Getting Started with Your GPIB-232CT-A and the NI-488.2™ Software for MS-DOS/Windows

March 1995 Edition

Part Number 320555B-01

National Instruments Corporate Headquarters
6504 Bridge Point Parkway
Austin, TX 78730-5039
(512) 794-0100
Technical support fax: (800) 328-2203
(512) 794-5678

Branch Offices:
Australia (03) 879 9422, Austria (0662) 435986, Belgium 02/757.00.20,
Canada (Ontario) (519) 622-9310, Canada (Québec) (514) 694-8521,
Denmark 45 76 26 00, Finland (90) 527 2321, France (1) 48 14 24 24,
Germany 089/741 31 30, Italy 02/48301892, Japan (03) 3788-1921,
Mexico 95 800 010 0793, Netherlands 03480-33466, Norway 32-84 84 00,
Singapore 2265886, Spain (91) 640 0085, Sweden 08-730 49 70,
Switzerland 056/20 51 51, Taiwan 02 377 1200, U.K. 0635 523545

Limited Warranty

The GPIB-232CT-A is warranted against defects in materials and workmanship for a period of two years from the date of shipment, as evidenced by receipts or other documentation. National Instruments will, at its option, repair or replace equipment that proves to be defective during the warranty period. This warranty includes parts and labor.

The media on which you receive National Instruments software are warranted not to fail to execute programming instructions, due to defects in materials and workmanship, for a period of 90 days from date of shipment, as evidenced by receipts or other documentation. National Instruments will, at its option, repair or replace software media that do not execute programming instructions if National Instruments receives notice of such defects during the warranty period. National Instruments does not warrant that the operation of the software shall be uninterrupted or error free.

A Return Material Authorization (RMA) number must be obtained from the factory and clearly marked on the outside of the package before any equipment will be accepted for warranty work. National Instruments will pay the shipping costs of returning to the owner parts which are covered by warranty.

National Instruments believes that the information in this manual is accurate. The document has been carefully reviewed for technical accuracy. In the event that technical or typographical errors exist, National Instruments reserves the right to make changes to subsequent editions of this document without prior notice to holders of this edition. The reader should consult National Instruments if errors are suspected. In no event shall National Instruments be liable for any damages arising out of or related to this document or the information contained in it.

EXCEPT AS SPECIFIED HEREIN, NATIONAL INSTRUMENTS MAKES NO WARRANTIES, EXPRESS OR IMPLIED, AND SPECIFICALLY DISCLAIMS ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. CUSTOMER'S RIGHT TO RECOVER DAMAGES CAUSED BY FAULT OR NEGLIGENCE ON THE PART OF NATIONAL INSTRUMENTS SHALL BE LIMITED TO THE AMOUNT THERETOFORE PAID BY THE CUSTOMER. NATIONAL INSTRUMENTS WILL NOT BE LIABLE FOR DAMAGES RESULTING FROM LOSS OF DATA, PROFITS, USE OF PRODUCTS, OR INCIDENTAL OR CONSEQUENTIAL DAMAGES, EVEN IF ADVISED OF THE POSSIBILITY THEREOF. This limitation of the liability of National Instruments will apply regardless of the form of action, whether in contract or tort, including negligence. Any action against

National Instruments must be brought within one year after the cause of action accrues. National Instruments shall not be liable for any delay in performance due to causes beyond its reasonable control. The warranty provided herein does not cover damages, defects, malfunctions, or service failures caused by owner's failure to follow the National Instruments installation, operation, or maintenance instructions; owner's modification of the product; owner's abuse, misuse, or negligent acts; and power failure or surges, fire, flood, accident, actions of third parties, or other events outside reasonable control.

Copyright

Trademarks

Warning Regarding Medical and Clinical Use of National Instruments Products

National Instruments products are not designed with components and testing intended to ensure a level of reliability suitable for use in treatment and diagnosis of humans. Applications of National Instruments products involving medical or clinical treatment can create a potential for accidental injury caused by product failure, or by errors on the part of the user or application designer. Any use or application of National Instruments products for or involving medical or clinical treatment must be performed by properly trained and qualified medical personnel, and all traditional medical safeguards, equipment, and procedures that are appropriate in the particular situation to prevent serious injury or death should always continue to be used when National Instruments products are being used. National Instruments products are NOT intended to be a substitute for any form of established process, procedure, or equipment used to monitor or safeguard human health and safety in medical or clinical treatment.

FCC/DOC Radio Frequency Interference Compliance

This equipment generates and uses radio frequency energy and, if not installed and used in strict accordance with the instructions in this manual, may cause interference to radio and television reception. This equipment has been tested and found to comply with the following two regulatory agencies:

Federal Communications Commission

This device complies with Part 15 of the Federal Communications Commission (FCC) Rules for a Class A digital device. Operation is subject to the following two conditions:

1. This device may not cause harmful interference in commercial environments.

2. This device must accept any interference received, including interference that may cause undesired operation.

Canadian Department of Communications

This device complies with the limits for radio noise emissions from digital apparatus set out in the Radio Interference Regulations of the Canadian Department of Communications (DOC).

Le présent appareil numérique n'émet pas de bruits radioélectriques dépassant les limites applicables aux appareils numériques de classe A prescrites dans le règlement sur le brouillage radioélectrique édicté par le ministère des communications du Canada.

Instructions to Users

These regulations are designed to provide reasonable protection against harmful interference from the equipment to radio reception in commercial areas. Operation of this equipment in a residential area is likely to cause harmful interference, in which case the user will be required to correct the interference at his own expense.

There is no guarantee that interference will not occur in a particular installation. However, the chances of interference are much less if the equipment is installed and used according to this instruction manual.

If the equipment does cause interference to radio or television reception, which can be determined by turning the equipment on and off, one or more of the following suggestions may reduce or eliminate the problem.

- Operate the equipment and the receiver on different branches of your AC electrical system.
- Move the equipment away from the receiver with which it is interfering.
- Reorient or relocate the receiver's antenna.
- Be sure that the equipment is plugged into a grounded outlet and that the grounding has not been defeated with a cheater plug.

Notice to user: Changes or modifications not expressly approved by National Instruments could void the user's authority to operate the equipment under the FCC Rules.

If necessary, consult National Instruments or an experienced radio/television technician for additional suggestions. The following booklet prepared by the FCC may also be helpful: *How to Identify and Resolve Radio-TV Interference Problems.* This booklet is available from the U.S. Government Printing Office, Washington, DC 20402, Stock Number 004-000-00345-4.

Contents

Figures

Tables

About This Manual

This manual contains instructions for installing and configuring the National Instruments GPIB-232CT-A RS-232-to-GPIB Controller and NI-488.2 software for MS-DOS/Windows. This manual is meant to be used with the *NI-488.2 Software Reference Manual for MS-DOS.*

Organization of This Manual

This manual is organized as follows:

- Chapter 1, *Introduction*, contains a description of the GPIB-232CT-A, lists what you need to get started and optional equipment you can order, contains instructions for inspecting your GPIB-232CT-A, and gives hardware and software descriptions.

- Chapter 2, *Install Your Hardware*, contains instructions for connecting your GPIB-232CT-A to a PC.

- Chapter 3, *Install the NI-488.2 Software for MS-DOS*, contains a description of the programs and files included with the NI-488.2 software for MS-DOS. This chapter also contains instructions for installing, configuring, and verifying your software as well as programming information on the `IBIC` program and the Applications Monitor program.

- Chapter 4, *Install the NI-488.2 Software for Windows*, contains a list of files that are copied to your destination and Windows directories when you install your software, and instructions for quick installation and interactive installation of the NI-488.2 software for Windows. This chapter also describes two methods you can use to communicate with GPIB devices from Windows: the Windows Interface Bus Interactive Control (`WIBIC`) program and a Windows application program that you develop.

- Chapter 5, *Configure Your Software with IBCONF*, contains a description of the programs `IBCONF.EXE`, a utility you can use to configure your NI-488.2 driver for MS-DOS, and `WIBCONF.EXE`, a utility you can use to configure your NI-488.2 for Windows DLL.

- Appendix A, *Hardware Configuration*, describes how to configure the GPIB-232CT-A RS-232 serial port.

- Appendix B, *Hardware Specifications*, specifies the electrical, environmental, and physical characteristics of the GPIB-232CT-A and the recommended operating conditions.

- Appendix C, *Troubleshooting*, suggests some areas to check if you have problems installing or using the GPIB-232CT-A or the NI-488.2 software after going through the procedures described in Chapters 2 through 5.

- Appendix D, *DLL Direct Entry NI-488 Functions and NI-488.2 Routines*, explains and give examples of how to use the DLL Direct Entry NI-488 functions and NI-488.2 routines to access the `GPIB.DLL` file. Following the examples are tables that list all NI-488.2 routines and NI-488 functions, including their calling syntax and ordinal entry values.

- Appendix E, *Interfacing to a Serial Device*, describes the RS-232 serial port on the GPIB-232CT-A and explains how to interface a DCE or DTE serial device to the RS-232 serial port.

- Appendix F, *Customer Communication*, contains forms you can use to request help from National Instruments or to comment on our products and manuals.

- The *Glossary* contains an alphabetical list and description of terms used in this manual including abbreviations, acronyms, metric prefixes, mnemonics, and symbols.

Conventions Used in This Manual

The following conventions are used in this manual.

bold	Bold text denotes commands, menus, menu items, options, screen button names, LEDs, and checkboxes.
italic	Italic text denotes emphasis, cross references, window names, or an introduction to a key concept.

bold italic	Bold italic text denotes a note, caution, or warning.
`monospace`	Text in this font denotes text or characters that are to be literally input from the keyboard, sections of code, programming examples, and syntax examples. This font is also used for the proper names of disk drives, directories, programs, subprograms, subroutines, device names, functions, variables, field names, and filenames.
`italic monospace`	Italic text in this font denotes that you must supply the appropriate words or values in the place of these items.
<>	Angle brackets enclose the name of a key on the keyboard–for example, <PageDown>.
<Enter>	Key names are capitalized.
-	A hyphen between two or more key names enclosed in angle brackets denotes that you should simultaneously press the named keys–for example, <Control-C>.
IEEE 488 and IEEE 488.2	IEEE 488 and IEEE 488.2 refer to the ANSI/IEEE Standard 488.1-1987 and the ANSI/IEEE Standard 488.2-1987, respectively, which define the GPIB.
RS-232	RS-232 refers to the ANSI/EIA-232-C standard.

Abbreviations, acronyms, metric prefixes, mnemonics, symbols, and terms are listed in the *Glossary*.

Related Documentation

The following documents contain information that you may find helpful as you read this manual:

- ANSI/EIA-232-C, *Interface Between Data Terminal Equipment and Data Circuit-Terminating Equipment Employing Serial Binary Data Interchange*
- ANSI/IEEE Standard 488.1-1987, *IEEE Standard Digital Interface for Programmable Instrumentation*
- ANSI/IEEE Standard 488.2-1987, *IEEE Standard Codes, Formats, Protocols, and Common Commands*
- *Microsoft MS-DOS User's Guide*, Microsoft Corporation
- *Microsoft Windows User's Guide*, Microsoft Corporation

Customer Communication

National Instruments wants to receive your comments on our products and manuals. We are interested in the applications you develop with our products, and we want to help if you have problems with them. To make it easy for you to contact us, this manual contains comment and configuration forms for you to complete. These forms are in Appendix F, *Customer Communication*, at the end of this manual.

Chapter 1
Introduction

This chapter contains a description of the GPIB-232CT-A, lists what you need to get started and optional equipment you can order, contains instructions for inspecting your GPIB-232CT-A, and gives hardware and software descriptions.

GPIB-232CT-A Overview

The GPIB-232CT-A is a high-performance serial-to-GPIB interface. It provides a computer with an RS-232 port, a means of controlling, talking, and listening on the GPIB. The GPIB-232CT-A is also capable of interfacing RS-232 instruments and peripherals to the GPIB.

The GPIB-232CT-A has all the software and logic required to implement the physical and electrical specifications of the IEEE 488 and RS-232 standards. It can interpret and execute high-level commands that you send to it over the serial port, and perform GPIB-to-RS-232 protocol conversion. The GPIB-232CT-A also conforms to all versions of the IEEE 488 standard, including IEEE 488.2. The NAT4882 Controller chip implements all IEEE 488 Talker/Listener/Controller functionality.

Figure 1-1 shows the GPIB-232CT-A.

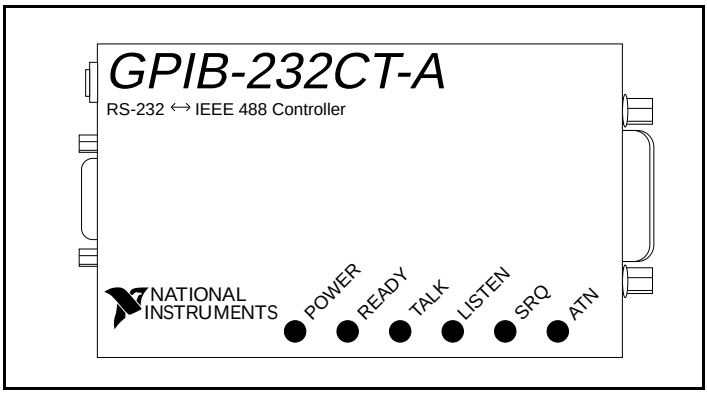

Figure 1-1. GPIB-232CT-A

What You Need to Get Started

❑ **One of the following boxes:**
GPIB-232CT-A (100-120 VAC)
GPIB-232CT-A (220-240 VAC)
GPIB-232CT-A (DC)

❑ **If you have the GPIB-232CT-A AC version, you need one of the following power cords:**
U.S.A. standard power cord
Switzerland power cord
Australian power cord
Universal European power cord
North American power cord
U.K. power cord

❑ **If you have the GPIB-232CT-A DC version, you need one of the following DC power supplies:**
100-120 VAC, 9 V 1 A, wall-mount power supply
220-240 VAC, 9 V 1 A, desktop power supply

❑ **NI-488.2 Distribution Disk for GPIB-232CT-A and MS-DOS/Windows**
3.5 in.
or
5.25 in. (Diskettes 1 and 2)

❑ **MS-DOS (version 3.0 or higher) or equivalent installed on your computer**

❑ **If you want to install the NI-488.2 software for Windows, you must have Windows (version 3.0 or higher) installed on your computer**

Optional Equipment

You can call National Instruments to order the following optional equipment.

Component	Part Number
RS-232 Shielded Cables, Compatible with IBM PC:	
DTE to DTE – 2 m	182238-02
DTE to DTE – 4 m	182238-04
Single-Shielded GPIB Cables:*	
GPIB Type X1 Cable – 1 m	763001-01
GPIB Type X1 Cable – 2 m	763001-02
GPIB Type X1 Cable – 4 m	763001-03
Double-Shielded GPIB Cables:*	
GPIB Type X2 Cable – 1 m	763061-01
GPIB Type X2 Cable – 2 m	763061-02
GPIB Type X2 Cable – 4 m	763061-03
* To meet FCC emission limits for this device you must use a shielded (Type X1 or X2) GPIB cable. Operating this equipment with a non-shielded cable may cause interference to radio and television reception.	

Inspection

Before you operate the GPIB-232CT-A, inspect the shipping container and its contents for damage. Keep the packaging material for possible inspection and/or reshipment.

If the equipment appears to be damaged, do not attempt to operate it. Contact National Instruments for instructions. If the damage appears to have been caused in shipment, file a claim with the carrier.

Hardware Description

The GPIB-232CT-A is a high-performance serial-to-GPIB interface product. It uses the NAT4882 GPIB Controller chip for complete IEEE 488.2 compatibility. You can connect the personal computer serial port to the

RS-232 port for controlling, talking, and listening on the GPIB. The GPIB-232CT-A is also capable of interfacing RS-232 instruments and peripherals to the GPIB.

Software Description

NI-488.2 software is part of your GPIB-232CT-A kit. It is a comprehensive set of programs and drivers for transforming the PC into a GPIB Controller with complete communications and bus management capability. The NI-488.2 software also includes the Microsoft C, Microsoft BASIC, QuickBASIC, and BASICA language interfaces.

National Instruments has developed three other software applications you can use with the GPIB-232CT-A and PC compatibles: LabVIEW for Windows, LabWindows® for DOS, and LabWindows/CVI.

LabVIEW for Windows is a software system with interactive graphics, state-of-the-art user interface concepts, and a powerful graphical programming language. You must order LabVIEW separately.

LabWindows for DOS enhances Microsoft QuickBASIC and C with an interactive development program, function panels to generate source code, and libraries for data acquisition, instrument control, data analysis, and presentation. You must order LabWindows separately.

LabWindows/CVI extends the power of the LabWindows DOS version with a more extensive, full-function professional C programming environment for Windows. You must order LabWindows/CVI separately.

Chapter 2
Install Your Hardware

This chapter contains instructions for connecting your GPIB-232CT-A to a PC.

The default hardware settings for the GPIB-232CT-A are compatible with the default software settings. If you need more information or want to change the defaults, follow the directions in Appendix A, *Hardware Configuration*. If you change the hardware settings, make sure that you also change the software settings so that they are compatible.

Step 1. Verify the Voltage Requirement

The GPIB-232CT-A AC version is shipped from the factory with a 100-120 V or 220-240 V internal power supply. The GPIB-232CT-A DC version is shipped with a 100-120 V or 220-240 V, wall-mount or desktop power supply.

Verify that the voltage marked on the GPIB-232CT-A or on the power supply matches the voltage that is supplied in your area.

Caution: ***Operating the GPIB-232CT-A at any voltage other than the one specified could damage the unit. Replacement fuses for the AC version must be the proper type and size. Refer to Appendix B,*** *Hardware Specifications****, for fuse specifications.***

Step 2. Shut Down Your System

Complete the following steps before connecting the cables:

1. Shut down your system.

2. Turn off your computer and unplug the power cord.

Step 3. Connect the Cables

Complete the following steps to connect the cables.

1. Connect the serial cable to the GPIB-232CT-A 9-pin D-Sub connector and securely fasten the holding screws. Connect the other end of the cable to your serial device. *Be sure* to use only shielded serial cables, and follow all RS-232 cabling restrictions.

2. Connect the GPIB cable to the GPIB-232CT-A 24-pin Champ connector and tighten the thumb screws on the connector. Connect the other end to your GPIB device(s). *Be sure* to follow all IEEE 488 cabling restrictions, and use only shielded GPIB cables.

3. If you have an AC version, connect the power cord to the power receptacle on the front panel of the GPIB-232CT-A, then plug the supply into an AC outlet of the correct voltage.

   If you have a DC version, connect the DC power plug of the DC power supply to the power jack on the serial end of the GPIB-232CT-A, then plug the supply into an AC outlet of the correct voltage.

Step 4. Power On Your System and Switch On the GPIB-232CT-A

1. Plug the power cords for your computer system into a power outlet and power on all devices.

2. If you have an AC version, use the front panel rocker switch to power on your GPIB-232CT-A. If you have a DC version, use the power switch on the rear panel to power on your GPIB-232CT-A.

   The **POWER** LED indicator should come on immediately. The **READY** LED indicator should come on after the GPIB-232CT-A has passed its power-on self test, indicating the unit is ready for operation. If the **READY** LED does not come on within seven seconds after the unit is powered on, recheck all connections and switch settings and retry the power-on sequence. If the **READY** LED still does not come on, refer to Appendix C, *Troubleshooting*, for information on areas to check if you have problems installing the GPIB-232CT-A.

Chapter 3
Install the NI-488.2 Software for MS-DOS

This chapter contains a description of the programs and files included with the NI-488.2 software for MS-DOS. This chapter also contains instructions for installing, configuring, and verifying your software as well as programming information on the `IBIC` program and the Applications Monitor program.

Before installing the software, you should understand the files that will be installed from the distribution disks and their purpose.

Main Programs and Files

You need the following programs and files to use the NI-488.2 software.

- `GPIB.COM` is the NI-488.2 driver that is loaded at system startup by MS-DOS.
- `IBTEST.EXE` is a program that you can use to test the NI-488.2 software.
- `IBCONF.EXE` is a software configuration program. It can be used to change the configuration of the NI-488.2 software.
- `IBIC.EXE` is an interactive control program that executes NI-488.2 functions that you enter from the keyboard. It helps you learn the functions, program instruments or other GPIB devices, and develop your application program.
- `APPMON.EXE` is the applications monitor program. It is a memory-resident program that is useful in debugging your application. The applications monitor can halt program execution (trap) on return from GPIB software calls, so that you can inspect function arguments, buffers, return values, GPIB global status variables, and other pertinent data. The applications monitor performs automatic error detection.

- IBTRAP.EXE is a program that configures the applications monitor.

- ULI.COM is the Universal Language Interface software file you need to use the Universal Language Interface option of the NI-488.2 software package.

Additional Programs and Files

The following programs and files provide programming examples and are required to program the GPIB interface from Microsoft C, Microsoft BASIC, QuickBASIC, or BASICA using the NI-488.2 software.

- MCIB.OBJ is a binary language interface file that gives an application program written in Microsoft C (version 4.0 or higher) access to the NI-488.2 software.

- MBIB.OBJ is a binary language interface file that gives an application program written in Microsoft Professional BASIC (version 7.0 or higher) access to the NI-488.2 software.

- QBIB.OBJ is a binary language interface file that gives an application program written in QuickBASIC (version 4.0 or higher) access to the NI-488.2 software.

- BIB.M is a binary language interface file that gives an application program written in BASICA access to the NI-488.2 software.

- DECL.H is a header file for use with Microsoft C applications.

- MBDECL.BAS is a declaration file that contains code you should place at the beginning of Microsoft BASIC application programs.

- QBDECL.BAS is a declaration file that contains code you should place at the beginning of QuickBASIC application programs.

- DECL.BAS is a declaration file that contains code you should place at the beginning of BASICA application programs.

- DCSAMP.C, DMBSAMP.BAS, DQBSAMP.BAS, DBSAMP.BAS and DIBSAMP are example programs for device calls in C, Professional BASIC, QuickBASIC, BASICA, and IBIC, respectively. BCSAMP.C, BMBSAMP.BAS, BQBSAMP.BAS, BBSAMP.BAS and BIBSAMP are

example programs for board calls in C, Professional BASIC, QuickBASIC, BASICA, and `IBIC`, respectively.

- `CSAMP488.C`, `MSAMP488.BAS`, `QBSAMP488.BAS`, `BSAMP488.BAS`, and `SAMP488` are example programs for NI-488.2 routines in C, BASIC, QuickBASIC, BASICA, and IBIC. For additional examples, refer to Chapter 4, *NI-488.2 Software Characteristics and Routines*, of the *NI-488.2 Software Reference Manual for MS-DOS.*

ReadMe files are included on the distribution disk. `ReadMe.DOC` and `ReadMe.DOS` discuss the NI-488.2 software. The remaining ReadMe files discuss programming considerations for the supported languages.

Terminology

The term *source disk* or *source directory* refers to the NI-488.2 distribution disk. The term *destination directory* refers to the location on your hard disk or disk where the software will be installed (usually `C:\GPIB-CT`). The term *boot drive* refers to the drive that is read by your computer when you power-on or restart your computer.

Step 1. Run INSTALL

The NI-488.2 distribution disk contains a program named `INSTALL.EXE` that installs and tests the NI-488.2 software for you. You can install the NI-488.2 software using one of two methods: quick installation or interactive installation.

Quick Installation

This quick version of the `INSTALL` program assumes that `C:` is your boot drive. `INSTALL` copies files to the GPIB destination directory, named `C:\GPIB-CT` and makes a change to your `C:\CONFIG.SYS` file.

Note: ***If your boot drive is not*** `C:`***, or you do not want the default destination directory to be created, you must install the NI-488.2 software interactively. Refer to the following section,*** *Interactive Installation****, for more information.***

After starting your computer, run the `INSTALL` program located on the distribution disk by entering the following command:

```
X:install /q
```

where *X* is the name of the drive containing the distribution disk (usually `A` or `B`).

The quick version of `INSTALL` copies the NI-488.2 files and then automatically modifies the `C:\CONFIG.SYS` file. If no error message appears, the NI-488.2 software is successfully installed and you can proceed to *Step 2. Configure the Software*. If an error occurs during the quick installation, you may need to run the `INSTALL` program interactively, as described in the next section. For more information on error codes, refer to Chapter 3, *Understanding the NI-488.2 Software*, in the *NI-488.2 Software Reference Manual for MS-DOS*.

Interactive Installation

If you choose to interactively install the NI-488.2 software, complete the following instructions.

After starting your computer, run the `INSTALL` program on the distribution disk by entering the following command:

```
X:install
```

where *X* is the name of the drive containing the distribution disk (this is usually `A`).

This is the interactive version of the `INSTALL` program. When the program prompts you for the type of software to install (**DOS** or **Windows**), select **DOS**. After you select the **DOS** option, `INSTALL` displays a main menu with three options: **Partial GPIB Installation**, **Full GPIB Installation**, and **Return to DOS**.

Select the type of installation that you want to use (**Partial** or **Full**). If you select **Partial GPIB Installation**, you are prompted to choose which parts of the NI-488.2 software to install. If you select **Full GPIB Installation**, all of the NI-488.2 software is installed.

Next, you are prompted to give your boot drive and the name of a new directory into which the files can be copied. The `INSTALL` program creates

the specified destination directory and copies the files listed in the previous section to their appropriate directories.

When the installation is complete, `INSTALL` asks to modify your `CONFIG.SYS` file. If you enter `yes`, `INSTALL` adds the following line to your `CONFIG.SYS` file:

`device=`*`dir`*`\gpib.com`

where *`dir`* is the directory to which `INSTALL` copied the NI-488.2 software files; for example, *`dir`* may be `C:\GPIB-CT`. If you have a previous version of the NI-488.2 software installed on your computer, `INSTALL` replaces the information in the old version file `CONFIG.SYS` with the new information.

If you enter `no` to the above prompt, `INSTALL` displays a message informing you of the correct line that you should add to your `CONFIG.SYS` file.

Step 2. Configure the Software

Before you can run the software diagnostics tests, the NI-488.2 software must be loaded. If you have just completed the installation procedure and have not restarted your computer, the software is not yet loaded. Exit `INSTALL` by pressing <Escape>. Before restarting your computer, you can run the software configuration program `IBCONF`.

You must run `IBCONF` if you have made any changes to the GPIB-232CT-A hardware switches or you need to change any of the default software configuration options. If you have not changed the hardware in any way and the default software configuration is acceptable, there is no need to run `IBCONF`. You can, however, run `IBCONF` just to examine the software configuration.

Refer to Chapter 5, *Configure Your Software with IBCONF*, for information on the configurable software options and their default values and how to run `IBCONF`.

Caution: ***In DOS, devices must NOT have the same names as files, directories, or subdirectories. The MS-DOS GPIB device driver uses the names*** `GPIB0`, `GPIB1`, `GPIB2`, `GPIB3` ***and*** `DEV1`, `DEV2`, `DEV3` ***through*** `DEV32`. ***If you have files,***

directories, or subdirectories with one these names, you must rename them.

Step 3. Verify the Software Installation

To load the NI-488.2 software into the memory of the computer, you must restart the computer. You will typically only have to do this once when you first install the NI-488.2 software and whenever you need to reconfigure the hardware settings.

Restart your computer by pressing <Ctrl-Alt-Del>. This restarts the computer and loads the NI-488.2 software into memory.

After the NI-488.2 software is installed, run `IBTEST`. Running `IBTEST` ensures that the NI-488.2 software is installed properly on your system.

Note: ***Before running*** `IBTEST`***, make sure that the GPIB-232CT-A is not connected to any GPIB devices.***

If an error occurs, check the following:

- Is the GPIB-232CT-A connected to a GPIB device? `IBTEST` requires that the GPIB-232CT-A *not* be connected to any GPIB devices.
- Is the GPIB-232CT-A box powered on?
- Did you change any of the hardware configurations on the GPIB-232CT-A box? If so, check the current software configuration of the software by running `IBCONF`. Make sure that the hardware settings match. For more information, refer to Chapter 5, *Configure Your Software with IBCONF*.
- Has the `CONFIG.SYS` file on your startup disk been correctly modified to contain the following line?

  `device=`*`dir`*`\gpib.com`

  where *`dir`* is the directory to which `INSTALL` copied the NI-488.2 software files; for example, `C:\GPIB-CT`.
- Did you restart your computer after installing and configuring the NI-488.2 software? If not, restart your computer and run `IBTEST` again.

If you have performed these steps and there is still a problem, refer to Appendix C, *Troubleshooting,* for more information on installing or using the NI-488.2 software.

If no errors occurred, you can proceed to learn how to use the software and how to develop your application program.

Step 4. Developing Your Application Program

There are two tools that are useful for developing your application program: `IBIC.EXE` and `APPMON.EXE`.

Interactive Control Program (IBIC)

The easiest way to learn to communicate with your instrument is by controlling it interactively. Located in your GPIB directory is the Interface Bus Interactive Control program called `IBIC.EXE`. You can use this program to communicate with your instrument; the program displays the status and any error after each NI-488.2 call.

With `IBIC`, you can program your instruments interactively from the keyboard rather than from an application program. Using `IBIC` helps you quickly understand how the instruments and the NI-488.2 software work. `IBIC` is fully described in Chapter 6, *IBIC*, of the *NI-488.2 Software Reference Manual for MS-DOS*.

While running `IBIC`, you should study the descriptions of each function given to fully understand their purpose or you can use the online help available if you have questions.

To use `IBIC.EXE`, connect your instrument to the bus and enter the following commands:

```
cd \dir
```

where *`dir`* is the name of the directory to which `INSTALL` copied the NI-488.2 software. By default, `dir` is `c:\GPIB-CT`.

```
IBIC
```

You are now ready to begin developing applications. Refer to Chapter 6, *IBIC*, of the *NI-488.2 Software Reference Manual for MS-DOS* to get started.

Applications Monitor

The program `APPMON.EXE` is the Applications Monitor. It is a DOS memory-resident program that is used to monitor and record GPIB calls that are made to the MS-DOS driver. It can also be used to halt the execution of an application when a particular bit, for example, ERR, is set in `ibsta`. The Applications Monitor stores up to 255 previous GPIB calls and is invaluable for debugging your application. The Applications Monitor is fully described in Chapter 7, *Applications Monitor*, of the *NI-488.2 Software Reference Manual for MS-DOS.*

The Application Program

When you decide to write your application program, be sure to refer to the appropriate language reference manuals and the *NI-488.2 Software Reference Manual for MS-DOS* for the proper syntax of the functions. Use `APPMON.EXE` and `IBIC.EXE` to help you develop your application.

Chapter 4
Install the NI-488.2 Software for Windows

This chapter contains a list of files that are copied to your destination and Windows directories when you install your software, and instructions for quick installation and interactive installation of the NI-488.2 software for Windows. This chapter also describes two methods you can use to communicate with GPIB devices from Windows: the Windows Interface Bus Interactive Control (`WIBIC`) program and a Windows application program that you develop.

Before installing the software, you should understand the files that will be copied from the distribution disk(s) and the purpose of each file. The following section describes the files contained on the distribution disk(s).

NI-488.2 Files for the Windows Operating Environment

The following files are required to run a Windows application using NI-488.2 routines. The `INSTALL` program copies these files to the specified Windows directory.

- `GPIB.DLL` is a dynamic link library (DLL) that is accessed by a NI-488.2 application for Windows as the application executes. The DLL contains all of the NI-488 functions and NI-488.2 routines.

- `GPIB.INI` is the private profile file which is used by `GPIB.DLL` to determine the software configuration parameters for each GPIB board and device in the system. You can modify `GPIB.INI` by using either the `WIBCONF.EXE` file or a text editor.

NI-488.2 Files for the Development of Windows Applications

The following files are required to test and begin programming with your NI-488.2 software. The `INSTALL` program copies the following files to the specified destination directory.

- `WIBCONF.EXE`, a DOS application, is a software configuration program that you can use to change the software parameters and other data used by the DLL. It has the same basic functionality as the NI-488.2 for MS-DOS `IBCONF` program, which is described in Chapter 5, *Configure Your Software with IBCONF*.

- `WIBCONF.PIF`, a Windows application, contains program information about the `WIBCONF.EXE` program that Windows uses when it runs `WIBCONF.EXE`.

- `WIBTEST.EXE`, a Windows application, is a program that tests the software installation. It verifies that the software configuration is consistent with the GPIB hardware configuration settings.

- `WIBIC.EXE`, a Windows application, is the Windows Interface Bus Interactive Control program that executes NI-488 functions and NI-488.2 routines that you enter from the keyboard. It can help you learn how to use the NI-488 functions and NI-488.2 routines, program instruments or other GPIB devices, and develop your particular Windows application program.

GPIB Sample Windows Application Program

The following files are required to make the sample Windows application. The `INSTALL` program also copies the following files into a new subdirectory, named `C` in the specified destination directory.

- `GPIB.LIB` is the import library for the DLL. You must link it to your NI-488.2 application for Windows just like any other library.

- `WINDECL.H` is an include file that contains prototypes of the NI-488 functions and NI-488.2 routines, and useful constants that you may want to use in your NI-488.2 application for Windows. You must include it at the beginning of any file that makes NI-488 function calls.

- `WINSAMP.EXE` is a compiled Windows application program for Windows that communicates over the GPIB. It is based primarily on the `GENERIC` Windows application example provided with the Windows Software Development Kit.
- `MAKEFILE` is the makefile used to compile and link the sample Windows application.
- `WINSAMP.C` is the C language source file containing the Windows functions `WinMain`, `MainWndProc`, `About`, `InitApplication`, and `InitInstance`.
- `GPIBSAMP.C` is the C language source file containing NI-488 function calls to the DLL.
- `WINSAMP.H` is the header file containing definitions and declarations required by `WINSAMP.C`.
- `WINSAMP.RC` is the resource script file that defines the menus and the dialog-box template for the `About` dialog box.
- `WINSAMP.DEF` is the module definition file that contains module definitions.

Step 1. Run INSTALL

You can install the NI-488.2 software for Windows using one of two methods: quick installation or interactive installation.

Quick Installation

This quick version of the `INSTALL` program assumes that Windows is installed in the default directory (`C:\WINDOWS`). `INSTALL` copies files to `C:\WINDOWS` and the GPIB destination directory, named `C:\GPIB-CTW`.

Note: ***If Windows is not in its default directory or you do not want the default destination directory to be created, you must install the NI-488.2 software for Windows interactively. Refer to the following section,*** *Interactive Installation*.

After starting your computer, run the INSTALL program located on the distribution disk by entering the following command:

```
X:install /qw
```

where *X* is the name of the drive containing the distribution disk (this is usually A).

The quick version of INSTALL copies the NI-488.2 files. If no error message appears, the NI-488.2 software is successfully installed and you can proceed to *Step 2. Set Up the Windows Applications*. If an error occurs during the quick installation, you may need to run the INSTALL program interactively, as described in the next section, *Interactive Installation*.

Interactive Installation

If you choose to interactively install the NI-488.2 software, complete the following instructions.

After starting your computer, run the INSTALL program on the distribution disk by entering the following command:

```
X:install
```

where *X* is the name of the drive containing the distribution disk (this is usually A).

This is the interactive version of the INSTALL program. When the program prompts you for the type of software to install (**DOS** or **Windows**), select **Windows**. After you select the **Windows** option, INSTALL displays a main menu with three options: **Partial GPIB Installation**, **Full GPIB Installation**, and **Return to DOS**.

Select the type of installation that you want to use (**Partial** or **Full**). If you select **Partial GPIB Installation**, you are prompted to choose which parts of the NI-488.2 software to install. If you select **Full GPIB Installation**, all of the NI-488.2 software is installed.

Next, you are prompted to give the location of your Windows directory and you are prompted for the name of the directory where Windows is stored and the name of a new directory into which the files can be copied. The INSTALL program creates the specified destination directory and copies the NI-488.2 files to the appropriate directories.

Step 2. Set Up the Windows Applications

To set up the NI-488.2 applications for Windows, complete the following steps:

1. Run **Windows Setup** in the **Main** window.
2. Select **Set Up Applications** from the **Options** pull-down menu.
3. Add `WIBIC` and `WIBTEST` to the **Windows Applications** window.

Refer to the *Microsoft Windows User's Guide* for a more detailed description of the Windows Setup procedure.

Step 3. Configure the Software

If you have made any changes to the GPIB-232CT-A hardware switches or you need to change any of the default software configuration options, you must run the program `WIBCONF`. If the default configuration of the hardware and software is acceptable, skip to Step 4. You can, however, run `WIBCONF` just to examine the software configuration.

Refer to Chapter 5, *Configure Your Software with IBCONF*, for information on the configurable software options and their default values and how to run `WIBCONF`.

Step 4. Verify the Software Installation

The `WIBTEST` program verifies that the software is properly installed and configured for your GPIB-232CT-A. `WIBTEST` is a Windows application and can be run by selecting the `WIBTEST` icon in the **Windows Applications** window.

`WIBTEST` requires no user interaction and takes about 10 seconds to complete. Disconnect any GPIB cables from the GPIB-232CT-A before running the program. If an error occurs, check the following:

- Is the GPIB-232CT-A connected to a GPIB device? `WIBTEST` requires that the GPIB-232CT-A *not* be connected to any GPIB devices.

- Is the GPIB-232CT-A powered on?
- Did you change any of the hardware configurations on the GPIB-232CT-A? If so, check the current software configuration of the software by running `WIBCONF`. Make sure that the hardware settings match. Refer to *Step 3. Configure the Software*, earlier in this chapter.
- Are the files `GPIB.DLL` and `GPIB.INI` located in your Windows directory (usually `C:\WINDOWS`)?

If you have performed these steps and there is still a problem, refer to Appendix C, *Troubleshooting,* for more information on installing the GPIB-232CT-A.

If no errors occur, you can proceed to learn how to use the software and how to develop your application program.

Using WIBIC

The easiest way to learn to communicate with your instrument is by controlling it interactively. Located in your GPIB directory is the Interface Bus Interactive Control program called `WIBIC.EXE`. You can use this program to communicate with your instrument; the program displays the status and any error after each NI-488.2 call.

With `WIBIC`, you can program your instruments interactively from the keyboard rather than from an application program. Using `WIBIC` helps you quickly understand how the instruments and the NI-488.2 software work. The `WIBIC` program is the Windows version of the NI-488.2 software for MS-DOS `IBIC` program. It has the same general appearance and same function as the DOS `IBIC` program. Refer to Chapter 6, *IBIC*, of your *NI-488.2 Software Reference Manual for MS-DOS* for a complete description of how to use `IBIC`.

While running `WIBIC`, you should study the descriptions of each function given to fully understand their purpose or you can use the online help available if you have questions.

To run `WIBIC`, change to the **Windows Applications** window and select the `WIBIC` icon.

Writing Windows Programs That Use the GPIB

There are two methods for writing a Windows application that uses the GPIB. The first method is to write an application that uses the standard NI-488 functions and NI-488.2 routines and is linked to one of the Windows language interfaces. The NI-488.2 software includes the Microsoft C language interface. Contact National Instruments for information on other language interfaces that you can use with Windows.

The second method of writing a NI-488.2 for Windows application is to use the DLL direct entry NI-488 functions and NI-488.2 routines. Using direct entry, you do not need to have a special language interface to link with your application. Refer to Appendix D, *DLL Direct Entry NI-488 Functions and NI-488.2 Routines*, for more information.

The remainder of this chapter describes the `WINSAMP` sample program that illustrates how GPIB calls can be made from a simple Microsoft C Windows application using the National Instruments Microsoft C language interface. It also lists a general set of rules to follow when using the DLL in your own Windows application.

The WINSAMP Sample

There are two primary parts to the `WINSAMP` sample: `WINSAMP.C` and `GPIBSAMP.C`. `WINSAMP.C` handles most of the details for interfacing with Windows and `GPIBSAMP.C` makes GPIB calls and then displays the results on the screen.

To execute `WINSAMP`, set it up as a Windows application. Refer to *Step 2. Set Up the Windows Applications*, change to the **Windows Applications** window, and select the `WINSAMP` icon.

To make changes to `WINSAMP`, add the desired changes and rebuild it by entering the following command.

```
make
```

General Rules for Using GPIB.LIB with Windows

By following these general rules, any application can use the `GPIB.DLL`.

- Make the same GPIB calls that you do under DOS (refer to the *NI-488.2 Software Reference Manual for MS-DOS* for a list of these calls).
- Add `GPIB.LIB` to the library list in the link command line.

  Note: ***All NI-488.2*** `GPIB.DLL` ***files for Windows share the same*** `.LIB` ***file; therefore, you do not have to relink applications to switch between GPIB boxes.***

- Ensure that the correct `GPIB.DLL` is in the directory in which Windows is installed or in the DOS search path when the application is run. Unlike the `GPIB.LIB` file, `GPIB.DLL` files are unique for each National Instruments GPIB box or interface board.
- Ensure that `GPIB.INI` is in the directory in which Windows is installed when the application is run so that it can be used to properly initialize the `GPIB.DLL` file. The `GPIB.INI` file is also unique for each GPIB box or interface board.

Chapter 5
Configure Your Software with IBCONF

This chapter contains a description of the programs IBCONF.EXE, a utility you can use to configure your NI-488.2 driver for MS-DOS, and WIBCONF.EXE, a utility you can use to configure your NI-488.2 driver for Windows DLL.

In this chapter, the term *IBCONF* is used to refer to both IBCONF.EXE and WIBCONF.EXE . When you complete your software configuration using IBCONF, you are ready to begin developing your application program. If you need more information on IBCONF, refer to the online help screens in the configuration utility.

Note: ***Throughout this chapter and*** IBCONF***, the terms*** *interface board*, *access board*, ***and*** *board* ***are used to refer the GPIB-232CT-A.***

Overview of IBCONF

IBCONF is a screen-oriented, interactive program that is used to modify the configuration parameters for your GPIB-232CT-A and the GPIB devices connected to it.

When used interactively, IBCONF reads in the GPIB configuration parameters and displays them for your inspection. You can alter any of the parameters to suit your requirements. When you have finished modifying the configurable parameters, these changes can be saved when you exit the IBCONF program.

In DOS, changes to the device driver using IBCONF.EXE take effect in the memory-resident device driver in two ways:

- The traditional method is to restart your computer so that DOS can reload the modified device driver into memory.
- A second, easier method, is to let IBCONF.EXE modify the memory-resident device driver when you exit IBCONF.EXE. This second method works only if the two copies of the device driver (the one stored on disk and the one loaded into memory by DOS) are compatible.

For DOS, the simplest way to use IBCONF.EXE is to change to the directory that contains the installed GPIB distribution files and enter the following command:

```
IBCONF
```

IBCONF.EXE finds a GPIB.COM file to configure by going through the following process:

1. If the file C:\CONFIG.SYS exists and contains a line of the format device=<path>gpib.com, that GPIB.COM file is configured.

2. If the file CONFIG.SYS exists on the root directory of the current drive and contains a line of the format device=<path>gpib.com, that GPIB.COM file is configured.

3. If a GPIB.COM file exists in the current directory, that file is configured.

In Windows, changes made using WIBCONF.EXE are recorded in the GPIB.INI file. The changes are effective immediately.

For Windows, you can run WIBCONF.EXE either from DOS or Windows. WIBCONF.EXE finds the GPIB.INI file that contains the configuration information by going through the following process:

1. Check C:\WINDOWS for a GPIB.INI file.

2. Check \WINDOWS for a GPIB.INI file.

3. Check the current directory for a GPIB.INI file.

Table 5-1. IBCONF Options

IBCONF Option	Action
driver	IBCONF configures the given driver file instead of following one of the search rules listed above.
-m	Monochrome mode. This option causes IBCONF to run in monochrome mode even though you have a color monitor.

Upper and Lower Levels of IBCONF

IBCONF operates at both an upper and a lower level. The upper level consists of the board device maps and gives a graphical picture of the GPIB system as defined in the handler. The lower level consists of screens describing the individual board and devices that make up the system.

Upper Level Device Map

Figure 5-1 shows the upper level of IBCONF.

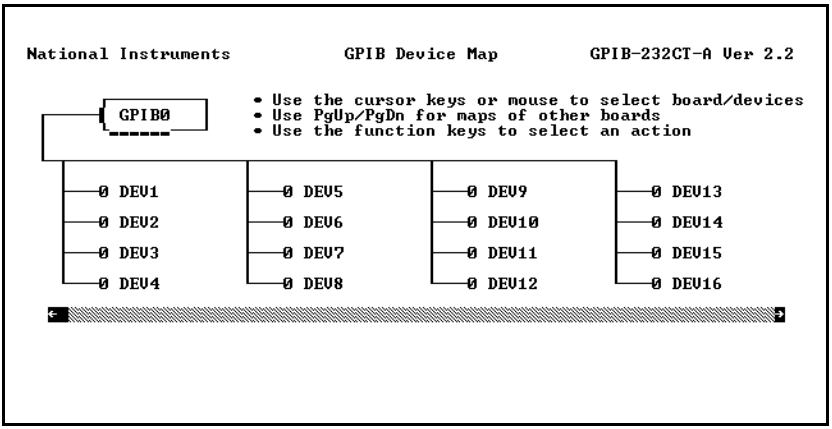

Figure 5-1. Upper Level of IBCONF

As shown in Figure 5-1, the upper-level screen of IBCONF displays the names of all devices controlled by the driver. It also indicates which devices, if any, are accessed through the interface or access board named GPIB*x*, where *x* is 0 for your first board, 1 for your second board, and so on. You can move around the map by using the cursor control keys. If you have a Microsoft compatible mouse, you can also use it to move around the map.

The following options are available at the upper level.

- Device Maps of the Boards
- Help
- Rename
- (Dis)connect
- Edit
- Output GPIB Driver Configuration (`IBCONF.EXE` only)
- Autoconfigure (`IBCONF.EXE` only)
- Exit

Device Maps of the Boards

Use <PageUp> or <PageDown> to toggle between the device maps for the different GPIB boards. Each board is referred to as an access board by `IBCONF`. The maps show which devices are assigned to each box. The default settings attach 16 devices to `GPIB0` and 16 devices to `GPIB1`.

Help

Use <F1> to access the comprehensive, online help feature of `IBCONF`. The help information describes the functions and common terms associated with the upper-level of `IBCONF`.

Rename

Use <F4> to rename a device. Move to the device you want to rename by using the cursor control keys. Press <F4> and enter the new name of the device. The device name may contain up to eight characters and uses the same rules as MS-DOS for naming files, except that suffixes (`.xxx`) are *not* allowed.

As specified by MS-DOS, the device name cannot use the following characters:

```
.       "       /       \       [       ]       :
|       <       >       +       =       ;       ,
```

(ASCII characters less than hex 21)

and cannot use the following reserved names:

CON NUL

Uppercase and lowercase letters are treated the same. The string `PLOTTER` is treated the same as the string `plotter`, for example. For this reason, `IBCONF` maps all lowercase letters to uppercase.

Caution: ***In DOS, devices must not have the same names as files, directories, or subdirectories. The MS-DOS GPIB device driver uses the names*** `GPIB0`, `GPIB1`, `GPIB2`, `GPIB3` ***and*** `DEV1`, `DEV2`, `DEV3` ***through*** `DEV32`. ***If you have files, directories, or subdirectories with one these names, you must rename them.***

The string representing a device or access board name is the first variable argument of the function `ibfind` called at the beginning of your application program. Refer to Chapter 4, *NI-488.2 Software Characteristics and Routines*, and Chapter 5, *NI-488 Software Characteristics and Functions*, of the *NI-488.2 Software Reference Manual for MS-DOS* for more explanations of `ibfind`.

(Dis)connect

Use <F5> to logically connect or disconnect a device from a board. Move the cursor to the device that you want to connect or disconnect by using the cursor control keys and press the <F5> key.

Edit

Use <F8> or <Enter> to edit or examine the characteristics of a particular GPIB board or device. Move to the GPIB board or device that you want to edit using the cursor control keys and press <F8>. This step puts you in the lower level of `IBCONF` and lists the characteristics for the particular GPIB

board or device that you want to edit. To exit edit mode, use <F9> or <Escape>.

Output GPIB Driver Configuration (IBCONF.EXE only)

When configuring a GPIB device driver, you can write a text version of the driver to a disk file. Use <F2> to direct `IBCONF` to create a text file named `GPIB.TXT` in the current directory. This file contains a description of the current GPIB driver and should be used only for information purposes.

Autoconfigure (IBCONF.EXE only)

Note: ***All the devices in the system must be connected and powered on before running Autoconfigure.***

Use <F3> to cause `IBCONF` to perform its automatic configuration. When asked to autoconfigure a particular GPIB board, `IBCONF` interrogates all the listen addresses on the GPIB to detect listening devices. `IBCONF` then adjusts the device map for the board so that only the responding devices are connected. It also adjusts the primary and secondary address fields of the devices to match the addresses that responded as Listeners. The entire operation only takes a few seconds. You may want to rename the connected devices with names that indicate their function. For more information on naming devices, refer to the section, *Rename*, earlier in this chapter.

`IBCONF` disconnects devices from higher numbered interface boards to configure the current board. For this reason, if you have more than one board in your system, you should plan to autoconfigure all of them, beginning with board 0 and increasing. If you want to autoconfigure only one board, and leave the others alone, you should arrange for board 3 to be the autoconfigured board, and for the others to be left alone.

Note: ***Do NOT use Autoconfigure if you are using LabWindows.***

Exit

Use <F9> or <Escape> to exit `IBCONF`. If you have made changes, `IBCONF` asks if you want to save the changes to the disk before exiting. Type a `y` (yes) to save changes, `n` (no) to lose changes, or `c` (cancel) and remain in `IBCONF`. For more information on exiting `IBCONF`, refer to the *Exiting IBCONF* section at the end of this chapter.

Lower Level Device/Board Characteristics

Figure 5-2 shows the lower level of `IBCONF`.

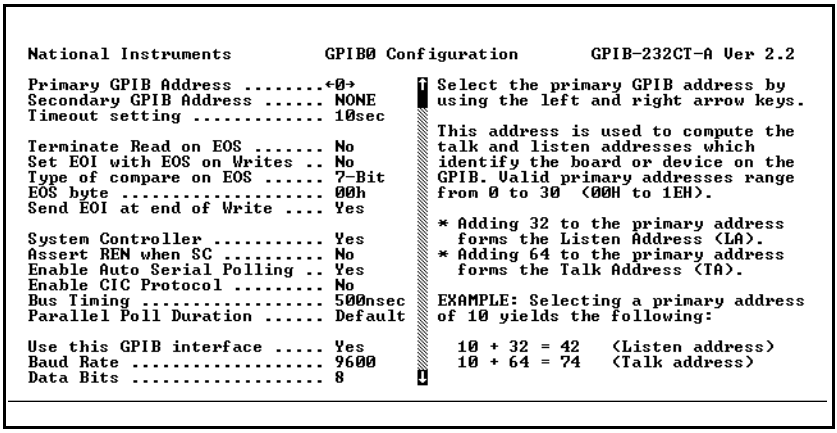

Figure 5-2. Lower Level of `IBCONF`

The lower level screens of `IBCONF` display the currently defined values for characteristics of a device or board, such as addressing and timeout information, as shown in Figure 5-2. You access these screens from the upper level of `IBCONF` by selecting a board or device and pressing <F8> or <Enter>. The configuration settings selected for each device and each board are a means of customizing the communications and other options used with that board or device. The settings for devices specify the characteristics used by the access board for that device when device functions are used. The settings for boards specify the characteristics used when board functions are used.

The following functions are available at the lower level.

- Change Characteristics
- Change Board or Device

- Help
- Reset Value
- Return to Map

Change Characteristics

To change a specific characteristic of a device or a board, move the cursor to or click the mouse on that characteristic. You can also use <PageUp>, <PageDown>, <Home>, or <End> to move around the characteristics of a device or a board. When the cursor is on the characteristic, either use the left/right arrow keys to select between different options or input the option directly from the keyboard. Instructions on the right side of the screen inform you which method is appropriate for the selected characteristic.

Change Board or Device

Use <Control-PageUp> and <Control-PageDown> to move to the next or previous GPIB board or device in your configuration. For example, if you are editing `DEV3` and press <Control-PageUp>, you will then be editing `DEV4`.

Help

Use <F1> to access the comprehensive, online help feature of `IBCONF`. The help information describes the functions and common terms associated with the lower level of `IBCONF`.

Reset Value

Use <F6> to reset a characteristic option to its previous value.

Return to Map

At the lower level, <F9> or <Escape> returns you to the upper level device map of `IBCONF`.

Default Configurations

The NI-488.2 software has factory default configurations. For example, the default device names of the 32 GPIB devices are `DEV1` through `DEV32`. You may want to change the names to more descriptive ones, such as `METER` for a digital multimeter.

You can use `IBCONF` to look at the current default settings in the software file.

If you do not use `IBCONF` to make changes to the NI-488.2 software, the default configurations of the software remain in effect.

Default Values

The following are the default values of the NI-488.2 software.

- There are 32 devices with symbolic names `DEV1` through `DEV32`.
- There are four access boards with symbolic names `GPIB0`, `GPIB1`, `GPIB2`, and `GPIB3`. You cannot change access board names.
- Access board `GPIB0` is enabled. `GPIB1`, `GPIB2`, and `GPIB3` are disabled.
- The GPIB addresses of the first 16 devices are the same as the device number. For example, `DEV1` is at address 1. These devices are assigned to the access board `GPIB0`.
- The GPIB addresses of the second 16 devices are 1 through 16, in order. For example, `DEV17` is at address 1 and `DEV18` is at address 2. These devices are assigned to access board `GPIB1`.
- Each GPIB interface board is System Controller of its independent bus and has a GPIB primary address of 0.
- The END message is sent with the last byte of each data message to a device. No End-of-String (EOS) character is recognized.

- The time limit on I/O and wait function calls is set for approximately 10 s.
- `GPIB0` serial communications settings:

  COM1, 8 Data Bits, 1 Stop Bit, No parity, 9600 baud

Device and Board Characteristics

The following explanations are for board and device characteristics in `IBCONF`. More extensive help for each characteristic is displayed on the `IBCONF` screen while the cursor is positioned in a field. Most of the following characteristics apply to both devices and boards although some, as indicated, only apply to boards.

Primary GPIB Address

All devices and boards must be assigned unique primary addresses in the range from hex 00 to hex 1E (0 to 30 decimal).

Secondary GPIB Address

Any device or board using extended addressing must be assigned a secondary address in the range from hex 60 to hex 7E (96 to 126 decimal), or you can select the `NONE` option to disable secondary addressing. The default option for this characteristic is `NONE`.

Timeout Settings

The timeout value is the approximate minimum length of time that I/O functions such as `ibrd`, `ibwrt`, and `ibcmd` can take before a timeout occurs. It is also the length of time that the `ibwait` function waits for an event before returning if the TIMO bit is set in the event mask. The default option for this characteristic is `10sec`.

Serial Poll Timeouts (Device Characteristic Only)

This timeout value controls the length of time the driver waits for a serial poll response from a device. The ANSI/IEEE 488 standard does not specify the length of time a Controller should wait for the response byte. The driver default of 1 s should work for most devices.

Terminate READ on EOS

Some devices send an EOS byte signaling the last byte of a data message. A `yes` response to this field causes the GPIB board to terminate a read operation when it receives the EOS byte. The default option for this characteristic is `no`. See also the *EOS Byte* section.

Set EOI with EOS on Write

A `yes` response to this field causes the GPIB board to assert the EOI line when the EOS byte is detected on a write operation. The default option for this characteristic is `no`. See also the *EOS Byte* section.

Type of Compare on EOS

This field specifies the type of comparison to be made with the EOS byte. You may indicate whether all eight bits are to be compared or just the seven least significant bits (ASCII or ISO format). This field is only valid if a `yes` response was given for either the Set EOI with EOS on Write field or the Terminate Read on EOS field. The default option for this characteristic is `7-bit`. See also the *EOS Byte* section.

EOS Byte

You can program some devices to terminate a read operation when a selected character is detected. A linefeed character (hex 0A) is a common EOS byte.

Note: ***The driver does not automatically append an EOS byte to the end of data strings on write operations. You must explicitly include this byte in your data string. The designation of the EOS byte is***

only for the purpose of informing the driver of its value so that I/O can terminate correctly.

The default option for this characteristic is `00H`.

Set EOI at End of Write

Some devices, as Listeners, require that the Talker terminate a data message by asserting the EOI line with the last byte. A `yes` response causes the GPIB interface board to assert the EOI line on the last data byte. The default option for this characteristic is `yes`.

System Controller (Board Characteristic Only)

This field appears only on the board characteristics screen. The System Controller in a GPIB system is the device that maintains ultimate control over the bus. There should only be one device designated as System Controller in any GPIB system. The default option for this characteristic is `yes`.

Assert REN when SC (Board Characteristic Only)

A `yes` response to this field causes Remote Enable (REN) to be asserted automatically whenever the board is placed online, provided that the board has been given System Controller capability. If a `no` response is provided, an explicit call to `ibsre` is required to assert REN. The default option for this characteristic is `no`.

Enable Auto Serial Polling (Board Characteristic Only)

This option enables or disables automatic serial polls of devices when the GPIB Service Request (SRQ) line is asserted. Positive poll responses are stored following the polls and can be read with the `ibrsp` device function. The default option for this characteristic is `no`.

Enable CIC Protocol (Board Characteristic Only)

If a device-level NI-488 call is made after control has been passed to another device, enabling this protocol causes the board to assert SRQ with a Serial Poll status byte of hex 42. The current Controller must recognize that the board wants to regain control. If the current Controller passes control back to the board, the device call proceeds as intended. If control is not passed within the timeout period, the error ECIC results from the call. If the CIC protocol is disabled, ECIC is returned immediately if a device call is made after control has been passed. The default option for this characteristic is `no`.

Bus Timing (Board Characteristic Only)

This field specifies the T1 delay of the source handshake capability for the board. This delay determines the minimum amount of time, after the data is placed on the bus, that the board may assert DAV during a write or command operation. If the total length of the GPIB cable length in the system is less than 15 m, then the value of `350nsec` is appropriate.

There are other factors that may affect the choice of the T1 delay, although they are unlikely to affect you. Refer to the ANSI/IEEE 488.2-1987 Standard, *IEEE Standard Codes, Formats, Protocols, and Common Commands*, for more information about these other factors.

The default option is `500nsec`.

Enable Repeat Addressing (Device Characteristic Only)

Normally, devices are addressed before a read or write operation is performed only if they are not already properly addressed for the read or write operation. If `yes` is selected, read or write operations always readdress even if the device is already properly addressed. The default option is `no`.

Use This Interface (Board Characteristic Only)

If you do not want the driver to try to access an interface (because you do not have an interface in the system), select `no` for this option. When this field is set to `no`, the driver does not try to access the interface hardware. It

returns the error ENEB as soon as a program tries to access the board. The default is `yes` for `GPIB0` and `no` for all other GPIB boards.

COM Port (Board Characteristic Only)

This field specifies the serial communications port to which the GPIB interface is connected. COM1 through COM4 define the standard PC-compatible serial communications ports. The Serial Port Base Address and IRQ Level fields change accordingly when you change the COM Port Selection. If you have non-standard serial communications ports, then modify the Serial Port Base Address and IRQ level fields separately. The default for `GPIB0` is `COM1`.

Serial Port Base Address (Board Characteristic Only)

This field specifies the base I/O address of the serial communications port and should only be used if you have a non-standard serial communications port. The GPIB-232CT-A software assumes that the serial controller hardware is compatible with the National Semiconductor INS8250 universal asynchronous receiver/transmitter (the standard for PC-compatibles).

Serial Port IRQ Level (Board Characteristic Only)

This field specifies the IRQ level of the serial communications port and should only be used if you have a non-standard serial communications port. The GPIB-232CT-A software assumes that the serial controller hardware is compatible with the National Semiconductor INS8250 universal asynchronous receiver/transmitter (the standard for PC-compatibles).

Baud Rate (Board Characteristic Only)*

This field specifies the baud rate of the serial communications port. It ranges from 300 baud to 38400 baud. For Windows, 38400 baud is not supported and 19200 baud does not work properly in all configurations; you should use the default rate of 9600 baud.

Parity (Board Characteristic Only)*

This field specifies the parity checking for the serial communications port. Parity checking can be odd, even, or none; the default is none.

Stop Bits (Board Characteristic Only)*

This field specifies the number of stop bits per character for the serial communications port. Stop bits can be set to 1 or 2; the default is 1.

Data Bits (Board Characteristic Only)*

This field specifies the number of data bits per character for the serial communications port. The number of data bits can be set to 7 or 8; the default is 8.

* If you change these software settings, you must change your hardware settings to match. Refer to Appendix A, *Hardware Configuration*, for more information on changing these settings.

Exiting IBCONF

After you have made all your changes, you can exit `IBCONF` by pressing <F9> or <Escape>. The program asks if you want to save the changes to the disk before exiting. Typing a `y` (yes) response causes the changes to be written to the file on disk.

Before exiting, the program checks for situations that can cause problems, such as the following:

- Is there a GPIB addressing conflict between a device and its access board?
- Are GPIB boards not present in the host machine at the specified address?
- Are timeouts disabled on a device or board?

If any of these situations are found, you are notified and given the option of re-entering or exiting `IBCONF`.

Appendix A
Hardware Configuration

This appendix describes how to configure the GPIB-232CT-A RS-232 serial port.

The default hardware settings for the GPIB-232CT-A are compatible with the default software settings. If you change the hardware settings, make sure that you also change the software settings so that they are compatible.

Configure the RS-232 Serial Port

The GPIB-232CT-A is shipped from the factory configured to operate in Serial (S) mode. In S mode, the computer attached to the RS-232 port of the GPIB-232CT-A is the controlling device. The RS-232 serial port is configured at 9600 baud, 1 stop bit, no parity, and 8 data bits. To change the parameters, you set the configuration switches on the back panel.

The DIP switch is used to configure the serial port parameters of the GPIB-232CT-A while in S mode. In G mode, the GPIB-232CT-A is used to interface an RS-232 device as a GPIB Talker/Listener.

Figure A-1 shows the factory default setting of the DIP switch. In Figure A-1, the black side of the switch represents the position of the switch handle.

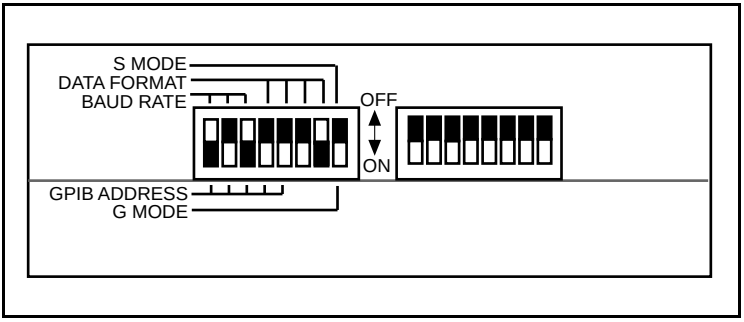

Figure A-1. Factory Default Setting (S Mode) for DIP Switch

Note: ***For the purpose of this explanation, the switches have been assigned numbers as a point of reference. These numbers do NOT appear on the GPIB-232CT-A itself. In this explanation, the three switches labeled BAUD RATE are switches 1 through 3, the DATA FORMAT switches are switches 4 through 7, and the S MODE switch is switch 8.***

In Figure A-1, switches 1 through 3 are ON, OFF, and ON, respectively, indicating that the serial port is operating at 9600 baud. Switches 4 and 5 are both OFF, which indicates that parity is disabled. Switch 6 is OFF, indicating 1 stop bit/character. Switch 7 is ON, indicating that the GPIB-232CT-A is using 8 bits per character for serial data transfers. Switch 8 is OFF, indicating that the GPIB-232CT-A is operating in S mode.

Tables A-1 and A-2 show the possible configurations of the eight switches and what each configuration indicates. The factory default settings are in ***bold italics***.

Table A-1. Configuration Parameters for Switches 1 through 3

Switches			Indication
1	**2**	**3**	
OFF	OFF	OFF	300 baud
ON	OFF	OFF	600 baud
OFF	ON	OFF	1200 baud
ON	ON	OFF	2400 baud
OFF	OFF	ON	4800 baud
ON	***OFF***	***ON***	***9600 baud***
OFF	ON	ON	19200 baud
ON	ON	ON	38400 baud

Table A-2. Configuration Parameters for Switches 4 through 8

Switch	Position	Indication
4	***OFF***	***odd parity***
	ON	even parity
5	***OFF***	***parity generation/checking disabled***
	ON	parity generation/checking enabled
6	***OFF***	***1 stop bit/character***
	ON	2 stop bits/character
7	OFF	7 bits/character
	ON	***8 bits/character***
8	***OFF***	***operates in S mode***
	ON	operates in G mode

Appendix B
Hardware Specifications

The appendix specifies the electrical, environmental, and physical characteristics of the GPIB-232CT-A and the recommended operating conditions.

Table B-1. Electrical Characteristics - AC Version

Characteristic	Specification
Power Supply Unit	100-120 VAC ± 10%, 50-60 Hz or 220-240 VAC ± 10%, 50-60 Hz
Current	100-120 VAC 23 mA or 220-240 VAC 20 mA
Fuse Rating and Type	100-120 VAC 300 mA, UL/CSA approved or 220-240 VAC 500 mA, IEC approved

Table B-2. Electrical Characteristics - DC Version

Characteristic	Specification
Power Supply Unit	Wall-mount type, 100-120 VAC ± 10% 50-60 Hz input , 9 VDC @ 1A output or Desktop type, 220-240 VAC ± 10% 50-60 Hz input, 9 VDC @ 1 A output
DC Input	+5 to +13 Regulated
Current	300 mA typical; 400 mA maximum

Table B-3. Environmental Characteristics- AC Version

Characteristic	Specification
Operating Temperature	10° to 40° C
Storage Temperature	0° to 70° C
Relative Humidity	10% to 95% noncondensing conditions
EMI	FCC Class A Verified

Table B-4. Environmental Characteristics- DC Version

Characteristic	Specification
Operating Temperature	10° to 40° C
Storage Temperature	0° to 70° C
Relative Humidity	10% to 95% noncondensing conditions
EMI	FCC Class B Certified

Table B-5. Physical Characteristics - AC Version

Characteristic	Specification
Case Size	4.65 in. by 3.0 in. by 1.74 in. (118.1 mm by 76.2 mm by 44.2 mm)
Case Material	All metal enclosure
Weight	12 oz (340 g)

Table B-6. Physical Characteristics - DC Version

Characteristic	Specification
Case Size	4.65 in. by 3.0 in. by 1.11 in. (118.1 mm by 76.2 mm by 28.2 mm)
Case Material	All metal enclosure
Weight	7 oz (198 g)

Appendix C
Troubleshooting

This appendix suggests some areas to check if you have problems installing or using the GPIB-232CT-A or the NI-488.2 software after going through the procedures described in Chapters 2 through 5.

If you still have problems after completing the steps in this appendix, complete the appropriate forms in Appendix F, *Customer Communication* and then contact National Instruments for technical support.

Troubleshooting Hardware Problems

Warning: ***The AC version of the GPIB-232CT-A contains circuitry that operates with hazardous voltages. Do NOT open the unit unless so instructed by National Instruments. Be sure to remove the power cord before opening the unit.***

- All cables must be securely connected to the GPIB-232CT-A.
- Check the DIP switch settings on the GPIB-232CT-A. This DIP switch selects the serial port configuration. Most applications require the default setting, which is shown in Figure A-1. Refer to Appendix A, *Hardware Configuration,* for information on setting the GPIB-232CT-A configuration using this switch.
- Make sure that the GPIB-232CT-A is powered on.
- If you have an AC version, check the fuse.

  Warning: ***For continued protection against fire, replace only with the same type and rating of fuse. See Appendix B,*** *Hardware Specifications****, for fuse specifications.***

Troubleshooting Software Problems

If the following three conditions apply, the GPIB-232CT-A and the NI-488.2 software are unable to communicate properly.

- GPIB calls return with the ERR bit set in `ibsta`.
- The value of `iberr` is EDVR.
- The value of `ibcntl` is 122541, 188077, or 253613 (`1DEAD`, `2DEAD`, or `3DEAD` in hexadecimal notation).

The solution depends on the value of `ibcntl`. The following paragraphs contain troubleshooting tips for each of the `ibcntl` values.

- If `ibcntl` is `1DEAD`, the software is unable to open and initialize the serial communications port. You should reconfigure your GPIB-232CT-A and software to verify that the settings match.
- If `ibcntl` is `2DEAD`, the software is encountering serial overrun errors when it attempts to access the GPIB-232CT-A box. Serial overrun errors occur when the software is not able to service the serial interrupt receiving a serial byte before the next serial byte arrives. One way to eliminate this problem is to slow down the baud rate that the GPIB-232CT-A and NI-488.2 software are using. Another way to eliminate this problem is to unload any device drivers, TSRs, or applications that you are using that might disallow interrupts over an extended period of time.
- If `ibcntl` is `3DEAD`, there is total miscommunication between the NI-488.2 software and the GPIB-232CT-A. The only solution is to power off the box and restart the computer. If the problem persists, try using a different serial cable.

Appendix D
DLL Direct Entry NI-488 Functions and NI-488.2 Routines

This appendix explains and gives examples of how to use the DLL Direct Entry NI-488 functions and NI-488.2 routines to access the `GPIB.DLL` file. Following the examples are tables that list all NI-488.2 routines and NI-488 functions, including their calling syntax and ordinal entry values.

The DLL Direct Entry NI-488 functions and NI-488.2 routines can be used to access the `GPIB.DLL` file from any language or programming environment that runs under Windows and supports access to standard Windows DLL functions. As with all functions exported by a DLL, these functions conform to the PASCAL calling conventions. A few examples of using these entry points follow. Tables D-1 and D-2 contain a complete list of all of the entry points.

For specific information on the variables `ibsta`, `iberr`, and `ibcntl`, refer to Chapter 3, *Understanding the NI-488.2 Software*, in the *NI-488.2 Software Reference Manual for MS-DOS*. For specific information on a routine or function, refer to the *NI-488.2 Software Reference Manual for MS-DOS*. For information about accessing dynamic link library DLL functions from a given language or environment or using ordinal entry values which some environments do not support, see the documentation provided with that package.

Example 1, accessing the `GPIB.DLL` file from Turbo Pascal for Windows:

```
(* First, import the DLL functions you plan to use. *)

function DLLibfind(udname: PChar;
                   var ibsta: integer;
                   var iberr: integer;
                   var ibcntl: longint) : integer; far;
external 'GPIB' index 22;

function DLLibsic(ud: integer;
                   var: ibsta: integer;
                   var: iberr: integer;
                   var: ibcntl: longint) : integer; far;
external 'GPIB' index 42;
```

```
procedure DLLSendIFC(board: integer;
                     var ibsta: integer;
                     var iberr: integer;
                     var ibcntl: longint) ; far;
external 'GPIB' index 119;

(* Your application can now use the functions. *)

var BoardHandle: integer;
var ibsta: integer;
var iberr: integer;
var ibcntl: longint;
var temp: integer;

BoardHandle:= DLLibfind('GPIB0', ibsta, iberr, ibcntl);
temp:= DLLibsic(BoardHandle, ibsta, iberr, ibcntl);

                  (* or *)

DLLSendIFC(0, ibsta, iberr, ibcntl);
```

Example 2, accessing the `GPIB.DLL` file from Microsoft Visual Basic:

```
'First declare the DLL functions you plan to use.

Declare function DLLibfind Lib "gpib.dll"
   (ByVal udname$, ibsta%, iberr%, ibcntl&) As Integer
Declare function DLLibsic Lib "gpib.dll"
   (ByVal ud%, ibsta%, iberr%, ibcntl&) As Integer
Declare procedure DLLSendIFC Lib "gpib.dll"
   (ByVal board%, ibsta%, iberr%, ibcntl&)

'Your application can now use the functions.

Global BoardHandle As Integer
Global ibsta As Integer
Global iberr As Integer
Global ibcntl As Long

BoardHandle% =
   DLLibfind("GPIB0", ibsta%, iberr%, ibcntl&)

temp% = DLLibsic(BoardHandle%, ibsta%, iberr%, ibcntl&)

   'or

call DLLSendIFC(0, ibsta%, iberr%, ibcntl&)
```

Note: ***All of the routines listed in Table D-1 are of type void _far _pascal.***

Table D-1. Direct Entry NI-488.2 Style Routines

Routine (ordinal entry value)	**Syntax**
AllSpoll (100)	DLLAllSpoll (short board, short _far* addresslist, short _far * resultlist, short _far *ibsta, short _far *iberr, unsigned long _far *ibcntl)
DevClear (101)	DLLDevClear (short board, short address, short _far *ibsta, short _far *iberr, unsigned long _far *ibcntl)
DevClearList (102)	DLLDevClearList (short board, short _far *addresslist, short _far *ibsta, short _far *iberr, unsigned long _far *ibcntl)
EnableLocal (103)	DLLEnableLocal (short board, short _far *addresslist, short _far *ibsta, short _far *iberr, unsigned long _far *ibcntl)
EnableRemote (104)	DLLEnableRemote (short board, short _far *addresslist, short _far *ibsta, short _far *iberr, unsigned long _far *ibcntl)
FindLstn (105)	DLLFindLstn (short board, short _far *addresslist, short _far *resultlist, short limit, short _far *ibsta, short _far *iberr, unsigned long _far *ibcntl)
FindRQS (106)	DLLFindRQS (short board, short _far *addresslist, short _far *result, short _far *ibsta, short _far *iberr, unsigned long _far *ibcntl)

Table D-1. Direct Entry NI-488.2 Style Routines (continued)

Routine (ordinal entry value)	**Syntax**
PassControl (107)	DLLPassControl (short board, short address, short _far *ibsta, short _far *iberr, unsigned long _far *ibcntl)
PPoll (108)	DLLPPoll (short board, short _far *result, short _far *ibsta, short _far *iberr, unsigned long _far *ibcntl)
PPollConfig (109)	DLLPPollConfig (short board, short address, short dataline, short sense, short _far *ibsta, short _far *iberr, unsigned long _far *ibcntl)
PPollUnconfig (110)	DLLPPollUnconfig (short board, short _far *addresslist, short _far *ibsta, short _far *iberr, unsigned long _far *ibcntl)
RcvRespMsg (111)	DLLRcvRespMsg (short board, char _far *data, long count, short termination, short _far *ibsta, short _far *iberr, unsigned long _far *ibcntl)
ReadStatusByte (112)	DLLReadStatusByte (short board, short address, short _far *result, short _far *ibsta, short _far *iberr, unsigned long _far *ibcntl)
Receive (113)	DLLReceive (short board, short address, char _far *data, unsigned long count, short termination, short _far *ibsta, short _far *iberr, unsigned long _far *ibcntl)

(continues)

Table D-1. Direct Entry NI-488.2 Style Routines (continued)

Routine (ordinal entry value)	**Syntax**
ReceiveSetup (114)	DLLReceiveSetup (short board, short address, short _far *ibsta, short _far *iberr, unsigned long _far *ibcntl)
ResetSys (115)	DLLResetSys (short board, short _far *addresslist, short _far *ibsta, short _far *iberr, unsigned long _far *ibcntl)
Send (116)	DLLSend (short board, short address, char _far *data, long count, short eotmode, short _far *ibsta, short _far *iberr, unsigned long _far *ibcntl)
SendCmds (117)	DLLSendCmds (short board, char _far *commands, unsigned long count, short eotmode, short _far *ibsta, short _far *iberr, unsigned long _far *ibcntl)
SendDataBytes (118)	DLLSendDataBytes (short board, char _far *data, long count, short eotmode, short _far *ibsta, short _far *iberr, unsigned long _far *ibcntl)
SendIFC (119)	DLLSendIFC (short board, short _far *ibsta, short _far *iberr, unsigned long _far *ibcntl)
SendList (120)	DLLSendList (short board, short _far *addresslist, char _far *data, long count, short eotmode, short _far *ibsta, short _far *iberr, unsigned long _far *ibcntl)

(continues)

Table D-1. Direct Entry NI-488.2 Style Routines (continued)

Routine (ordinal entry value)	**Syntax**
SendLLO (121)	DLLSendLLO (short board, short _far *ibsta, short _far *iberr, unsigned long _far *ibcntl)
SendSetup (122)	DLLSendSetup (short board, short _far *addresslist, short _far *ibsta, short _far *iberr, unsigned long _far *ibcntl)
SetRWLS (123)	DLLSetRWLS (short board, short _far *addresslist, short _far *ibsta, short _far *iberr, unsigned long _far *ibcntl)
TestSRQ (124)	DLLTestSRQ (short board, short _far *result, short _far *ibsta, short _far *iberr, unsigned long _far *ibcntl)
TestSys (125)	DLLTestSys (short board, short _far *addresslist, short _far *resultlist, short _far *ibsta, short _far *iberr, unsigned long _far *ibcntl)
Trigger (126)	DLLTrigger (short board, short address, short _far *ibsta, short _far *iberr, unsigned long _far *ibcntl)
TriggerList (127)	DLLTriggerList (short board, short _far *addresslist, short _far *ibsta, short _far *iberr, unsigned long _far *ibcntl)
WaitSRQ (128)	DLLWaitSRQ (short board, short _far *result, short _far *ibsta, short _far *iberr, unsigned long _far *ibcntl)

Note: ***All of the functions listed in Table D-2 are of type short _far _pascal.***

Table D-2. Direct Entry NI-488 Style Functions

Functions (**ordinal entry value**)	**Syntax**
ibbna (10)	DLLibbna (short ud, char _far *bname, short _far *ibsta, short _far *iberr, unsigned long _far *ibcntl)
ibcac (11)	DLLibcac (short ud, short v, short _far *ibsta, short _far *iberr, unsigned long _far *ibcntl)
ibclr (12)	DLLibclr (short ud, short _far *ibsta, short _far *iberr, unsigned long _far *ibcntl)
ibcmd (13)	DLLibcmd (short ud, char _far *cmd, long cnt, short _far *ibsta, short _far *iberr, unsigned long _far *ibcntl)
ibcmda (14)	DLLibcmda (short ud, char _far *cmd, long cnt, short _far *ibsta, short _far *iberr, unsigned long _far *ibcntl)
ibconfig (15)	DLLibconfig (short ud, unsigned short option, unsigned short cnt, short _far *ibsta, short _far *iberr, unsigned long _far *ibcntl)
ibdev (16)	DLLibdev (short boardindex, short pad, short sad, short tmo, short eot, short eos, short _far *ibsta, short _far *iberr, unsigned long _far *ibcntl)
ibdma (18)	DLLibdma (short ud, short v, short _far *ibsta, short _far *iberr, unsigned long _far *ibcntl)

Table D-2. Direct Entry NI-488 Style Functions (continued)

Functions (**ordinal entry value**)	**Syntax**
ibeos (19)	DLLibeos (short ud, short v, short _far *ibsta, short _far *iberr, unsigned long _far *ibcntl)
ibeot (20)	DLLibeot (short ud, short v, short _far *ibsta, short _far *iberr, unsigned long _far *ibcntl)
ibfind (22)	DLLibfind (char _far *udname, short _far *ibsta, short _far *iberr, unsigned long _far *ibcntl)
ibgts (23)	DLLibgts (short ud, short v, short _far *ibsta, short _far *iberr, unsigned long _far *ibcntl)
ibist (24)	DLLibist (short ud, short v, short _far *ibsta, short _far *iberr, unsigned long _far *ibcntl)
iblines (25)	DLLiblines (short ud, short _far *clines, short _far *ibsta, short _far *iberr, unsigned long _far *ibcntl)
ibln (26)	DLLiblin (short ud, short pad, short sad, short _far *listen, short _far *ibsta, short _far *iberr, unsigned long _far *ibcntl)
ibloc (27)	DLLibloc (short ud, short _far *ibsta, short _far *iberr, unsigned long _far *ibcntl)
ibonl (28)	DLLibonl (short ud, short v, short _far *ibsta, short _far *iberr, unsigned long _far *ibcntl)
ibpad (29)	DLLibpad (short ud, short v, short _far *ibsta, short _far *iberr, unsigned long _far *ibcntl)

(continues)

Table D-2. Direct Entry NI-488 Style Functions (continued)

Functions (**ordinal entry value**)	**Syntax**
ibpct (30)	DLLibpct (short ud, short _far *ibsta, short _far *iberr, unsigned long _far *ibcntl)
ibppc (32)	DLLibppc (short ud, short v, short _far *ibsta, short _far *iberr, unsigned long _far *ibcntl)
ibrd (33)	DLLibrd (short ud, short _far *rd, unsigned long cnt, short _far *ibsta, short _far *iberr, unsigned long _far *ibcntl)
ibrda (34)	DLLibrda (short ud, char _far * rd, unsigned long cnt, short _far *ibsta, short _far *iberr, unsigned long _far *ibcntl)
ibrdf (35)	DLLibrdf (short ud, char _far * flname, short _far *ibsta, short _far *iberr, unsigned long _far *ibcntl)
ibrpp (37)	DLLibrpp (short ud, char _far * ppr, short _far *ibsta, short _far *iberr, unsigned long _far *ibcntl)
ibrsc (38)	DLLibrsc (short ud, short v, short _far *ibsta, short _far *iberr, unsigned long _far *ibcntl)
ibrsp (39)	DLLibrsp (short ud, char _far * spr, short _far *ibsta, short _far *iberr, unsigned long _far *ibcntl)
ibrsv (40)	DLLibrsv (short ud, short v, short _far *ibsta, short _far *iberr, unsigned long _far *ibcntl)
ibsad (41)	DLLibsad (short ud, short v, short _far *ibsta, short _far *iberr, unsigned long _far *ibcntl)

Table D-2. Direct Entry NI-488 Style Functions (continued)

Functions (**ordinal entry value**)	**Syntax**
ibsic (42)	DLLibsic (short ud, short _far *ibsta, short _far *iberr, unsigned long _far *ibcntl)
ibsre (43)	DLLibsre (short ud, short v, short _far *ibsta, short _far *iberr, unsigned long _far *ibcntl)
ibstop (44)	DLLibstop (short ud, short _far *ibsta, short _far *iberr, unsigned long _far *ibcntl)
ibtmo (45)	DLLibtmo (short ud, short v, short _far *ibsta, short _far *iberr, unsigned long _far *ibcntl)
ibtrg (46)	DLLibtrg (short ud, short _far *ibsta, short _far *iberr, unsigned long _far *ibcntl)
ibwait (47)	DLLibwait (short ud, short mask, short _far *ibsta, short _far *iberr, unsigned long _far *ibcntl)
ibwrt (48)	DLLibwrt (short ud, char _far *wrt, unsigned long cnt, short _far *ibsta, short _far *iberr, unsigned long _far *ibcntl)
ibwrta (49)	DLLibwrta (short ud, char _far *wrt, unsigned long cnt, short _far *ibsta, short _far *iberr, unsigned long _far *ibcntl)
ibwrtf (50)	DLLibwrtf (short ud, char _far *flname, short _far *ibsta, short _far *iberr, unsigned long _far *ibcntl)

Appendix E
Interfacing to a Serial Device

This appendix describes the RS-232 serial port on the GPIB-232CT-A and explains how to interface a DCE or DTE serial device to the RS-232 serial port.

The GPIB-232CT-A transfers serial data using the electrical signals, mechanical connections, data format, and control protocols defined in the RS-232C standard. The RS-232 port on the GPIB-232CT-A provides an asynchronous serial communication link to a serial peripheral device.

The RS-232C Standard

The RS-232C standard (international standard CCITT V.24) was formulated in 1969 largely from the efforts of the Electronic Industries Association (EIA) and Bell Laboratories. The standard describes the electrical specifications and arrangement of control and data signals on both sides of a serial communications interface. Its original intent was to interface terminals to modems. Many manufacturers of computers and instruments have adopted the standard for their serial communications needs. For more information on the RS-232C standard, contact

Global Engineering Documents
7730 Carondelet Avenue, Suite 4007
St. Louis, MO 63105
(800) 854-7179

Description of the RS-232 Port

The RS-232 serial port on the GPIB-232CT-A uses a male 9-pin D-Subminiature connector with a DTE interface configuration. Table E-1 shows the signal lines supported on the GPIB-232CT-A.

Table E-1. RS-232 Serial Port Signal Configuration

Pin Number	Signal Description	RS-232 Code	Function
2	RXD (Receive Data)	BB	This signal carries serial data from the serial device to the GPIB-232CT-A.
3	TXD (Transmit Data)	BA	This signal carries serial data from the GPIB-232CT-A to the serial device.
4	DTR (Data Terminal Ready)	CD	This signal is asserted by the GPIB-232CT-A to signal that it has been powered on and is ready to operate.
5	GND (Ground)	AB	This signal establishes a reference point for all interface voltages.
7	RTS (Request to Send)	CA	This signal is driven by the GPIB-232CT-A. When asserted, it indicates that the GPIB-232CT-A is ready to accept serial data. When unasserted, it indicates that the GPIB-232CT-A is no longer ready to accept serial data because the buffer is full.
8	CTS (Clear to Send)	CB	This signal is sensed by the GPIB-232CT-A. When asserted, it indicates that the serial device is ready to accept serial data. When unasserted, it indicates that data transmission should be disabled.

Interfacing Serial Devices to the RS-232 Serial Port

To interface other serial devices to the RS-232 serial port on the GPIB-232CT-A, first refer to the manual that came with your serial device to determine if the device is configured as a DTE or DCE. Also, determine from the manual how the control lines are used and whether they must be driven for the serial port to operate.

Note: ***The GPIB-232CT-A serial port is configured to be a DTE.***

Figure E-1 shows the location of the RS-232 connector.

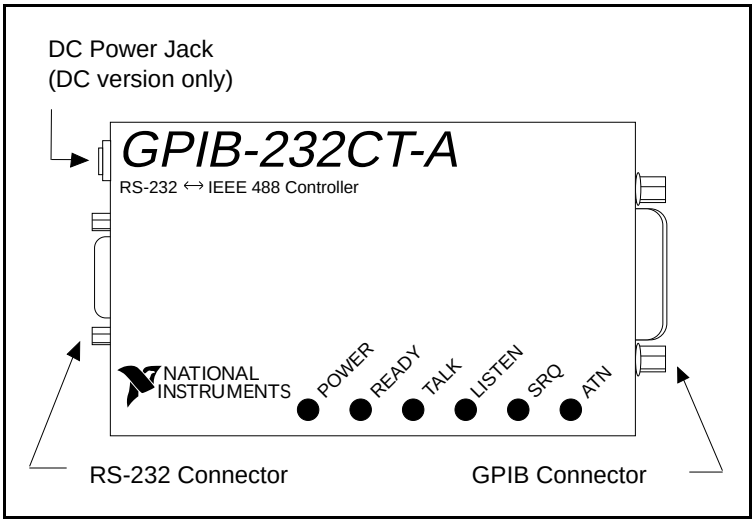

Figure E-1. Location of the RS-232 Connector

Interfacing the GPIB-232CT-A to a DCE with Handshaking

A correctly configured DTE-to-DCE interface is wired *straight across*: GPIB-232CT-A TXD to DCE TXD, GPIB-232CT-A RXD to DCE RXD, and so on as shown in Table E-2.

Table E-2. Cable Wiring Scheme for GPIB-232CT-A DTE to Serial Device DCE

GPIB-232CT-A Signal to DCE Serial Device Signal	**Standard DTE to DCE 9-pin 9-pin**	**Standard DTE to DCE 9-pin 25-pin**
RXD to RXD	***2 to 2***	***2 to 3***
TXD to TXD	***3 to 3***	***3 to 2***
DTR to DTR	4 to 4	4 to 20
GND to GND	***5 to 5***	***5 to 7***
RTS to RTS	7 to 7	7 to 4
CTS to CTS	8 to 8	8 to 5
Note: Connections in ***bold italics*** must be implemented.		

Figure E-2 shows a properly configured 9-pin DTE to 9-pin DCE cable including the hardware handshake lines RTS, CTS, and DTR. With this configuration, the GPIB-232CT-A can function properly (handshake) on buffer full conditions. Figure E-3 shows an equivalent 9-pin DTE to 25-pin DCE cable configuration.

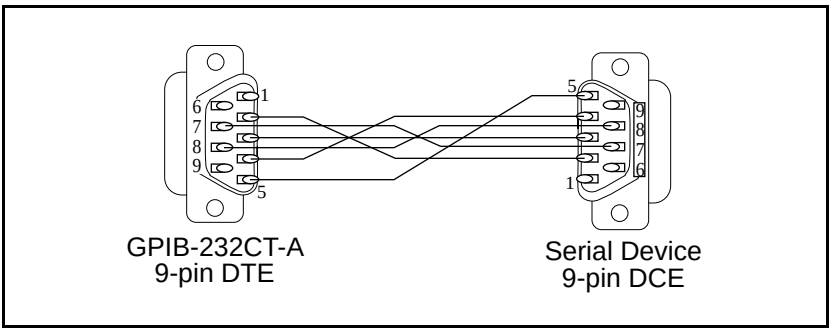

Figure E-2. Cable Configuration for 9-pin DTE to 9-pin DCE with Handshaking

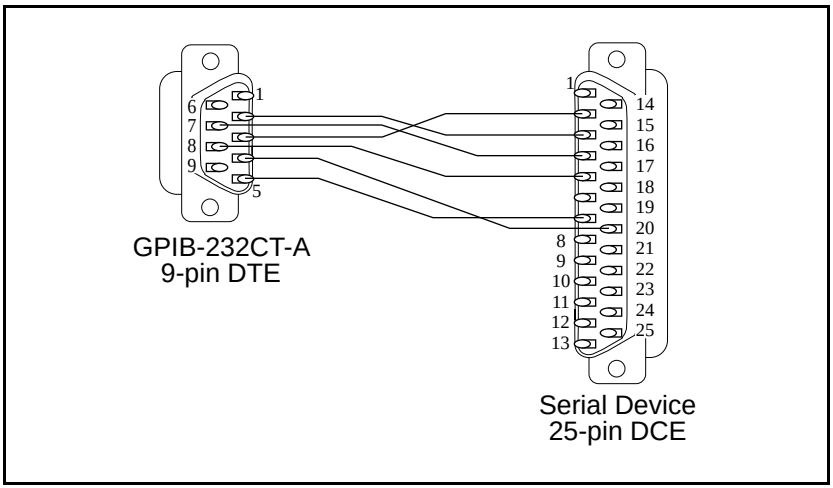

Figure E-3. Cable Configuration for 9-pin DTE to 25-pin DCE with Handshaking

Interfacing the GPIB-232CT-A to a DCE without Handshaking

If your serial device does not require or use the hardware handshaking protocol used by the GPIB-232CT-A, you have two options:

- Use a minimum configuration cable which does not support hardware handshaking and use XON/XOFF software handshaking (if necessary). To create a minimum configuration cable, connect the signals shown in bold italics in Table E-2.

- Wire a custom cable that properly interfaces the GPIB-232CT-A hardware handshaking protocol to the handshaking protocol of your serial device.

Minimum Configuration Cable

The minimum configuration cable assumes that the DCE does not require external hardware handshaking. The minimum configuration for a DTE 9-pin to DCE 9-pin cable is shown in Figure E-4. Figure E-5 shows an equivalent 9-pin to 25-pin cable.

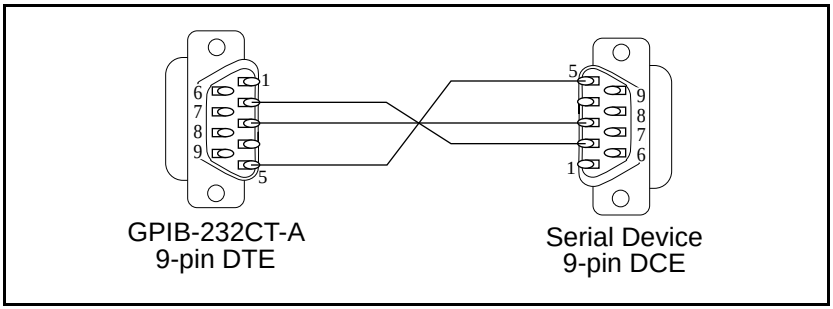

Figure E-4. Minimum Configuration for 9-pin DTE to 9-pin DCE

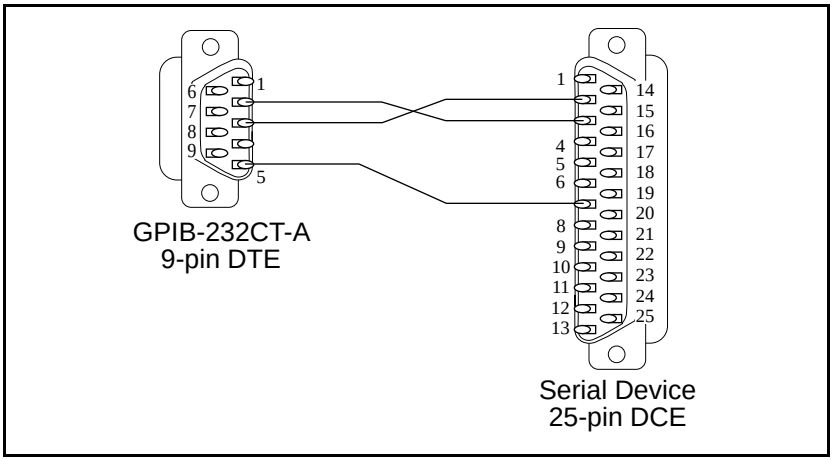

Figure E-5. Minimum Configuration for 9-pin DTE to 25-pin DCE

Custom Cables

If your application requires a custom cable, you can construct one if you have a thorough knowledge of the handshaking protocols involved. Review the RS-232 characteristics of your serial device and build the cable to properly connect the handshake lines of the two devices. Because the DTE-to-DCE connection is a straight across connection, it often involves only connecting RTS to RTS, CTS to CTS, DTR to DTR, and so on. If the documentation for your serial device does not provide a thorough explanation of its handshaking protocol, the ANSI/EIA-232-C standard is a good reference, provided your device conforms to the RS-232 protocol.

Caution: ***Although handshaking might not be required, it is best to use some form of handshaking to prevent loss of data.***

Interfacing the GPIB-232CT-A to a DTE with Handshaking

For serial devices set up as DTEs, you must wire a DTE-to-DTE interface cable, commonly known as a null modem cable. The cable allows the GPIB-232CT-A to act as though it is communicating with a DCE, but it swaps the appropriate pins to achieve a DTE configuration. This wiring configuration is shown in Table E-3.

Table E-3. Cable Wiring Scheme for GPIB-232CT-A DTE to Serial Device DTE

GPIB-232CT-A Signal to DTE Serial Device Signal	**Standard DTE to DTE 9-pin 9-pin**	**Standard DTE to DTE 9-pin 25-pin**
RXD to TXD	***2 to 3***	***2 to 2***
TXD to RXD	***3 to 2***	***3 to 3***
DTR to DSR	4 to 6	4 to 6
GND to GND	***5 to 5***	***5 to 7***
RTS to CTS	7 to 8	7 to 5
CTS to RTS	8 to 7	8 to 4
Note: Connections in ***bold italics*** must be implemented.		

Figure E-6 shows a typical 9-pin to 9-pin null modem cable with the RTS, CTS, and DTR handshake lines implemented. Figure E-7 shows an equivalent 9-pin DTE to 25-pin DTE cable configuration.

The cable configuration in Figure E-6 allows you to connect to the 9-pin serial port of a personal computer. The cable in Figure E-7 shows how to connect to the 25-pin serial port of a personal computer.

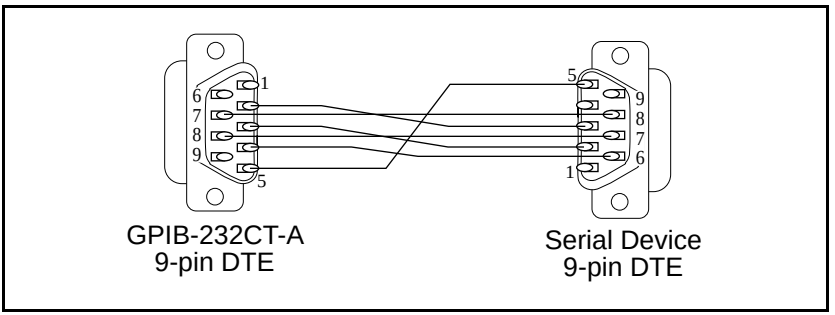

Figure E-6. Cable Configuration for 9-pin DTE to 9-pin DTE with Handshaking

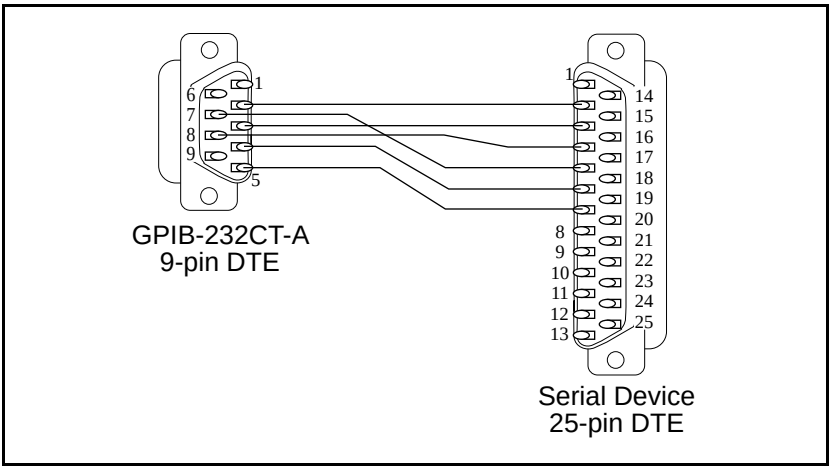

Figure E-7. Cable Configuration for 9-pin DTE to 25-pin DTE with Handshaking

Interfacing the GPIB-232CT-A to a DTE without Handshaking

If your serial device does not require or use the same hardware handshaking protocol used by the GPIB-232CT-A, you have two options:

- Use a minimum configuration null modem cable which does not support the hardware handshake lines and use XON/XOFF software handshaking (if necessary). To create a minimum configuration cable, connect the signals shown in bold italics in Table E-3.
- Wire a custom cable that properly interfaces the GPIB-232CT-A hardware handshaking protocol to the handshaking protocol of your serial device.

Minimum Configuration Cable

The minimum cable configuration assumes that the serial device does not require external hardware handshaking. The minimum configuration for a 9-pin to 9-pin null modem cable is shown in Figure E-8. Figure E-9 shows an equivalent 9-pin to 25-pin cable.

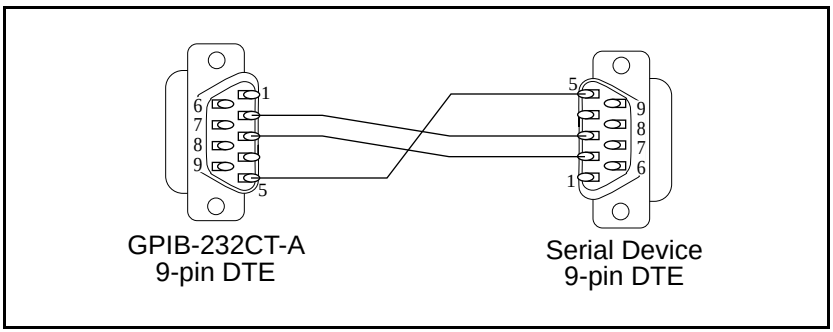

Figure E-8. Minimum Configuration for 9-pin DTE to 9-pin DTE

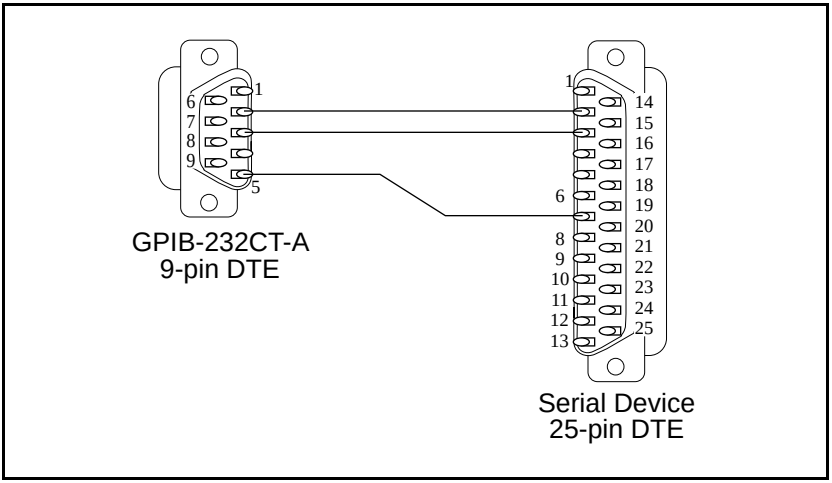

Figure E-9. Minimum Configuration for 9-pin DTE to 25-pin DTE

Custom Cables

If your application requires a custom cable, you can construct one if you have a thorough knowledge of the handshaking protocols involved. Review the RS-232 characteristics of your serial device and build the cable to properly connect the handshake lines of the two devices. If the documentation for your serial device does not provide a thorough explanation of its handshaking protocol, the ANSI/EIA-232-C standard is a good reference, provided your device conforms to the RS-232 protocol.

Caution: ***Although handshaking might not be required, it is best to use some form of handshaking to prevent loss of data.***

Appendix F
Customer Communication

For your convenience, this appendix contains forms to help you gather the information necessary to help us solve technical problems you might have as well as a form you can use to comment on the product documentation. Filling out a copy of the *Technical Support Form* before contacting National Instruments helps us help you better and faster.

National Instruments provides comprehensive technical assistance around the world. In the U.S. and Canada, applications engineers are available Monday through Friday from 8:00 a.m. to 6:00 p.m. (central time). In other countries, contact the nearest branch office. You may fax questions to us at any time.

Corporate Headquarters
(512) 795-8248
Technical support fax: (800) 328-2203
(512) 794-5678

Branch Offices	**Phone Number**	**Fax Number**
Australia	(03) 879 9422	(03) 879 9179
Austria	(0662) 435986	(0662) 437010-19
Belgium	02/757.00.20	02/757.03.11
Denmark	45 76 26 00	45 76 71 11
Finland	(90) 527 2321	(90) 502 2930
France	(1) 48 14 24 00	(1) 48 14 24 14
Germany	089/741 31 30	089/714 60 35
Italy	02/48301892	02/48301915
Japan	(03) 3788-1921	(03) 3788-1923
Mexico	95 800 010 0793	95 800 010 0793
Netherlands	03480-33466	03480-30673
Norway	32-848400	32-848600
Singapore	2265886 2265887	
Spain	(91) 640 0085	(91) 640 0533
Sweden	08-730 49 70	08-730 43 70
Switzerland	056/20 51 51	056/20 51 55
Taiwan	02 377 1200	02 737 4644
U.K.	0635 523545	0635 523154

Technical Support Form

Technical support is available at any time by fax. Include the information from your configuration form. Use additional pages if necessary.

Name _____

Company _____

Address _____

_____

Fax (___)_____ Phone (___)_____

Computer brand _____

Model _____ Processor _____

Operating system _____

Speed _____MHz RAM _____MB

Display adapter _____

Mouse _____yes _____no

Other adapters installed_

Hard disk capacity _____MB Brand _____

Instruments used _____

National Instruments hardware product model _____

Revision _____

Configuration _____

National Instruments software product _____

Version _____

Configuration _____

(continues)

The problem is _____

_____

_____

_____

_____

_____

_____

_____

List any error messages _____

_____

_____

_____

_____

_____

_____

_____

The following steps will reproduce the problem _____

_____

_____

_____

_____

_____

_____

GPIB-232CT-A Hardware and Software Configuration Form

Record the settings and revisions of your hardware and software on the line to the right of each item. Update this form each time you revise your software or hardware configuration, and use this form as a reference for your current configuration.

National Instruments Products

- GPIB-232CT-A Model and Revision _____
- NI-488.2 Software Version Number on Disk __
- RS-232 Port Configuration

  Standard RS-232 (factory setting) _____

  PC _____

Other Products

- Computer Make and Model _____
- Memory Capacity on Computer _____
- Operating System Version _____
- Number of GPIB Devices on Bus _____
- Other Hardware Devices in System _____
- Type of Monitor _____

Documentation Comment Form

National Instruments encourages you to comment on the documentation supplied with our products. This information helps us provide quality products to meet your needs.

Title: **Getting Started with Your GPIB-232CT-A and the NI-488.2™ Software for MS-DOS/Windows**

Edition Date: **March 1995**

Part Number: **320555B-01**

Please comment on the completeness, clarity, and organization of the manual.

(continues)

If you find errors in the manual, please record the page numbers and describe the errors.

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

Thank you for your help.

Name _____

Title _____

Company _____

Address _____

_____

Phone (_____) _____

Mail to: Technical Publications
National Instruments Corporation
6504 Bridge Point Parkway, MS 53-02
Austin, TX 78730-5039

Fax to: Technical Publications
National Instruments Corporation
MS 53-02
(512) 794-5678

Glossary

Prefix	Meaning	Value
n-	nano-	10^{-9}
m-	milli-	10^{-3}
k-	kilo-	10^{3}
M-	mega-	10^{6}

°	degrees
%	percent
A	amperes
AC	alternating current
ANSI	American National Standards Institute
ASCII	American Standard Code for Information Interchange
boot drive	Refers to the drive your computer reads when you power on or restart your computer
C	Celsius
CIC	Controller-in-Charge
CSA	Canadian Standards Association
CTS	Clear to Send
DAV	Data Valid
destination directory	Refers to the location on your hard disk where you install the NI-488.2 software
DIP	dual inline package
DLL	dynamic link library
DTR	Data Terminal Ready
ECIC	Enable Controller-In-Charge
EDVR	DOS error
EIA	Electronic Industries Association
EMI	electromagnetic interference
ENEB	Nonexistent GPIB board
EOI	end or identify

EOS	end of string
ERR	GPIB error
FCC	Federal Communications Commission
g	grams
GND	ground
GPIB	General Purpose Interface Bus
hex	hexadecimal
Hz	hertz
IBIC	Interface Bus Interactive Control
IEC	International Electrotechnical Commission
IEEE	Institute for Electrical and Electronic Engineers
in.	inches
I/O	Input/Output
IRQ	Interrupt Request
ISO	International Standards Organization
LED	light-emitting diode
m	meters
MB	megabytes of memory
oz	ounces
PC	personal computer
RAM	random-access memory
REN	Remote Enable
RTS	Request to Send
RXD	Receive Data
SC	System Controller
s	seconds
source directory	Refers to the NI-488.2 distribution disk
source disk	Refers to the NI-488.2 distribution disk
SRQ	Service Request
TIMO	Time limit exceeded
TSR	terminate-stay resident
TXD	Transmit Data
UL	Underwriters Laboratories
ULI	Universal Language Interface
V	volts
VAC	volts alternating current
VDC	volts direct current