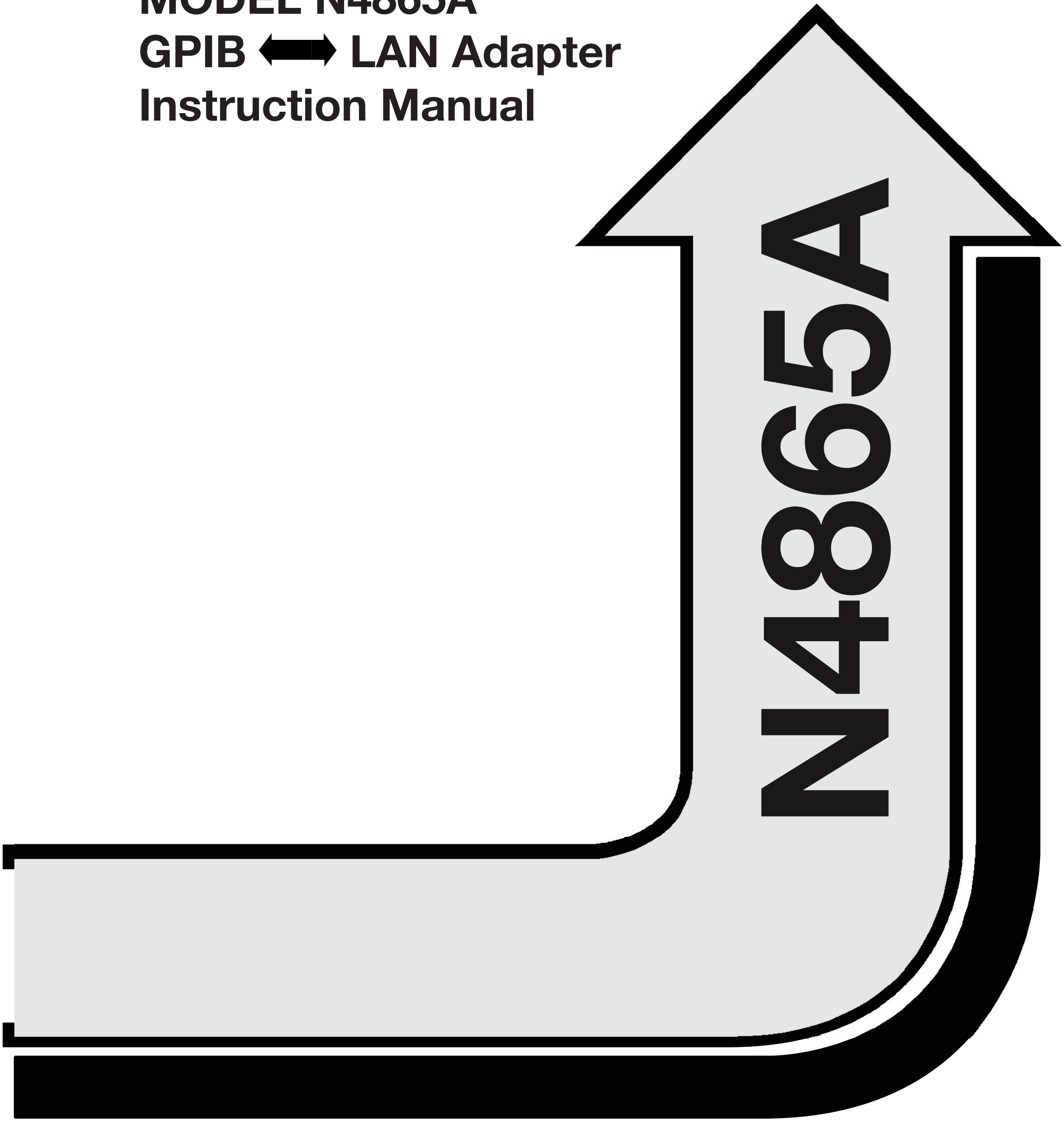
ICS
ELECTRONICS
a division of Systems West Inc.

MODEL N4865A
GPIB ⟷ LAN Adapter
Instruction Manual



MODEL N4865A
GPIB⟺LAN Adapter
Instruction Manual

ICS
ELECTRONICS
division of Systems West Inc.

7034 Commerce Circle, Pleasanton, CA 94588
Phone 925.416.1000, Fax 925.416.0105
Web Site http://www.icselect.com

Publication Number 120203
August 2009 Edition Rev x3

LIMITED WARRANTY

Within 12 months of delivery, ICS Electronics will repair or replace this product, at our option, if any part is found to be defective in materials or workmanship (labor is included). Return this product to ICS Electronics, or other designated repair station, freight prepaid, for prompt repair or replacement. Contact ICS for a return material authorization (RMA) number prior to returning the product for repair.

CERTIFICATION

ICS Electronics Corporation certifies that this product was carefully inspected and tested at the factory prior to shipment and was found to meet all requirements of the specification under which it was furnished.

EMI/RFI WARNING

This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause interference to radio communications. The Model N4865A has been tested and found to comply with the limits for a Class A computing device pursuant to Subpart J of Part 15 of the FCC Rules and to comply with the EEC Standards EN 55022 and EN 50082-1, which are designed to provide reasonable protection against such interference when operated in a commercial environment. Operation of this equipment in a residential area is likely to cause interference, in which case the user, at his own expense, will be required to take whatever measures may be required to correct the interference.

CE Certificate of Compliance reproduced in Figure 1-3.

TRADEMARKS

The following trademarks referred to in this manual are the property of the following companies:

VEE is a trademark of Agilent, Palo Alto, CA
LabView is a Trademark of National Instruments, Austin, TX
ICS and GPIB AnyWhere are trademarks of ICS Electronics, Pleasanton, CA

Contents

1

General Information

1.1 INTRODUCTION

This section provides a description and specifications for ICS's Model N4865A GPIB⟷LAN Adapter.

1.2 DESCRIPTION

The Model N4865A GPIB⟷LAN Adapter is a hardware module that provides an GPIB Interface for Agilent DSO/MSO7000 Series and DSO/MSO9000 Series oscilloscopes. Using the N4865A, GPIB applications running on a computer can transparently interface to a LAN-based oscilloscope just as if it were a GPIB oscilloscope

The Model N4865A is designed to interface with and control Agilent DSO/MSO7000 Series and DSO/MSO9000 Series oscilloscopes. The N4865A has two ways to connect to an oscilloscope after power turn-on and selftest. The user can specify an IP address for the oscilloscope and the N4865A will attempt to connect to it. If the N4865A's AutoFind function is enabled, the N4865A will do an UDP broadcast and attempt to connect to the first VXI-11.3 compatible oscilloscope that replies to the broadcast. When the N4865A has established VXI-11 channel connections to the oscilloscope, the N4865A is then ready to accept GPIB bus commands and device specific messages for its companion oscilloscope.

The N4865A follows the IEEE-488.2 Message Exchange Protocol when handling GPIB commands and device messages and should only be used with IEEE-488.2 compatible GPIB Controllers. GPIB commands are converted to VXI-11.3 equivalents and transmitted to the companion oscilloscope. Device specific data messages are transparently passed to the oscilloscope and oscilloscope replies are transparently passed back to the GPIB Controller.

The N4865A can accept reverse channel Service Requests and will assert SRQ on the GPIB bus after reading the oscilloscopes Status Byte Register.

The N4865A's operation and network settings are completely programmable by commands from its Ethernet interface, eliminating the need to open the unit to change or verify a function setting. The setup parameters can be changed with any Internet browser. All settings are stored in the N4865A's internal nonvolatile memory. A rear panel LAN Reset button allows the user to return the N4865A to its default network settings at any time.

The N4865A and its companion oscilloscope can be placed anywhere on the GPIB bus following the cabling guidelines in the IEEE-488.1 Standard. In addition, you can have multiple N4865A's on a GPIB bus, each with its unique GPIB address.

ICS's N4865A Interface is packaged in ICS's small metal Minibox™ case which provides proven EMI/RFI protection and rack mounting capability. Rear panel RJ-45 and GPIB connectors provide access to the network and GPIB bus. Front panel LEDs provide visual network and GPIB bus status and diagnostic help for troubleshooting system problems.

1.3 MODEL N4865A SPECIFICATIONS

The following specifications apply to all N4865A models. Options for your unit may be found by comparing the list below to those listed on the Ethernet label on your unit.

N4865A - X	General Model Number
	Option Codes

-6	Special settings
-7	Special Program
-8	Hardware modification
-9	Factory Rack Mounted

1.4 IEEE 488 INTERFACE

1.4.1 488.1 Capabilities

The N4865A's 488 Bus interface meets the IEEE STD 488.1-1987 standard and has the following capabilities:

SH1, AH1, T6, L4, SR1, PP0, DC0, RL0, DT0, C1 - C4 and C9

The N4865A's GPIB Bus drivers incorporate powerup/down protection to prevent sending invalid data to the bus. GPIB fanout is up to 14 devices.

1.4.2 GPIB Addresses

The N4865A's GPIB primary address is entered and saved in flash memory when the unit is configured. Configuring is done via the N4865A's Ethernet Interface.

Primary address range is 0 - 30. The default GPIB address is 4.

1.4.3 Conversion Operation

The N4865A follows the IEEE-488.2 Message Exchange Protocol when handling GPIB commands and device messages.

1.4.4 Device Messages

All device data messages sent to the N4865A must be terminated with EOI asserted on the last character. The N4865A sends data received from the GPIB bus to the companion oscilloscope with a *device_write* RPC. When the N4865A is addressed to talk, it sends a *device_read* RPC to the companion oscilloscope to read the expected reply. The N4865A will assert EOI when it outputs the last character of the received response if the END bit was asserted in the *device_read* response packet.

Large Listen messages are handled by holding off the GPIB bus while a buffer of data is transferred to the oscilloscope. Large data responses are processed by reading multiple talk buffers worth of data from the oscilloscope and outputting them to the GPIB bus until a response packet is received from the oscilloscope with the END bit asserted.

GPIB Listen buffer	1024 bytes (see Note 1)
GPIB Talk buffer	1024 bytes (see Note 1)

Note 1. Working buffer size may be smaller if the companion oscilloscope reports smaller maximum packet sizes in its *create_link* response.

When the N4865A is transistioned to the Listen Active State from the Talk Active State, it discards any data in the GPIB talk buffer. Similarly, when the N4865A is transistioned to the Talk Active State from the Listen Active State, it discards any data in the GPIB listen buffer

1.4.5 Data Transfer Rate

The following are short term data rates:

> 17 kbytes/second to the GPIB bus from 4865 memory.
> 20 kbytes/second from the GPIB bus to 4865 memory.

1.4.6 Transfer Delays

The following times are averages and vary depending upon what the 4865 was doing when the GPIB command or query occurred, the Ethernet network loading and oscilloscope reaction time.

Short device commands	18 milliseconds typical
*IDN? Queries	46 milliseconds typical

1.4.7 Remote-Local Operation

The N4865A uses the *device_remote* and *device_local* RPCs to transmit the GPIB Remote and Local states to the companion oscilloscope. The N4865A does not support the GPIB Remote with Lockout or the Local with Lockout states.

1.4.8 Device Clear and Selected Device Clear

When the N4865A receives a Selected Device Clear or a Device Clear on the GPIB bus, it sends a *device_clear* RPC to the companion oscilloscope and clears its GPIB talk and GPIB listen buffers.

1.4.9 Group Execute Trigger (GET)

When the 4896 receives a Group Execute Trigger (GET) on the GPIB Bus, any data in the GPIB listen buffer is sent to the companion oscilloscope and then the N4865A sends a *device_trigger* RPC to the companion oscilloscope.

1.4.10 SRQs

When the N4865A receives a Reverse Channel Service Request, it sends a *device_readstb* RPC to the companion oscilloscope to obtain the Status Register response and assert the GPIB SRQ signal. The SRQ signal is deasserted when the N4865A is Serial Polled or when the N4865A learns that the companion oscilloscope has deasserted its MSS bit.

1.4.11 Serial Poll Responses.

The N4865A periodically reads the companion oscilloscope's Status Byte Register and use the response to update its Status Byte. The oscilloscope polling rate is 3 Hz.

1.5 ETHERNET INTERFACE

The N4865A supports the Internet Protocol Suite, including the Transmission Control Protocol (TCP) and Internet Protocol (IP). TCP is used as the transport layer. The N4865A also supports RPC port mapper commands over UPD or over TCP and includes a web server capability.

1.5.1 Type and Speed

IEEE-802.3 Compliant. 10 Mbs with 10BaseT, 100 Mbs with 100BaseT

1.5.2 Network Address Ranges

The IP Address, Subnet Mask, and Gateway values can be set to static values from 0.0.0.0 to 255.255.255.255. Default values are listed in Table 1-2.

1.5.3 DHCP Capability

The N4865A has the capability to accept an address from a DHCP Server. If the N4865A is set to DHCP and does not receive a IP address, it defaults to a Class B AutoIP address of 169.254.48.65 with a Net Mask of 255.255.0.0.

1.5.4 MAC Address

The N4865A's MAC Address is saved internally and can be read with ICS's VXI-11 Configuration Utility or with a Web Browser. See Section 2.

1.5.5 Port Usage

The N4865A uses the ports listed in Table 1-1

TABLE 1-1 N4865A PORT USAGE

Port	Usage	Protocols	Notes
111	RPC Port Mapper	UDP, TCP	ICS Config Services
5556	Configuration Port, Error Logger	TCP	

1.5.6 HTML Pages

The N4865A's standard HTML pages conform to HTML version 4.01 or XHTML version 1.0. The required pages are needed for correct WebServer operation. User can redefine the other page names, page layouts or add additional pages. The WebServer serves the stored pages after substituting values for the variable placeholders. The standard N4865A pages are:

404.html	404 Error Page (required page)
501.html	501 Error Page (required page)
index.html	Welcome Page (required page)
config.html	Configuration Page
confirm.html	Confirmation Page
reboot.html	Reboot Page

1.5.7 Graphics

Image files with .jpg or .gif extensions are served as graphics

1.5.8 HTML User Configurability

The user can replace the standard HTML pages and image files with modified pages or add additional pages and images to the card. User is responsible for assuring that any substituted HTML pages conform to HTML version 4.01 or XHTML version 1.0. Guidelines for modifying the pages, HTML files and the Update Utility are described in Application Bulletin AB80-5.

File types supported	.html, .gif and .jpg
Number of files	32 maximum
File size	63 kbytes maximum for all files 32 kbytes maximum for a single file
File name size	27 characters

1.6. VXI-11 CONFORMANCE

The N4865A operates as a VXI-11 client and is fully compliant to the VXI-11 and VXI-11.3 Specifications.

1.6.1 RPC Protocol

The N4865A's RPC protocol conforms to ONC RPC Version 2 per the VXI-11 Standard.

1.6.2 VXI-11 Interface Names

The N4865A addresses its companion LAN oscilloscope as *inst0*.

1.6.3 Oscilloscope Linkage

If AutoFind is On, the N4865A does a UDP broadcast to the entire IP universe to find an VXI-11.3 compatible oscilloscope. The extent of the broadcast is determined by the network configuration. The N4865A will try to connect to the first VXI-11.3 compatible oscilloscope that replies. Connection capability is determined by the network topography.

If AutoFind is Off, no UDP broadcast is issued and the N4865A will try to link to an oscilloscope at the user designated ip address.

TABLE 1-2 N4865A PROGRAMMABLE FUNCTIONS AND FACTORY SETTINGS

Function	Choices	Default	Command Source (1)
IP Address Mode	Static or Dynamic	Dynamic	E
IP Address	0.0.0.0 to 255.255.255.255	169.254.48.65	E
Net Mask	0.0.0.0 to 255.255.255.255	255.255.0.0	E
Gateway IP	0.0.0.0 to 255.255.255.255	169.254.0.1	E
IP KeepAlive	On or Off	On	E
GPIB Address	0-30	0	E
Auto Disconnect Sockets	On or Off	Off	E
Oscilloscope IP Addresses	0.0.0.0 to 255.255.255.255	none	E
AutoFind	On of Off	On	E

Notes:
1. E = Ethernet Interface as the configuration port for Browsers or for ICS's VXI-11 Configuration Utility.
2. The N4865A's MAC Address is factory set and is not user changeable. The MAC Address can be read with the VXI-11 Configuration Utility or with a Web Browser.
3. Net Mask defaults to a Class B range.
4. Setting AutoFind ON causes the N4865A to ignore the Oscilloscope IP Address setting.

1.7 INDICATORS

The N4865A has eight front panel LEDs that normally display the following conditions:

PWR - Indicates power on
LAN - Indicates the unit is connected to an active Ethernet oscilloscope or to the network.
ACT - Indicates activity over the Ethernet connection.
RDY - Blinks at a 1 Hz rate while doing an AutoFind of the oscilloscope. Blinks at a 2 Hz rate while looking for the oscilloscope at a designated IP address. Solid on indicates the unit has passed self test and is linked to a LAN oscilloscope.
TALK - Indicates the unit is an active GPIB Talker.
LSTN - Indicates the unit is an active GPIB Listener.
SRQ - On when the N4865A has received a Service Request and is asserting SRQ on the GPIB bus.
ERR - Solid on when the N4865A runs into a problem that prevents it from continuing.

When the N4865A is turned on, it performs an internal selftest which takes about 4 seconds. Only the PWR LED is on during self test. At the end of a successful selftest, the N4865A blinks its GPIB address for 2 seconds. The LED bit weights are:

RDY	TALK	LSTN	SRQ	ERR
16	8	4	2	1

If DHCP is selected, the N4865A will wait 15-20 seconds for a DHCP server response before defaulting to its Auto IP address. It then starts looking for a companion oscilloscope. If the N4865A is connected to a network or oscilloscope, the LAN LED turns on and the ACT LED displays the N4865A's network activity.

If the N4865A finds a selftest error, it blinks the error code on its front panel LEDs. Refer to paragraph 5.4 for a description of the selftest errors and their possible causes.

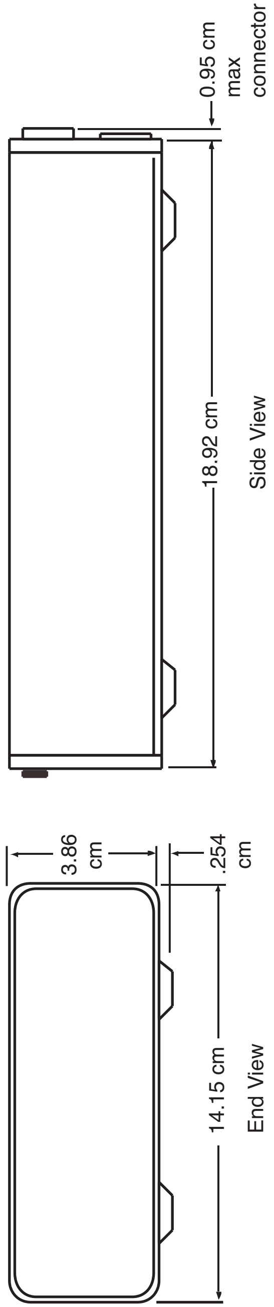


Figure 1-1 N4865A Outline Dimensions

1.8 PHYSICAL

Size	-	7.45" L x 5.57" W x 1.52" H (18.92 cm L x 14.15 cm W x 3.86 cm H) (See Figure 1-1)
Weight	-	1.37 lbs. (.6 kg.)
Temperature	-	Operating -10 °C to +55 °C Storage -40 °C to +70 °C
Humidity	-	0-90% RH without condensation
Shock/Vibration	-	Normal handling only
Construction	-	All metal case
Power	-	5 ± 0.2 Vdc @ 2.5 VA
Connectors	-	IEEE 488 Interface GPIB Connector with metric studs Ethernet Interface RJ-45 Connector Power Jack 2.45 mm plug, center positive
Controls	-	Power switch on front panel LAN Reset button on rear panel

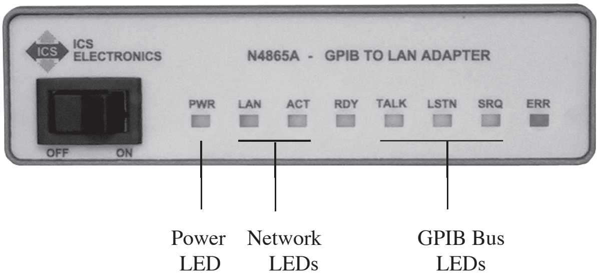


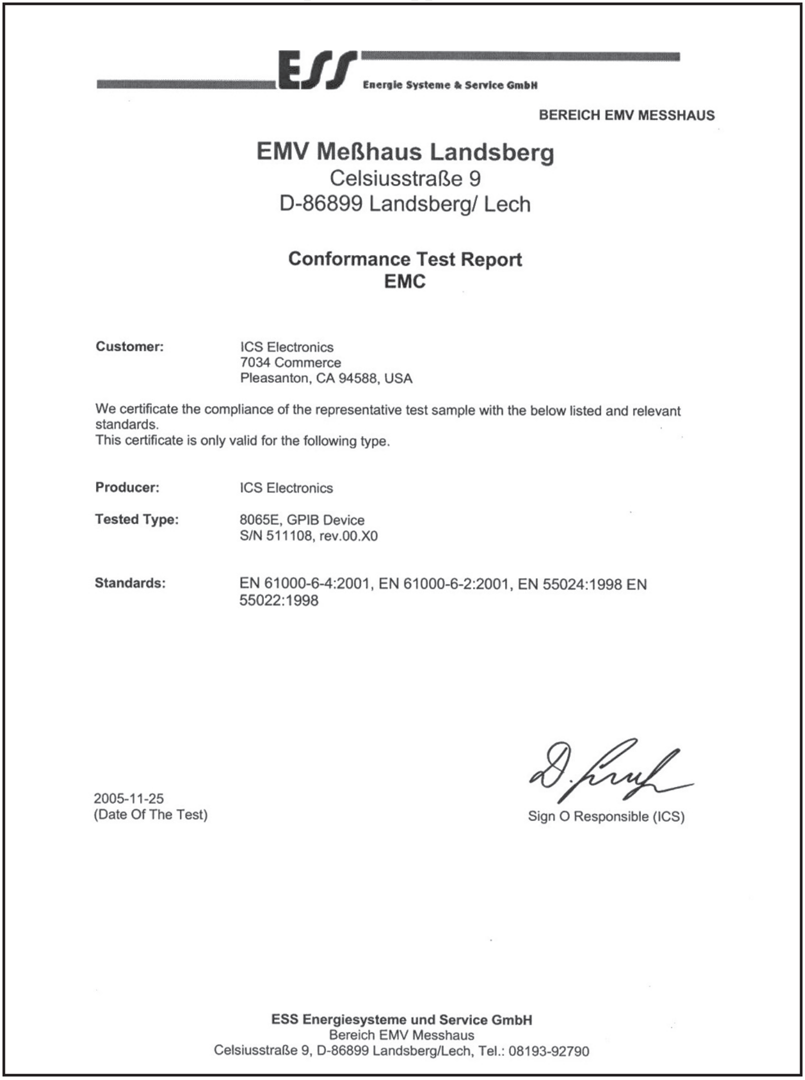
Figure 1-2 N4865A Front Panel Indicators

1.9 CERTIFICATIONS OR APPROVALS

EMI/RFI — Meets limits for part 15, Class A of US FCC Docket 20780 and complies with EEC Standards EN 55022 and 50082-1.

CE Certificate of Compliance reproduced in Figure 1-3. The N4865A uses the 8065 PCB assembly and 8065 Certificate.

UL/CSA/VDE — AC Wall adapter has applicable UL/CSA/VDE



ESS Energie Systeme & Service GmbH

BEREICH EMV MESSHAUS

EMV Meßhaus Landsberg
Celsiusstraße 9
D-86899 Landsberg/ Lech

Conformance Test Report
EMC

Customer: ICS Electronics
7034 Commerce
Pleasanton, CA 94588, USA

We certificate the compliance of the representative test sample with the below listed and relevant standards.
This certificate is only valid for the following type.

Producer: ICS Electronics

Tested Type: 8065E, GPIB Device
S/N 511108, rev.00.X0

Standards: EN 61000-6-4:2001, EN 61000-6-2:2001, EN 55024:1998 EN 55022:1998

2005-11-25
(Date Of The Test)

Sign O Responsible (ICS)

ESS Energiesysteme und Service GmbH
Bereich EMV Messhaus
Celsiusstraße 9, D-86899 Landsberg/Lech, Tel.: 08193-92790

Figure 1-3 Certificate of Conformity

1.10 INCLUDED ACCESSORIES

120204	Quick Start Guide
895011	Ethernet Crossover Cable
895014	USB Power Cable
123170	Support CD-ROM with Model N4865A Instruction Manual

1.11 OPTIONAL ACCESSORIES

120170	N4865A Instruction Manual
114210	Single Rack Mounting Kit
114211	Dual Rack Mounting Kit
104705	GPIB Cable, 0.5 meters long
104710	GPIB Cable, 1 meters long
104720	GPIB Cable, 2 meters long
104740	GPIB Cable, 4 meters long
895011	Ethernet Crossover Cable
895014	USB Power Cable

This page left intentionally blank

2

Installation

2.1 UNPACKING

When unpacking, check the unit for signs of shipping damage (damaged box, scratches, dents, etc.) If the unit is damaged or fails to meet specifications, notify ICS Electronics or your local sales representative immediately. Also, call the carrier immediately and retain the shipping carton and packing material for the carrier's inspection. ICS will make arrangements for the unit to be repaired or replaced without waiting for the claim against the carrier to be settled.

2.2 SHIPMENT VERIFICATION

Take a moment to verify that the following items were included with your unit:

(1)	Model N4865A GPIB to LAN Adapter
(1)	Quick Start Guide
(1)	USB Power Cord
(1)	Ethernet Crossover Cable
(1)	Support CD-ROM with Instruction Manual PDF

2.3 FACTORY CONFIGURATION

When shipped, the N4865A is configured to operate with a dynamic IP address that will fall back to an IP address of 169.254.48.65 if no DHCP server is found. Refer to Table 1-2 for all N4865A factory settings.

2.4 N4865A INSTALLATION GUIDE

The following steps should be used as a guide to setting up and using your N4865A. **Note - The N4865A is only configured through its Ethernet port with a Web Browser.**

1. The factory set N4865A defaults to an auto IP address of 169.254.48.65 if not connected to a DHCP server. If the companion LAN oscilloscope has an IP address in the range of 169.254.xxx.xxx it should successfully communicate with the N4865A without any changes to either's IP settings. If the oscilloscope has an IP address outside of the N4865A's range, either the N4865A or the oscilloscope needs to have its IP address changed. See paragraphs 3.1.1 and 2.5.

2. The N4865A's factory GPIB address is 4.

3. If the N4865A's settings are okay, skip step 3. Else use a Web Browser to configure the N4865A's network and/or GPIB settings as directed in paragraph 2.5. If you are resetting the oscilloscope's IP address, do not set the oscilloscope to the N4865A's own IP address.

3. If the N4865A is to be rack mounted, turn the N4865A off and disconnect all cables from the N4865A. Follow the instructions in Section 2.8 to install the N4865A in the rack mounting kit. Mount the rack mount assembly in the equipment rack.

4. After configuration, the N4865A can be connected to its companion LAN oscilloscope with the supplied Ethernet crossover cable as shown in Figure 2-1. Plug the Ethernet cable into the RJ-45 receptacle on the N4865A's rear panel and to the oscilloscope's RJ-45 connector. Use a GPIB Cable to connect the N4865A to the GPIB Controller.

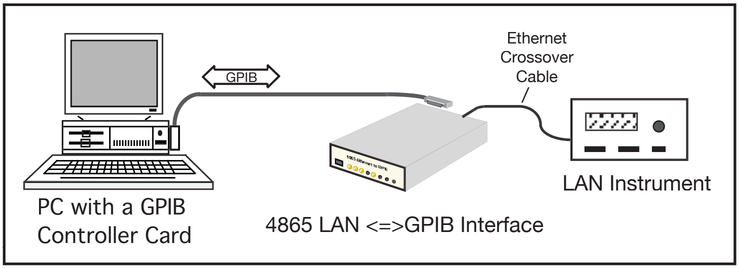


Figure 2-1 N4865A Direct Oscilloscope Connection

TABLE 2-1 N4865A CONFIGURATION SETTINGS

Setting	Choices	Comments
IP Address Mode	Static or DHCP	Static lets the user set the N4865A's IP Address, Net Mask and Gateway IP values. DHCP enables the N4865A to accept the IP values supplied by a DHCP Server. If there is no Server, the N4865A defaults to IP address 169.254.48.65 and Net Mask of 255.255.0.0. The factory setting is DHCP.
IP Address	Any	Sets the N4865A's own IP address. Any valid IP address setting. Must be four groups of numbers between 0.0.0.0 and 255.255.255.255. The default for static assignments is 192.168.0.254 .
Net Mask	Any	254.254.0.0. for B range operation.
Gateway IP	Any	Set to match the static IP. Recommend using 1 for the last octet. i.e. 192.168.0.1.
IP KeepAlive	On or Off	Enables the N4865A's socket layer to send the client a short test message after 120 minutes if there has been no activity on the socket. If the client fails to reply, the N4865A will close the socket, release any locks and reclaim all associated oscilloscope links. Do not enable Keep Alive if the network or the client does not support Keep Alive messages. Factory setting is On.
N4865A's GPIB Address	0 to 30	Sets the GPIB address the N4865A uses for itself on the GPIB bus. Values are 0 to 30. Recommended values are 0 and 21. Factory setting is 4.
Instrument IP	Any	Sets the oscilloscope's IP address. Not required if AutoFind is On. If AutoFind is Off, the N4865A will look for an oscilloscope at just this IP address. The oscilloscope IP address can be any valid IP address setting between 0.0.0.0 and 255.255.255.255. The default value is: 192.168.0.200.
AutoFind	On or Off	Enables the N4865A to ignore the Oscilloscope IP address setting and search for a VXI-11.3 compatible oscilloscope in the N4865A's B address range. The N4865A will attempt to open channels and link to the first oscilloscope it finds. Enable AutoFind for back-to-back connections only. Disable AutoFind when connecting to the oscilloscope over a network. Factory setting is On.

Note: Default values are listed in Table 1-2

2.5 N4865A CONFIGURATION METHODS

This paragraph provides two methods to configure the N4865A for operation with the companion oscilloscope. When shipped, N4865As are configured as listed in Table 1-2. AutoFind is enabled and the N4865A's GPIB address is 4.

Check the LAN oscilloscope to see how it is configured. If the oscilloscope is set to AutoIP or to an IP address in the AutoIP B Range of 169.254.xxx.xxx, it does not have to be changed. The N4865A will find it and link to it in back-to-back connections or when connected through a hub or switch. If the oscilloscope's IP address is not in the AutoIP B range, either it or the N4865A should be reconfigured for successful linkage. You have two choices:

1. Set the oscilloscope to use a static IP address in the N4865A's B address range. **Caution: Do not set the oscilloscope to the N4865A's own IP address. A suggested value is 169.254.90.00.**

2. Reset the N4865A's IP address and network settings to match the oscilloscope's settings or those used on your company network. For dedicated oscilloscope linkage, turn AutoFind off and enter the Oscilloscope's IP address in the Instrument IP Address window on the N4865A's Configuration Page.

2.5.1 Web Browser Configuration Method

The Web Browser method uses a standard browser such as Firefox, Internet Explorer or Netscape to view and change the N4865A's current settings.

1. Temporarily disconnect the computer from the company network and use the supplied Ethernet Crossover Cable to connect the computer to the unit as shown in Figure 2-2. This will eliminate any potential network conflicts while configuring the unit. Use the supplied USB power adapter to power the N4865A.

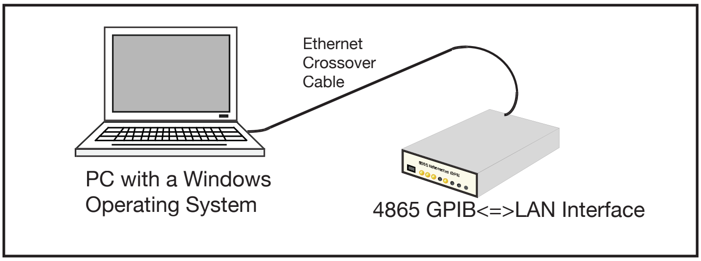


Figure 2-2 N4865A Connected to PC with a Crossover Cable

An alternate connection is to use a standard Ethernet Cable to connect the card to the same hub or switch that the computer running the browser is connected to as shown in Figure 2-3. Temporarily disconnect the switch from the company network.

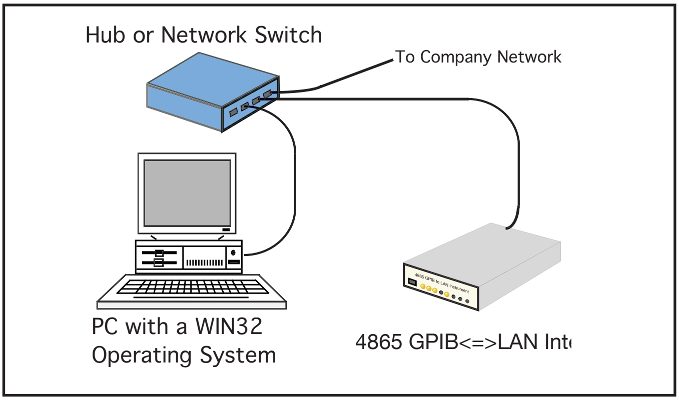


Figure 2-3 N4865A Connected to the local hub

2 Apply power to the unit. After 25 seconds, the RDY LED should start blinking which indicates that the N4865A is ready to communicate with the PC.

3 Check your computer's network settings to be sure its IP address is in the N4865A's B range of 169.254.xxx.xxx range so it can communicate with the unit. (To check, run ipconfig in the Command Prompt box). If the computer's IP address is not in the B range, its network settings must be changed before proceeding. For Windows PCs, right-click on My Network Places and click on Properties. Right-click on Local Area Connection and click Properties. Scroll down and highlight Internet Protocol (TCP/IP) and click the Properties button. If your PC's IP address is in a different range, record the current settings and temporarily set the following network values:

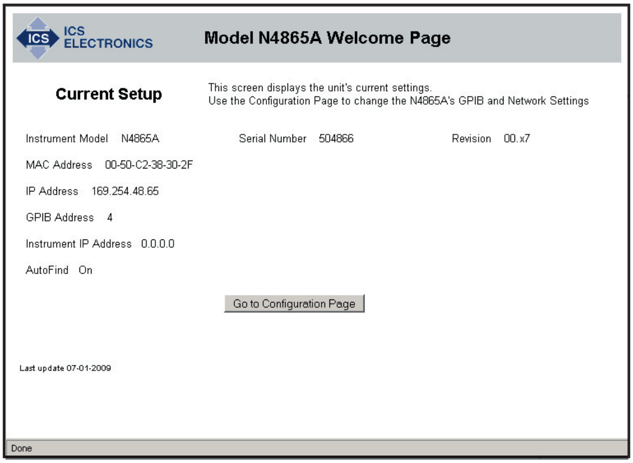
Check	'Use the following IP Address'
IP Address	169.254.0.2
Subnet mask	255.255.0.0

Click Okay twice to apply the new settings and close the open windows. Minimize the Network Places window.

4. Open the browser and enter the N4865A's default IP address of 169.254.48.65 in the browser address window. Press Enter.

5. A Welcome Page similar to the one shown Figure 2-4 should appear in your browser. Here you can read all of the information about the N4865A including its MAC number.

6. If you want to change any of the settings, press the 'Update Configuration' button. A Configuration Page similar to the one shown in Figure 2-5 should appear in your browser.



ICS ELECTRONICS

Model N4865A Welcome Page

Current Setup

This screen displays the unit's current settings.
Use the Configuration Page to change the N4865A's GPIB and Network Settings

Instrument Model N4865A   Serial Number 504866   Revision 00.x7

MAC Address 00-50-C2-38-30-2F

IP Address 169.254.48.65

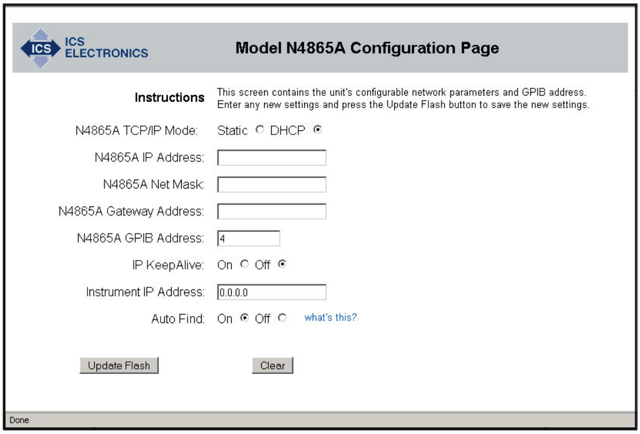
GPIB Address 4

Instrument IP Address 0.0.0.0

AutoFind On

Last update 07-01-2009

Figure 2-4 N4865A Welcome Page



ICS ELECTRONICS

Model N4865A Configuration Page

Instructions This screen contains the unit's configurable network parameters and GPIB address.
Enter any new settings and press the Update Flash button to save the new settings.

N4865A TCP/IP Mode: Static ○ DHCP ●

N4865A IP Address:

N4865A Net Mask:

N4865A Gateway Address:

N4865A GPIB Address: 4

IP KeepAlive: On ○ Off ●

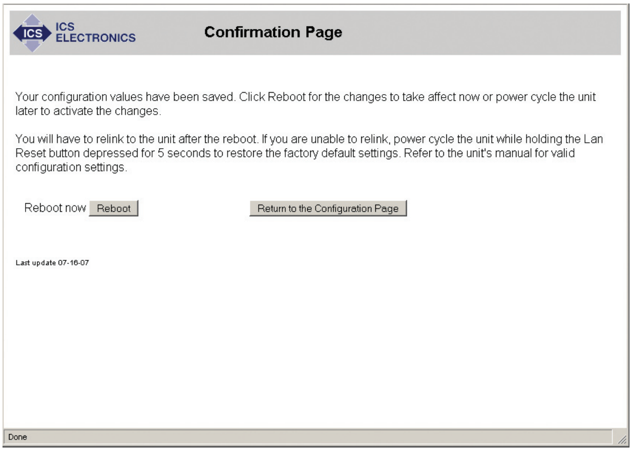
Instrument IP Address: 0.0.0.0

Auto Find: On ● Off ○ what's this?

Figure 2-5 N4865A Configuration Page

7. Enter the new network and GPIB Address settings as desired. If you change the N4865A to a Static IP address, record it so you can link to it

later on. Also set the Net Mask and Gateway values. Check the entered values carefully as the unit's webserver does minimal error checking. Press the 'Update Flash' button when done. A Confirmation Page similar to the one shown in Figure 2-6 will appear in your browser.



Confirmation Page

Your configuration values have been saved. Click Reboot for the changes to take affect now or power cycle the unit later to activate the changes.

You will have to relink to the unit after the reboot. If you are unable to relink, power cycle the unit while holding the Lan Reset button depressed for 5 seconds to restore the factory default settings. Refer to the unit's manual for valid configuration settings.

Last update 07-16-07

Figure 2-6 N4865A Confirmation Page

8. The Confirmation Page affirms that the new settings have been saved in the unit's flash memory. Press the 'Reboot' button to reboot the unit and activate the new settings. A Rebooting Page similar to the confirmation page will appear in your browser.

9. Relink to the N4865A and verify that all of your settings are correct. You may have to change the PC's Local Area Network settings if you selected a Static IP outside of the AutoIP range of 169.254.xxx.xxx.

10. Connect the PC back to the company network if it had been disconnected and restore its Local Area Network settings to the original values recorded in Step 3.

2.5.2 RPC Configuration Method

The RPC Configuration method involves using selected RPC commands from those listed in Appendix 3 to change the N4865A settings. Refer to Appendix 2 and ICS's Application Notes for more information about using RPC commands.

2.6 LED HEADER (J4)

The N4865A has an internal header for driving external LEDs. The N4865A provides 5 Vdc to power the external LEDs. User should provide current limiting resistors to keep the LED current below 10 mA per LED. Cathode side of the LEDs connects to the N4865A's active low LED drive signals. J4 pinouts are listed in Table 2-3

TABLE 2-3 LED HEADER PINOUTS (J4)

Pin	Signal	Pin	Signal
1	5 Vdc	6	LSTN
2	RDY	7	SRQ
3	LAN	8	ERROR
4	ACT	9	no connection
5	TALK	10	Ground

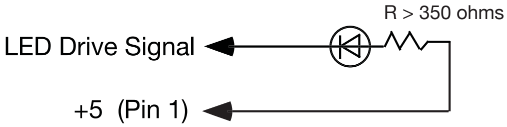


Figure 2-7 External LED Circuit

2.7 JUMPER SETTINGS

The N4865A has the following jumper positions. See Figure 2-8 on the next page:

TABLE 2-2 N4865A JUMPERS

Jumper	Function	Factory Setting
W1	Default Restore. See paragraph 5.6	Open
W2	Option Jumper. Selects N4865A firmware	Installed
W5	SRQ LED signal selection SRQ selects GPIB bus SRQ signal. LAT selects N4865A processor output latch	LAT
W7 W8	JTAG Loading Jumpers. For factory test use only.	Open ARM

Notes: Open jumpers may be parked on a single post.

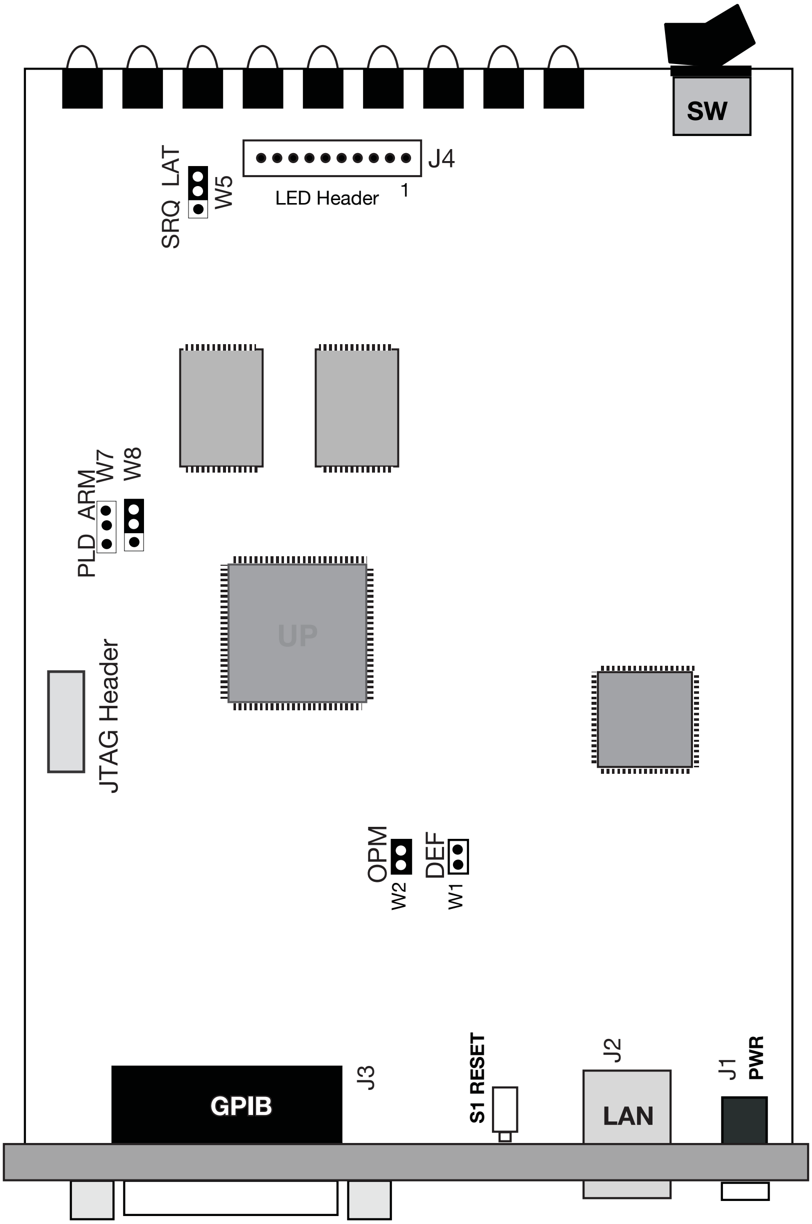


Figure 2-8 N4865A Jumper Locations

2.8 N4865A RACK MOUNTING INSTRUCTIONS

The Model N4865A is held in its rack mounting kit with a winged-'U' shaped bracket. Perform the following steps to install a N4865A in a rack mounting kit:

1. Hold the N4865A at a 30 degree nose down angle and place the front bezel through the rack mount kit from the rear of the kit. Push it forward through the opening until the rubber feet line up with the holes in the rack mounting kit. Push the unit down until it rests flat on the kit and the feet are in the four holes.

2. Repeat step 1 for a second unit if two units are being held in one rack mounting kit.

3. Align the unit(s) so the bezels are parallel with the front of the rack mount kit and protrude equally through the front panel of the rack mounting kit.

4. Set the bracket so its two holes line up with the holes in the rack mounting kit extrusion. Use the supplied 4-40 screws to hold the bracket to the extrusion. Do not overtighten.

5. Use the supplied 10-32 screws to bolt the rack mounting kit into the rack.

This page left intentionally blank

3

Operation

3.1 INTRODUCTION

This section describes the operation of the N4865A GPIB⟺LAN Adapter.

3.2 OPERATION

3.2.1 Oscilloscope Discovery and Linkage

At power turn-on, the N4865A performs a selftest for 4 seconds and then blinks its GPIB address. If the N4865A is set for dynamic IP addressing, it attempts to locate a server for 15-20 seconds before defaulting to its Auto IP address of 169.254.48.65. If the N4865A is set to a static IP, it immediately assumes that IP address.

If AutoFind is enabled, the N4865A does a UDP broadcast and waits for a reply. It then queries any replying oscilloscope to see if it has a VXI-11.3 RPC Service. If one is found, the N4865A attempts to open communication channels with it. The N4865A will attempt to open communication channels with the first VXI-11.3 oscilloscope it finds.

While the N4865A can communicate with an oscilloscope at any legal IP address, the N4865A's ability to communicate with an oscilloscope is limited by the network topology, its Net Mask and Gateway settings. The factory set N4865A can communicate to an oscilloscope in the AutoIP B address range of 169.254.xxx.xxx. To improve chances of a successful linkage, connect the N4865A to its companion oscilloscope in a back-to-back configuration or through a hub. Check the Net Mask and Gateway settings of all of the devices in the communication path for their affect on communication between the N4865A and the oscilloscope.

If AutoFind is disabled and the N4865A has been configured to use a specific oscilloscope IP address, then the N4865A does not do a UDP broadcast and instead attempts to communicate directly with a VXI-11.3 oscilloscope at the designated IP address.

When the N4865A finds a VXI-11.3 compatible oscilloscope it opens a core and abort channel to the oscilloscope. The N4865A will also set up a reverse Interrupt channel if the oscilloscope supports it.

The N4865A's RDY LED blinks during the oscilloscope discovery process, which can last an indefinite period of time, and becomes solid on when the N4865A links with a LAN oscilloscope and becomes ready to transmit GPIB Commands and Device Messages. The RDY LED will stay on until the N4865A fails to receive a reply to a device RPC. If the N4865A looses connection with the oscilloscope, it will attempt to reconnect to the oscilloscope.

3.2.2 GPIB Operation

The N4865A only operates correctly with an IEEE-488.2 compatible GPIB Controller. All device messages sent to the N4865A must be terminated with EOI asserted on the last character as the N4865A does not check for message terminators. The N4865A will assert EOI when it outputs the last character in the response received from the LAN oscilloscope if the END bit was asserted in the response packet.

When the N4865A receives a terminated device message from the GPIB bus the N4865A sends the data to the companion oscilloscope with the *device_write* RPC. The N4865A holds the GPIB bus while transmitting the data to the companion oscilloscope. In the case of large messages, the N4865A sends the data to the companion oscilloscope when its GPIB listen buffer becomes full. When the first buffer load of data has been sent to the companion oscilloscope, the N4865A accepts the next buffer worth of data. The process repeats with the N4865A handling a buffer worth of data at a time until the N4865A receives the last portion of the data with EOI asserted on the last byte. Unterminated device messages are normally not sent to the companion oscilloscope and are discarded when the N4865A becomes an active talker or is sent a GPIB clear or trigger command.

When the N4865A is made an active talker, it sends a *device_read* RPC to the companion oscilloscope to read the expected data. When the N4865A receives response data from the companion oscilloscope that is terminated with END asserted in the response packet, the data is output on the GPIB bus with EOI asserted on the last byte. When the data size exceeds the N4865A's GPIB talk buffer, the N4865A outputs the first buffer of data to the GPIB bus and sends an additional *device_read* RPC to the oscilloscope to read the next buffer's worth of data. This process repeats until all of the data has been read from the companion oscilloscope and the N4865A receives a response data packet with END asserted. If the companion oscilloscope returns a zero length response, the N4865A will not have any data for the GPIB bus and the GPIB read operation will timeout.

When the N4865A is addressed to talk and has received a response packet with END asserted, it will not fetch additional data from the companion oscilloscope until it has been addressed to listen and sent a device message, a GPIB Device Trigger, a GPIB Device Clear command or an Interface Clear.

When the N4865A is transistioned to the Listen Active State from the Talk Active State, it discards any data in the GPIB talk buffer. Similarly, when the N4865A is transistioned to the Talk Active State from the Listen Active State, it discards any data left in the GPIB listen buffer.

If the N4865A receives a Service Request from the oscilloscope over the Reverse Interrupt channel, the N4865A will query the oscilloscope's Status Byte Register with a *read_statusstb* RPC when it is safe to do so. If bit 6 in the Status Byte is set, the N4865A will assert the SRQ line on the GPIB bus and turn on its front panel SRQ LED. If the N4865A is Serial Polled, it will immediately output the oscilloscope's Status Byte when addressed to talk. The SRQ line will be deasserted and the SRQ LED will be turned off. Future register queries and the response to the Service Request are dependent upon the program in the GPIB Controller.

The N4865A periodically queries the companion oscilloscope's Status Byte to update its own Status Byte at a low 3 Hz rate. If the N4865A discovers that the oscilloscope is no longer asserting its MSS bit, then the N4865A will deassert the SRQ signal on the GPIB bus and turn off its SRQ LED. The N4865A's SRQ signal turn off will lag behind the companion oscilloscope due to the N4865A's low query rate and the user's program.

Status Byte queries are inhibited if the N4865A was addressed to listen and a device message was written to the companion oscilloscope, to avoid corrupting any expected oscilloscope response. Status Byte queries resume when the N4865A has finished outputting response data, after a GPIB Device Clear or a GPIB Device Trigger command or after an Interface Clear.

3.2.3 Web Server

The N4865A includes an HTML server that provides a Welcome html page that lets a user view the N4865A's properties, its network and GPIB Bus settings and the AutoFind status. The Welcome page also displays the companion oscilloscope's IP address if the N4865A is linked to an oscilloscope or the user set oscilloscope address if AutoFind is disabled and an IP address has been entered. A link is provided to a configuration page for changing the N4865A's network, GPIB address, oscilloscope IP address and AutoFind settings.

The Configuration page is self explanatory. Select DHCP for dynamic IP operation that will cause the N4865A search for a DHCP server for 20 seconds before defaulting to an IP address of 169.254.48.65. Select Static to operate the N4865A at a fixed IP address and then enter the desired IP address, Net Mask and Gateway values. Enable AutoFind for easy linking to any oscilloscope.

A Confirmation page lets the user save the new settings and restart the N4865A so the changes can take affect. For more details, see the instructions in Section 2.5 on configuring the N4865A with a browser.

3.3 PROGRAMMING GUIDELINES

The N4865A can be used successfully to operate a LAN oscilloscope with most test applications if a few precautions are taken. Applications that do normal oscilloscope command or query sequences should have no problems. Applications that depend upon oscilloscope timing may have to be adjusted to compensate for delays in converting the commands and transmitting them over the Ethernet.

The following multiple sequences should be avoided:

3.3.1 Multiple Triggers

Device Triggers can be sent by using the IEEE-488.1 GET command or with the 488.2 '*TRG' Common Command. The N4865A may miss multiple GET commands when sent to it at high speed. Multiple triggers are not found in normal test applications as the trigger is typically followed by some GPIB commands to read the data resulting from the trigger action. If multiple triggers are required, use the 488.2 '*TRG' Common Command instead of the 488.1 GET.

3.3.2 Multiple Device Clears

Device Clear and Selected Device Clear are sent by using the IEEE-488.1 DC or SDC commands. Normally a Device Clear is followed by a delay to allow the oscilloscope(s) to clear their buffers and output lines. Many oscilloscopes will not be able to input a new GPIB command for several hundred milliseconds since they also clear their GPIB input buffer as part of the clearing process. The N4865A may miss multiple DC or SDC commands when sent to it at high speed. If multiple Device Clears are required, provide a minimum 100 milliseconds delay between commands.

3.3.3 Multiple Serial Polls

Multiple high speed Serial Polls will return the same value. The N4865A's GPIB Interface responds to a GPIB Serial Poll by outputting its current value when polled. The N4865A updates its Status Byte after receiving a Service Request and before asserting the SRQ signal and periodically thereafter at a 3 Hz rate when the N4865A is left in a talk address or not-addressed state. Serial Poll updates are inhibited while the N4865A is in the listen addressed state and resume after talking out an oscilloscope response message, after a Device Trigger, after a Device Clear command or after an Interface Clear.

3.3.4 Message Terminators

Check existing programs for the correct termination sequence. The N4865A requires that all Device data messages sent to it be terminated with EOI asserted on the last byte. The LAN oscilloscopes want their messages terminated with a linefeed character. Therefore all messages to the N4865A should be terminated with both a linefeed character and EOI asserted. (The IEEE-488.2 NL terminator)

The N4865A's GPIB responses are terminated with EOI asserted on the last byte. Any terminating character is supplied by the companion oscilloscope as part of its response message.

3.3.5 Transferring Large Files

The N4865A transfers large data files in blocks of one Talk buffer or one Listen buffer worth of data at a time. These buffers are a maximum of 1,024 bytes. Users should set their program to read data in blocks that are multiples of 1,024 bytes for best performance. The user's program should not untalk or unlisten the N4865A in the middle of transferring large data files. Setting readdress off is recommended for applications that transfer only portions of the data at a time. Timeouts should be set for 1 second per kbyte of data transfer.

3.3.6 Linkage Lost

When the N4865A links to a oscilloscope, it turns the RDY LED solid on after it receives a good reply to the *create_link* RPC. The N4865A only knows the oscilloscope is alive an well when it receives a response to the data transfer RPCs. There is nothing to alert the N4865A to the oscilloscope being turned off or loosing connection until it fails to respond to a RPC. The time between RPCs depends upon the Test Application. If the Test Application is quiescent and the N4865A is not a listener, the N4865A polls the oscilloscope at a 3 HZ rate.

When the N4865A does not receive a valid RPC response, it turns off the RDY LED and attempts to relink to the oscilloscope. If it is successful, the N4865A will resume transferring data between the GPIB bus and its companion oscilloscope. The N4865A has no way to alert the GPIB Controller to this problem except for letting it timeout.

Any data being transferred when the oscilloscope fails to respond to the RPC should be considered lost. The user will have to consider this in his program.

3.4 OEM DOCUMENTATION AND CONFIGURATION

3.4.1 WebServer Pages

The OEM can customize the N4865A's WebServer configuration pages to identify the product and incorporate the company logo by following the guidelines in Application Bulletin AB80-5.

3.4.2 End-User Documentation

OEM users of the this interface N4865A should provide the end-user with the instructions and utility programs necessary to operate the complete system. This is not done by passing on the N4865A manual to the end-user since it does not relate to the end product. In most cases the end-user needs directions for:

1. Setting the N4865A's GPIB Address.
3. Using GPIB commands to control the companion LAN oscilloscope. (Includes sending outputs and reading inputs if applicable). The OEM needs to define the commands in terms of what they do to the overall product and show the end-user how to use them.
4. Using special commands like the trigger functions if applicable.
5. Using the 488.2 Status Reporting Structure. The OEM needs to define how to enable Service Requests (SRQs) and how to read the oscilloscope's Status Registers.

The N4865A is transparent and does not have any SCPI commands.

3.4.3 Utility Programs and Drivers

None associated with the N4865A.

3.4.4 Copyright Release

OEM users of the N4865A GPIB⬌LAN Adapter are hereby given permission to copy any portion of this manual, referenced ICS material and utility or example programs for the purpose of documenting systems, maintaining or enhancing sales of systems that incorporate ICS's interfaces. Reproduction of this manual for other purposes without the expressed written consent of ICS Electronics is forbidden.

This page left intentionally blank

4

Theory of Operation

4.1 INTRODUCTION

This section describes the theory of operation of the N4865A.

4.2 FUNCTIONAL DESCRIPTION

A block diagram of the N4865A is shown in Figure 4-1. The N4865A is a microprocessor based device that performs the VXI-11 client functions to control an Agilent oscilloscope (instrument) from a GPIB bus. The N4865A also functions as a HTML server for configuration.

The N4865A is made up of seven major elements, most of which are interconnected to the microprocessor by a common data, address and control signal bus.

Incoming GPIB commands or device messages are received by the GPIB Interface Chip. Each received command or message interrupts the microprocessor so it can fetch the message from the GPIB Interface chip. If the decoded message is an GPIB device command, it is converted into the equivalent VXI-11 command for the Ethernet Interface chip. Device messages are packetized and output through the Ethernet Interface chip to the LAN instrument. Response packets from the LAN instrument are received by the LAN Interface chip and placed in the GPIB talk buffer. The response data in the GPIB talk buffer is output on the GPIB bus when the N4865A is addressed to talk.

Other commands like Device Clear and Device Trigger are converted into their VXI-11 equivalents and sent through the Ethernet interface to the LAN instrument. A GPIB Serial Poll causes the N4865A to execute a Read Status Byte on the LAN instrument.

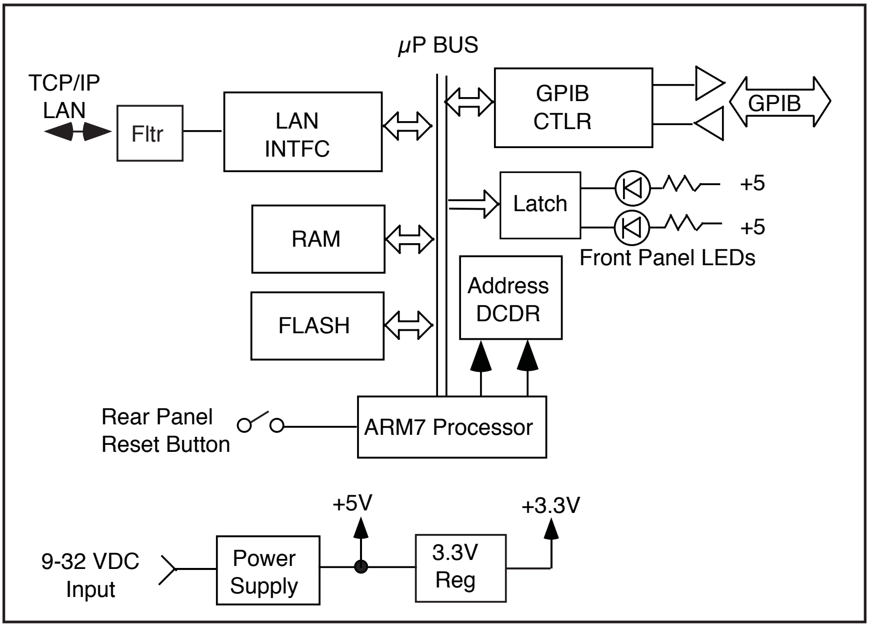


Figure 4-1 N4865A Block Diagram

4.3 BLOCK DIAGRAM DESCRIPTION

Figure 4-1 shows a block diagram of the N4865A's internal logic. Most of the major logic elements center around the 16-bit data bus from the ARM7 Processor. The N4865A has a LAN Interface, a static 1 Mbyte RAM chip, a FLASH, a GPIB Interface and a LED Latch chip, all controlled by an ARM7 Processor. The FLASH chip stores the program code and the configuration parameters. At power turn-on time, the ARM7 processor loads the program code from the FLASH into the RAM chip. The program is then run out of the RAM chip.

At power turn-on, the processor clears the LEDs, does a brief check of the logic elements and then checks the FLASH to find the correct program. If the processor had found a hardware error, the loading process would stop and the LEDs would blink the appropriate error code. New units default to using the program stored in the program0 space. Units that have been updated in the field use the program stored in the program1 space. If the code in program1 space is corrupted, the N4865A reverts to the factory installed code in the program0 space.

After the correct program code is transferred to RAM, the LAN and GPIB Interfaces are initialized. If the N4865A has a good LAN connection, it starts the instrument discovery process. When the N4865A has linked to an instrument, it is ready to receive messages from a GPIB Controller. The operating system handles all of the GPIB communication issues and decides what actions are required on the GPIB bus.

The power supply is a switching regulator that converts the unregulated DC input to +5 volts DC to run the N4865A's peripheral logic chips. A 3.3 regulator regulates the 5 Vdc down to 3.3 volts to power the processor and major logic chips. (RAM, FLASH and LAN Interface). Satisfactory DC input range is +9 to +32 Vdc.

This page purposely left blank.

5

Troubleshooting and Repair

5.1 INTRODUCTION

This section describes the maintenance testing, troubleshooting, and repair procedures for ICS's Model N4865A GPIB ⟷ LAN Adapter.

5.2 MAINTENANCE

The N4865A does not require periodic calibration and has no internal adjustments. However, if the N4865A is used in an application where the IEEE 488 bus cables are frequently changed, the N4865A's IEEE 488 Bus Connector may occasionally require cleaning to remove wax and dirt buildup. New bus connectors are shipped with a brightener (thin wax like film) on them. Depending upon usage, enough of the brightener may buildup on the N4865A's bus connector to cause intermittent operation.

The brightener is an organic compound and may be cleaned off by washing the connector with a mild detergent solution followed by an alcohol wash.

5.3 TROUBLESHOOTING

Table 5-1 lists several common fault indications and suggests actions that can be done to either clear the fault or isolate the problem to a faulty piece of equipment. Table 5-2 lists the N4865A's self test error codes and most probable faulty component.

WARNING

If the fault isolation procedure requires internal measurements, always remove power when disassembling or assembling the unit. Use extreme caution during troubleshooting, adjustments, or repair to prevent shorting components and causing further damage to the unit.

TABLE 5-1 TROUBLESHOOTING GUIDE

Symptom	Possible Fault	Action or Check
Unit will not turn on PWR LED off	Power cord not plugged in	Push power cord into DC receptacle
	Power at AC outlet	Check outlet and power adapter
	Wrong DC adapter	Use ICS supplied adapter
	DC plug polarity wrong	N4865A requires + polarity on center pin.
All LEDs on	Internal fault	Check TP4 (3.3V) and TP5 (5 V) for proper voltage. Return unit for repair
Unit shows blinking LEDs	Self test fault	Refer to Self Test errors in Table 5-2.
	ERR LED blinks.	Instrument communication error. The N4865A is unable to communicate with the LAN instrument. Usually means no response from the instrument.
	ACT LED blinks	Normal operation
	RDY LED blinks	The N4865A is looking for a VXI-11.3 compatible LAN instrument. If blinking continues more than 20 seconds, check instrument connections and its network settings. Be sure it is VXI-11.3 compatible.

TABLE 5-1 TROUBLESHOOTING GUIDE (CONT.)

Symptom	Possible Fault	Action or Check
Unit does not respond to client PC	LAN LED off after 30 seconds	Network not detected. No network cable or local hub/switch. Check hub/switch power. Replace Ethernet cable. Wrong cable type. Use a cross-over cable to connect directly to a PC. Use a standard cable to connect to a hub or network switch. No network. Check hub, switch or PC for proper operation
	LAN LED on ACT LED never on	No network messages received by the N4865A. Check PC IP settings. Settings must match N4865A's settings for proper communication. N4865A not set to factory defaults. Hold Rest button in while power cycling the unit. See para 5.5.
	LAN LED on ACT LED blinks	Possible link error. Link to correct N4865A or IP address.
Not linked to a LAN Instrument	RDY LED blinking	Instrument LAN connections. Be sure LAN instrument is connected directly to the N4865A or to it via a local hub Hub or switch error. Power cycle the hub or switch. Instrument IP settings wrong. Check settings. See para 2.4-1
LAN Instrument not in Remote	N4865A in Local	Send the N4865A a Selected Device Clear command with REN on or send the N4865A an Enable Remote command.

TABLE 5-1 TROUBLESHOOTING GUIDE (CONT.)

Symptom	Possible Fault	Action or Check
Message not sent to LAN instrument	Missing terminator	All device messages must be terminated with a LF and EOI asserted on the last byte. Verify the GPIB Controller settings.
No response from N4865A	Instrument not Responding	Verify that the instrument's error indicator is not on.
	Response Terminator	Device message to the GPIB Controller only terminated with EOI asserted on the last character. Check GPIB Controller read terminator settings.
	Missing instrument response data	Instrument did not get a query. or has not response data. See program error above Instrument did not respond. Check GPIB program and instrument manual for correct programming and commands.
	Does not get all data in a large message	Instrument was not asked for additional data. Verify that the N4865A was not addressed as listener and sent a device message, a Device Trigger or a Device Clear after talking out the first data.
	Missing data bytes when reading large files.	Controller using ATN holdoff. when doing partial reads. Change read size to a multiple of 1024 bytes
	Device command error	Serial poll the device or read its Event Status Register to determine reason for no response.
N4865A reverts to factory settings at power turn-on	Configuration settings not saved	Press Save and Reboot after changing the N4865A's configuration

5.4 SELF TEST ERROR CODES

At power turn on, the N4865A conducts a selftest of its major components. The test takes about 5 seconds. During the selftest the PWR LED is on and the RDY LED is off. A successful test ends when the RDY LED on. Test failures are indicated by the blinking LED patterns shown in Table 5-3. If a self test failure occurs, turn the unit off for 10 seconds and turn power back on. If the failure persists, refer to paragraph 5.5 for repair instructions. Note that some of the failures could occur while the N4865A is running.

TABLE 5-2 N4865A SELF TEST ERROR CODES

		Front Panel LEDs						
RDY	PWR	LAN	ACT	TALK	LSTN	SQR	ERR	Fault
⊕	x	x	x	x	x	x	x	fatal error (CPU, FLASH, RAM..etc.)
–	–	–	–	–	–	–	–	fatal error (power supply)
⊕	–	–	–	–	–	–	B	LAN IC, Network Socket Failure or DHCP
⊕	–	–	–	–	–	B	-	GPIB IC or GPIB Transceivers
⊕	–	–	–	–	–	B	B	Configuration Error or Flash Failure
⊕	–	–	–	–	B	–	–	OS Issued Exit
⊕	–	–	–	–	B	–	B	RAM IC or Memory overflow error
⊕	–	–	–	–	B	B	–	OS Error
⊕	–	–	–	–	B	B	B	Flash Error

Symbols: ⊕ = solid on, B = blinking, x = solid on or off

5.5 RESETTING TO DEFAULT NETWORK SETTINGS

The N4865A can be reset to the default network settings listed in Table 1-2 by holding the rear panel LAN Reset Button in for 10 seconds while turning the N4865A's Power Switch on.

1. Connect the supplied AC adapter to the N4865A and to an AC power outlet. Be sure the N4865A Power Switch is turned off.

2. Find a blunt non-metallic stick about 1/16 inch diameter (1.6 mm) that you can use to depress the rear panel LAN Reset button. (A wooden Q-tip handle works well.) You should be able to feel the Reset button move as you gently depress it.

3. Hold the Reset button depressed and turn the Power Switch on. Keep the button depressed until the three front panel yellow LEDs blink.

4. Release the Reset button and turn Power off before connecting any cables to the unit.

5.6 UPDATING THE FIRMWARE

The N4865A's firmware can be updated in the field without returning the unit to the factory.

1. Connect the N4865A to a Windows PC as described in paragraph 2.5.1. Connect the supplied USB power cable to the N4865A and to one of the computer's powered USB ports. Do not connect the N4865A to a port on the keyboard as they only have enough power for a mouse.

2. Locate the latest ICS_N4865A_Update file on ICS Electronics website at http://www.icselect.com. Highlight the Download tab and select 'VXI-11 Support' to get to the correct page. Download the .zip file to an temporary directory and unzip the file.

3. Disconnect the PC from your company network and change its Local Area Network setting to the N4865A's AutoIP range as described in paragraph 2.5.1-3.

4. Run the N4865A Update Utility. If it will not run, download the Visual Basic 6 runtime package from ICS's website or from the Support CD supplied with the N4865A.

5. Turn the N4865A on. It will take about 25 seconds before the RDY LED starts blinking and the N4865A is ready to communicate with the PC.

6. Enter the N4865A's default IP address, 169.254.48.65, in the IP Address window of the Update Program, highlight it and link to it. If you are unable to link to the N4865A reset its LAN settings as described in Section 5.5. Retry the linking process when the RDY LED starts blinking.

7. Press the Prgm Flash button to start the update process. The three yellow LEDs (TALK, LSTN and SRQ) will blink while the new program is being downloaded. Follow the directions on the screen and re-link to the N4865A as required to complete the update process.

CAUTION
Do not exit the Update Program until told to do so. It will ask you to re- link to the unit twice to finish the update.

8. After exiting the Update Program, connect the PC back to the company network and restore the Local Area Network settings to their original values.

5.7 REPAIR

For service or repair information, please contact your local Agilent Sales Office or Support Center. Refer to www.agilent.com for contact information.

Appendix

APPENDIX **PAGE**

A1 IEEE 488 BUS DESCRIPTION (IEEE 488.1, IEEE 488.2, SCPI)

The IEEE Std 488 Bus is a convenient means of connecting instruments and computers together to form a test system or to transfer data between two computers. The IEEE Std 488.1 covers the electrical and mechanical bus specifications and the state diagrams for each bus function. The IEEE Std 488.2 expanded on the original specification and established data formats, common commands for each 488.2 device and controller protocols. The SCPI standard developed a tree like series of standard commands for programmable instruments so that similar instruments by different manufacturers can be controlled by the same program.

A1.1 IEEE 488.1 BUS

The IEEE Std 488 Bus, or GPIB as it is commonly referred to, provides a means of transferring data and commands between devices. The physical portion of the bus is governed by IEEE -Std 488.1 - 1978. The interface functions for each device are contained within that device itself, so only passive cabling is needed to interconnect the devices. The cables connect all instruments, controllers and other components of the system in parallel to the signal line as shown in Figure A-1. Eight of the lines (DIO1-DIO8) are reserved for the transfer of data and other messages in a byte-serial, bit-parallel manner. Data and message transfer is asynchronous, coordinated by the three handshake lines (DAV, NRFD, NDAC). The other five lines control Bus activity.

A1

Two types of messages are transferred over the bus:

Interface messages - for bus management

Device-dependent messages - for device control and data transfer

Devices connected to the bus may act as talkers, listeners, controllers, or combinations of the three functions, depending upon their internal capability. The system controller is a controller that becomes active at power turn-on. It is the Bus manager and the initial controller-in-charge.

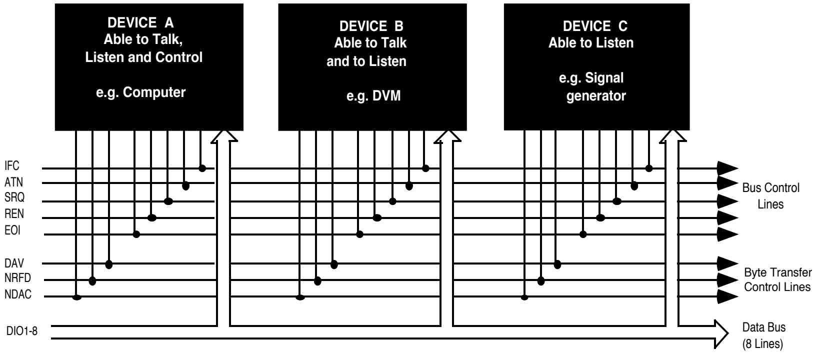


Figure A-1 IEEE 488 Bus

A controller **can send interface messages to manage the other devices, address devices to talk or listen and command specific actions within devices.**

A talker **sends device dependent messages, i.e., data, status.**

A listener **accepts interface messages, bus commands and device-dependent messages, i.e., setup commands, data.**

Bus systems can be as simple as two devices; one a talker always sending data to a second device which listens to the data. Larger systems can have one or more controllers and many devices (the IEEE 488 driver specifications limit the total number of units on one bus system to 15). Only one controller can be the controller-in-charge at any given time. Control originates with the system controller and is passed back to other controller(s) as required. Control can be passes back to the system controller or to another controller after the completion of the task. The system controller has the capability of taking control back at any time and resetting all addressed devices to their unaddressed state.

Each bus device is identified by a five-bit binary address. There are 31 possible primary addresses 0 through 30. Address 31 is reserved as the 'untalk' or 'unlisten' command. Some devices contain sub-functions, or the devices themselves may be addressed by a secondary five-bit binary address immediately following the primary address, i.e. 1703. This secondary address capability expands the bus address range to 961 addresses. The GPIB addresses for most devices are set at the time the system is configured by setting rocker switches on each devices' rear panel. Devices that are

SCPI 1991 compatible, can have their bus address set by a GPIB SYSTEM configuration command.

Information is transmitted on the data lines under sequential control of the three handshake lines. No step in the sequence can be initiated until the previous step is completed. Information transfer proceeds as fast as the devices respond (up to 1 Mbs), but no faster than that allowed by the slowest addressed device. This permits several devices to receive the same message byte at the same time. Although several devices can be addressed to listen simultaneously, only one device at a time can be addresses as a talker. When a talk address is put on the data lines, all other talkers are normally unaddressed.

ATN (attention) is one of the five control lines and is set true by the controller-in-charge while it is sending interface messages or device addresses. The messages are transmitted on the seven least significant data lines and are listed in the MSG columns in Table A-1. When a device is addressed as a talker, it is allowed to send device-dependent messages (e.g., data) when the controller-in-charge sets the ATN line false. The data messages are typically a series of ASCII characters ending in a CR, LF, or CR LF sequence. The data messages often consist of eight-bit binary characters and end on a predetermined count or when the talker asserts the EOI line simultaneously with the last data byte. The controller-in-charge must be programmed to correctly respond to each device's message termination sequence to avoid hanging-up the system or leaving characters that will be output when the device is addressed as a talker again.

IFC (interface clear) is sent by the system controller and places the interface system in a known quiescent state with all devices unaddressed.

REN (remote enable) is sent by the system controller and is used with other interface messages or device addresses to select either local or remote control of each device.

SRQ (service request) is sent by any device on the bus that wants service, such as counter that has just completed a time-interval measurement.

EOI (end or identify) is used by a device to indicate the end of a multiple-byte transfer sequence. When a controller-in-charge sets both the ATN and EOI lines true, each device configured to respond to a parallel poll indicates its current status on the DIO line assigned to it.

Bus Commands are transmitted when ATN is asserted. The commands are listed in the message columns in Table A-1 (on the left hand page) which shows the relationship between the commands and ASCII data characters. ASCII data characters have the same code values as bus commands but are transmitted with ATN off. The following chart lists the standard command and address mnemonics.

Address Commands

MLA	My listen address (controller to self)
MTA	My talk address (controller to self)
LAD	Device listen address
TAD	Device talk address
SAD	Secondary Device address (device optional address)
UNL	Unlisten
UNT	Listen

Universal Commands (to all devices)

LLO	Local Lockout
DCL	Device Clear
PPU	Parallel Poll Unconfigure
SPE	Serial Poll Enable
SPD	Serial Poll Disable

Addressed Commands (to addressed listeners only)

SDC	Selected Device Clear
GTL	Go to Local
GET	Device Trigger
PPC	Parallel Poll Configure
TCT	Take Control

A1

Devices on the bus are normally interconnected by cables with dual male/female connectors at each end to allow easy cable stacking. The 24 conductor cable pinouts are shown in Figure A-2. Signal levels are 0 and 3.3 Vdc with 0 being the logic true level. Cable connectors are modified Amphenol 24 pin Blue ribbon style connectors (57-30240) with metric jack screws.

TABLE A-1 IEEE 488 COMMAND AND ADDRESS MESSAGES

A1

ASCII -- IEEE 488 BUS MESSAGES (COMMANDS AND ADDRESS) HEX CODES

MSD / LSD	0		1		2		3		4		5		6		7	
	ASCII	MSG	ASCII	MSG	ASCII	MSG1	ASCII	MSG1	ASCII	MSG1	ASCII	MSG1	ASCII	MSG	ASCII	MSG
0	NUL		DLE		SP	00	0	16	@	00	P	16	`	▲ MEANING DEFINED BY PCG CODE ▼	p	▲ MEANING DEFINED BY PCG CODE ▼
1	SOH	GTL	DC1	LLO	!	01	1	17	A	01	Q	17	a		q	
2	STX		DC2		"	02	2	18	B	02	R	18	b		r	
3	ETX		DC3		#	03	3	19	C	03	S	19	c		s	
4	EOT	SDC	DC4	DCL	$	04	4	20	D	04	T	20	d		t	
5	ENQ	PPC	NAK	PPU	%	05	5	21	E	05	U	21	e		u	
6	ACK		SYN		&	06	6	22	F	06	V	22	f		v	
7	BEL		ETB		'	07	7	23	G	07	W	23	g		w	
8	BS	GET	CAN	SPE	(	08	8	24	H	08	X	24	h		x	
9	HT	TCT	EM	SPD	)	09	9	25	I	09	Y	25	i		y	
A	LF		SUB		*	10	:	26	J	10	Z	26	j		z	
B	VT		ESC		+	11	;	27	K	11	[	27	k		(	
C	FF		FS		,	12	<	28	L	12	\	28	l		|	
D	CR		GS		-	13	=	29	M	13	]	29	m		)	
E	SO		RS		.	14	>	30	N	14	^	30	n		~	
F	SI		US		/	15	?	UNL	O	15	_	UNT	o		DEL	

ADDRESSED COMMAND GROUP — UNIVERSAL COMMAND GROUP — LISTEN ADDRESS GROUP — TALK ADDRESS GROUP — SECONDARY COMMAND GROUP

PRIMARY COMMAND GROUP (PCG)

Notes:
1. Device Address messages shown in decimal
2. Message codes are:

DCL -- Devices Clear	LLO -- Local Lockout	SDC -- Selected Device Clear
GET -- Device Trigger	PPC -- Parallel Poll Configure	SPD -- Serial Poll Disable
GTL -- Go to Local	PPU -- Parallel Poll Unconfigure	SPE -- Serial Poll Enable

3. ATN off, Bus data is ASCII; ATN on, Bus data is an IEEE MSG.

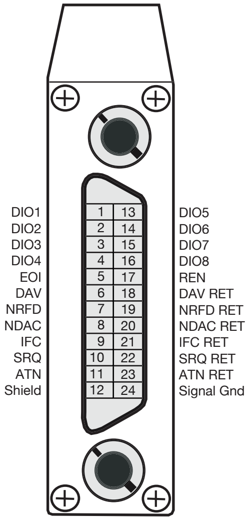


Figure A-2   GPIB Signal-Pin Assignments

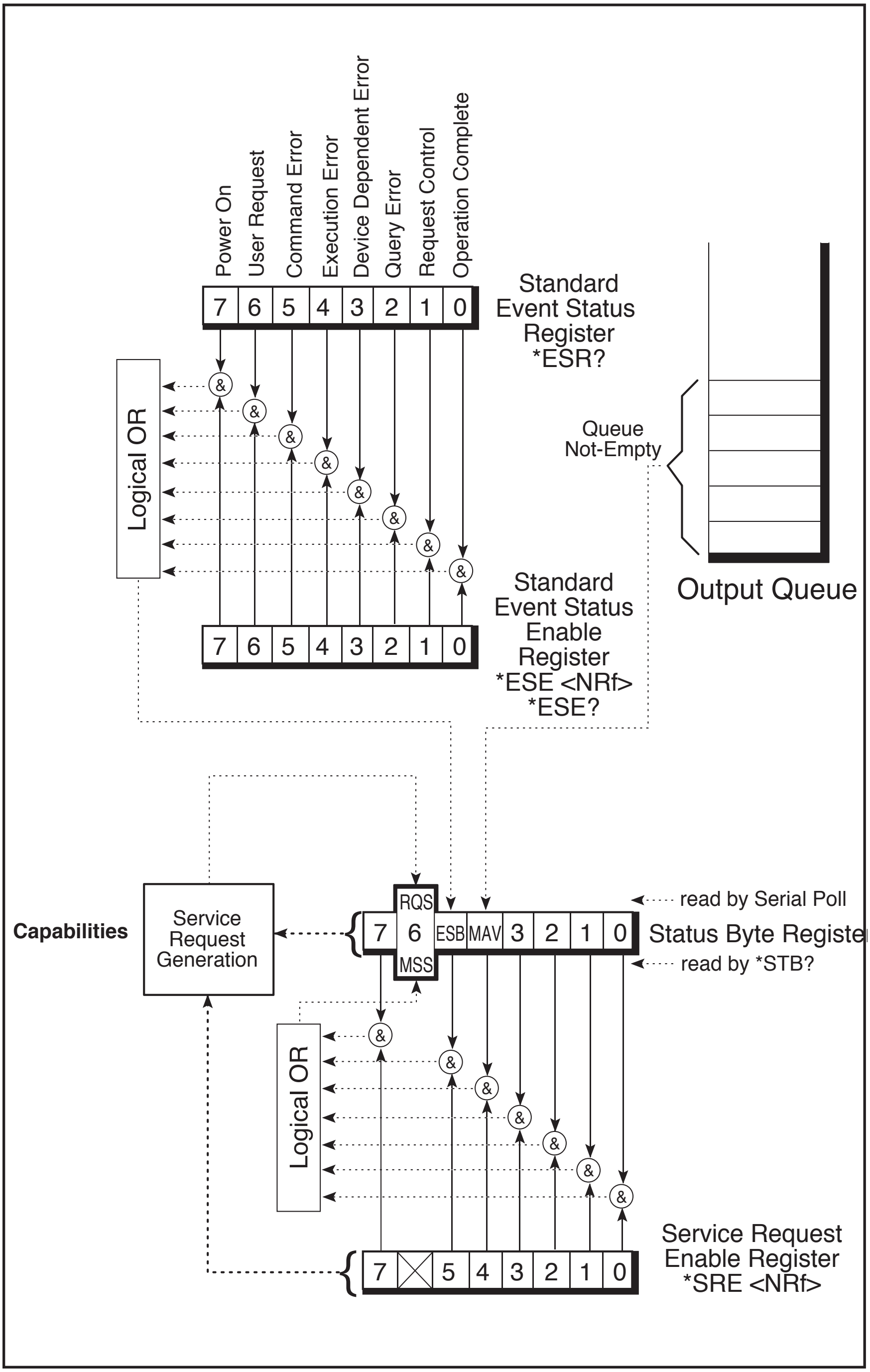


Figure A-3 488.2 Required Status Reporting Capabilities

A1.2 IEEE 488.2 STANDARD

A1.2.1 IEEE 488.2 Message Formats

The IEEE 488.2 Standard was established in 1987 to standardize message protocols, status reporting and define a set of common commands for use on the IEEE 488 bus. IEEE 488.2 devices are supposed to receive messages in a more flexible manner than they send. A message sent from GPIB controller to GPIB device is called: PROGRAM MESSAGE. A message sent from device to controller is called: RESPONSE MESSAGE. As part of the protocol standardization the following rules were generated:

(;) Semicolons are used to separate messages.
(:) Colons are used to separate command words.
(,) Commas are used to separate data fields.
<nl> Line feed and/or EOI on last character terminates a 'program message'. Line feed (ASCII 10) and EOI terminates a RESPONSE MESSAGE.
(*) Asterisk defines a 488.2 common command.
(?) Ends a query where a reply is expected.

A1.2.2 IEEE 488.2 Reporting Structure

With IEEE 488.2, status reporting was enhanced from the simple serial poll response byte in IEEE 488.1 to the multiple register concept shown in Figure A-3. The IEEE 488.2 Standard standardized the bit assignments in the Status Byte Register, added eight more bits of information in the Event Status Register and introduced the concept of summary bits reporting to the Status Byte Register. The Status and Event registers have enabling registers that can control the generation of their summary reporting bits and ultimately SRQ generation. Each 488.2 device must implement a Status Byte Register, a Standard Event Status Register and an Output Message Queue as a minimum status reporting structure. A device may include any number of additional condition registers, event registers and enabling registers providing they follow the model shown in Figure A-3.

A1.2.3 IEEE 488.2 Common Commands

The IEEE 488.2 Standard also mandated a list of required and optional Common Commands that all 488.2 devices could support. All of the Common

Commands start with an asterisk. Commands that end with a question mark are queries. Query responses can be an ASCII number or an ASCII string. Other numerical formats are legal as long as the device supports the required ASCII format. Table A-2 lists the IEEE 488.2 Common Commands.

A1.2.4 IEEE 488.2 Differences From IEEE 488.1

The user who is familiar with the older 488.1 devices should take the following differences into account when programming a 488.2 device.

A 488.2 device outputs the Status Byte Register contents plus the RQS bit in response to a serial poll. The RQS bit is reset by the serial poll. The same 488.2 device outputs the Status Byte Register contents plus the MSS bit in response to a *STB? query. The MSS bit is cleared when the condition is cleared.

488.2 restricts the Device Clear to only clearing the device's buffers and pending operations. It does not clear the Status Reporting Structure or the output lines. Use *CLS to clear the Status Structure and *RST or *RCL to reset the outputs.

488.2 commands are really special data messages and are executed by the device's parser. Always allow sufficient time for the parser to execute the commands before sending the device a 488.1 command. i.e. a Device Clear sent too soon will erase any pending commands and reset the parser.

Enable Register values are only saved and restored if the *PSC command is 0. A *PSC command of 1 causes zeros to be loaded into the enable registers when the unit is next reset or powered on.

TABLE A-2 IEEE 488.2 COMMON COMMANDS

Required common commands are:

***CLS**	Clear Status Command
***ESE**	Standard Event Status Enable Command
***ESE?**	Standard Event Status Enable Query
***ESR?**	Standard Event Status Register Query
***IDN?**	Identification Query
***OPC**	Operation Complete Command
***OPC?**	Operation Complete Query
***RST**	Reset Command
***SRE**	Service Request Enable Command
***SRE?**	Service Request Enable Query
***STB?**	Status Byte Query
***TST?**	Self-Test Query
***WAI**	Wait-to-Continue Command

Devices that support parallel polls must support the following three commands:

***IST?**	Individual Status Query?
***PRE**	Parallel Poll Register Enable Command
***PRE?**	Parallel Poll Register Enable Query

Devices that support Device Trigger must support the following commands:

***TRG**	Trigger Command

Controllers must support the following command:

***PCB**	Pass Control Back Command

Devices that save and restore settings support the following commands:

***RCL**	Recall configuration
***SAV**	Save configuration

Devices that save and restore enable register settings support the following commands:

***PSC**	Saves enable register values and enables/disables recall

A1.3 SCPI COMMANDS

A1.3.1 Introduction

SCPI (Standard Commands for Programmable Instruments) builds on the programming syntax of 488.2 to give the programmer the capability handling a wide variety of instrument functions in a common manner. This gives all instruments a common "look and feel".

SCPI commands use common command words defined in the SCPI specification. Control of any instrument capability that is described in SCPI shall be implemented exactly as specified. Guidelines are included for adding new defined commands in the future as new instruments are introduced without causing programming problems.

SCPI is designed to be laid on top of the hardware - independent portion of the IEEE 488.2 and operates with any language or graphic instrument program generators. The obvious benefits of SCPI for the ATE programmer is in reducing the learning time on how to program multiple SCPI instruments since they all use a common command language and syntax.

A second benefit of SCPI is that its English like structure and words are self documenting, eliminating the needs for comments explaining cryptic instrument commands. A third benefit is the reduction in programming effort to replace one manufacturer's instrument with one from another manufacturer, where both instruments have the same capabilities.

This consistent programming environment is achieved by the use of defined program messages, instrument responses and data formats for all SCPI devices, regardless of the manufacturer.

A1.3.2 Command Structure and Examples

SCPI commands are based on a hierarchical structure that eliminates the need for most multi-word mnemonics. Each key word in the command steps the device parser out along the decision branch - similar to a squirrel hopping from the tree trunk out on the branches to the leaves. Subsequent keywords are considered to be at the same branch level until a new complete command is sent to the device. SCPI commands may be abbreviated as shown by the capital letters in Figure A-4 or the whole key word may be used when entering a command. Figure A-4 shows some single SCPI commands for setting up and querying a serial interface.

```
SYSTem:COMMunicate:SERial:BAUD 9600 <nl>
                                     'Sets the baud rate

SYST:COMM:SER:BAUD? <nl>             'Queries the current
                                     baud setting

SYST:COMM:SER:BITS 8 <nl>            'Sets character format
                                     to 8 data bits
```

Figure A-4 SCPI Command Examples

Multiple SCPI commands may be concatenated together as a compound command using semi colons as command separators. The first command is always referenced to the root node. Subsequent commands are referenced to the same tree level as the previous command. Starting the subsequent command with a colon puts it back at the root node. IEEE 488.2 common commands and queries can be freely mixed with SCPI messages in the same program message without affecting the above rules. Figure A-5 shows some compound command examples.

```
SYST:COMM:SER:BAUD 9600; BAUD? <nl>

SYST:COMM:SER:BAUD 9600; :SYST:COMM:SER:BITS 8
<nl>

SYST:COMM:SER:BAUD 9600; BAUD?; *ESR?; BIT 6; BIT?;
PACE XON; PACE?; *ESR?<nl>
```

Figure A-5 Compound Command Examples

A typical response would be: 9600; 0; 8; XON; 32 <nl>

The response includes five items because the command contains 5 queries. The first item is 9600 which is the baud rate, the second item is ESR=0 which means no errors (so far). The third item is 8 (bit/word) which is the current setting. The BIT 6 command was not accepted because only 7 or 8 are valid for this command. The fourth item XON means that XON is active. The last item is 32 (ESR register bit 5) which means execution error - caused by the BIT 6 command.

A1.3.3 Variables and Channel Lists

SCPI variables are separated by a space from the last keyword in the SCPI command. The variables can be numeric values, boolean values or ASCII strings. Numeric values are typically decimal numbers unless otherwise stated. When setting or querying register values, the decimal variable represents the sum of the binary bit weights for the bits with a logic '1' value. e.g. a decimal value of 23 represents 16 + 4 + 2 + 1 or 0001 0111 in binary. Boolean values can be either 0 or 1 or else OFF or ON. ASCII strings can be any legal ASCII character between 0 and 255 decimal except for 10 which is the Linefeed character.

Channel lists are used as a way of listing multiple values. Channel lists are enclosed in parenthesis and start with the ASCII '@' character. The values are separated with commas. The length of the channel list is determined by the unit. A range of values can be indicated by the starting and stopping values separated by a colon.

```
(@1,2,3,4)                'lists sequential values
(@ 1:4)             'shows a range of sequential values
(@ 1,5,7,34)              'lists random values
```

Figure A-6 Channel List Examples

A1.3.4 Error Reporting

SCPI provides a means of reporting errors by responses to the SYST:ERR? query. If the SCPI error queue is empty, the unit responds with 0, "No error" message. The error queue is cleared at power turn-on, by a *CLS command or by reading all current error messages. The error messages and numbers are defined by the SCPI specification and are the same for all SCPI devices.

A1

A1.3.5 Additional Information

For more information about SCPI refer to the SCPI Standard or to the SCPI section in any SCPI compatible instrument manual.

A3 ICS CONFIGURATION RPC PROTOCOL

The following document describes ICS's Configuration RPC Protocol. This information is supplied to enable a RPC programmer to configure ICS devices that have an Ethernet interface with RPC commands.

A3.1 INTRODUCTION

This document defines the configuration interface to the ICS Ethernet devices (hereafter referred to as the Edevice). The purpose of this document is to allow the communication between the controlling computer and the Edevice, for the purposes of modifying the operational characteristics of the Edevice. Edevices are ICS products whose Model number is in the 8xxx range. Not all commands are supported by all Edevices.

A3.2 SCOPE

This specification addresses the Edevice communication for the purposes of operational configuration.

This specification is to be considered an addendum to the VXI-11 specification for communication to the VXI-11 compliant ICS Edevice Interfaces. The Edevice follows the VXI-11.2 and/or VXI-11.3 specifications.

It is assumed the reader is conversant with ONC/RPC and XDR specifications as published by Sun Microsystems. All client/Edevice communication is performed through ONC/RPC and thus requires knowledge of both (ONC/RPC and XDR) specifications. In addition, it is assumed the reader is conversant with the VXI-11 and VXI-11.2 specifications as published by the VXIbus Consortium. In some cases, the reader will require an understanding of the GPIB (IEEE-488) specification as published in the IEEE Standards.

A3.3 SPECIFICATION OBJECTIVES

This specification has the following objectives:

1. To enable the creation of tools to perform Edevice configuration.
2. To enable applications to perform temporary modifications to the Edevice configuration.
3. To define the ONC/RPC protocol used by the Edevice configuration.

A3.4 REFERENCES

[1] IEEE Std 488.1-1987, IEEE Standard Digital Interface for Programmable Instrumentation.
[2] IEEE Std 488.2-1992, IEEE Standard Codes, Formats, Protocols, and Common Commands For Use With IEEE Std 488.1-1987, IEEE Standard Digital Interface for Programmable Instrumentation.
[3] XDR: External Data Representation Standard, Request for Comments 1014, Sun Microsystems, DDN Network Information Center, SRI International, June, 1987.
[4] RPC: Remote Procedure Call Protocol Specification, Request for Comments, 1057, Sun Microsystems, DDN Network Information Center, SRI International, June, 1988.

B EDEVICE CONFIGURATION PROTOCOL

The Edevice Configuration protocol uses the ONC remote procedure call (RPC) model. This model allows an application executing on one computer to conceptually call a function on another computer.

The client identifies the remote procedure by means of a program ID, program version, and procedure number. This information is encoded into an RPC communication packet with the procedure argument values. The message is then sent to the RPC service running on the server device, where the target procedure is then executed. The server is required to respond to all procedure calls with an RPC reply message containing any/all procedure return values.

Table B.1 lists the RPC messages used by the Edevice configuration protocol. Messages that apply to the 8065 are marked by an 'x' in the 8065 column.

B.1 PROTOCOL BEHAVIOR

The client shall issue an RPC command to the Edevice directing the action to be taken. The Edevice shall attempt to execute the action and will then reply with an RPC reply.

All configuration messages pertaining to values or modes shall contain an Action boolean indicating whether the action is to be a read of the current setting or a modification of the current setting. Edevice default values are listed in Section 1 of this manual.

TABLE B-1 CONFIGURATION RPC MESSAGES

Message	ID	Description	8065	Reboot Req'd
interface_name	1	VXI-11 logical name	X	No
rpc_port_number	2	RPC TCP port		
core_port_number	3	VXI-11 core TCP port		
abort_port_number	4	VXI-11 abort TCP port		
config_port_number	5	configuration TCP port		
comm_timeout	6	TCP timeout	X	No
hostname	7	Edevice TCP hostname		
static_ip_mode	8	static/dynamic IP	X	Yes
ip_number	9	network IP number	X	Yes
netmask	10	network netmask	X	Yes
gateway	11	network gateway	X	Yes
keepalive	12	keepalive time	X	No
gpib_address	13	Edevice GPIB bus address	X	No
system_controller	14	system controller	X	Yes
ren_mode	15	REN active at boot	X	Yes
eos_8_bit_mode	16	EOS 8 bit comparison		
auto_eos_mode	17	automatic EOS on EOI		
eos_active	18	EOS active		
eos_char	19	EOS character		
reload_config	20	force reload of default config	X	No
reload_factory	21	reload factory config settings	X	No
commit_config	22	commit (write) current config	X	No
reboot	23	cause a reboot of the Eth488	X	No
fw_revision	24	firmware revision	X	No
idn_string	25	read IDN type string	X	No
error_logger	26	read current error log contents	X	No

All configuration messages pertaining to actions shall respond with an RPC reply and then (if the status is No Error) execute the action.

All Edevice configuration commands will reply with statuses corresponding to Error Codes as defined by the VXI-11 (section B.5.2).

All Edevice configuration command data will use XDR encoding. Numerical values will be of 4-byte unsigned integer format. String and binary fields will be of opaque array format. Variable length string values will be NULL terminated and will contain a leading length numerical value defining the total length (inclusive of the NULL).

All Edevice configuration commands and replies will result in RPC messages which are multiples of 4-byte lengths. Padding will occur following the last data field and may consist of any byte value.

When the Action boolean signals a read of a mode/value setting, the RPC command must contain a dummy mode/value. While the mode/value in the RPC command is not used, it must exist. If the mode/value is not contained within the RPC command, an error status will result.

The successful modification of a configuration setting will result in the change taking effect immediately, except where noted. Thus, it is strongly advisable to not make configuration changes if VXI-11 device links are currently active. Doing so can cause unpredictable results and Edevice misbehavior. However, such dynamic modifications may be desirable and are possible at the discretion of the user. Messages that require rebooting will not take affect until the Edevice is rebooted.

B.2 Edevice PROGRAM ID AND VERSION

The Edevice configuration procedures shall use an RPC program ID of 1515151515 and an RPC version number of 1.

C.1 interface_name

The interface_name procedure is used to read/modify the current VXI-11 logical interface name.

```
struct Int_Name_Parms {
          unsigned          int       action;
          unsigned          int       length;
          opaque            name<>;
          };
struct Int_Name_Resp {
          unsigned          int       error;
          unsigned  int     length;
          opaque            name<>;
          };

Int_Name_Resp  interface_name (Int_Name_Parms) = 1;
```

The action value determines whether the client wishes to execute a read of the current setting, or a modification of the current value.
action = 0 = read of current value
action = 1 = modify current value

If the action value is other than 0 or 1, then an error value of 5 is returned.

If the RPC message is of insufficient length to satisfy the required length, an error value of 1 is returned.

The name string must be a NULL terminated string with a 32-byte maximum length (exclusive of the NULL). An error of 5 is returned and the Interface Name is unchanged if the name field exceeds 32-bytes.

The returned Int_Name_Resp structure will always contain the current Interface Name, irrespective of the error value.

error	Meaning
0	No error
1	Syntax error
5	Parameter error

C.2 rpc_port_number

The rpc_port_number procedure is used to read/modify the TCP port used by the RPC server.

```
struct Rpc_Port_Parms {
          unsigned         int       action;
          unsigned         int       port;
          };
struct Rpc_Port_Resp {
          unsigned         int       error;
          unsigned  int    port;
          };
```

```
Rpc_Port_Resp rpc_port_number (Rpc_Port_Parms) = 2;
```

The action value determines whether the client wishes to execute a read of the current setting, or a modification of the current value.
action = 0 = read of current value
action = 1 = modify current value

If the action value is other than 0 or 1, then an error value of 5 is returned.

If the RPC message is of insufficient length to satisfy the required length, an error value of 1 is returned.

The port value must be within the range of 0x0001 and 0xFFFF. An error of 5 is returned and the RPC Port value is unchanged if the port value is outside of this range.

The returned Rpc_Port_Resp structure will always contain the current RPC Port value, irrespective of the error value.

error	Meaning
0	No error
1	Syntax error
5	Parameter error

C.3 core_port_number

The core_port_number procedure is used to read/modify the TCP port used by the VXI-11 core channel.

```
struct Core_Port_Parms {
        unsigned        int     action;
        unsigned        int     port;
        };
struct Core_Port_Resp {
        unsigned        int     error;
        unsigned  int   port;
        };
```

Core_Port_Resp core_port_number (Core_Port_Parms) = 3;

The action value determines whether the client wishes to execute a read of the current setting, or a modification of the current value.
action = 0 = read of current value
action = 1 = modify current value

If the action value is other than 0 or 1, then an error value of 5 is returned.

If the RPC message is of insufficient length to satisfy the required length, an error value of 1 is returned.

The port value must be within the range of 0x0001 and 0xFFFF. An error of 5 is returned and the VXI-11 Core Port value is unchanged if the port value is outside of this range.

The returned Core_Port_Resp structure will always contain the current VXI-11 Core Port value, irrespective of the error value.

error	Meaning
0	No error
1	Syntax error
5	Parameter error

C.4 abort_port_number

The abort_port_number procedure is used to read/modify the TCP port used by the VXI-11 abort channel.

```
struct Abort_Port_Parms {
          unsigned        int       action;
          unsigned        int       port;
          };
struct Abort_Port_Resp {
          unsigned        int       error;
          unsigned  int   port;
          };
```

Abort_Port_Resp abort_port_number (Abort_Port_Parms) = 4;

The action value determines whether the client wishes to execute a read of the current setting, or a modification of the current value.
action = 0 = read of current value
action = 1 = modify current value

If the action value is other than 0 or 1, then an error value of 5 is returned.

If the RPC message is of insufficient length to satisfy the required length, an error value of 1 is returned.

The port value must be within the range of 0x0001 and 0xFFFF. An error of 5 is returned and the VXI-11 Abort Port value is unchanged if the port value is outside of this range.

The returned Abort_Port_Resp structure will always contain the current VXI-11 Abort Port value, irrespective of the error value.

error	Meaning
0	No error
1	Syntax error
5	Parameter error

C.5 config_port_number

The config_port_number procedure is used to read/modify the TCP port used by the Edevice configuration channel.

```
struct Config_Port_Parms {
          unsigned       int      action;
          unsigned       int      port;
          };
struct Config_Port_Resp {
          unsigned       int      error;
          unsigned  int  port;
          };
```

```
Config_Port_Resp config_port_number (Config_Port_Parms) = 5;
```

The action value determines whether the client wishes to execute a read of the current setting, or a modification of the current value.
action = 0 = read of current value
action = 1 = modify current value

If the action value is other than 0 or 1, then an error value of 5 is returned.

If the RPC message is of insufficient length to satisfy the required length, an error value of 1 is returned.

The port value must be within the range of 0x0001 and 0xFFFF. An error of 5 is returned and the VXI-11 Abort Port value is unchanged if the port value is outside of this range.

The returned Configt_Port_Resp structure will always contain the current VXI-11 Abort Port value, irrespective of the error value.

error	Meaning
0	No error
1	Syntax error
5	Parameter error

C.6 comm_timeout

The comm_timeout procedure is used to read/modify the TCP timeout value. An inactive TCP channel will be left open this length of time before being closed. A value of zero means no timeout checking.

```
struct Comm_Timeout_Parms {
        unsigned        int       action;
        unsigned        int       timeout;
        };
struct Comm_Timeout_Resp {
        unsigned        int       error;
        unsigned  int   timeout;
        };
```

Comm_Timeout_Resp comm_timeout (Comm_Timeout_Parms) = 6;

The action value determines whether the client wishes to execute a read of the current setting, or a modification of the current value.
action = 0 = read of current value
action = 1 = modify current value
If the action value is other than 0 or 1, then an error value of 5 is returned.

If the RPC message is of insufficient length to satisfy the required length, an error value of 1 is returned.

A2

The timeout value is not range checked, thus it is possible to define an impossible timeout period. A timeout value of zero prevents timeout checking. If a channel remains inactive for the specified timeout period, then the channel is closed in the belief that the TCP connection is broken.

The comm_timeout procedure applies only to the VXI-11 core and Edevice configuration channels. The timeout period is defined as the number of seconds until a timeout is detected. The returned Comm_Timeout_Resp structure will always contain the current communication timeout value, irrespective of the error value.

error	Meaning
0	No error
1	Syntax error
5	Parameter error

C.7 hostname

The hostname procedure is used to read/modify the hostname used by the Edevice. The hostname is only applicable if a dynamic DNS service is available.

```
struct Hostname_Parms {
        unsigned        int     action;
        unsigned        int     length;
        opaque          name<>;
        };
struct Hostname_Resp {
        unsigned        int     error;
        unsigned  int   length;
        opaque          name<>;
        };
```

Hostname_Resp hostname (Hostname_Parms) = 7;

The action value determines whether the client wishes to execute a read of the current setting, or a modification of the current value.
action = 0 = read of current value
action = 1 = modify current value
If the action value is other than 0 or 1, then an error value of 5 is returned.

If the RPC message is of insufficient length to satisfy the required length, an error value of 1 is returned.

The name string must be a NULL terminated string with a 32-byte maximum length (exclusive of the NULL). An error of 5 is returned and the Interface Name is unchanged if the name field exceeds 32-bytes.

The returned Hostname_Resp structure will always contain the current hostname value, irrespective of the error value.

error	Meaning
0	No error
1	Syntax error
5	Parameter error

C.8 static_ip_mode

The static_ip_mode procedure is used to read/modify the static IP mode. If static_ip_mode is set TRUE, then the Edevice will use a static IP and will need a netmask and gateway IP.

```
struct Static_IP_Parms {
            unsigned        int      action;
            unsigned        int      mode;
            };
struct Static_IP_Resp {
            unsigned        int      error;
            unsigned        int      mode;
            };
```

Static_IP_Resp static_ip_mode (Static_IP_Parms) = 8;

The action value determines whether the client wishes to execute a read of the current setting, or a modification of the current value.
action = 0 = read of current value
action = 1 = modify current value

If the action value is other than 0 or 1, then an error value of 5 is returned.

If the RPC message is of insufficient length to satisfy the required length, an error value of 1 is returned.

A2

The mode must be either 0 (dynamic) or 1 (static). An error of 5 is returned and the static IP mode is unchanged if the mode field is any other value.

The returned Static_IP_Resp structure will always contain the current static IP mode, irrespective of the error value.

error	Meaning
0	No error
1	Syntax error
5	Parameter error

C.9 ip_number

The ip_number procedure is used to read/modify the static IP number. If static_ip_mode is set TRUE, then the Edevice will use a static IP (see the static_ip_mode function) and will need a netmask and gateway IP.

```
struct IP_Number_Parms {
        unsigned        int     action;
        unsigned        char    ip[4];
        };
struct IP_Number_Resp {
        unsigned        int     error;
        unsigned        char    ip[4];
        };

IP_Number_Resp ip_number (IP_Number_Parms) = 9;
```

The action value determines whether the client wishes to execute a read of the current setting, or a modification of the current value.
action = 0 = read of current value
action = 1 = modify current value

If the action value is other than 0 or 1, then an error value of 5 is returned.

If the RPC message is of insufficient length to satisfy the required length, an error value of 1 is returned.

The ip must be exactly 4-bytes in length. An error of 5 is returned and the current IP is unchanged if the IP is determined to be invalid.

The returned IP_Number_Resp structure will always contain the current IP, irrespective of the error value.

* Note that the IP will only be used if Static IP is selected.

error	Meaning
0	No error
1	Syntax error
5	Parameter error

C.10 netmask

The netmask procedure is used to read/modify the netmask. If static_ip_mode is set TRUE, then the Edevice will use a static IP (see the static_ip_mode function) and will need a netmask and gateway IP.

```
struct Netmask_Parms {
          unsigned        int       action;
          unsigned        char      netmask[4];
          };
struct Netmask_Resp {
          unsigned        int       error;
          unsigned        char      netmask[4];
          };
```

Netmask_Resp netmask (Netmask_Parms) = 10;

The action value determines whether the client wishes to execute a read of the current setting, or a modification of the current value.
action = 0 = read of current value
action = 1 = modify current value

If the action value is other than 0 or 1, then an error value of 5 is returned.

If the RPC message is of insufficient length to satisfy the required length, an error value of 1 is returned.
The netmask must be exactly 4-bytes in length. An error of 5 is returned and the current netmask is unchanged if the netmask is determined to be invalid.
The returned Netmask_Resp structure will always contain the current netmask, irrespective of the error value.

* Note that the IP will only be used if Static IP is selected.

error	Meaning
0	No error
1	Syntax error
5	Parameter error

C.11 gateway

The gateway procedure is used to read/modify the gateway IP. If static_ip_mode is set TRUE, then the Edevice will use a static IP (see the static_ip_mode function) and will need a netmask and gateway IP.

```
struct Gateway_Parms {
          unsigned          int       action;
          unsigned          char      gateway[4];
          };
struct Gateway_Resp {
          unsigned          int       error;
          unsigned          char      gateway[4];
          };
```

Gateway_Resp gateway (Gateway_Parms) = 11;

The action value determines whether the client wishes to execute a read of the current setting, or a modification of the current value.
action = 0 = read of current value
action = 1 = modify current value
If the action value is other than 0 or 1, then an error value of 5 is returned.

If the RPC message is of insufficient length to satisfy the required length, an error value of 1 is returned.

The gateway must be exactly 4-bytes in length. An error of 5 is returned and the current gateway IP is unchanged if gateway is determined to be invalid. The returned Gateway_Resp structure will always contain the current gateway, irrespective of the error value.

* Note that the IP will only be used if Static IP is selected.

error	Meaning
0	No error
1	Syntax error
5	Parameter error

C.12 keepalive

The keepalive procedure is used to read/modify the keepalive value. If set to zero, then keepalives will not be used. If used, then this is the time (in seconds) of inactivity prior to a keepalive being sent.

```
struct Keepalive_Parms {
          unsigned        int       action;
          unsigned        int       time;
          };
struct Keepalive_Resp {
          unsigned        int       error;
          unsigned        int       time;
          };
```

```
Keepalive_Resp keepalive (Keepalive_Parms) = 12;
```

The action value determines whether the client wishes to execute a read of the current setting, or a modification of the current value.
action = 0 = read of current value
action = 1 = modify current value
If the action value is other than 0 or 1, then an error value of 5 is returned.

If the RPC message is of insufficient length to satisfy the required length, an error value of 1 is returned.

The time value is not range checked, thus it is possible to define an impossible timeout period. A time value of zero prevents keepalive from being used. If a channel remains inactive for the specified time period, then a keepalive is sent (assuming time is non-zero). The returned Keepalive_Resp structure will always contain the current keepalive value, irrespective of the error value.

* Note that the Keepalive time may be fixed and not variable. If non-zero time is specified, Keepalive will be active regardless of time.

error	Meaning
0	No error
1	Syntax error
5	Parameter error

C.13 gpib_address

The gpib_address procedure is used to read/modify the Edevice GPIB bus address.

```
struct Gpib_Addr_Parms {
          unsigned       int       action;
          unsigned       int       address;
          };
struct Gpib_Addr_Resp {
          unsigned       int       error;
          unsigned       int       address;
          };

Gpib_Addr_Resp gpib_address (Gpib_Addr_Parms) = 13;
```

The action value determines whether the client wishes to execute a read of the current setting, or a modification of the current value.
action = 0 = read of current value
action = 1 = modify current value

If the action value is other than 0 or 1, then an error value of 5 is returned.

If the RPC message is of insufficient length to satisfy the required length, an error value of 1 is returned.

The address must be within the range of 0 and 30. An error of 5 is returned and the current GPIB address is unchanged if address is determined to be invalid.

The returned Gpib_Addr_Resp structure will always contain the current Edevice GPIB bus address, irrespective of the error value.

error	Meaning
0	No error
1	Syntax error
5	Parameter error

C.14 system_controller

The system_controller procedure is used to read/modify the system controller mode. If the system controller mode is set TRUE, then the Edevice will initialize at boot time as the GPIB bus controller.

```
struct Sys_Control_Parms {
        unsigned        int     action;
        unsigned        int     controller;
        };
struct Sys_Control_Resp {
        unsigned        int     error;
        unsigned        int     controller;
        };

Sys_Control_Resp system_controller (Sys_Control_Parms) = 14;
```

The action value determines whether the client wishes to execute a read of the current setting, or a modification of the current value.
action = 0 = read of current value
action = 1 = modify current value
If the action value is other than 0 or 1, then an error value of 5 is returned.

If the RPC message is of insufficient length to satisfy the required length, an error value of 1 is returned.

The controller mode must be either 0 or 1. An error of 5 is returned and the current system controller mode is unchanged if controller is determined to be invalid.

The returned Sys_Control_Resp structure will always contain the current system controller mode, irrespective of the error value.

error	Meaning
0	No error
1	Syntax error
5	Parameter error

C.15 ren_mode

The ren_mode procedure is used to read/modify the REN mode. If the REN mode is TRUE, then REN will be asserted at boot time.

```
struct Ren_Parms {
          unsigned       int       action;
          unsigned       int       ren;
          };
struct Ren_Resp {
          unsigned       int       error;
          unsigned       int       ren;
          };

Ren_Resp ren_mode (Ren_Parms) = 15;
```

The action value determines whether the client wishes to execute a read of the current setting, or a modification of the current value.
action = 0 = read of current value
action = 1 = modify current value

If the action value is other than 0 or 1, then an error value of 5 is returned.

If the RPC message is of insufficient length to satisfy the required length, an error value of 1 is returned.

The ren mode must be either 0 or 1. An error of 5 is returned and the current REN mode is unchanged if ren is determined to be invalid.

The returned Ren_Resp structure will always contain the current REN mode, irrespective of the error value.

* Note that this requires System Controller mode.

error	Meaning
0	No error
1	Syntax error
5	Parameter error

C.16 eos_8bit_mode

The eos_8bit_mode procedure is used to read/modify the 8-bit EOS compare mode. If the 8-bit compare mode is TRUE, then EOS compare will be 8-bits. If 8-bit compare mode is FALSE, then EOS compare will be 7-bits.

```
struct Eos_8bit_Parms {
          unsigned       int       action;
          unsigned       int       eos8bit;
          };
struct Eos_8bit_Resp {
          unsigned       int       error;
          unsigned       int       eos8bit;
          };
```

```
Eos_8bit_Resp eos_8bit_mode (Eos_8bit_Parms) = 16;
```

The action value determines whether the client wishes to execute a read of the current setting, or a modification of the current value.
action = 0 = read of current value
action = 1 = modify current value
If the action value is other than 0 or 1, then an error value of 5 is returned.

If the RPC message is of insufficient length to satisfy the required length, an error value of 1 is returned.

The eos8bit mode must be either 0 or 1. An error of 5 is returned and the current 8-bit EOS compare mode is unchanged if eos8bit is determined to be invalid.

The returned Eos_8bit_Resp structure will always contain the current 8-bit EOS compare mode, irrespective of the error value.

error	Meaning
0	No error
1	Syntax error
5	Parameter error

C.17 auto_eos_mode

The auto_eos_mode procedure is used to read/modify the automatic EOS on EOI mode. If the autoEos mode is TRUE, then an EOS character will be sent with EOI.

```
struct Auto_Eos_Parms {
        unsigned        int     action;
        unsigned        int     autoEos;
        };
struct Auto_Eos_Resp {
        unsigned        int     error;
        unsigned        int     autoEos;
        };
```

```
Auto_Eos_Resp auto_eos_mode (Auto_Eos_Parms) = 17;
```

The action value determines whether the client wishes to execute a read of the current setting, or a modification of the current value.
action = 0 = read of current value
action = 1 = modify current value

If the action value is other than 0 or 1, then an error value of 5 is returned.

If the RPC message is of insufficient length to satisfy the required length, an error value of 1 is returned.

The autoEos mode must be either 0 or 1. An error of 5 is returned and the current automatic EOS mode is unchanged if autoEos is determined to be invalid.

The returned Auto_Eos_Resp structure will always contain the current automatic EOS mode, irrespective of the error value.

error	Meaning
0	No error
1	Syntax error
5	Parameter error

C.18 eos_active_mode

The eos_active_mode procedure is used to read/modify the EOS active mode. If the EOS mode is TRUE, then an EOS character will terminate reads.

```
struct Eos_Active_Parms {
          unsigned        int       action;
          unsigned        int       eosActive;
          };
struct Eos_Active_Resp {
          unsigned        int       error;
          unsigned        int       eosActive;
          };
```

EosActive_Resp eos_active_mode (Eos_Active_Parms) = 18;

The action value determines whether the client wishes to execute a read of the current setting, or a modification of the current value.
action = 0 = read of current value
action = 1 = modify current value

If the action value is other than 0 or 1, then an error value of 5 is returned.

If the RPC message is of insufficient length to satisfy the required length, an error value of 1 is returned.

A2

The eosActive mode must be either 0 or 1. An error of 5 is returned and the current automatic EOS mode is unchanged if eosActive is determined to be invalid.

The returned Eos_Active_Resp structure will always contain the current EOS mode, irrespective of the error value.

error	Meaning
0	No error
1	Syntax error
5	Parameter error

C.19 eos_char

The eos_char procedure is used to read/modify the EOS character.

```
struct Eos_Char_Parms {
        unsigned     int     action;
        unsigned     int     eos;
        };
struct Eos_Char_Resp {
        unsigned     int     error;
        unsigned     int     eos;
        };
```

Eos_Char_Resp eos_char (Eos_Char_Parms) = 19;

The action value determines whether the client wishes to execute a read of the current setting, or a modification of the current value.
action = 0 = read of current value
action = 1 = modify current value

If the action value is other than 0 or 1, then an error value of 5 is returned.

If the RPC message is of insufficient length to satisfy the required length, an error value of 1 is returned.

The eos character must be in the range of 0x00 through 0xFF. An error of 5 is returned and the current EOS char is unchanged if eos is determined to be invalid.

The returned Eos_Char_Resp structure will always contain the current automatic EOS character, irrespective of the error value.

error	Meaning
0	No error
1	Syntax error
5	Parameter error

C.20 reload_config

The reload_config procedure is used to cause a reload of the configuration settings. Any modified configuration settings will be restored to default settings.

```
struct Reload_Config_Resp {
        unsigned        int        error;
        };
```

```
Reload_Config_Resp reload_config (void) = 20;
```

If the RPC message is of insufficient length to satisfy the required length, an error value of 1 is returned.

The returned Reload_Config_Resp.error value determines whether the default configuration was reloaded.

error	Meaning
0	No error
1	Syntax error

C.21 reload_factory

The reload_factory procedure is used to cause the Edevice to reset the default configuration back to factory loaded defaults. Any/all modifications to the default configuration are lost as a result. Note that dynamic in-memory configuration settings are not modified until a reload_config or reboot is executed.

```
struct Reload_Factory_Resp {
        unsigned        int        error;
        };

Reload_Factory_Resp reload_factory (void) = 21;
```

If the RPC message is of insufficient length to satisfy the required length, an error value of 1 is returned.

The returned Reload_Factory_Resp.error value determines whether the Edevice has reset the default configurations to the factory default settings.

error	Meaning
0	No error
1	Syntax error

C.22 commit_config

The commit_config procedure is used to cause the current configuration settings to be saved. Any modified configuration settings now become default settings and will be reloaded as the default settings with either reload_config or a reboot.

```
struct Commit_Config_Resp {
          unsigned        int       error;
          };
```

```
Commit_Config_Resp commit_config (void) = 22;
```

If the RPC message is of insufficient length to satisfy the required length, an error value of 1 is returned.

The returned Commit_Config_Resp.error value determines whether the current configuration was saved as the default configuration.

error	Meaning
0	No error
1	Syntax error

C.23 reboot

The reboot procedure is used to cause the Edevice to reboot. This causes all device links to be cleared, all connections closed, all resources released, and the default configuration to be loaded and used during initialization.

```
struct Reboot_Resp {
        unsigned        int       error;
        };

Reboot_Resp reboot (void) = 23;
```

If the RPC message is of insufficient length to satisfy the required length, an error value of 1 is returned.

The returned Reboot_Resp.error value determines whether the Edevice has initiated a reboot process. Note that the timing of the reboot process may block the RPC reply.

* Note that certain configuration settings are only set at boot time. Thus when setting configuration settings, it is recommended that the Reboot command always terminates the configuration setting.

error	Meaning
0	No error
1	Syntax error

C.24 idnReply

The idnReply procedure is used to obtain a response similar to the GPIB *IDN? string. It contains the FW revision, the ICS product model number, and other miscellaneous information.

```
struct idn_Parms {
          };
struct idn_Resp {
          unsigned      int      error;
          unsigned      int      length;
          unsigned      char     reply[128];
          };

idn_Resp  idnReply (idn_Parms) = 25;
```

The length determines the length of the data buffer. The maximum length is 128 bytes.

Error	Meaning
0	No error

C.25 errorLogger

The errorLogger procedure is used to obtain the current contents of the error log.

```
struct error_log_Parms {
          };
struct error_log_Resp {
          unsigned        int       error;
          unsigned        int       count;
          unsigned        int       errors[100];
          };

idn_Resp  errorLogger (error_log_Parms) = 26;
```

The error log will contain 100 entries. The count will signify how many are valid. The remaining values will be of indeterminate values.

Note this function returns all entries and flushes the error log. Do not run this function more than 5 times per second to avoid impacting the 8065's performance and overloading the network.

Refer to the ErrorLogger utility for the error value definitions.

Error	Meaning
0	No error

This page purposely left blank.

Index

This page left intentionally blank