the RS-232 port to nine data bits and one stop bit. This command works in conjunction with the B9MOFF command. Syntax: SEND_COMMAND <dev>, 'B9MON' Example: SEND_COMMAND RS232_1, 'B9MON' Resets the RS-232 port's communication parameters to nine data bits, one stop bit, and locks-in the baud rate.</dev>
CHARD	Sets the delay time between transmitted characters in 100 microsecond increments. Syntax: SEND_COMMAND <dev>, 'CHARD<time>' Variable: Time: 0-255 in 100 microsecond increments Example: SEND_COMMAND RS232_1, 'CHARD10' Sets a 1mS delay between all transmitted characters.</time></dev>
CHARDM	Sets the delay time between transmitted characters in 1 millisecond increments. Syntax: SEND_COMMAND <dev>, 'CHARDM<time>' Variable: Time: 0-255 in 1 millisecond increments Example: SEND_COMMAND RS232_1, 'CHARDM10' Sets a 10 mS delay between all transmitted characters.</time></dev>
CTSPSH	Enables Pushes, Releases, and status information to be reported via channel 255. If Clear To Send (CTS) is high, the channel is on. Syntax: SEND_COMMAND <dev>, 'CTSPSH' Example: SEND_COMMAND RS232_1, 'CTSPSH' Sets the RS232_1 port to detect changes on the CTS input.</dev>

CTSPSH OFF Detables Publics, Release, and status information to be reported via channel 255. Turns CTSPSH off. Synta;	RS-232/422/4	RS-232/422/485 SEND_COMMANDs (Cont.)		
Turns CTSPSH off. Synta: SIRE_COMMAND FLEW-, CTSPSE OFF: Sample: Start St	CTSPSH OFF	Disables Pushes, Releases, and status information to be reported via channel 255.		
Synta: SINDE_COMMON FORMULT_CENTRY_CENTRY OFF SET BAUD Sets the 65-322422458 port's communication parameters. Synta: SET BAUD Sets the 65-322422458 port's communication parameters. Synta: SET BAUD SET BAUD Sets the 65-322422458 port's communication parameters. Synta: SET (NOND) CREW, SET BAUD (Baud), (Pacity), (Data), (Stop) (485 DISABLE/EXABLE)' Variable: Baud = 150, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400 (factory set default), 57600, 76600, 115200, 230400 Parity = N (none), 0 (odd), E (even), M (mark), S (space) Data = 7 or 6 data bits Stop = 1 or 2 dop bits A85 Disable = Disables RS-485 mode and disables RS-422. Example: Sets the 85222, 1 parts communication parameters to 9,600 baud, no parity, 8 data bits, 1 stop bit, and enables RS-485 mode and subles RS-422. Example: Sets the 85222, 1 parts communication parameters to 9,600 baud, no parity, 8 data bits, 1 stop bit, and enables RS- 485 mode. TEET BAUD Temporarily sets the 85-232/422/485 port's communication parameters. Syntax: Syntax: Sets the 85202, 0000000 RS22, 1 (stor), (stor), (stor), (stor), (dtsp) (485 DISANLE/FMBRIAR)' HSOFF Disables hardware handshaking (default). Syntax: Syntax: Sets the 85232, 1 (stor): Syntax: Sets the 85232, 1 (stor): Syntax: Sets the 85232, 1 (stor): Syntax: Sets the 85232, 1 (stor): Sets the 85232, 1		Turns CTSPSH off.		
SIND_CONVADD_CREATE CONV_CREATES COPY Sample: Sample: SET BAUD Sets the R5-232/42/485 ports communication parameters. Syntax Syntax Syntax <		Syntax:		
Example. SET BAUD SET BAUD Sets the R5-232/42/485 port's communication parameters. SYNtax SSED_COMMON (PEV», (SET BAUD (Baud), (Pacity), (Data), (Scop)(485 DISABLE/EXAMPLE)' Variables: Boud = 150, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400 (factory set default), 57600, 76800, 115200, 230400 Party = N (none), 0 (odd), E (even), M (mark), S (space) Data = 7 or 8 data bits Stop = 1 or 2 stop bits 4485 Tobable SR-485 mode and deables R5-422. Example: SEXD_COMMON PARSA, 1, SET BAUD (Sec0, #, 1, 445 EXAMPLE') Stop = 1 or 2 stop bits 495 Tobable R5-435 mode and deables R5-422. Example: SEXD_COMMON PARSA, 1, SET BAUD (Sec0, #, 1, 415 EXAMPLE') Stop = 1 or 2 stop bits Gatable R5-232/42/445 port's communication parameters. TSET BAUD works the same as SET BAUD except that the changes are not permanent, and the previous values will be restored if the power is cycled on the device. Syntax SSED_COMMAD +DEV*, 'TBET BAUD (Baud), (Parity), (Data1, (Stop))(485 DISABLE/EXAMPLE)' Disables hardware handshaking (default). Syntax SSED_COMMAD +DEV*, 'TBET BAUD (Baud), (Parity), (Data1, (Stop))(485 DISABLE/EXAMPLE)' Disables hardware handshaking on the R5232_1 device. Enables R17 (ready-to-send) and CTS (clear-to-send) hardware handshaking. SYNTax SSED_COMMAD +DE		SEND_COMMAND <dev>, 'CTSPSH OFF'</dev>		
Turns off CTSPSH on the specified device. SET BAUD Sets the RS-322/422/485 port's communication parameters. Syntax: 3200_COMARD +DEV+, GET BAUD (Baud), (Parley), (Daca), (SLOP)(485 DIGABLE/ENADLE)* Variables: Baud = 150, 900, 600, 1200, 2400, 4800, 9600, 19200, 38400 (factory set default), 57600, 76800, 115200, 220400 Party = N (none), 0 (odd), E (even), M (mark), S (space) Data = 7 or 8 data bits Stop = 1 or 2 stop bits 485 Orable = Disable SS-485 mode and enables RS-422. 485 Chable = Disable SS-485 mode and disables RS-422. 485 mode: TSET BAUD Remportantly sets the RS-232, 1/ SET BAUD (900, R, 8.1, 485 Disable); Sets the RS-322, 1 port's communication parameters to 9,600 baud, no party, 8 data bits, 1 stop bit, and enables RS- 485 mode. TSET BAUD Remportantly sets the RS-232/422/485 port's communication parameters. TSET BAUD works the same as SET BAUD, sets the RS-322, 1 port's communication parameters to 9,600 baud, no party, 8 data bits, 1 stop bit, and enables RS- 485 mode. TSET BAUD Demography sets the RS-232, 1/ SET BAUD (Baud), (Parley), (Data), (Stop) (485 DISABLE/ENABLE)* Disables hardware handshaking (default). Syntax: 3280_COMMAD 402V+, 'ISON* Baub_be hardware handshaking (default). Syntax: 3280_COMMAD 402V+, 'ISON* Example: 3280_COMMAD 402V+, 'ISON* E		SEND COMMAND RS232 1. CTSPSH OFF		
SET BAUD Sets the R5-232/422/485 port's communication parameters. Syntax: SWDD. COMMAD: <devp., 'set="" (baud),="" (deta),="" (parisy),="" (stop)(485="" badd.="" disaelf="" emale)'<="" td=""> Variables: Baud = 150, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400 (factory set default), 57600, 75800, 115200, 230400 Party = N (none), 0 (odd), E (even), M (mark), S (space) Data = 7 or 8 data bits Stop = 1 or 2 stop bits 4455 Disables R5-485 mode and enables R5-422. Hample: Sets the R5232_1 port's communication parameters to 9,600 baud, no party, 8 data bits, 1 stop bit, and enables R5-485 mode. TEET BAUD Temporarily sets the R5-232/422/485 port's communication parameters. TSET BAUD works the same as SET BAUD, except that the changes are not permanent, and the previous values will be restored if the power is cycled on the device. Syntax: Simp_COMMAD CDEVP. (TSET BAUD (Baud), (Part Ly), (Data), (Stop)(485 DISAELE/EMARCE)' HSOFF Disables hardware handshaking (default). Syntax: Syntax: Simp_COMMAD CDEVP. (TSET BAUD (Baud), (Part Ly), (Data), (Stop)(485 DISAELE/EMARCE)' HSOF Disables hardware handshaking (default). Syntax: Syntax: Simp_COMMAD CDEVP. (TSET BAUD (Baud), (Part Ly), (Data), (Stop)(485 DISAELE/EMARCE)' HSOF Disables hardware handshaking on the R5322_1 device. RXCLR Clears al characters in the R522</devp.,>		Turns off CTSPSH on the specified device.		
Syntax: SIMD_COMMAD: <1070*, 'SITT_BAID_(Bould), (Parity), (Data), (Stop) (485_DISABLE/PRAFILE)' Variables: Baud = 150, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400 (factory set default), 57600, 76800, 115200, 230400 Parity = N (none), 0 (odd), E (even), M (mark), S (space) Data = 7 or 8 data bits Stop = 1 or 2 stop bits 485 Disble = Disbles RS-485 mode and enables RS-422. 485 Disble = Disbles RS-485 mode and disbles RS-422. 485 Disble = Disbles RS-485 mode and disbles RS-422. 485 Binble = Disbles RS-485 mode and disbles RS-422. 485 Binble = Disbles RS-485 mode and enables RS-422. 485 Binble = Disbles RS-425 mode and disbles RS-422. 485 mode. TSET BAUD Temporarity sets the RS-232, 1 orts communication parameters to 5600 baud, no parity, 8 data bits, 1 stop bit, and enables RS-428. 485 mode. TSET BAUD Temporarity sets the RS-232, 1 device, communication parameters. Bisbles hardware handshaking (default). Syntax Syntax Bisbles hardware handshaking on the RS232, 1 device. HSON Enables RTS (ready-to-send) and CTS (clear-to-send) hardware handshaking. Syntax Bisble bardware handshaking on the RS232, 1 device. <td< th=""><th>SET BAUD</th><th>Sets the RS-232/422/485 port's communication parameters.</th></td<>	SET BAUD	Sets the RS-232/422/485 port's communication parameters.		
SBID_CONMAD -REST_BAUD_(Tead.), (Farl(y), (Dat.a), (Stop)(455_DISABLE/INVALE)' Variables: Baud = 150, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400 (factory set default), 57600, 76800, 115200, 230400 Data = 7 or 8 data bits Stop = 1 or 2 etop bits 485_Disble = Disbles/RS-485 mode and disbles RS-422. 485_Enable = Enables/RS-485 mode and disbles RS-422. Example: SSID_COMMAD SSID_COMMAD F8232_1, 'STT_BAUD_9000, N, 8.1 455_ENABLE' SSID_COMMAD F8232_10, 'STT_BAUD 9000, N, 8.1 455_ENABLE' SSID_COMMAD F8232_10, 'STT_BAUD (Saud), (Parity), (Data), (Stop)(455_DISABLE/ENABLE)' Syntac Syntac		Syntax:		
Baud = 150, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400 (factory set default), 57600, 76800, 115200, 230400 Parity = N (none), 0 (odd), E (even), M (mark), S (space) Data = 7 or 8 data bits Stop = 1 or 2 stop bits 485 Disable = Bitables RS-485 mode and disables R5-422. Example: Set the RS232_1 port's communication parameters to 9,600 baud, no parity, 8 data bits, 1 stop bit, and enables RS-485 mode. TSET BAUD Temporarity sets the RS-232/422/485 port's communication parameters to 18,600 baud, no parity, 8 data bits, 1 stop bit, and enables RS-485 mode. TSET BAUD Temporarity sets the RS-232/422/485 port's communication parameters. TSET BAUD works the same as SET BAUD, except that the changes are not permanent, and the previous values will be restored if the power is cycled on the device. Status_context Status_context		<pre>SEND_COMMAND <dev>,'SET BAUD (Baud),(Parity),(Data),(Stop)(485 DISABLE/ENABLE)' Variables:</dev></pre>		
Parity = N (noe), 0 (odd), E (even), M (mark), S (space) Data = 7 or 8 data bits Stop = 1 or 2 stop bits 445 Disable = Disables RS-485 mode and disables RS-422. Example: SIDD_COMMAND RE232_1_IST BAUD 9600,M, 8, 1, 445 EXALE! Sets the RS332_1 port's communication parameters to 9,600 baud, no parity, 8 data bits, 1 stop bit, and enables RS-485 mode. TSET BAUD Temporarily sets the RS-322/422/485 port's communication parameters. TSET BAUD works the same as SET BAUD, except that the changes are not permanent, and the previous values will be restored if the power is cycled on the device. Status SISD_COMMAND <pev2, 'tset="" (485="" (baud),="" (data),="" (earlty),="" (stop)="" baud="" disable="" enable)'<="" td=""> HSOFF Disables hardware handshaking (default). SISD_COMMAND <pev2, 'tset<="" td=""> SISD_COMMAND POV2, 'TSET SISD_COMMAND POV2, 'TSET SISD_COMMAND POV2, 'TSEN' SISD_COMMAND POV2, 'TSEN' SISD_COMMAND POV2, 'TSEN'</pev2,></pev2,>		Baud = 150, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400 (factory set default), 57600, 76800, 115200, 230400		
bata = 7 or 8 data bits Stop = 1 or 2 stop bits 445 Disable = Disables RS-485 mode and disables RS-422. 445 Enables RS-485 mode and disables RS-422. 445 Enables RS-485 mode and disables RS-422. 455 Enables RS-485 mode and disables RS-422. 455 Enables RS-485 mode and disables RS-422. 455 mode. Sets the RS-322_1 / rest aton 9600, N, 8.1.4 sts ENABLE! Sets the RS-322_1 port's communication parameters to 9,600 baud, no parity, 8 data bits, 1 stop bit, and enables RS-485 mode. 455 mode. TSET BAUD cxcept that the changes are not permanent, and the previous values will be restored if the power is cycled on the device. Syntax: SSID_COMMADD ODEV>, 'ISST BAUD (Baud), (Parity), (Data), (Stop)(485 DISABLE/ENABLE)' HSOFF Disables hardware handshaking on the RS232_1 device. HSON Enables RS'(ready-to-send) and CTS (clear-to-send) hardware handshaking. Syntax: SSID_COMMAD ODEV>, 'ISSOT SSID_COMMAD ODEV>, 'ISSOT SSID_COMMAD ODEV>, 'ISSOT SSID_COMMAD ODEV>, 'ISSOT SSID_SD_COMMAD ODEV>, 'ISSOT Syntax: SSID_COMMAD ODEV>, 'ISSOT SSID_COMMAD ODEV>, 'ISSOT SSID_SD_COMMAD ODEV>, 'ISSOT SSID_COMMAD ODEV>, 'ISSOT SSID_SD_COMMAD ODEV>, 'ISSOT <th></th> <th>Parity = N (none), O (odd), E (even), M (mark), S (space)</th>		Parity = N (none), O (odd), E (even), M (mark), S (space)		
Stop = 1 or 2 stop bits 485 Disables RS-485 mode and enables RS-422. 485 Disables RS-485 mode and disables RS-422. Example: SERE_COMMAND R2323_1, 'SET RALD 9600, N, 8, 1 485 EXALLS' Sets the RS232_1 port's communication parameters to 9,600 baud, no parity, 8 data bits, 1 stop bit, and enables RS-485 mode. TSET BAUD Temporarily sets the RS-232/422/485 port's communication parameters. TSET BAUD works the same as SET BAUD, except that the changes are not permanent, and the previous values will be restored if the power is cycled on the device. Syntax: SSRD_COMMAND <b222, 'issrt="" (bata),="" (baud),="" (parity),="" (stop)(485="" baud="" disable="" enable)'<="" td=""> HSOFF Disables hardware handshaking (default). Syntax: SSRD_COMMAND <b222, 'issrt="" (bata),="" (parity),="" (stop)(485="" disable="" enable)'<="" td=""> HSON Enables RS (ready-to-send) and CTS (clear-to-send) hardware handshaking. Syntax: SSRD_COMMAND >DEV>, 'HSOP' Disables hardware handshaking on the RS232_1 device. Enables RS', Casta, 'SSRT, COMMAND >DEV>, 'HSOP' Example: SSRD_COMMAND >DEV>, 'HSOP' SSRD_COMMAND >DEV>, 'HSOP' SSRD_COMMAND >DEV>, 'HSOP' Example: SSRD_COMMAND >DEV>, 'HSOP' SSRD_COMMAND >DEV>, 'HSOP' SSRD_COMMAND >DEV>, 'HSOP' Example: SSRD_COMMAND >DEV>, 'HSOP' SSRD_COMMAND >DEV>, 'HSOP' SSRD_COMMAND >DEV>,</b222,></b222,>		Data = 7 or 8 data bits		
45 Disables PL-485 mode and enables RS-422. 485 Enable = Enables RS-485 mode and disables RS-422. Example: BSED_COMMAD R232_1, 'BSET BADD 9600, N, 8, 1, 485 ENABLE' SEST BAUD TSET BAUD TET FAUD SET DEADD SET DEADDE SET DEADD SET		Stop = 1 or 2 stop bits		
#45 Enable = Enables R5-485 mode and disables RS-422. Example: SETE FAUDE Sett for R5232_1, F33T_BAUD_9600, N, 8, 1, 485_ENABLE* Set the R5232_1 port's communication parameters to 9,000 baud, no parity, 8 data bits, 1 stop bit, and enables RS-485 mode. TSET BAUD Temporarily sets the R5-232/422/485 port's communication parameters. TSET BAUD works the same as SET BAUD, except that the changes are not permanent, and the previous values will be restored if the power is cycled on the device. Syntax: SEND_COMMAND <dev>, 'TSET_BAUD (Baud), (Parity), (Data), (Stop)(485_DISABLE/ENABLE)* HSOFF Disables hardware handshaking (default). Syntax: SEND_COMMAND *DEV>, 'ISEOF' Example: SEND_COMMAND *DEV>, 'ISEOF' Example:</dev>		485 Disable = Disables RS-485 mode and enables RS-422.		
Example: SEND_COMMAND R8332_1, 'SET BAID 9600, N. 8.1 485 ENABLE' Sets the R5232_1 port's communication parameters to 9,600 baud, no parity, 8 data bits, 1 stop bit, and enables RS-485 mode. TSET BAUD Temporarily sets the R5-232/422/485 port's communication parameters. TSET BAUD works the same as SET BAUD, except that the changes are not permanent, and the previous values will be restored if the power is cycled on the device. Syntax: SEND_COMMAND CREV>, 'TSET BAUD (Baud), (Parity), (Data), (Stop) (485 DISABLE/ENABLE)' HSOFF Disables hardware handshaking (default). Syntax: SEND_COMMAND CREV>, 'HSOPF' Example: SEND_COMMAND CREV>, 'HSOPF' Disables hardware handshaking on the RS232_1 device. HSON Enables RTS (ready-to-send) and CTS (clear-to-send) hardware handshaking. Syntax: SEND_COMMAND CREV>, 'HSON' Example: SEND_COMMAND ADEV>, 'HSON' Example: SEND_COMMAND ADEV>, 'HSON' Example: Example: SEND_COMMAND ADEV>, 'HSON' Example: SEND_COMMAND ADEV>, 'HSON' Example: SEND_COMMAND ADEV>, 'HSON' Example: SEND_COMMAND ADEV>, 'HSON' SEND_COMMAND ADEV>, 'HSON' SEND Command to Adarcha in the R5232_1, 'HSON'		485 Enable = Enables RS-485 mode and disables RS-422.		
SERD_COMMAND_RS232_1, VIST_BADD_9600, X, 81, 485_EXABLE* Setts the R5232_1 port's communication parameters to 9,600 baud, no parity, 8 data bits, 1 stop bit, and enables R5-485 TEFT BAUD Temporarily sets the R5-232/422/485 port's communication parameters. TSET BAUD works the same as SET BAUD, except that the changes are not permanent, and the previous values will be restored if the power is cycled on the device. Syntax: SERD_COMMAND_OPEV>, 'TSET BAUD (Baud), (Parity), (Data), (Stop)(485_DISABLE/ENABLE)' HSOFF Disables hardware handshaking (default). Syntax: SERD_COMMAND_OPEV>, 'HSOFF' Example: SERD_COMMAND_OPEV>, 'HSOFF' Disables hardware handshaking on the R5232_1 device. HSON Enables RTS (ready-to-send) and CTS (clear-to-send) hardware handshaking. Syntax: SERD_COMMAND_OPEV>, 'HSOFF' Clears all characters in the receive buffer waiting to be sent to the Master Card. Syntax: SERD_COMMAND_OPEV>, 'RXCER' Clears all characters in the R5232_1 device's receive buffer waiting to be sent to the Master Card. Syntax: SERD_COMMAND_OPEV>, 'RXCER' Example: SERD_COMMAND_OPEV>, 'RXCER' Clears all characters in the R5232_1 device's receive buffer waiting to be sent to the Master Card. Syntax: SERD_COMMAND_OPEV>, 'RXCER' Example: SERD		Example:		
Sets the R5252_1 ptbl 5 Communication parameters to 9,000 datab, the parity, e data bits, 1 stup bit, and enables R5-485 mode. TSET BAUD Temporarily sets the R5-232/422/485 port's communication parameters. TSET BAUD works the same as SET BAUD, except that the changes are not permanent, and the previous values will be restored if the power is cycled on the device. Syntax: SND_COMMAND <dev5, 'tset="" (485="" (baud),="" (data),="" (parity),="" (stop)="" baud="" disable="" enable)'<="" td=""> HSOFF Disables hardware handshaking (default). Syntax: SND_COMMAND R5232_1, 'HSOFF' Example: SNN_COMMAND CDEV5, 'HECLR' Example: SNN_COMMAND CDEV5, 'HECLR' Example: SNN_COMMAND CDEV5, 'HECLR' Example: SNN_COMMAND CDEV5, 'HECLR' SNN_COMMAND CDEV5, 'HECLR' SNN_COMMAND C</dev5,>		SEND_COMMAND RS232_1, 'SET BAUD 9600, N, 8, 1 485 ENABLE'		
TSET BAUD Temporarily sets the RS-232/422/485 port's communication parameters. TSET BAUD, works the same as SET BAUD, except that the changes are not permanent, and the previous values will be restored if the power is cycled on the device. Syntax: SEND_COMMAND <dev>, 'TSET BAUD (Baud), (Parity), (Data), (Stop)(485 DISABLE/ENABLE)' HSOFF Disables hardware handshaking (default). Syntax: SEND_COMMAND x0EV>, 'HSOFF' Disables hardware handshaking on the RS232_1 device. Enables RTS (ready-to-send) and CTS (clear-to-send) hardware handshaking. Syntax: SEND_COMMAND x0EV>, 'HSON' Example: SEND_COMMAND x0EV>, 'HSON' Example: SEND_COMMAND x232_1, 'HSOFF' Example: SEND_COMMAND x232_1, 'HSOFF' Example: SEND_COMMAND x232_1, 'RXOFF' Stops the R523_1 device from transmitting received characters to the Master Card. RXOFF Stops and R5332_1, 'RXOFF' <</dev>		485 mode.		
Syntax: SRMD_COMMAND <dev>, 'TSET_BALD' (Baud), (Parity), (Data), (Stop)(485 DISABLE/ENABLE)' HSOFF Disables hardware handshaking (default). Syntax: SRMD_COMMAND <dev>, 'HSOFF' Example: SEND_COMMAND R2322_1, 'HSOFF' Disables hardware handshaking on the RS232_1 device. HSON Enables RTS (ready-to-send) and CTS (clear-to-send) hardware handshaking. Syntax: SEND_COMMAND <dev>, 'HSON' Example: SEND_COMMAND X232_1, 'HSOLE' Example: SEND_COMMAND R232_1, 'HSOLE' SEND_COMMAND X232_1, 'HSOFF' Example: SEND_C</dev></dev></dev>	TSET BAUD	Temporarily sets the RS-232/422/485 port's communication parameters. TSET BAUD works the same as SET BAUD, except that the changes are not permanent, and the previous values will be restored if the power is cycled on the device.		
SEND_COMMAND <dev>, 'TSET BADD (Baud), (Parity), (Data), (Stop)(485 DISABLE/ENABLE)' HSOFF Disables hardware handshaking (default). Synta: SEND_COMMAND <dev>, 'HSOFF' Example: SEND_COMMAND <dev>, 'HSOFF' Disables hardware handshaking on the R5232_1 device. HSON Enables RTS (ready-to-send) and CTS (clear-to-send) hardware handshaking. Syntax: SEND_COMMAND CDEV>, 'HSON' Example: Example: SEND_COMMAND CDEV>, 'HSON' Example: SEND_COMMAND CDEV>, 'HSON' Example: SEND_COMMAND CDEV>, 'HSON' Example: SEND_COMMAND CDEV>, 'RXCLR' Example: SEND_COMMAND CDEV>, 'RXCLR' Eleas all characters in the R5232_1 device's receive buffer waiting to be sent to the Master Card. RXOFF Stops transmitting received characters to the Master Card (default). Syntax: SEND_COMMAND R5232_1, 'RXCFF' SEND_COMMAND CDEV>, 'RXOFF' Stops the R5232_1 device from transmitting received characters to the Master Card. RXON Starts transmitting received characters to the Master Card. This command is sent automatically when issuing a CREATE BUFFER SEND_COMMAND CDEV>, 'RXOFF' Starts transmitting received characters to the Master Card. SEND_COMMAND CDEV>, 'RXOFF' Stops the R5232_1 device from tra</dev></dev></dev>		Syntax:		
HSOFF Disables hardware handshaking (default). Syntax: SEND_COMMAND <dev>, 'HSOFF' Example: SEND_COMMAND F2332_1, 'HSOFF' Disables hardware handshaking on the RS232_1 device. HSON Enables RTS (ready-to-send) and CTS (clear-to-send) hardware handshaking. Syntax: SEND_COMMAND F2332_1, 'HSON' Example: SEND_COMMAND F2332_1, 'HSON' Enables hardware handshaking on the RS232_1 device. RXCLR Clears all characters in the receive buffer waiting to be sent to the Master Card. Syntax: SEND_COMMAND F2332_1, 'RXCLR' Example: SEND_COMMAND F2332_1, 'RXCLR' Clears all characters in the RS232_1 device's receive buffer waiting to be sent to the Master Card. RXOFF Stops transmitting received characters to the Master Card (default). Syntax: SEND_COMMAND F2332_1, 'RXCLR' Clears all characters in the RS232_1 device's receive buffer waiting to be sent to the Master Card. RXOFF Stops transmitting received characters to the Master Card (default). Syntax: SEND_COMMAND F2332_1, 'RXOFF' Example: SEND_COMMAND F2332_1, 'RXOFF' Stops the RS232_1 device from transmitting received characters to the Master Card. RXON Starts transmitting received characters to the Master Card. This command is sent automatically when issuing a CREATE_BUFFER SEND_COMMAND Syntax: SEND_COMMAND <dev>, 'RXOF' Example: SEND_COMMAND <dev>, 'RXON' Example: SEND_COMMAND F8232_1, 'RXON' Sets the RS232_1 device to transmit received characters to the Master Card. TXCLR Stops and clears all characters waiting in the transmit buffer. SEND_COMMAND F8232_1, 'RXON' Clears and stops all characters waiting in the RS232_1 device's transmit buffer.</dev></dev></dev>		<pre>SEND_COMMAND <dev>,'TSET BAUD (Baud),(Parity),(Data), (Stop)(485 DISABLE/ENABLE)'</dev></pre>		
Syntax: Syntax: SEND_COMMAND FS232_1, 'HSOFF' Disables hardware handshaking on the R5232_1 device. HSON Enables RTS (ready-to-send) and CTS (clear-to-send) hardware handshaking. Syntax: SEND_COMMAND <pev>, 'HSON' Example: SEND_COMMAND <pev>, 'HSON' Example: SEND_COMMAND <pev>, 'HSON' Example: SEND_COMMAND PARCE SEND_COMMAND PARCE PACACH RXCLR Clears all characters in the receive buffer waiting to be sent to the Master Card. Syntax: SEND_COMMAND PARCE SEND_COMMAND PARCE 'RXCLR' Elears all characters in the RS232_1 device's receive buffer waiting to be sent to the Master Card. RXOFF Stops transmitting received characters to the Master Card (default). Syntax: SEND_COMMAND PARCE SEND_COMMAND PARCE 'RXOFF' Stops the RS232_1 device from transmitting received characters to the Master Card. RXON Starts transmitting received characters to the Master Card. SIND_COMMAND PR232_1, 'RXOFF' SEND_COMMAND PR233_1, 'RXOFF' SEND_COMMAND PR233_1, 'RXOFF' SEND_COMMAND PR233_1, 'RXOFF' SEND_COMMAND PR233_1, 'RXOFF' SEND_COMMAND PR233_1, 'RXOFF' SEND_CO</pev></pev></pev>	HSOFF	Disables hardware handshaking (default).		
SEND_COMMAND <dev>, 'HSOFF' Example: SEND_COMMAND RE232_1, 'HSOFF' Disables hardware handshaking on the RS23_1 device. HSON Enables RTS (ready-to-send) and CTS (clear-to-send) hardware handshaking. Syntax: SEND_COMMAND <dev>, 'HSON' Example: SEND_COMMAND <dev>, 'HSON' Enables hardware handshaking on the RS232_1 device. RXCLR Clears all characters in the receive buffer waiting to be sent to the Master Card. Syntax: SEND_COMMAND <dev>, 'RXCLR' Example: SEND_COMMAND DEV2, 'RXCLR' SEND_COMMAND DEV2, 'RXCLR' Clears all characters in the RS232_1 device's receive buffer waiting to be sent to the Master Card. RXOFF Stops transmitting received characters to the Master Card (default). Syntax: SEND_COMMAND <dev>, 'RXOFF' SEND_COMMAND <dev>, 'RXOFF' Stops the RS232_1 device from transmitting received characters to the Master Card. RXON Starts transmitting received characters to the Master Card. Syntax: SEND_COMMAND <dev>, 'RXON' SEND_COMMAND <dev>, 'RXON' Example: SEND_COMMAND <dev>, 'RXON' SEND_COMMAND <dev>, 'RXON' SEND_COMMAND <dev>, 'RXON' SEND_COMMAND <dev>, 'RXON' SEND_COMMAND <dev>, 'RXON'</dev></dev></dev></dev></dev></dev></dev></dev></dev></dev></dev></dev></dev>		Syntax:		
Example: SEND_COMMAND_R232_1, 'HSOPF' Disables hardware handshaking on the RS232_1 device. HSON Enables RTS (ready-to-send) and CTS (clear-to-send) hardware handshaking. Syntax: SEND_COMMAND_R2332_1, 'HSON' Example: SEND_COMMAND_R2332_1, 'HSON' Example: SEND_COMMAND_R2332_1, 'HSON' Example: SEND_COMMAND_R2332_1, 'HSON' Enables hardware handshaking on the RS232_1 device. RXCLR Clears all characters in the receive buffer waiting to be sent to the Master Card. SYND_COMMAND_rEV2-, 'RXCLR' Example: SEND_COMMAND_rEV2-, 'RXCLR' Clears all characters in the RS232_1 device's receive buffer waiting to be sent to the Master Card. RXOFF Stops transmitting received characters to the Master Card (default). Syntax: SEND_COMMAND_rDV2-, 'RXOFF' Example: SEND_COMMAND_rD232_1, 'RXOFF' SEND_COMMAND_rDV2-, 'RXOFF' SEND_COMMAND_rDV2-, 'RXOFF' SEND_COMMAND_rDV2-, 'RXOFF'		SEND_COMMAND <dev>, 'HSOFF'</dev>		
Disables hardware handshaking on the RS232_1 device. HSON Enables RTS (ready-to-send) and CTS (clear-to-send) hardware handshaking. Syntax: SERD_COMMAND <pev>,'HSON' Example: SERD_COMMAND <pev>,'HSON' Enables hardware handshaking on the RS232_1 device. RXCLR Clears all characters in the receive buffer waiting to be sent to the Master Card. Syntax: SERD_COMMAND <pev>, 'RXCLR' Example: SERD_COMMAND <pev>, 'RXCLR' Clears all characters in the RS232_1 device's receive buffer waiting to be sent to the Master Card. RXOFF Stops transmitting received characters to the Master Card (default). Syntax: SERD_COMMAND <pev>, 'RXOFF' Example: SERD_COMMAND <pev>, 'RXOFF' Stops the RS232_1, 'RXOFF' Stops and clears all characters to the Master Card. This command is sent automatically when issuing a CREATE_BUFFER SEND_COMMAND. Syntax: SEND_COMMAND <pev>, 'RXON' Example: SEND_COMMAND <pev>, 'RXON' Example: SEND_COMMAND <pev>, 'RXON' Example: SEND_COMMAND <pev>, 'RXON' Example: SEND_COMMAND <pev>, 'TXCLR' Clears and stops all characters waiting in the transmit buffer. Syntax: SEND_COMMAND <pev>, 'TXCLR' Clears and stops all characters waiting in the RS232_1 device's transmit buffer.</pev></pev></pev></pev></pev></pev></pev></pev></pev></pev></pev></pev>		EXAMPLE.		
HSON Enables RTS (ready-to-send) and CTS (clear-to-send) hardware handshaking. Syntax: SEND_COMMAND <dev>, 'HSON' Example: SEND_COMMAND R5232_1, 'HSON' Enables hardware handshaking on the RS232_1 device. RXCLR Clears all characters in the receive buffer waiting to be sent to the Master Card. Syntax: SEND_COMMAND <dev>, 'RXCLR' Example: SEND_COMMAND <dev>, 'RXCLR' Clears all characters in the RS232_1 device's receive buffer waiting to be sent to the Master Card. RXOFF Stops transmitting received characters to the Master Card (default). Syntax: SEND_COMMAND <dev>, 'RXOFF' Example: SEND_COMMAND <dev>, 'RXOFF' Example: SEND_COMMAND <dev>, 'RXOFF' Example: SEND_COMMAND <dev>, 'RXOFF' Stops the RS232_1 device from transmitting received characters to the Master Card. RXON Starts transmitting received characters to the Master Card. This command is sent automatically when issuing a CREATE_BUFFER SEND_COMMAND. Syntax: SEND_COMMAND <dev>, 'RXON' Example: SEND_COMMAND <dev>, 'RXON' Sets the RS232_1 device to transmit received characters to the Master Card. TXCLR Stops and clears all characters waiting in the transmit buffer. Syntax: SEND_COMMAND <dev>, 'TXCLR' Example: SEND_COMMAND <s232_1, 'txclr'<br="">Clears and stops all characters waiting in the RS232_1 device's transmit buffer.</s232_1,></dev></dev></dev></dev></dev></dev></dev></dev></dev></dev>		Disables hardware handshaking on the RS232_1 device.		
Syntax: SEND_COMMAND <dev>, 'HSON' Example: SEND_COMMAND RS232_1, 'HSON' Enables hardware handshaking on the RS232_1 device. RXCLR Clears all characters in the receive buffer waiting to be sent to the Master Card. Syntax: SEND_COMMAND <dev>, 'RXCLR' Example: SEND_COMMAND <dev>, 'RXCLR' Clears all characters in the RS232_1 device's receive buffer waiting to be sent to the Master Card. RXOFF Stops transmitting received characters to the Master Card (default). Syntax: SEND_COMMAND <dev>, 'RXOFF' Example: SEND_COMMAND RS232_1, 'RXOFF' Stops the RS232_1 device from transmitting received characters to the Master Card. RXON Starts transmitting received characters to the Master Card. This command is sent automatically when issuing a CREATE_BUFFER SEND_COMMAND. Syntax: SEND_COMMAND <dev>, 'RXOFF' SEND_COMMAND <dev>, 'RXOFF' Starts transmitting received characters to the Master Card. Syntax: SEND_COMMAND <dev>, 'RXOFF' SEND_COMMAND <dev>, 'RXOFF' Starts transmitting received characters to the Master Card. SYntax: SEND_COMMAND <dev>, 'RXON' SEND_COMMAND <dev>, 'RXON' SEND_COMMAND <dev>, 'RXON' SEND_COMMAND <dev>, 'RXON' <th>HSON</th><th>Enables RTS (ready-to-send) and CTS (clear-to-send) hardware handshaking.</th></dev></dev></dev></dev></dev></dev></dev></dev></dev></dev></dev></dev>	HSON	Enables RTS (ready-to-send) and CTS (clear-to-send) hardware handshaking.		
SEND_COMMAND <dev>, 'HSON' Example: SEND_COMMAND R5232_1, 'HSON' Enables hardware handshaking on the RS232_1 device. RXCLR Clears all characters in the receive buffer waiting to be sent to the Master Card. Synta: SEND_COMMAND R5232_1, 'RXCLR' Clears all characters in the RS232_1 device's receive buffer waiting to be sent to the Master Card. RXOFF Stops transmitting received characters to the Master Card (default). Syntax: SEND_COMMAND <dev>, 'RXCLR' Clears all characters in the RS232_1 device's receive buffer waiting to be sent to the Master Card. RXOFF Stops transmitting received characters to the Master Card (default). Syntax: SEND_COMMAND <dev>, 'RXOFF' Example: SEND_COMMAND R5232_1, 'RXOFF' Stops the RS232_1 device from transmitting received characters to the Master Card. RXON Starts transmitting received characters to the Master Card. This command is sent automatically when issuing a CREATE_BUFFER SEND_COMMAND. Syntax: SEND_COMMAND <dev>, 'RXON' SEND_</dev></dev></dev></dev></dev></dev></dev></dev></dev></dev>		Syntax:		
Example: SEND_COMMAND RS232_1, 'HSON' Enables hardware handshaking on the RS232_1 device. RXCLR Clears all characters in the receive buffer waiting to be sent to the Master Card. Syntax: SEND_COMMAND RS232_1, 'RXCLR' Example: SEND_COMMAND RS232_1, 'RXCLR' Clears all characters in the RS232_1 device's receive buffer waiting to be sent to the Master Card. RXOFF Stops transmitting received characters to the Master Card (default). Syntax: SEND_COMMAND RS232_1, 'RXOFF' Example: SEND_COMMAND RS232_1, 'RXOFF' Stops the RS232_1 device from transmitting received characters to the Master Card. RXON Starts transmitting received characters to the Master Card. This command is sent automatically when issuing a CREATE_BUFFER SEND_COMMAND. Syntax: SEND_COMMAND RS232_1, 'RXOFF' Stops the RS232_1 device from transmitting received characters to the Master Card. RXON Starts transmitting received characters to the Master Card. This command is sent automatically when issuing a CREATE_BUFFER SEND_COMMAND. Syntax: SEND_COMMAND RS232_1, 'RXON' Starts the RS232_1 device to transmit received characters to the Master Card. TXCLR Stops and clears all characters waiting in the transmit buffer. Syntax: SEND_COMMAND <dev>, 'TXCLR' SEND_COM</dev>		SEND_COMMAND <dev>, 'HSON'</dev>		
Enables hardware handshaking on the RS232_1 device. RXCLR Clears all characters in the receive buffer waiting to be sent to the Master Card. Syntax: SEND_COMMAND <dev>, 'RXCLR' Example: SEND_COMMAND <dev>, 'RXCLR' Clears all characters in the RS232_1 device's receive buffer waiting to be sent to the Master Card. RXOFF Stops transmitting received characters to the Master Card (default). Syntax: SEND_COMMAND <dev>, 'RXOFF' Example: SEND_COMMAND <dev>, 'RXOFF' Example: SEND_COMMAND RS232_1, 'RXOFF' Stops the RS232_1 device from transmitting received characters to the Master Card. RXON Starts transmitting received characters to the Master Card. This command is sent automatically when issuing a CREATE_BUFFER SEND_COMMAND. Syntax: SEND_COMMAND <dev>, 'RXON' Example: SEND_COMMAND RS232_1, 'RXON' Example: SEND_COMMAND RS232_1, 'RXON' Example: SEND_COMMAND RS232_1, 'RXON' Sets the RS232_1 device to transmit received characters to the Master Card. TXCLR Stops and clears all characters waiting in the transmit buffer. Syntax: SEND_COMMAND RS232_1, 'TXCLR' Clears and stops all characters waiting in the RS232_1 device's transmit buffer.</dev></dev></dev></dev></dev>		Example:		
RXCLR Clears all characters in the receive buffer waiting to be sent to the Master Card. Syntax: SEND_COMMAND <dev>, 'RXCLR' Example: SEND_COMMAND RS232_1, 'RXCLR' Clears all characters in the RS232_1 device's receive buffer waiting to be sent to the Master Card. RXOFF Stops transmitting received characters to the Master Card (default). Syntax: SEND_COMMAND <dev>, 'RXOFF' Example: SEND_COMMAND RS232_1, 'RXOFF' Stops the RS232_1 device from transmitting received characters to the Master Card. RXON Starts transmitting received characters to the Master Card. This command is sent automatically when issuing a CREATE_BUFFER SEND_COMMAND. Syntax: SEND_COMMAND <dev>, 'RXON' Sets the RS232_1 device to transmit received characters to the Master Card. TXCLR Stops and clears all characters waiting in the transmit buffer. Syntax: SEND_COMMAND <dev>, 'RXON' Sets the RS232_1 device to transmit received characters to the Master Card. TXCLR Stops and clears all characters waiting in the transmit buffer. Syntax: SEND_COMMAND <dev>, 'TXCLR' Example: SEND_COMMAND <dev>, 'TXCLR' Clears and stops all characters waiting in the RS232_1 device's transmit buffer. </dev></dev></dev></dev></dev></dev>		Enables hardware handshaking on the RS232_1 device.		
Syntax: SEND_COMMAND <dev>, 'RXCLR' Example: SEND_COMMAND RS232_1, 'RXCLR' Clears all characters in the RS232_1 device's receive buffer waiting to be sent to the Master Card. RXOFF Stops transmitting received characters to the Master Card (default). Syntax: SEND_COMMAND <dev>, 'RXOFF' Example: SEND_COMMAND RS232_1, 'RXOFF' Stops the RS232_1 device from transmitting received characters to the Master Card. RXON Starts transmitting received characters to the Master Card. This command is sent automatically when issuing a CREATE_BUFFER SEND_COMMAND. Syntax: SEND_COMMAND <dev>, 'RXON' Sets the RS232_1, 'RXON' Sets the RS232_1, 'RXON' Sets the RS232_1 device to transmit received characters to the Master Card. TXCLR Stops and clears all characters waiting in the transmit buffer. Syntax: SEND_COMMAND <dev>, 'RXON' Sets the RS232_1 device to transmit received characters to the Master Card. TXCLR Stops and clears all characters waiting in the transmit buffer. Syntax: SEND_COMMAND <dev>, 'TXCLR' Example: SEND_COMMAND RS232_1, 'TXCLR' Clears and stops all characters waiting in the RS232_1 device's transmit buffer.</dev></dev></dev></dev></dev>	RXCLR	Clears all characters in the receive buffer waiting to be sent to the Master Card.		
SEND_COMMAND <dev>, 'RXCLR' Example: SEND_COMMAND RS232_1, 'RXCLR' Clears all characters in the RS232_1 device's receive buffer waiting to be sent to the Master Card. RXOFF Stops transmitting received characters to the Master Card (default). Syntax: SEND_COMMAND <dev>, 'RXOFF' Example: SEND_COMMAND <dev>, 'RXOFF' Stops the RS232_1 device from transmitting received characters to the Master Card. RXON Starts transmitting received characters to the Master Card. This command is sent automatically when issuing a CREATE_BUFFER SEND_COMMAND. Syntax: SEND_COMMAND <dev>, 'RXON' Starts transmitting received characters to the Master Card. This command is sent automatically when issuing a CREATE_BUFFER SEND_COMMAND. Syntax: SEND_COMMAND RS232_1, 'RXON' Example: SEND_COMMAND RS232_1, 'RXON' Stops and clears all characters waiting in the transmit buffer. Syntax: SEND_COMMAND <dev>, 'TXCLR' SenD_COMMAND RS232_1, 'TXCLR' Example: SEND_COMMAND RS232_1, 'TXCLR' Clears and stops all characters waiting in the RS232_1 device's transmit buffer.</dev></dev></dev></dev></dev>		Syntax:		
Example: SEND_COMMAND_RS232_1, 'RXCLR' Clears all characters in the RS232_1 device's receive buffer waiting to be sent to the Master Card. RXOFF Stops transmitting received characters to the Master Card (default). Syntax: SEND_COMMAND <dev>, 'RXOFF' Example: SEND_COMMAND RS232_1, 'RXOFF' Stops the RS232_1 device from transmitting received characters to the Master Card. RXON Starts transmitting received characters to the Master Card. This command is sent automatically when issuing a CREATE_BUFFER SEND_COMMAND. Syntax: SEND_COMMAND <dev>, 'RXON' SEND_COMMAND rS232_1, 'RXOFF' Starts transmitting received characters to the Master Card. This command is sent automatically when issuing a CREATE_BUFFER SEND_COMMAND. Syntax: SEND_COMMAND <dev>, 'RXON' SEND_COMMAND rS232_1, 'RXON' Sets the RS232_1 device to transmit received characters to the Master Card. TXCLR Stops and clears all characters waiting in the transmit buffer. Syntax: SEND_COMMAND <dev>, 'TXCLR' SEND_COMMAND rS232_1, 'TXCLR' SEND_COMMAND rS232_1, 'TXCLR' SEND_COMMAND rS232_1, 'TXCLR' Clears and stops all characters waiting in the RS232_1 device's transmit buffer.</dev></dev></dev></dev>		SEND_COMMAND <dev>, 'RXCLR'</dev>		
SEND_COMMAND RS232_1, 'RXCLR' Clears all characters in the RS232_1 device's receive buffer waiting to be sent to the Master Card. RXOFF Stops transmitting received characters to the Master Card (default). Syntax: SEND_COMMAND <dev>, 'RXOFF' Example: SEND_COMMAND RS232_1, 'RXOFF' Stops the RS232_1 device from transmitting received characters to the Master Card. RXON Starts transmitting received characters to the Master Card. This command is sent automatically when issuing a CREATE_BUFFER SEND_COMMAND. Syntax: SEND_COMMAND <dev>, 'RXON' Example: SEND_COMMAND RS232_1, 'RXON' Sets the RS232_1 device to transmit received characters to the Master Card. TXCLR Stops and clears all characters waiting in the transmit buffer. Syntax: SEND_COMMAND <dev>, 'TXCLR' Example: SEND_COMMAND <dev>, 'TXCLR' Clears and stops all characters waiting in the RS232_1 device's transmit buffer.</dev></dev></dev></dev>		Example:		
RXOFF Stops transmitting received characters to the Master Card (default). Syntax: SEND_COMMAND <dev>, 'RXOFF' Example: SEND_COMMAND RS232_1, 'RXOFF' Stops the RS232_1 device from transmitting received characters to the Master Card. RXON Starts transmitting received characters to the Master Card. This command is sent automatically when issuing a CREATE_BUFFER SEND_COMMAND. Syntax: SEND_COMMAND <dev>, 'RXON' SEND_COMMAND <dev>, 'RXON' SEND_COMMAND RS232_1, 'TXCLR' SEND_COMMAND RS232_1, 'TXCLR' Clears and stops all characters waiting in the RS232_1 device's transmit buffer.</dev></dev></dev>		SEND_COMMAND_RS232_1, 'RXCLR'		
KXOFF Stops transmitting received characters to the Master Card (default). Syntax: SEND_COMMAND <dev>, 'RXOFF' Example: SEND_COMMAND RS232_1, 'RXOFF' Stops the RS232_1 device from transmitting received characters to the Master Card. RXON Starts transmitting received characters to the Master Card. This command is sent automatically when issuing a CREATE_BUFFER SEND_COMMAND. Syntax: SEND_COMMAND <dev>, 'RXON' Example: SEND_COMMAND RS232_1, 'RXON' Sets the RS232_1 device to transmit received characters to the Master Card. TXCLR Stops and clears all characters waiting in the transmit buffer. Syntax: SEND_COMMAND <dev>, 'TXCLR' Example: SEND_COMMAND <dev>, 'TXCLR' Clears and stops all characters waiting in the RS232_1 device's transmit buffer.</dev></dev></dev></dev>	BYOFF	Clears an characters in the KS2S2_1 device's receive burlet waiting to be sent to the Master Card.		
Syntax. SEND_COMMAND <dev>, 'RXOFF' Example: SEND_COMMAND RS232_1, 'RXOFF' Stops the RS232_1 device from transmitting received characters to the Master Card. RXON Starts transmitting received characters to the Master Card. This command is sent automatically when issuing a CREATE_BUFFER SEND_COMMAND. Syntax: SEND_COMMAND <dev>, 'RXON' Example: SEND_COMMAND RS232_1, 'RXON' Sets the RS232_1 device to transmit received characters to the Master Card. TXCLR Stops and clears all characters waiting in the transmit buffer. Syntax: SEND_COMMAND <dev>, 'TXCLR' Example: SEND_COMMAND RS232_1, 'TXCLR' Clears and stops all characters waiting in the RS232_1 device's transmit buffer.</dev></dev></dev>	RXUFF	Stops transmitting received characters to the Master Card (default).		
Example: SEND_COMMAND_RS232_1, 'RXOFF' Stops the RS232_1 device from transmitting received characters to the Master Card. RXON Starts transmitting received characters to the Master Card. This command is sent automatically when issuing a CREATE_BUFFER SEND_COMMAND. Syntax: SEND_COMMAND SEND_COMMAND OEV>, 'RXON' Example: SEND_COMMAND RS232_1, 'RXON' Sets the RS232_1 device to transmit received characters to the Master Card. TXCLR Stops and clears all characters waiting in the transmit buffer. Syntax: SEND_COMMAND SEND_COMMAND SEV>, 'TXCLR' Example: SEND_COMMAND SEND_COMMAND RS232_1, 'TXCLR' Clears and stops all characters waiting in the RS232_1 device's transmit buffer.		SEND COMMAND <dev>, 'RXOFF'</dev>		
SEND_COMMAND RS232_1, 'RXOFF' Stops the RS232_1 device from transmitting received characters to the Master Card. RXON Starts transmitting received characters to the Master Card. This command is sent automatically when issuing a CREATE_BUFFER SEND_COMMAND. Syntax: SEND_COMMAND <dev>, 'RXON' Example: SEND_COMMAND RS232_1, 'RXON' Sets the RS232_1 device to transmit received characters to the Master Card. TXCLR Stops and clears all characters waiting in the transmit buffer. Syntax: SEND_COMMAND <dev>, 'TXCLR' Example: SEND_COMMAND <dev>, 'TXCLR' Command RS232_1, 'TXCLR' Clears and stops all characters waiting in the RS232_1 device's transmit buffer.</dev></dev></dev>		Example:		
Stops the RS232_1 device from transmitting received characters to the Master Card. RXON Starts transmitting received characters to the Master Card. This command is sent automatically when issuing a CREATE_BUFFER SEND_COMMAND. Syntax: SEND_COMMAND <dev>, 'RXON' Example: SEND_COMMAND RS232_1, 'RXON' Sets the RS232_1 device to transmit received characters to the Master Card. TXCLR Stops and clears all characters waiting in the transmit buffer. Syntax: SEND_COMMAND <dev>, 'TXCLR' Example: SEND_COMMAND <dev>, 'TXCLR' Clears and stops all characters waiting in the RS232_1 device's transmit buffer.</dev></dev></dev>		SEND_COMMAND RS232_1, 'RXOFF'		
RXON Starts transmitting received characters to the Master Card. This command is sent automatically when issuing a CREATE_BUFFER SEND_COMMAND. Syntax: SEND_COMMAND <dev>, 'RXON' Example: SEND_COMMAND RS232_1, 'RXON' Sets the RS232_1 device to transmit received characters to the Master Card. TXCLR Stops and clears all characters waiting in the transmit buffer. Syntax: SEND_COMMAND <dev>, 'TXCLR' Example: SEND_COMMAND <dev>, 'TXCLR' Clears and stops all characters waiting in the RS232_1 device's transmit buffer.</dev></dev></dev>		Stops the RS232_1 device from transmitting received characters to the Master Card.		
Syntax: SEND_COMMAND <dev>, 'RXON' Example: SEND_COMMAND RS232_1, 'RXON' Sets the RS232_1 device to transmit received characters to the Master Card. TXCLR Stops and clears all characters waiting in the transmit buffer. Syntax: SEND_COMMAND <dev>, 'TXCLR' Example: SEND_COMMAND RS232_1, 'TXCLR' Clears and stops all characters waiting in the RS232_1 device's transmit buffer.</dev></dev>	RXON	Starts transmitting received characters to the Master Card. This command is sent automatically when issuing a CREATE_BUFFER SEND_COMMAND.		
SEND_COMMAND <dev>, 'RXON' Example: SEND_COMMAND RS232_1, 'RXON' Sets the RS232_1 device to transmit received characters to the Master Card. TXCLR Stops and clears all characters waiting in the transmit buffer. Syntax: SEND_COMMAND <dev>, 'TXCLR' Example: SEND_COMMAND RS232_1, 'TXCLR' Clears and stops all characters waiting in the RS232_1 device's transmit buffer.</dev></dev>		Syntax:		
Example. SEND_COMMAND RS232_1, 'RXON' Sets the RS232_1 device to transmit received characters to the Master Card. TXCLR Stops and clears all characters waiting in the transmit buffer. Syntax: SEND_COMMAND <dev>, 'TXCLR' Example: SEND_COMMAND RS232_1, 'TXCLR' Clears and stops all characters waiting in the RS232_1 device's transmit buffer.</dev>		SEND_COMMAND <dev>, 'RXON'</dev>		
Sets the RS232_1 device to transmit received characters to the Master Card. TXCLR Stops and clears all characters waiting in the transmit buffer. Syntax: SEND_COMMAND <dev>, 'TXCLR' Example: SEND_COMMAND RS232_1, 'TXCLR' Clears and stops all characters waiting in the RS232_1 device's transmit buffer.</dev>		SEND COMMAND RS232 1. 'RXON'		
TXCLR Stops and clears all characters waiting in the transmit buffer. Syntax: SEND_COMMAND <dev>, 'TXCLR' Example: SEND_COMMAND RS232_1, 'TXCLR' Clears and stops all characters waiting in the RS232_1 device's transmit buffer.</dev>		Sets the RS232_1 device to transmit received characters to the Master Card.		
Syntax: SEND_COMMAND <dev>, 'TXCLR' Example: SEND_COMMAND RS232_1, 'TXCLR' Clears and stops all characters waiting in the RS232_1 device's transmit buffer.</dev>	TXCLR	Stops and clears all characters waiting in the transmit buffer.		
SEND_COMMAND <dev>, 'TXCLR' Example: SEND_COMMAND RS232_1, 'TXCLR' Clears and stops all characters waiting in the RS232_1 device's transmit buffer.</dev>		Syntax:		
Example: SEND_COMMAND RS232_1, 'TXCLR' Clears and stops all characters waiting in the RS232_1 device's transmit buffer.		SEND_COMMAND <dev>, 'TXCLR'</dev>		
Clears and stops all characters waiting in the RS232_1 device's transmit buffer.		Example:		
		Clears and stops all characters waiting in the RS232_1 device's transmit buffer.		

RS-232/42	RS-232/422/485 SEND_COMMANDs (Cont.)	
XOFF	Disables software handshaking (default). Syntax: SEND_COMMAND <dev>, 'XOFF' Example: SEND_COMMAND RS232_1, 'XOFF' Disables software handshaking on the RS232_1 device.</dev>	
XON	Enables software handshaking. Syntax: SEND_COMMAND <dev>, 'XON' Example: SEND_COMMAND RS232_1, 'XON' Enables software handshaking on the RS232_1 device.</dev>	

RS-232/422/485 SEND_STRING Escape Sequences

RS-232/422	RS-232/422/485 Send_String Escape Sequences		
27,17, 27,18,1	Sends device-specific break characters for a specified duration. Syntax: SEND_STRING <dev>, "27,17, <time>" Variable: Time = 1-255 in 100 microsecond increments Example: SEND_STRING RS232_1, "27,17,10" Sends a break character of 1 millisecond to the RS232_1 device. Sets the ninth data bit to 1 on all character transmissions. You can use this escape sequence with the B9MON</time></dev>		
	command. Syntax: SEND_STRING <dev>,"27,18,1" Example: SEND_STRING RS232_1,"27,18,1" Sets the RS232_1 device's ninth data bit to 1 on all character transmissions.</dev>		
27,18,0	Sets the ninth data bit to 0 on all character transmissions. You can use this escape sequence with the B9MON command. Syntax: SEND_STRING <dev>, "27,18,0" Example: SEND_STRING RS232_1, "27,18,0" Sets the RS232_1 devices ninth data bit to 0 on all character transmissions.</dev>		
27,19,	Inserts time delays before transmitting the next character. Syntax: SEND_STRING <dev>, "27,19, <time>" Variable: Time = 1-255 in 1 millisecond increments Example: SEND_STRING RS232_1, "27,19,10" Inserts a 10 millisecond delay before transmitting characters to the RS232_1 device.</time></dev>		
27,20,0	Sets the RTS hardware handshaking output to Low/Inactive. Syntax: SEND_STRING <dev>,"27,20,0" Example: SEND_STRING RS232_1,"27,20,0" Sets the RTS hardware handshaking output to Low on the RS232_1 device.</dev>		
27,20,1	Sets the RTS hardware handshaking output to High/Active. Syntax: SEND_STRING <dev>,"27,20,1" Example: SEND_STRING RS232_1,"27,20,1" Sets the RTS hardware handshaking output to High on the RS232_1 device.</dev>		

IR / Serial Ports (8 - 15) Channels

IR / Serial Ports Channels	
00001 - 00229	IR commands.
00229 - 00253	May be used for system call feedback.
00254	Power Fail. (Used with the 'PON' and 'POF' commands).
00255	Power status. (Shadows I/O Link channel status).
00256 - 65000	IR commands.

IR/Serial SEND_COMMANDs

The following IR and IR/Serial SEND_COMMANDs generate control signals for external equipment.

IR/Serial SE	IR/Serial SEND_COMMANDs		
CAROFF	Disables the carrier signal until a CARON command is received.		
	Syntax:		
	SEND_COMMAND <dev>, 'CAROFF'</dev>		
	Example:		
	SEND_COMMAND IR_1, 'CAROFF' Stops transmitting IP carrier signals to the IP 1 port		
CARON			
CARON	Enables Carrier Signais (detault setting).		
	SUILAX.		
	Example:		
	SEND_COMMAND IR_1, 'CARON'		
	Starts transmitting IR carrier signals to the IR_1 port.		
СН	Sends IR pulses to select a channel. All channels below 100 are transmitted as two digits.		
	• If the IR code for ENTER (#21) is loaded, an Enter will follow the number.		
	• If the channel is greater than or equal to 100, the IR function 127 is generated for the one hundred digit.		
	Syntax:		
	SEND_COMMAND <dev>, " 'CH', <number>"</number></dev>		
	Variable:		
	Number = 0-199		
	Example:		
	The NXI performs the following:		
	 Transmits IR signals for 1 (IR code 11). The transmit time is set with the CTON command. 		
	Waits until the time set with the CTOF command elapses.		
	Transmits IR signals for 8 (IR code 18).		
	• Waits for the time set with the CTOF command elapses. If the IR code for Enter (IR code 21) is programmed, the		
	NXI performs steps 5 and 6.		
	Transmits IR signals for Enter (IR code 21).		
	Waits for the time set with the CTOF command elapses.		
СР	Clears buffered IR commands, and sends a single IR pulse. You can set the Pulse and Wait times with the CTON and		
	Syntax:		
	SEND COMMAND <dev>, "'CP', <number>"</number></dev>		
	Variable:		
	Number = 1-252 and 256-65,000 (253-255 reserved)		
	Example:		
	SEND_COMMAND IR_1,"'CP',2"		
	Clears the active/buffered commands and pulses IR_1 port's channel 2.		
CTOF	Sets the duration of off time (no signal) between IR pulses for channel and IR function transmissions. Off time		
	settings are stored in non-volatile memory.		
	Ine factory default for channel off time is 5 (.5 second). This command is accessibled with the CD (single pulse) and CD (clear pulse) commands		
	This command is associated with the SP (single pulse) and CP (clear pulse) commands.		
	SYILLAR.		
	Variable:		
	Time = 0-255 in tenths of a second increments		
	Example:		
	SEND_COMMAND IR_1,"'CTOF',10"		
	Sets the off time between each IR pulse to 1 second.		

IR/Serial SEN	IR/Serial SEND_COMMANDs (Cont.)		
СТОМ	Sets the total time of IR pulses transmitted, and is stored in non-volatile memory. Syntax: SEND_COMMAND <dev>, " 'CTON', <time>" Variable:</time></dev>		
	Time = 0-255 in tenths of a second increments; default = 5 (.5 second). Example:		
	Sets the IR pulse duration to 2 seconds.		
GET MODE	Polls the IR/Serial ports and reports the active mode settings to the device requesting the information. Syntax: SEND_COMMAND <dev>, 'GET MODE' Example: SEND_COMMAND IR_1,'GET MODE' System response example: PORT 4 PD GUDDEDE DO LEVELO</dev>		
IROFF	Halts and clears all IR output on the designated port. Syntax: SEND_COMMAND <dev>, 'IROFF' Example: SEND_COMMAND IR_1, 'IROFF' Immediately halts and clears all IR output signals on the IR_1 port.</dev>		
POD	Disables active PON (power on) or POF (power off) command settings. Channel 255 changes are enabled. This command is used in conjunction with the I/O Link command. Syntax: SEND_COMMAND <dev>, 'POD' Example: SEND_COMMAND IR_1, 'POD' Disables PON and POF command settings on the IR_1 device.</dev>		
POF	 Turns off a device, based on input Link. If at any time the IR sensor reads that the device is on (such as if one turned it on manually at the front panel), the card automatically attempts to turn the device back off. If three attempts fail, the card will continue executing commands in the buffer. If there are no commands in the buffer, the card will continue to try until a 'PON' or 'POD' command is received. If it fails to turn the device off, a PUSH and RELEASE is made on channel 254 to indicate a power failure error. Channel 255 changes are disabled after receipt of this command. You can only use the PON and POF commands when an IR device has a linked I/O channel. Syntax: SEND_COMMAND <dev>, 'POF'</dev> Example: SEND_COMMAND IR_1, 'POF' Sends power down IR commands 28 (if present) or 9 to the IR_1 device. 		
PON	 Turns on a device, based on input Link. If at any time the IR sensor reads that the device is off (such as if one turned it off manually at the front panel), the card automatically attempts to turn the device back on. If three attempts fail, card will continue executing commands in the buffer. If there are no commands in the buffer, the card will continue to try until a 'POF' or 'POD' command is received. If it fails to turn the device on, a PUSH and RELEASE is made on channel 254 to indicate a power failure error. Channel 255 changes are disabled after receipt of this command. You can only use the PON and POF commands when an IR device has a linked I/O channel. Syntax: SEND_COMMAND <dev>, 'PON'</dev> Example: SEND_COMMAND IR_1, 'PON' Sends power up IR commands 27 or 9 to the IR_1 port. 		
PTOF	Sets the time between power pulses in .10-second increments, and is stored in permanent memory. Syntax: SEND_COMMAND <dev>, " 'PTOF', <time>" Variable: Time = 0-255 in tenths of a second increments; default = 15 (1.5 seconds). Example: SEND_COMMAND IR_1, " 'PTOF', 15" Sets the time between power pulses to 1.5 seconds for the IR_1 device.</time></dev>		

IR/Serial SEND_COMMANDs (Cont.)		
PTON	Sets the duration of power pulses in .10-second increments. Time is stored in permanent memory. Syntax: SEND_COMMAND <dev>, " 'PTON', <time>" Variable:</time></dev>	
	Time = 0-255 in tenths of a second increments; default = 5 (.5 seconds). Example: SEND_COMMAND IR_1, " 'PTON', 15" Sets the duration of the power pulse to 1.5 seconds for the IR_1 device.	
SET IO LINK	Links an IR or Serial port to an I/O channel for use with DE, POD, PON and POF commands. The I/O status is automatically reported on channel 255 on the IR port. Syntax: SEND_COMMAND <dev>, "'SET IO LINK <number>' Variable: Number = 1-8; set the I/O channel to 0 to disable I/O link settings.</number></dev>	
	Example: SEND_COMMAND IR_1," 'SET IO LINK 1'" Sets the IR_1 port link to I/O channel 1. The IR port uses the specified I/O input as power status for processing PON and POF commands.	
SET MODE	Sets the IR/Serial ports for IR or Serial-controlled devices connected to a Card Frame or NetModule. Syntax: SEND_COMMAND <dev>, 'SET MODE <mode>' Variable: Mode = IR or Serial Example: SEND_COMMAND IR_1, 'SET MODE IR' Sets the IR_1 port to IR mode for IR control.</mode></dev>	
SP	Generates a single IR pulse. You can use the CTON to set pulse lengths and CTOF for time off between pulses. Syntax: SEND_COMMAND <dev>, " 'SP', <ir out="">" Variable: IR OUT = 1-252 and 256-65,000 Example: SEND_COMMAND IR_1, " 'SP', 25" Pulses IR code 25 on IR_1 device.</ir></dev>	
ХСНМ	Changes the IR output pattern for the XCH command. Syntax: SEND_COMMAND <dev>, 'XCH-<mode>' Variable: Mode = 0-4 Example: SEND_COMMAND IR_1, 'XCH-3' Sets the IR_1 device's extended channel command to mode 3. Mode 0 Example (default): [x] [x] <x> <enter> SEND_COMMAND IR_1, 'XCH-3' Transmits the IR code as 3-enter. SEND_COMMAND IR_1, 'XCH-34' Transmits the IR code as 3-4-enter. SEND_COMMAND IR_1, 'XCH-343' Transmits the IR code as 3-4-3-enter. Mode 1 Example: <x> <x> <x> <enter> SEND_COMMAND IR_1, 'XCH-3 Transmits the IR code as 0-0-3-enter. SEND_COMMAND IR_1, 'XCH-3 Transmits the IR code as 0-3-4-enter. SEND_COMMAND IR_1, 'XCH-3 Transmits the IR code as 0-3-4-enter. SEND_COMMAND IR_1, 'XCH-34 Transmits the IR code as 3-4-3-enter.</enter></x></x></x></enter></x></mode></dev>	

IR/Serial SE	ND_COMMANDs (Cont.)
ХСНМ	Mode 2 Example:
	<x> <x> <x></x></x></x>
	SEND_COMMAND IR_1, 'XCH-3
	Transmits the IR code as 0-0-3.
	SEND_COMMAND IR_1, 'XCH-34
	Transmits the IR code as 0-3-4.
	SEND_COMMAND IR_1, 'XCH-343
	Transmits the IR code as 3-4-3.
	Mode 3 Example:
	[[100][100]] <x> <x></x></x>
	SEND_COMMAND IR_1, 'XCH-3
	Transmits the IR code as 0-3.
	SEND_COMMAND IR_1, 'XCH-34
	Transmits the IR code as 3-4.
	SEND_COMMAND IR_1, 'XCH-343
	Transmits the IR code as 100-100-4-3.
	Mode 4:
	Mode 4 sends the same sequences as the CH command. Only use Mode 4 with channels 0-199.
хсн	Transmits IR code in the format set with the XCHM mode command.
	Syntax:
	SEND_COMMAND <dev>,'XCH <channel>'</channel></dev>
	Variable:
	Channel = 0-999
ZAP HIGH	Deletes all IR data stored in the NXI ports 12-15.
	Syntax:
	SEND_COMMAND <dev>, 'ZAP HIGH'</dev>
	SEND_COMMAND IR_4, 'ZAP HIGH' Delates IP, commands in parts 12, 15 of the IP, 4 device
745101	Polotes all ID data stars d is the NVI starts 0.44
ZAP LOW	Deletes all IR data stored in the NXI ports 8-11.
	Syntax:
	SEND_COMMAND <dev>, 'ZAP LOW'</dev>
	SEND COMMAND TR 1. 'ZAP LOW'
	Deletes IR commands in ports 8-11 of the IR 1 device.
1	

Input/Output SEND_COMMANDs

The following SEND_COMMANDs program the I/O ports on the NXI.

GET INPUT	Cets the input channels active state
021 111 01	An active state can be high (logic high) or low (logic low or contact closure)
	Channel changes Pushes and Releases generate reports based on their active state
	Svntax:
	SEND COMMAND <dev>,'GET INPUT <chan>'</chan></dev>
	Variable:
	CHAN = 1-8
	Example:
	SEND_COMMAND IO, 'GET INPUT 1'
	Gets the I/O port's active state.
	System response:
	INPUT1 ACTIVE HIGH
SET INPUT	Sets the input channel's active state.
	An active state can be high (logic high) or low (logic low or contact closure).
	Channel changes, Pushes, and Releases generate reports based on their active state.
	Setting an input to ACTIVE HIGH will disable the output for that channel.
	Syntax:
	SEND_COMMAND <dev>,'SET INPUT <channel> <state>'</state></channel></dev>
	Variable:
	State = LOW or HIGH
	Example:
	SEND_COMMAND IO, 'SET INPUT 1 HIGH'
	Sets the I/O channel to detect a high state change, and disables output on the channel.



© 2016 Harman. All rights reserved. NetLinx, AMX, AV FOR AN IT WORLD, HARMAN, and their respective logos are registered trademarks of HARMAN. Oracle, Java and any other company or brand name referenced may be trademarks/registered trademarks of their respective companies. AMX does not assume responsibility for errors or omissions. AMX also reserves the right to alter specifications without prior notice at any time. The AMX Warranty and Return Policy and related documents can be viewed/downloaded at www.amx.com.
 3000 RESEARCH DRIVE, RICHARDSON, TX 75082 AMX.com | 800.222.0193 | 469.624.8000 | +1.469.624.7400 | fax 469.624.7153 AMX (UK) LTD, AMX by HARMAN - Unit C, Auster Road, Clifton Moor, York, Y030 4GD United Kingdom • +44 1904-343-100 • www.amx.com/eu/