Getting Started with Your GPIB-PCII/IIA and the NI-488.2™ Software for MS-DOS

June 1992 Edition

Part Number 320320-01

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,
Netherlands 03480-33466, Norway 32-848400, Spain (91) 640 0085,
Sweden 08-730 49 70, Switzerland 056/20 51 51, U.K. 0635 523545

Limited Warranty

The GPIB-PCII/IIA 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 B digital device. A Class B device is distinguishable from a Class A device by the appearance of an FCC ID number located on the Class B device.

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 B 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 interference from the equipment to radio and television reception in residential areas.

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.

Bescheinigung des Herstellers/Importeurs

Hiermit wird bescheinigt, daß die GPIB-PCII/IIA in Übereinstimmung mit den Bestimmungen der Vfg. 1046/1984 funk-entstört ist.

Der Deutsche Bundespost wurde das Inverkehrbringen dieses Gerätes angezeigt und die Berechtigung zur Überprüfung der Serie auf Bestimmungen eingeräumt.

Preface

This manual contains instructions for installing and configuring the National Instruments GPIB-PCII/IIA interface board and NI-488.2 MS-DOS handler. This manual is meant to be used with the *NI-488.2™ MS-DOS Software Reference Manual*, Part Number 320282-01.

Organization of This Manual

This manual is organized as follows:

- Chapter 1, *Introduction*, contains a picture of the GPIB-PCII/IIA interface board, lists the contents of your GPIB-PCII/IIA kit and optional equipment, and contains instructions for unpacking your GPIB-PCII/IIA.
- Chapter 2, *Installation and Elementary Programming Example*, contains instructions for installing your GPIB-PCII/IIA hardware and NI-488.2 software. It also contains an elementary programming example.
- Chapter 3, *Writing an Advanced Program Using NI-488.2 Routines*, contains an introduction to the NI-488.2 routines and step-by-step instructions for writing an NI-488.2 program. The end of the chapter contains some example programs.
- Appendix A, *Changing Hardware and Software Configuration Settings*, contains instructions for changing the configuration settings of your GPIB-PCII/IIA interface board.
- Appendix B, *Customer Communication*, contains forms for you to complete to facilitate communication with National Instruments concerning our products.

Conventions Used in This Manual

Throughout this manual, the following conventions are used to distinguish elements of text:

italic	Italic text denotes emphasis, a cross reference, or an introduction to a key concept.
`monospace`	Lowercase 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, and filenames, and for statements and comments taken from program code.
`italic monospace`	Italic lowercase 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, <Break>.
-	A hyphen between two or more key names enclosed in angle brackets denotes that you should simultaneously press the named keys–for example, <Ctrl-Alt-Del>.
<Enter>	Key names are capitalized.

Throughout this manual, the word *enter* is reserved to mean that the commands immediately following the word must be typed into the computer, and then executed by pressing the <Enter> key on the keyboard.

Abbreviations

The following are the abbreviations for units of measure used in this manual:

A	amperes
hex	hexadecimal
in.	inches
kbytes	1,000 bytes
MHz	megahertz
µsec	microseconds
sec	seconds

Acronyms

The following acronyms are used in this manual:

AC	alternating current
DMA	direct memory access
GPIB	General Purpose Interface (IEEE-488) Bus
EGA	enhanced graphics adapter
EMI	electromagnetic interference
IEEE-488	Institute of Electrical and Electronic Engineers Standard 488-1978, which defines the GPIB
I/O	input/output
PC	personal computer
VAC	volts alternating current

Related Documentation

The following document contains information that you may find helpful as you read this manual:

The *IBM Personal Computer Technical Reference* manual

Customer Communication

We appreciate communicating with the people who use our products. We are also very interested in hearing about the applications you develop using our products. To make it easy for you to communicate with us, this manual contains forms for you to complete. These forms are located in Appendix B, *Customer Communication*, at the back of this manual.

Contents

Figures

Tables

Chapter 1
Introduction

This chapter contains a picture of the GPIB-PCII/IIA interface board, lists the contents of your GPIB-PCII/IIA kit and optional equipment, and contains instructions for unpacking your GPIB-PCII/IIA.

The GPIB-PCII/IIA is a half-size IEEE-488 interface board for the IBM PC, PC/XT, PC AT and compatible computers (herein referred to as the PC). It is built using the custom-designed NAT4882 integrated circuit to achieve functionality and dependability previously unavailable.

The GPIB-PCII/IIA interface board combines the functionality of the National Instruments GPIB-PCII and GPIB-PCIIA interface boards. This interface board can be configured to function as either a GPIB-PCII or a GPIB-PCIIA, depending on the setting of the configuration switches.

Figure 1-1 shows the GPIB-PCII/IIA interface board in PCII mode.

Figure 1-1. GPIB-PCII/IIA Interface Board

To determine how your board was configured to function at the factory, check the identifying label on the mounting bracket of the interface board, beside the GPIB connector.

The default configuration of the board can also be determined by checking the assembly number located on the component side of the board. The assembly numbers for the board configurations are as follows:

GPIB-PCIIA	181065-01
GPIB-PCII	181065-02

Check that the setting of the GPIB-PC mode switch is consistent with the configuration indicated on the identifying label. For information on the GPIB-PC mode switch, refer to Appendix A, *Changing the Hardware and Software Configuration Settings*, later in this manual.

What Your Kit Should Contain

Your kit should contain the following components:

Component	Part Number
• GPIB-PCII/IIA interface board set for one of the following modes:	
- GPIB-PCIIA	181065-01
or	
- GPIB-PCII	181065-02
• For the GPIB-PCIIA, one of the following sets of diskettes:	
- 3.5 in. NI-488.2 Distribution Diskette for GPIB-PCIIA MS-DOS/Windows Handler, BASICA, QuickBASIC, BASIC, C & Universal Interfaces	422764-66
- Two 5.25 in. NI-488.2 Distribution Diskettes for GPIB-PCIIA MS-DOS/Windows Handler, BASICA, QuickBASIC, BASIC, C & Universal Interfaces	420761-66 and 420762-66
• For the GPIB-PCII, one of the following sets of diskettes:	
- 3.5 in. NI-488.2 Distribution Diskette for GPIB-PCII MS-DOS/Windows Handler, BASICA, QuickBASIC, BASIC, C & Universal Interfaces	422763-66
- Two 5.25 in. NI-488.2 Distribution Diskettes for GPIB-PCII MS-DOS/Windows Handler, BASICA, QuickBASIC, BASIC, C & Universal Interfaces	420759-66 and 420760-66
NI-488.2 MS-DOS Software Reference Manual	320282-01
Getting Started with Your GPIB-PCII/IIA and the NI-488.2 MS-DOS Handler	320320-01
Universal Language Interface Using HP-Style Calls	320135-90
Using Your GPIB Software with Microsoft Windows	320319-01

Make sure each of these items is in your kit. If any item is missing, contact National Instruments.

Optional Equipment

Equipment	Part Number
Single-Shielded GPIB Cables*:	
Type X1 Cable – 1 m	763001-01
Type X1 Cable – 2 m	763001-02
Type X1 Cable – 4 m	763001-03
Double-Shielded GPIB Cables*:	
Type X2 Cable – 1 m	763061-01
Type X2 Cable – 2 m	763061-02
Type X2 Cable – 4 m	763061-03
GPIB Connector Extender	760402-01

*To meet FCC emission limits for this Class B 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 in residential areas.

Unpacking Your GPIB-PCII/IIA

Follow these steps when unpacking your GPIB-PCII/IIA.

1. Verify that the pieces contained in the package you received match the kit parts list given earlier in this chapter. Do not remove the board from its plastic bag at this point.

2. Your GPIB-PCII/IIA board is shipped packaged in an antistatic plastic bag to prevent electrostatic damage to the board. Several components on the board can be damaged by electrostatic discharge. To avoid such damage in handling the board, touch the plastic bag to a metal part of your computer chassis before removing the board from the bag.

3. Remove the board from the bag and inspect the board for loose components or any other sign of damage. Notify National Instruments if the board appears damaged in any way. *Do not* install a damaged board into your computer.

Chapter 2
Installation and Elementary Programming Example

This chapter contains instructions for installing your GPIB-PCII/IIA hardware and NI-488.2 software. It also contains an elementary programming example.

Table 2-1 shows the default settings of the GPIB-PCII/IIA interface board.

Table 2-1. GPIB-PCII/IIA Default Settings

Parameters	GPIB-PCIIA	GPIB-PCII
Base I/O Address	2E1	2B8
Interrupt Line	7	7
DMA Channel	1	1

If these settings are known to conflict with your existing hardware, refer to Appendix A, *Changing Hardware and Software Configuration Settings*, to change these settings and proceed to *Step 1 - Install the Hardware.* Otherwise, record these settings in the *GPIB-PCII/IIA Hardware and Software Configuration Form* in Appendix B and proceed to *Step 1 - Install the Hardware.*

Step 1 – Install the Hardware

The following steps are general installation instructions. Consult the user manual or technical reference manual of your personal computer for specific instructions and warnings.

Install the GPIB-PCII/IIA board by completing the following steps:

1. Turn off your computer and all external devices, such as monitors or tape drives.
2. Unplug the power cord from the wall outlet.
3. Remove the top cover or access port to the I/O channel.
4. Remove the expansion slot cover on the back panel of the computer.
5. Insert the GPIB-PCII/IIA board into any unused slot with the IEEE-488 connector protruding out of the back panel.
6. Screw the GPIB-PCII/IIA mounting bracket to the back panel rail.
7. Verify that the interface board is securely installed.
8. Replace the retaining screw of the expansion slot cover if there is one.
9. Replace the cover on the computer.
10. Plug the power cord into the wall outlet.
11. Turn on your computer and external devices.

Step 2 - Install the Software

Complete the following steps to install the NI-488.2 software.

Note: If you are using Microsoft Windows 3 applications software that uses the GPIB, refer to *Using Your GPIB Software with Microsoft Windows* for a description of the GPIB Windows software package and instructions for software installation and configuration. You must have approximately 600 kilobytes of free disk space to install the NI-488.2 software files.

1. Insert the NI-488.2 distribution diskette into an unused drive.

2. Change to the drive containing the distribution diskette by entering the following command:

   ```
   x:
   ```

   where `x` is the letter of the drive containing the distribution diskette.

3. Install the software by entering the following command:

   ```
   install /q
   ```

   The `INSTALL` utility installs the NI-488.2 software quickly (`/q` minimizes user interaction). `INSTALL` creates a directory, `C:\GPIB-PC`, and copies the NI-488.2 software files to that directory. It also modifies the `C:\CONFIG.SYS` file to include the following line:

   ```
   device = \gpib-pc\gpib.com
   ```

   `INSTALL` then runs the GPIB-PCII/IIA hardware diagnostic program, `IBDIAG`. `IBDIAG` confirms that the hardware is functioning properly, and verifies that the DMA and interrupt configuration settings are set correctly. The default settings that the NI-488.2 handler uses are DMA channel 1 and no interrupt line.

   Direct memory access (DMA) is a method of moving data in a computer from one location to another that allows the GPIB-PCII/IIA interface board to operate at top speed. Operating the interface board without using DMA does not limit the functionality of the interface board, but it does limit its speed. The interface board often operates at a higher speed than the other instruments in a system without using DMA, but can only achieve top performance by using a DMA channel.

   The GPIB-PCII/IIA uses interrupts to notify the computer of certain specific events. However, if you configure the NI-488.2 software to run without interrupts, the software monitors the interface board to detect these events. (Using interrupts may result in slightly higher performance.)

4. Restart your computer.

If any one of the software installation steps fails, you may have to change a configuration setting. Refer to Appendix A, *Changing Hardware and Software Configuration Settings*, for more information on changing these settings.

If you do not have enough disk space to install the NI-488.2 software files, or if you want to change the default file names and installation settings used by the `INSTALL` utility, refer to Chapter 2 of the *NI-488.2 MS-DOS Software Reference Manual* for more information on `INSTALL`.

Step 3 - Test the Software Installation

Verify that the software is installed and configured correctly for your interface board by running the `IBTEST` utility. Complete the following steps:

1. Change to the `GPIB-PC` directory created by `INSTALL` by entering the following command:

   ```
   CD \GPIB-PC
   ```

2. Run `IBTEST` by entering the following command:

   ```
   IBTEST
   ```

If `IBTEST` fails, follow the instructions on your screen. If you need to change a hardware configuration setting, refer to Appendix A, *Changing the Hardware Configuration Settings.*

Step 4 - Install the GPIB Application Monitor

Install the GPIB Application Monitor by entering the following command:

```
APPMON
```

The GPIB Application Monitor is a memory-resident program that monitors all GPIB device driver activity in the background.

Step 5 - Trap GPIB Errors

Type `IBTRAP -err -dis` to signal `APPMON` to search for GPIB errors and pop up a screen if it finds an error.

Step 6 - Write a Program

The following program is a simple Microsoft QuickBASIC program written with the new NI-488.2 routines. This program reads a value from a Fluke 45 multimeter which is at GPIB address 6 and is connected to the PC through GPIB-PCII/IIA board number 0 (`gpib0`). Refer to the *NI-488.2 MS-DOS Software Reference Manual* for specific information about each routine.

```
REM $INCLUDE: 'C:\GPIB-PC\QBDECL.BAS'
DIM READING AS STRING*30
'
' Assert Interface Clear (IFC) for more than 100 µsec on GPIB
' interface board number zero (0).
'
CALL SendIFC(0)           'initialize bus.
'
' Selectively clear the multimeter:  address it to listen (GPIB
' address 6 on board number 0) and send it an SDC multiline
' command.
'
CALL DevClear(0, 6)       'clear the multimeter.
'
' Instruct the multimeter (board number 0, GPIB address 6) to reset
' itself (*RST); measure volts direct current (VDC); select range
' setting number 2 (RANGE 2); select trigger type number 2
' (TRIGGER 2); trigger and start a measurement (*TRG); send the
' measurement results back (VAL?).  Instruct the GPIB-PC board
' to append a linefeed character with EOI asserted as the last byte
' sent to the meter.
'
CALL Send(0, 6, "*RST; VDC; RANGE 2; TRIGGER 2; *TRG; VAL?", NLend)
'
' Read the last measured value from the multimeter (board 0, GPIB
' address 6) into a string.  Stop reading when the meter sends you
' the END message.
'
CALL Receive(0,6, Reading$, STOPend)
PRINT Reading$
' Call the IBONL function to disable the hardware and software.
' This step also frees the unit descriptor.
CALL IBONL (0,0)
END
```

The same results can be obtained using standard NI-488 functions, as shown in the following Microsoft QuickBASIC program.

```
REM $INCLUDE: 'C:\GPIB-PC\QBDECL.BAS'
DIM Reading AS STRING*30
CALL IBDEV(0,6,0,12,1,0,Fluke45%)     'open device
CALL IBCLR(Fluke45%)                  'clear device
CALL IBWRT(Fluke45%,"*RST; VDC; RANGE 2; TRIGGER 2; *TRG; VAL?")
CALL IBRD(Fluke45%,Reading$)
PRINT Reading$
CALL IBONL (Fluke45%,0)
END
```

A more detailed example program is given in Chapter 3, and several more examples are presented in the *NI-488.2 MS-DOS Software Reference Manual*.

Chapter 3
Writing an Advanced Program Using NI-488.2 Routines

This chapter contains an introduction to the NI-488.2 routines and step-by-step instructions for writing an NI-488.2 program. The end of the chapter contains some example programs.

You can take full advantage of the IEEE-488.2-1987 standard by using the NI-488.2 routines. These routines are completely compatible with the controller commands and protocols defined in IEEE-488.2. The NI-488.2 routines are described more completely in the *NI-488.2 MS-DOS Software Reference Manual*.

The NI-488.2 routines are easy to learn and use. Only a few routines are needed for most application programs.

Interface Boards

The NI-488.2 handler supports two GPIB-PCII/IIA interface boards. These boards are referenced by number from your application program. The reference number is zero (0) for the first GPIB-PCII/IIA board and one (1) for the second board. If you installed two boards in your computer, and you do not know which board is 0 and which board is 1, run the configuration utility, `IBCONF`. `IBCONF` will show you the relationship between the board number and the base address of the board, thereby identifying the board by its base address. Refer to Chapter 2 in the *NI-488.2 MS-DOS Software Reference Manual* for additional information about running and using `IBCONF`.

Calling Syntax

The calling syntax for NI-488.2 routines varies slightly according to the language used. For the sake of brevity, the syntax shown in this manual is for Microsoft QuickBASIC and C. Syntax for other languages is described in the *NI-488.2 MS-DOS Software Reference Manual*, and in associated language interface manuals.

Steps for Writing an NI-488.2 Program

This section demonstrates how to use the NI-488.2 routines. An in-depth example program written in Microsoft QuickBASIC is developed step-by-step. The example program configures a Fluke model 45 digital multimeter, reads back 10 voltage measurements, and computes the average of these measurements. Error detection and reporting is handled explicitly in the program as opposed to using the Applications Monitor. (In-line error checking has a much faster execution time.) For this example, it is assumed that you are using GPIB-PCII/IIA interface board number zero (0). In addition, several more of the NI-488.2 routines are used than were shown in Chapter 2. This program finds and identifies all the Listeners on the bus prior to controlling the Fluke 45 instrument.

Step 1 – Preparation

The first step in writing the program is to load in the definitions of the NI-488.2 routines from a file that is on your distribution diskette. Include the declaration file for the NI-488.2 QuickBASIC language interface as follows:

```
REM $INCLUDE: 'C:\GPIB-PC\QBDECL.BAS'
```

To check for errors, declare a subroutine to handle all possible GPIB errors.

```
DECLARE SUB gpiberr (msg$)
```

The following arrays are used to hold lists of IEEE-488 addresses, which are used to find all of the Listeners on the bus.

```
DIM instruments% (31)
DIM result% (31)
DIM Reading AS STRING*30
```

Step 2 – Initialization

Initialize the IEEE-488 bus and the GPIB-PCII/IIA Controller circuitry so that the IEEE-488 interface for each device is quiescent, and so that the GPIB-PCII/IIA is Controller-In-Charge and is in the Active Controller State (CACS). This is accomplished with the `SendIFC` procedure. The first and only argument of `SendIFC` is the GPIB-PCII/IIA interface board number.

```
CALL SendIFC(0)
IF IBSTA% < 0 THEN
      CALL gpiberr("SendIFC(0) error")
      CALL IBONL (0,0)
      STOP
END IF
```

Step 3 – Find All Listeners

Find all of the instruments attached to the IEEE-488 bus. Create an array that contains all of the IEEE-488 primary addresses that could possibly be connected to the IEEE-488 bus. Assume that you are only attaching devices with primary addresses (0 to 30) to the bus. The end of the list of addresses must be marked with the `NOADDR` constant, which is defined in the `QBDECL.BAS` file that was included at the beginning of this program.

```
FOR K% = 0 to 30
      instruments%(K%) = K%
NEXT K%
instruments%(31) = NOADDR
```

Next find out which, if any, device or devices are connected. This is easily accomplished by using the `FindLstn` procedure. The first argument is the GPIB-PCII/IIA board number, the second argument is the list of instruments that was created above, the third argument is a list of instrument addresses that the procedure actually found, and the last argument is the maximum number of devices that the procedure may find (that is, it must stop if it reaches the limit).

```
PRINT "Finding all listeners on the bus..."

CALL FindLstn(0, instruments%(), result%(), 31)
IF IBSTA% < 0 THEN
      CALL gpiberr("FindLstn error")
      CALL IBONL (0,0)
      STOP
END IF
num.listeners% = ibcnt%-1

PRINT "No. of instruments found = ", num.listeners%
```

Step 4 – Identify the Instrument

Now that you have determined the addresses of the devices, you should send an identification query to each device for identification. For this example, assume that all of the instruments are IEEE-488.2 compatible and can accept the 488.2 identification query, `*IDN?`. In addition, assume that `FindLstn` found the GPIB-PCII/IIA interface board at primary address 0 (which is where it is by default) and, therefore, you can skip the first entry in the `result` array.

```
FOR K% = 1 TO num.listeners%
    CALL Send(0, result%(K%), "*IDN?", NLend)
    IF IBSTA% < 0 THEN
          CALL gpiberr("Send error")
          CALL IBONL (0,0)
          STOP
    END IF

    CALL Receive(0,result%(K%),Reading$,STOPend)
    IF IBSTA% < 0 THEN
          CALL gpiberr("Receive error")
          CALL IBONL (0,0)
          STOP
    END IF

    pad% = result%(K%) AND &HFF
    PRINT "The instrument at address ";pad%; " is: ",
        LEFT$(Reading$,IBCNT%)

    IF LEFT$(Reading$,9)="FLUKE, 45" THEN
        fluke% = result%(K%)
        PRINT "**** We found the Fluke 45 ****"
        GOTO found
    END IF
NEXT K%
```

```
PRINT "Did not find the Fluke!"
CALL IBONL (0,0)
STOP

found:
```

The constant NLend signals that the new line character with EOI will be automatically appended to the data.

The constant STOPend indicates that the read is stopped when EOI is detected.

Step 5 – Initialize the Instrument

Now that you have found the multimeter, clear it. Initialization of the devices attached to the IEEE-488 bus is accomplished with the DevClear procedure. The first argument is the GPIB-PCII/IIA interface board number. The second argument is the IEEE-488 address of the multimeter.

```
CALL DevClear(0, fluke%)
IF IBSTA% < 0 THEN
        CALL gpiberr("DevClear error")
        CALL IBONL (0,0)
        STOP
END IF
```

Send the IEEE-488.2 Reset command to the meter.

```
CALL Send(0, fluke%, "*RST", NLend)
IF IBSTA% < 0 THEN
        CALL gpiberr("Send *RST error")
        CALL IBONL (0,0)
        STOP
END IF
```

Step 6 – Configure the Instrument

After initialization, the instrument is ready to receive instructions. To configure the multimeter, device-specific commands are sent using the Send routine. The first argument of the Send routine is the GPIB interface board number. The second argument is the IEEE-488 address of the multimeter. The third argument is a string of bytes to send to the multimeter.

The bytes in this example instruct the meter to measure volts alternating current (VAC) using auto-ranging (AUTO), to wait for a trigger from the Controller before starting a measurement (TRIGGER 2), and to assert the IEEE-488 Service Request signal line, SRQ, when the measurement has been completed and the meter is ready to send the result (*SRE 16). The last argument, NLend, is a constant that is defined in QBDECL.BAS which tells Send to append a linefeed character with EOI asserted to the end of the message sent to the multimeter.

The calls to actually trigger the instrument and receive the measurement are placed in a loop to repeat 10 times.

```
CALL Send(0, fluke%, "VAC; AUTO; TRIGGER 2; *SRE 16", NLend)
IF IBSTA% < 0 THEN
      CALL gpiberr("Send setup error")
      CALL IBONL (0,0)
      STOP
END IF
SUM = 0
FOR M% = 1 TO 10
```

Step 7 – Trigger the Instrument

In the previous step, the multimeter was instructed to wait for a trigger before conducting a measurement. Now send a trigger command to the multimeter. You could use the Trigger routine to accomplish this, but because the Fluke 45 is IEEE-488.2 compatible, you can just send it the trigger command, *TRG. The other command, VAL1?, instructs the meter to send the next triggered reading to its IEEE-488.1 output buffer.

```
CALL Send(0, fluke%, "*TRG; VAL?", NLend)
IF IBSTA% < 0 THEN
      CALL gpiberr("Send trigger error")
      CALL IBONL (0,0)
      STOP
END IF
```

Step 8 – Wait for the Instrument to Complete the Measurement

After the meter is triggered, it will take a measurement and display it on its front panel. When this is finished, the meter asserts SRQ. You can detect the assertion of SRQ using either the `TestSRQ` or `WaitSRQ` routine. If you have processing that you want to do while you are waiting for the measurement, you would use `TestSRQ`. For this example, there is nothing in particular to do at this point, so you can use the `WaitSRQ` routine. The first argument to `WaitSRQ` is the GPIB-PCII/IIA board number. The second argument is a flag returned by `WaitSRQ` that indicates whether or not SRQ is asserted.

```
CALL WaitSRQ(0, SRQasserted%)
IF SRQasserted% = 0 THEN
      CALL gpiberr("WaitSRQ error")
      CALL IBONL (0,0)
      STOP
END IF
```

After you have detected SRQ, poll the meter to ascertain its status. This is accomplished by using the `ReadStatusByte` procedure. The first argument for this procedure is the GPIB-PCII/IIA board number, the second argument is the IEEE-488 address of the instrument, and the last argument is a variable that `ReadStatusByte` uses to store the status byte of the instrument.

```
CALL ReadStatusByte(0, fluke%, status%)
IF IBSTA% < 0 THEN
      CALL gpiberr("ReadStatusByte error")
      CALL IBONL (0,0)
      STOP
END IF
```

After you have obtained the status byte, you must check to see if the meter has a message to send. You can find this out by checking the message available (MAV) bit, bit 4, in the status byte.

```
IF (status% AND &H010) <> &H010 THEN
      CALL gpiberr("Improper status byte")
      PRINT "Status byte: "; status%
      CALL IBONL (0,0)
      STOP
END IF
```

Step 9 – Read the Measurement

Use the `Receive` function to read the measurement over the IEEE-488 bus. Again, the first argument is the GPIB interface board number, and the second argument is the IEEE-488 address of the multimeter. The third argument is a string into which the `Receive` function places the data bytes from the multimeter. The last argument specifies that the `Receive` function is to terminate upon receiving a byte accompanied with the `END` message. The loop is then repeated. The average value is printed at the end of the 10 measurements.

```
CALL Receive(0, fluke%, Reading$, STOPend)
IF IBSTA% < 0 THEN
      CALL gpiberr("Receive error")
      CALL IBONL (0,0)
      STOP
END IF

PRINT "Reading: "; Reading$
SUM = SUM + VAL(Reading$)
NEXT M%
PRINT "The average of the 10 readings is", SUM/10
CALL IBONL (0,0)
END
```

The Complete Application Program

The following program combines the previous steps into one program.

```
REM $INCLUDE: 'C:\GPIB-PC\QBDECL.BAS'

DECLARE SUB gpiberr (msg$)

DIM instruments% (31)
DIM result% (31)
DIM Reading AS STRING*30

CLS
CALL SendIFC(0)
IF IBSTA% < 0 THEN
      CALL gpiberr("SendIFC(0) error")
      CALL IBONL (0,0)
      STOP
END IF
```

```
FOR K% = 0 to 30
       instruments%(K%) = K%
NEXT K%
instruments%(31) = NOADDR

PRINT "Finding all listeners on the bus..."

CALL FindLstn(0, instruments%(), result%(), 31)
IF IBSTA% < 0 THEN
     CALL gpiberr("FindLstn error")
     CALL IBONL (0,0)
     STOP
END IF
num.listeners% = ibcnt%-1

PRINT "No. of instruments found = ", num.listeners%

FOR K% = 1 TO num.listeners%
    CALL Send(0, result%(K%), "*IDN?", NLend)
    IF IBSTA% < 0 THEN
         CALL gpiberr("Send error")
         CALL IBONL (0,0)
         STOP
    END IF

    CALL Receive(0,result%(K%),Reading$,STOPend)
    IF IBSTA% < 0 THEN
         CALL gpiberr("Receive error")
         CALL IBONL (0,0)
         STOP
    END IF

    pad% = result%(K%) AND &HFF
    PRINT "The instrument at address"; pad%; "is:",
        LEFT$(Reading$,IBCNT%)
    IF LEFT$(Reading$,9)="FLUKE, 45" THEN
        fluke% = result%(K%)
        PRINT "**** We found the Fluke 45 ****"
        GOTO found
    END IF
NEXT K%
PRINT "Did not find the Fluke!"
CALL IBONL (0,0)
STOP

found:

CALL DevClear(0, fluke%)
IF IBSTA% < 0 THEN
     CALL gpiberr("DevClear error")
     CALL IBONL (0,0)
     STOP
END IF
```

```
CALL Send(0, fluke%, "*RST", NLend)
IF IBSTA% < 0 THEN
      CALL gpiberr("Send *RST error")
      CALL IBONL (0,0)
      STOP
END IF

CALL Send(0, fluke%, "VAC; AUTO; TRIGGER 2; *SRE 16", NLend)
IF IBSTA% < 0 THEN
      CALL gpiberr("Send setup error")
      CALL IBONL (0,0)
      STOP
END IF
SUM = 0
FOR M% = 1 TO 10

CALL Send(0, fluke%, "*TRG; VAL?", NLend)
IF IBSTA% < 0 THEN
      CALL gpiberr("Send trigger error")
      CALL IBONL (0,0)
      STOP
END IF

CALL WaitSRQ(0, SRQasserted%)
IF SRQasserted% = 0 THEN
      CALL gpiberr("WaitSRQ error")
      CALL IBONL (0,0)
      STOP
END IF

CALL ReadStatusByte(0, fluke%, status%)
IF IBSTA% < 0 THEN
      CALL gpiberr("ReadStatusByte error")
      CALL IBONL (0,0)
      STOP
END IF

IF (status% AND &H010) <> &H010 THEN
      CALL gpiberr("Improper status byte")
      PRINT "Status byte: "; status%
      CALL IBONL (0,0)
      STOP
END IF

CALL Receive(0, fluke%, Reading$, STOPend)
IF IBSTA% < 0 THEN
      CALL gpiberr("Receive error")
      CALL IBONL (0,0)
      STOP
END IF
```

```
Reading$ = Left%(Reading$, IBCNT%)
PRINT "Reading: "; Reading$
SUM = SUM + VAL(Reading$)
NEXT M%
PRINT "The average of the 10 readings is", SUM/10
CALL IBONL (0,0)
END
```

The Error Handling Subroutine

```
SUB gpiberr(msg$) STATIC

LOCATE 15,1
PRINT msg$

PRINT "IBSTA=&H"; HEX$(IBSTA%); " < ";
IF IBSTA% AND EERR THEN PRINT " ERR";
IF IBSTA% AND TIMO THEN PRINT " TIMO";
IF IBSTA% AND EEND THEN PRINT " END";
IF IBSTA% AND SRQI THEN PRINT " SRQI";
IF IBSTA% AND RQS THEN PRINT " RQS";
IF IBSTA% AND CMPL THEN PRINT " CMPL";
IF IBSTA% AND LOK THEN PRINT " LOK";
IF IBSTA% AND RREM THEN PRINT " REM";
IF IBSTA% AND CIC THEN PRINT " CIC";
IF IBSTA% AND AATN THEN PRINT " ATN";
IF IBSTA% AND TACS THEN PRINT " TACS";
IF IBSTA% AND LACS THEN PRINT " LACS";
IF IBSTA% AND DTAS THEN PRINT " DTAS";
IF IBSTA% AND DCAS THEN PRINT " DCAS";
PRINT ">                              "

PRINT "IBERR="; IBERR%;
IF IBERR% = EDVR THEN PRINT " EDVR <DOS Error>       "
IF IBERR% = ECIC THEN PRINT " ECIC <Not CIC>         "
IF IBERR% = ENOL THEN PRINT " ENOL <No Listener>     "
IF IBERR% = EADR THEN PRINT " EADR <Address error>   "
IF IBERR% = EARG THEN PRINT " EARG <Invalid argument>"
IF IBERR% = ESAC THEN PRINT " ESAC <Not Sys Ctrlr>   "
IF IBERR% = EABO THEN PRINT " EABO <Op. aborted>     "
IF IBERR% = ENEB THEN PRINT " ENEB <No GPIB board>   "
IF IBERR% = EOIP THEN PRINT " EOIP <Async I/O in prg>"
IF IBERR% = ECAP THEN PRINT " ECAP <No capability>   "
IF IBERR% = EFSO THEN PRINT " EFSO <File sys. error> "
IF IBERR% = EBUS THEN PRINT " EBUS <Command error>   "
IF IBERR% = ESTB THEN PRINT " ESTB <Status byte lost>"
IF IBERR% = ESRQ THEN PRINT " ESRQ <SRQ stuck on>    "
IF IBERR% = ETAB THEN PRINT " ETAB <Table Overflow>  "

PRINT "IBCNT="; IBCNT%; "                            "

END SUB
```

Compiling and Linking

To create an executable program, complete the following steps:

1. Copy the QuickBASIC 4.5 language interface (`QBIB.OBJ`) from the `C:\GPIB-PC` directory, which was set up during installation, into your QuickBASIC directory (`C:\QB45` in this example).

2. Compile the application program (assume the example program you just created is in a file called `ACVOLTS.BAS`).

   ```
   bc /o acvolts.bas;            <Enter>
   ```

3. Link in the language interface by entering the following command:

   ```
   link acvolts.obj+qbib.obj;    <Enter>
   ```

   `link` creates an executable file named `ACVOLTS.EXE.`

4. If you have a Fluke 45 DVM, run the program by entering the following command:

   ```
   acvolts                       <Enter>
   ```

The Complete Application Program in C

The same program written in Microsoft C is shown below for those of you who are programming in C.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "c:\gpib-pc\decl.h"

#define  MAVbit   0x10   /* Position of the Message Available
bit.*/

   char buffer[101];
   int loop, m;
   int num_listeners;
   double sum;
   unsigned int instruments[32], result[32], fluke;
   unsigned int statusByte;
   int SRQasserted;
```

```
main() {
   /*  Our board needs to be the Controller-In-Charge in order to
    *  perform the Find All Listeners protocol.
    */
   SendIFC(0);
   if (ibsta < 0) {
        gpiberr ("SendIFC(0) error");
        ibonl (0,0);
        exit(1);
   }


   /*  Create an array with all of the valid GPIB primary
addresses.
    *  This array will be given to the Find All Listeners protocol.
    */
   for (loop = 0; loop <= 30; loop++) {
      instruments[loop] = loop;
   }
   instruments[31] = NOADDR;       /* Mark the end of the array.*/


   /*  Find all of the listeners on the bus.
    */
   printf("Finding all listeners on the bus...\n");

   FindLstn(0, instruments, result, 31);
   if (ibsta < 0) {
        gpiberr("FindLstn error");
        ibonl (0,0);
        exit(1);
   }

   num_listeners = ibcnt - 1;

   printf("Number of instruments found = %d\n", num_listeners);


   /*
    *  Now send the *IDN? command to each of the devices that we
    *  found.
    *
    *  Our board is at address 0 by default.  Our board does not
    *  respond to *IDN?, so skip it.
    */
   for (loop = 1; loop <= num_listeners; loop++) {
      Send(0, result[loop], "*IDN?", 5L, NLend);
      Receive(0, result[loop], buffer, 100L, STOPend);
      buffer[ibcnt] = '\0';

      printf("The instrument at address %d, %d is a %s\n",
        GetPAD(result[loop]),
                GetSAD(result[loop]), buffer);
```

```
   if (strncmp(buffer, "FLUKE, 45", 9) == 0) {
      fluke = result[loop];
      printf("**** We found the Fluke ****\n");
      break;
   }
}

if (loop > num_listeners) {
     printf("Did not find the Fluke!\n");
     ibonl (0,0);
     exit(1);
}

/*  Reset the Fluke.
 */
DevClear(0, fluke);
if (ibsta < 0) {
     gpiberr("DevClear error");
     ibonl (0,0);
     exit(1);
}
Send(0, fluke, "*RST", 4L, NLend);


/*  Setup for a test.  Allow the Fluke to assert
 *  SRQ when it has a message to send.
 */
Send(0, fluke, "VAC; AUTO; TRIGGER 2; *SRE 16", 29L, NLend);
if (ibsta < 0) {
   gpiberr("Send setup error");
   ibonl (0,0);
   exit(1);
}

sum = 0.0;
for (m=0; m < 10 ; m++) {

     /*  Trigger the Fluke.
     */
      Send(0, fluke, "*TRG; VAL?", 10L, NLend);
      if (ibsta < 0) {
         gpiberr("Send trigger error");
         ibonl (0,0);
         exit(1);
      }

     /*  Wait for the Fluke to assert SRQ, meaning it is ready
      *  with the measurement.
     */
     WaitSRQ(0, &SRQasserted);
     if (!SRQasserted) {
         printf("SRQ is not asserted.  The Fluke is not
    ready.\n");
```

```
            ibonl (0,0);
            exit(1);
         }
         /*  Read its status byte.  Be sure that the MAV
          *  (Message Available) bit is set.
          */
          ReadStatusByte(0, fluke, &statusByte);

          if ((ibsta & ERR) || !(statusByte & MAVbit)) {
              printf("The Serial Poll failed.  ibsta = 0x%x  ");
              printf(" statusByte = 0x%x\n", ibsta, statusByte);
              ibonl (0,0);
              exit(1);
          }

         /*  Read the measurement.
          */
          Receive(0, fluke, buffer, 100L, STOPend);
          if (ibsta < 0) {
             gpiberr("Receive error");
             ibonl (0,0);
             exit(1);
          }

          buffer[ibcnt] = '\0';

          printf("Reading :  %s\n", buffer);
          sum = sum + atof(buffer);
   }
   printf("   The average of the 10 readings is : %f\n", sum/10);

   sendIFC (0);

   ibonl (0,0);

}


gpiberr(char *msg)
{

   printf ("%s\n", msg);
```

```
   printf ( "ibsta=&H%x ", ibsta, "< ");
   if (ibsta & ERR ) printf (" ERR");
   if (ibsta & TIMO) printf (" TIMO");
   if (ibsta & END ) printf (" END");
   if (ibsta & SRQI) printf (" SRQI");
   if (ibsta & RQS ) printf (" RQS");
   if (ibsta & CMPL) printf (" CMPL");
   if (ibsta & LOK ) printf (" LOK");
   if (ibsta & REM ) printf (" REM");
   if (ibsta & CIC ) printf (" CIC");
   if (ibsta & ATN ) printf (" ATN");
   if (ibsta & TACS) printf (" TACS");
   if (ibsta & LACS) printf (" LACS");
   if (ibsta & DTAS) printf (" DTAS");
   if (ibsta & DCAS) printf (" DCAS");
   printf (">\n");

   printf ("iberr= %d", iberr);
   if (iberr == EDVR) printf (" EDVR <DOS Error>\n");
   if (iberr == ECIC) printf (" ECIC <Not CIC>\n");
   if (iberr == ENOL) printf (" ENOL <No Listener>\n");
   if (iberr == EADR) printf (" EADR <Address error>\n");
   if (iberr == EARG) printf (" EARG <Invalid argument>\n");
   if (iberr == ESAC) printf (" ESAC <Not Sys Ctrlr>\n");
   if (iberr == EABO) printf (" EABO <Op. aborted>\n");
   if (iberr == ENEB) printf (" ENEB <No GPIB board>\n");
   if (iberr == EOIP) printf (" EOIP <Async I/O in prg>\n");
   if (iberr == ECAP) printf (" ECAP <No capability>\n");
   if (iberr == EFSO) printf (" EFSO <File sys. error>\n");
   if (iberr == EBUS) printf (" EBUS <Command error>\n");
   if (iberr == ESTB) printf (" ESTB <Status byte lost>\n");
   if (iberr == ESRQ) printf (" ESRQ <SRQ stuck on>\n");
   if (iberr == ETAB) printf (" ETAB <Table Overflow>\n");

   printf ("ibcnt= %d\n", ibcntl);
   printf ("\n");
}
```

Helpful Hint

Now that your software and hardware are installed and you have used some NI-488.2 routines, read the *NI-488.2 MS-DOS Software Reference Manual* for a description of all of the NI-488.2 routines available for the GPIB-PCII/IIA. After reading about these functions and their capabilities, practice using them with your programmable instrument or device in an interactive environment using the `IBIC` program described in Chapter 6 of the *NI-488.2 MS-DOS Software Reference Manual*.

Appendix A
Changing Hardware and Software Configuration Settings

This appendix contains instructions for changing the configuration settings of your GPIB-PCII/IIA interface board. You only need to change the configuration settings if the default settings conflict with another board in your system.

Step 1 – Hardware Configuration

Figure A-1 shows the location of the GPIB-PCII/IIA configuration jumpers and switches.

Figure A-1. GPIB-PCII/IIA Parts Locator Diagram

GPIB-PC Mode Selection

The GPIB-PCII/IIA interface board is set at the factory to function as either a GPIB-PCII or a GPIB-PCIIA board. Using the board in its default configuration ensures that the software configuration is consistent with the hardware. To change the GPIB-PC mode of the board, use switch 9 in switch block U2. To select GPIB-PCII mode, push the switch down on the side labelled PCII. To select GPIB-PCIIA mode, push the switch down on the side labelled PCIIA.

Figure A-2 shows the GPIB-PC mode selection switch set for GPIB-PCII mode and GPIB-PCIIA mode.

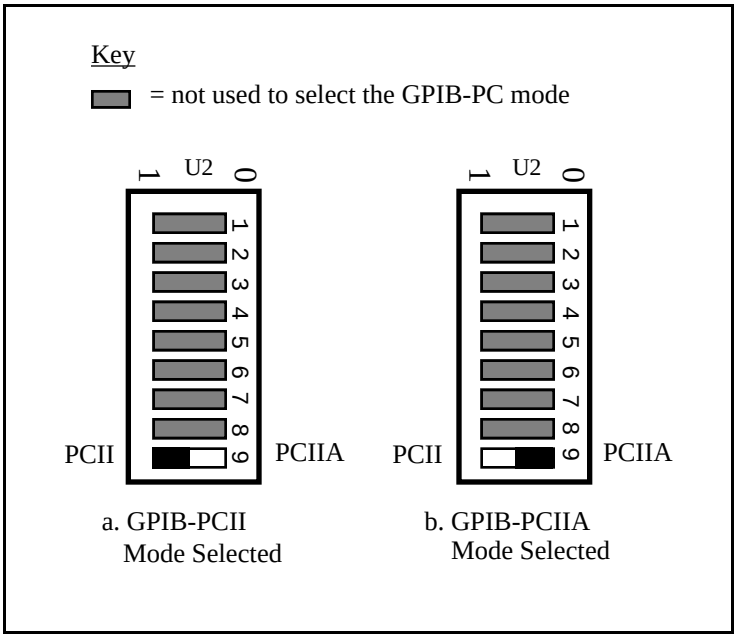


Figure A-2. GPIB-PC Mode Selection Settings

Switch and Jumper Settings

Table A-1 shows the factory settings and available configurations for the switches and jumpers on the GPIB-PCII/IIA in GPIB-PCII mode. Table A-2 shows the factory settings and available configurations for the switches and jumpers in GPIB-PCIIA mode.

Table A-1. Factory Default Settings and Available Configurations for GPIB-PCII Mode

GPIB-PCII	Default	Available
Base I/O Address (hex)	2B8	000 to 3F8
DMA Channel	1	1, 2, 3, and Not Used
Interrupt Line (IRQ)	7	2, 3, 4, 5, 6, 7, and Not Used
7210/9914 Mode	7210	7210 and 9914
Shield Ground	Connected	Connected, disconnected

Table A-2. Factory Default Settings and Available Configurations for GPIB-PCIIA Mode

GPIB-PCIIA	Default	Available
Base I/O Address (hex)	2E1	2E1, 22E1, 42E1, and 62E1
DMA Channel	1	1, 2, 3, and Not Used
Interrupt Line (IRQ)	7	2, 3, 4, 5, 6, 7, and Not Used
7210/9914 Mode	7210	7210 and 9914
Shield Ground	Connected	Connected, disconnected

If you make any changes to these settings, record the new settings in the *GPIB-PCII/IIA Hardware and Software Configuration Form* in Appendix B and go on to *Step 2 - Software Configuration* at the end of this appendix.

7210/9914 Mode Selection

The GPIB-PCII/IIA can emulate many IEEE-488 interface boards that use the TI 9914A GPIB Controller chip as well as the 7210 chip. Use switch 8 in switch block U2 to select either TI 9914A mode or NEC 7210 mode. For normal operation with National Instruments software, leave this switch in the 7210 position. Figure A-3 shows the 7210/9914 mode selection switch settings for 7210 mode and 9914 mode.

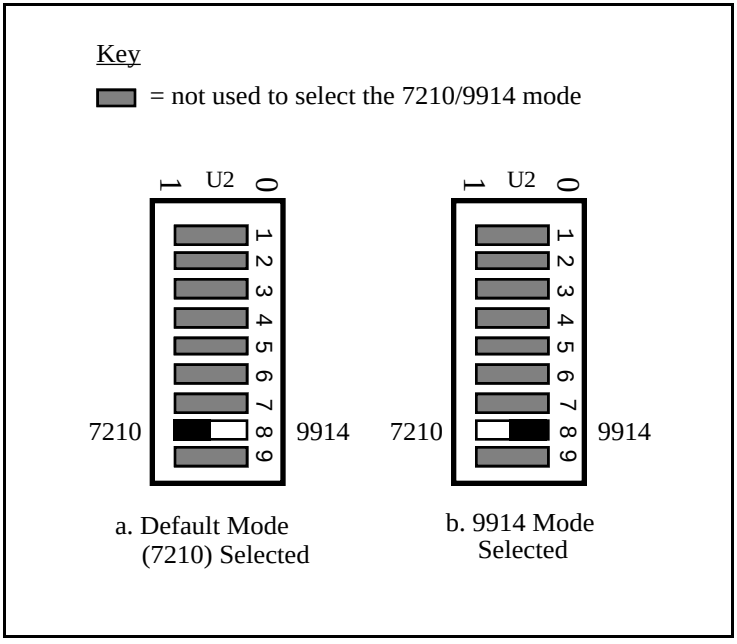


Figure A-3. 7210/9914 Mode Selection Settings

Base I/O Address Selection

GPIB-PCII Mode

The GPIB-PCII base I/O address is set using the switches at position U2. The switch block is used to set the address for address lines A3 through A9. The addresses are in a consecutive block of eight beginning on any multiple of 8 between 000 and 3F8 hex. For example, for the default address, 2B8 hex, the GPIB-PCII uses the address space 2B8 through 2BF hex.

Press the side marked 1 to select a binary value of 1 for the corresponding address bit. Press the 0 side of the switch to select a value of 0 for the corresponding address bit.

To change the base I/O address, press each switch to the desired position, then check each switch to make sure it is pressed down all the way.

Figure A-4 shows two possible switch settings. In this figure, the black side of the switch is the side you press down. Each of the address selections shows how the base I/O address was calculated from the switch positions.

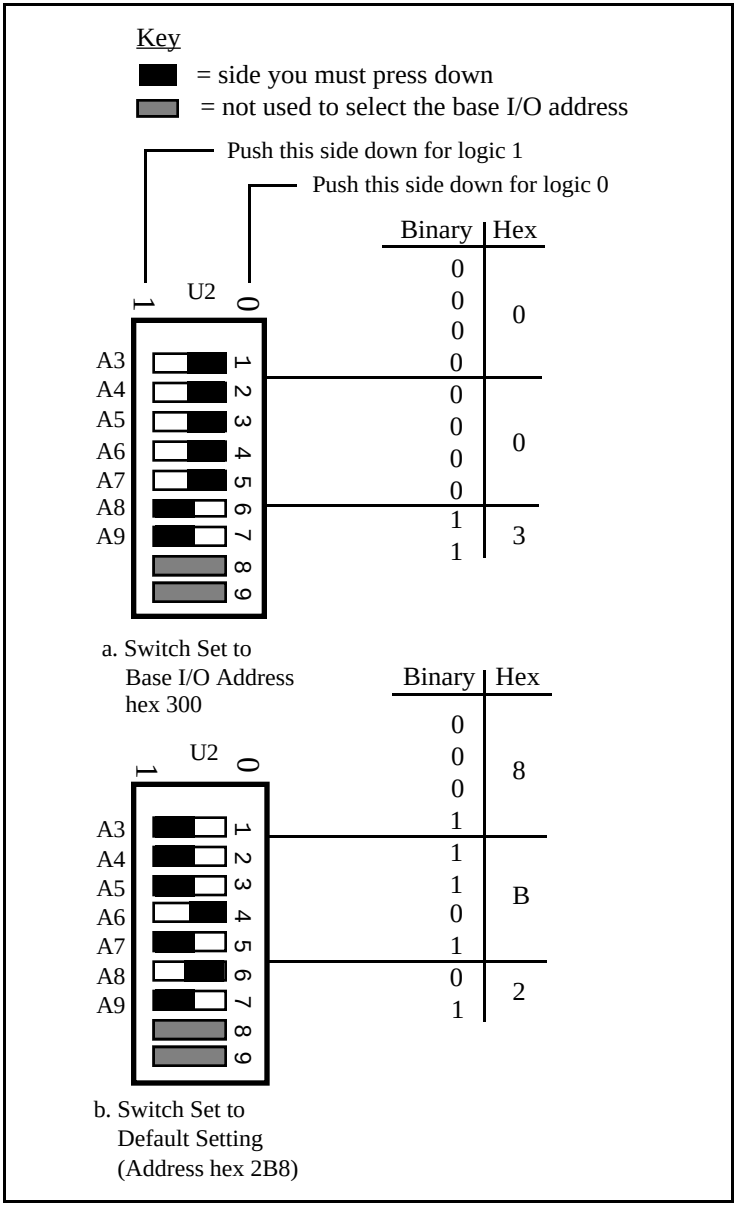


Figure A-4. Base I/O Address Switch Settings for GPIB-PCII

GPIB-PCIIA Mode

The GPIB-PCIIA base I/O address is set using switches 4 and 5 of the switch block at U2. The four possible base I/O addresses are 2E1, 22E1, 42E1, and 62E1 hex.

Figure A-5 shows the switch settings for the four possible base I/O addresses and the address space used for each setting. Figure A-5a shows how the base I/O address was calculated from the switch positions.

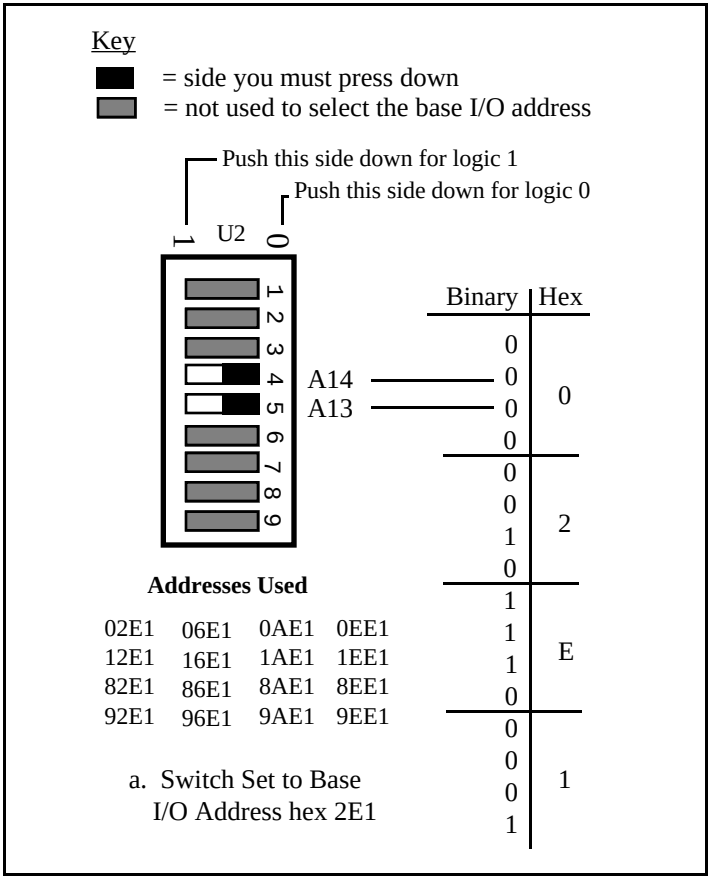
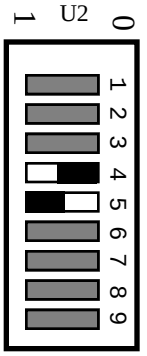


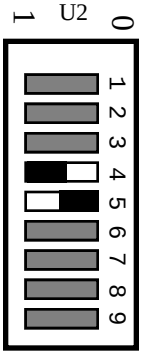
Figure A-5. Base I/O Address Switch Settings for GPIB-PCIIA (continues)



Addresses Used

22E1	26E1	2AE1	2EE1
32E1	36E1	3AE1	3EE1
A2E1	A6E1	AAE1	AEE1
B2E1	B6E1	BAE1	BEE1

b. Switches Set to Base I/O Address hex 22E1



Addresses Used

42E1	46E1	4AE1	4EE1
52E1	56E1	5AE1	5EE1
C2E1	C6E1	CAE1	CEE1
D2E1	D6E1	DAE1	DEE1

c. Switches Set to Base I/O Address hex 42E1

Addresses Used

62E1	66E1	6AE1	6EE1
72E1	76E1	7AE1	7EE1
E2E1	E6E1	EAE1	EEE1
F2E1	F6E1	FAE1	FEE1

d. Switches Set to Base I/O Address hex 62E1

Figure A-5. Base I/O Address Switch Settings (continued)

To change the base I/O address, locate the switches at U2, press each switch to the desired position, and check each switch to make sure it is pressed down all the way.

If you change the base I/O address setting from the default setting, record the new setting in the *GPIB-PCII/IIA Hardware and Software Configuration Form* in Appendix B and proceed to *Step 2 - Software Configuration* at the end of this appendix.

Possible Conflicts

Table A-3 lists some of the I/O addresses used by other PC plug-in interface boards and adapters. This is not a complete list, but it may help in determining possible address conflicts. Symptoms of I/O address conflicts vary widely. At one extreme, conflicts can prevent the computer from booting. At the other extreme, conflicts can cause problems that do not surface until a considerable amount of time has elapsed. When they do surface, the problem can exhibit itself simply as strange behavior.

National Instruments has made every effort to select a default base I/O address that will work. However, because of the numerous different interface boards available for use in the PC, it is not possible to select a base I/O address that is guaranteed to work in all systems.

Note: In GPIB-PCII mode, eight consecutive addresses are used, while in GPIB-PCIIA mode, sixteen addresses spread throughout the upper address space are used.

Table A-3. PC I/O Address Map

I/O Address Range (Hex)	Device
100 to 1EF 1F0 to 1F8	IBM PC Fixed Disk
200 to 20F 208 210 to 217 210 to 213	PC and PC AT Game Controller, reserved LIM Expanded Memory Card PC Expansion Unit AT-DIO-24

(continues)

Table A-3. PC I/O Address Map (continued)

I/O Address Range (Hex)	Device
218	LIM Expanded Memory Card
219 to 21E	
21F	Reserved
220 to 23F	AT-MIO-16
240 to 25F	AT-DIO-32F
248	LIM Expanded Memory Card
258	LIM Expanded Memory Card
260 to 27F	LabPC (default)
259 to 267	
268	LIM Expanded Memory Card
269 to 277	
278 to 27F	AT Parallel Printer Port 2
280 to 29F	WD EtherCard+ (default)
2A0 to 2A7	
2A8	LIM Expanded Memory Card
2A9 to 2AF	
2B0 to 2DF	PC, AT EGA (alternate)
2B8	LIM Expanded Memory Card, GPIB-PCII (base)
2B9 to 2BF	
2C0 to 2DF	AT-GPIB board 0 (default)
2E0 to 2FF	AT-GPIB board 1 (default)
2E1	IBM GPIB Adapter 0, GPIB-PCIIA (base)
2E2 to 2E3	IBM Data Acquisition Adapter 0
2E4 to 2E7	
2E8	LIM Expanded Memory Card
2E9 to 2F7	
2F8 to 2FF	PC, AT Serial Port 2 (COM2)

(continues)

Table A-3. PC I/O Address Map (continued)

I/O Address Range (Hex)	Device
300 to 31F	PC, AT Prototype card
300 to 30F	3Com EtherLink (default)
320 to 32F	IBM PC/XT Fixed Disk Controller
330 to 347	
348 to 357	DCA 3278
358 to 35F	
360 to 363	PC Network (low address)
364 to 367	Reserved
368 to 36B	PC Network (high address)
36C to 36F	Reserved
370 to 377	
378 to 37F	PC, AT Parallel Printer Port 1
380 to 38C	SDLC Communications
380 to 389	Bisynchronous (BSC) Communications (alternate)
390 to 393	Cluster Adapter 0
394 to 39F	
3A0 to 3A9	Bisynchronous (BSC) Communications (primary)
3AA to 3AF	
3B0 to 3BF	Monochrome Display/Parallel Printer Adapter 0
3C0 to 3CF	Enhanced Graphics Adapter, VGA
3D0 to 3DF	Color/Graphics Monitor Adapter, VGA
3E0 to 3EF	
3F0 to 3F7	Diskette Controller
3F8 to 3FF	Serial Port 1 (COM1)

Interrupt Selection

The GPIB-PCII/IIA interface board can use any of the six interrupt lines available on the PC, or no interrupts at all. The interrupt line is selected using the jumper sets labeled IRQ2 through IRQ7 (see Figure A-1). The GPIB-PCII/IIA is set at the factory to use line 7. To select another interrupt line, place the supplied jumper across the two pins adjacent to the label designating that interrupt line. Figure A-6 shows the selection of interrupt line 7.

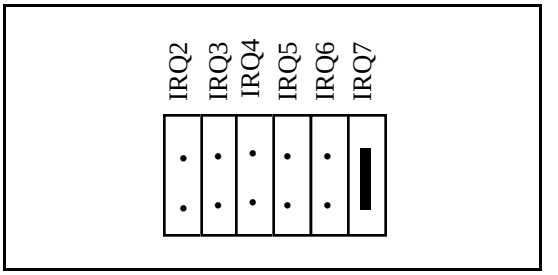


Figure A-6. Default Interrupt Jumper Setting for GPIB-PCII

If you do not want to use interrupts, you must logically disconnect the GPIB-PCII/IIA from the IRQ lines by selecting NONE for the interrupt line when you run IBCONF in *Step 2 - Configure the Software* later in this chapter. The board can remain in the backplane and no jumpers have to be moved or changed.

Shared Interrupts in GPIB-PCIIA Mode

The GPIB-PCIIA uses a special feature of the PC called shared interrupts. This feature allows more than one device to share the same interrupt line, if both devices have the ability to share interrupts. If you use the GPIB-PCII/IIA in GPIB-PCIIA mode and you want to change the interrupt line, you must set switches I0, I1, and I2 in switch set U2 to the line setting in addition to setting the interrupt jumpers.

Figure A-7 shows the switch and jumper settings for the default interrupt setting, IRQ7, and shows how the interrupt setting was calculated from the switch positions. Figure A-8 shows the switch and jumper settings for the five remaining interrupt lines. In both figures, the black side of the switch is the side you press down. The shaded switches are used to set the base I/O address and are not used in determining the interrupt line.

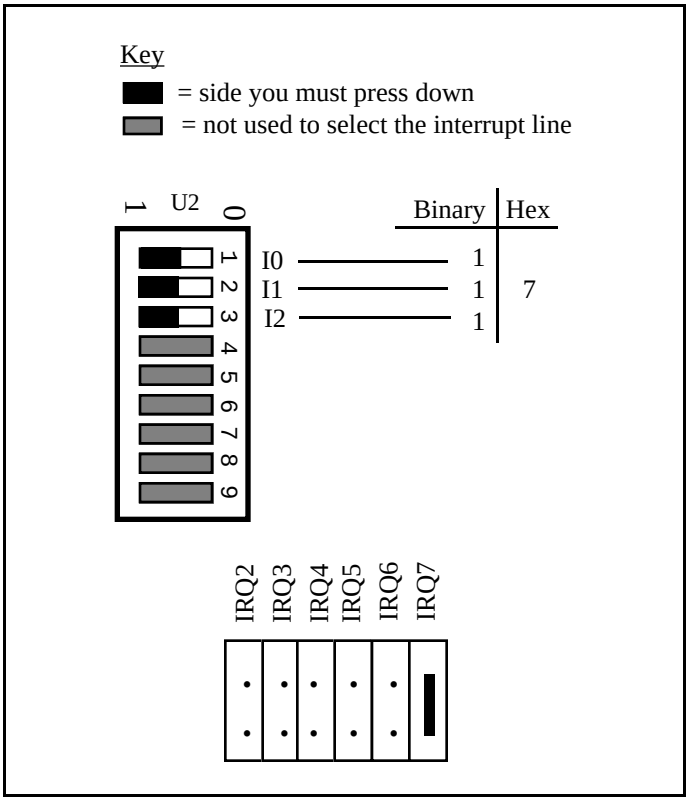


Figure A-7. Default Interrupt Jumper Setting for GPIB-PCIIA

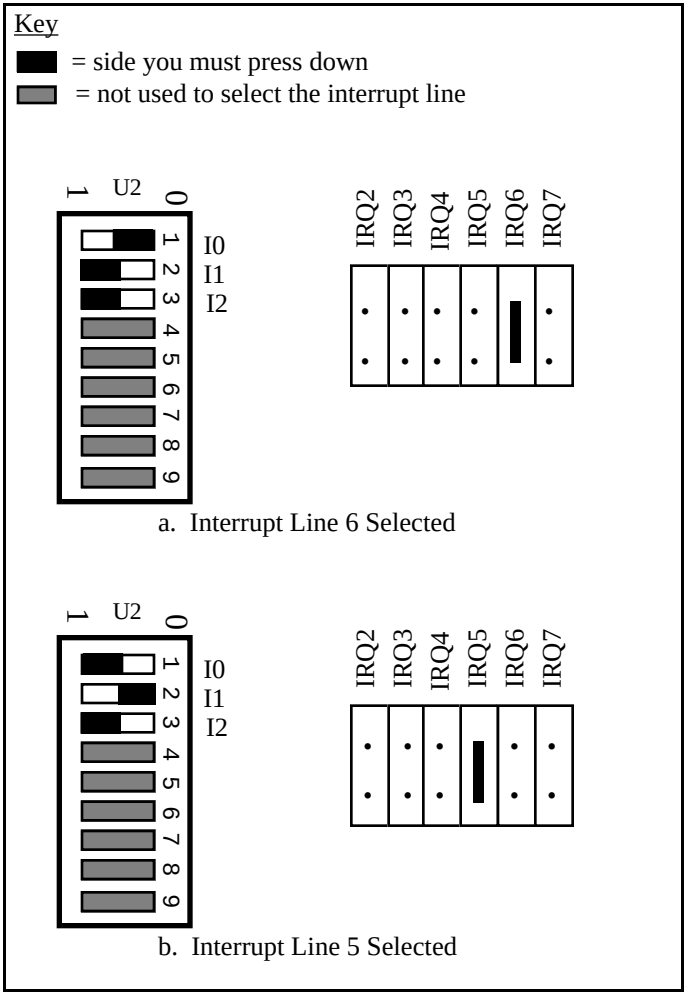


Figure A-8. Interrupt Jumper Settings for GPIB-PCIIA (continues)

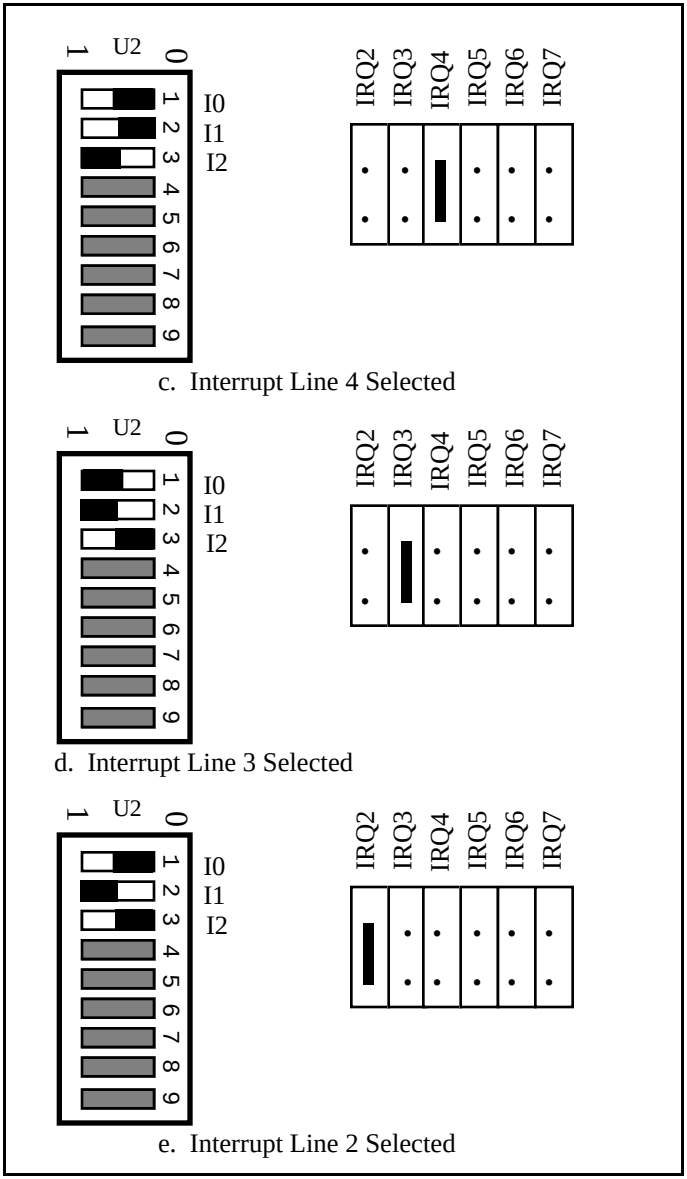


Figure A-8. Interrupt Jumper Settings for GPIB-PCIIA (continued)

If you change the interrupt jumper setting from the default setting, record the new setting in the *GPIB-PCII/IIA Hardware and Software Configuration Form* in Appendix B and proceed to *Step 2 - Software Configuration* later in this appendix.

Possible Conflicts

Table A-4 lists some of the interrupt lines used by other PC plug-in interface boards and adapters. This is by no means a complete list, but it may help in determining possible interrupt conflicts. Symptoms of interrupt conflicts vary widely. At one extreme, conflicts can prevent the computer from booting. Interrupt conflicts may also cause repeated time outs on GPIB function calls. When they do surface, the problems can exhibit themselves simply as strange behavior.

National Instruments has made every effort to select a default interrupt line that will work. However, because of the numerous different interface boards available for use in the PC, it is not possible to select an interrupt line that is guaranteed to work in all systems. Therefore, be certain of your system's interrupt assignments before proceeding with installation.

Table A-4. PC Interrupt Assignment Map

IRQ	Device
7	Parallel Port 1 Data Acquisition and Control (default) GPIB-PCII/IIA
6	Diskette Controller Fixed Disk and Diskette Drive
5	Parallel Port 2 PC-DIO-24 (default) LabPC (default)

(continues)

Table A-4. PC Interrupt Assignment Map (continued)

IRQ	Device
4	Serial Port 1 BSC BSC Alt. SDLC
3	Serial Port 2 BSC BSC Alt. Cluster (Primary) PC Network (default) PC Network Alt. (default) SDLC WD EtherCard+ (default) 3Com EtherLink (default)
2	IRQ Chain for PC AT
1	Keyboard Controller Output Buffer Full
0	Timer Channel 0 Output

DMA Channel Selection

The GPIB-PCII/IIA can use DMA channels 1, 2, or 3, or no DMA at all. The DMA channel is selected by the jumper sets labeled DRQ1 through DACK 3 (see Figure A-1).

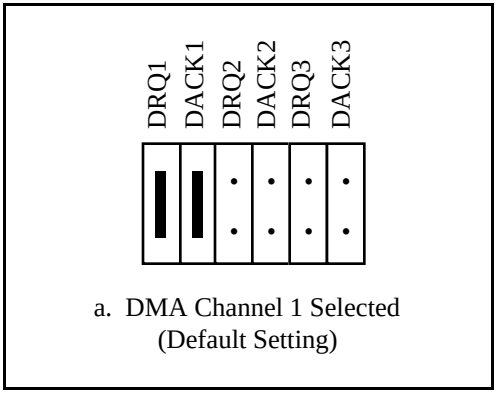
Each DMA channel consists of two signal lines, as shown in Table A-5.

Table A-5. DMA Channels for the GPIB-PCII/IIA

DMA Channel	Signal Lines	
	DMA Acknowledge	DMA Request
1	DACK1	DRQ1
2	DACK2	DRQ2
3	DACK3	DRQ3

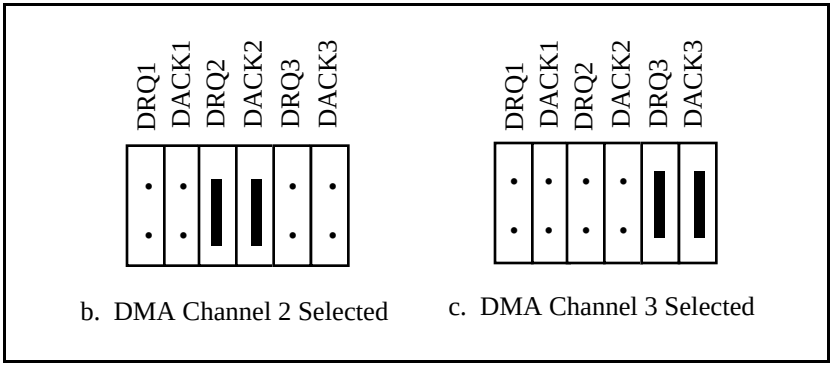
You must position two jumpers to select a DMA channel. One jumper selects the DMA Request line, and the other selects the DMA Acknowledge line. You must move these two jumpers as a pair, and the DMA Acknowledge and DMA Request lines that you select must have the same numeric suffix for proper operation.

Figure A-9 shows the jumper position for selecting each DMA channel.



a. DMA Channel 1 Selected
(Default Setting)

Figure A-9. DMA Channel Jumper Settings (continues)



b. DMA Channel 2 Selected c. DMA Channel 3 Selected

Figure A-9. DMA Channel Jumper Settings (continued)

If you do not want to use DMA for GPIB transfers (the GPIB-PCII/IIA alternatively can use programmed I/O transfers), you must logically disconnect the GPIB-PCII/IIA from the DMA lines by selecting NONE for the DMA line when you run `IBCONF` in *Step 2 - Configure the Software* later in this chapter. The board can remain in the backplane and no jumpers have to be moved or changed.

If you change the DMA jumper setting from the default setting, record the new setting in the *GPIB-PCII/IIA Hardware and Software Configuration Form* in Appendix B and proceed to *Step 2 - Software Configuration* at the end of this appendix.

Possible Conflicts

There are only three DMA channels that can be used by other PC plug-in interface boards and adapters. If any device uses DMA channel 1, change the DMA channel used by either the GPIB-PCII/IIA or the other device to DMA channel 2 or 3. If no DMA channel is available, configure the software to run without DMA using the software configuration program, `IBCONF`.

Shield Ground Configuration

The GPIB-PCII/IIA is set at the factory with the jumper in place to connect the logic ground of the GPIB-PCII/IIA to its shield ground. This configuration minimizes the EMI emitted from a PC equipped with a GPIB-PCII/IIA. However, if your application requires that logic ground be disconnected from shield ground, remove the jumper (W1) and place it across only one of the jumper pins. Jumper settings for logic ground connected and disconnected from shield ground are shown in Figure A-10.

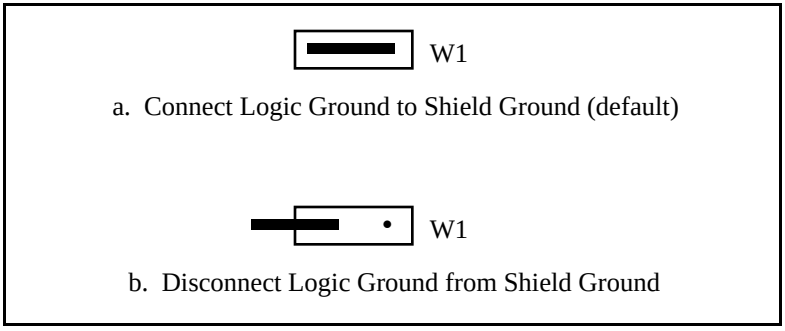


Figure A-10. Ground Configuration Jumper Settings

Step 2 - Software Configuration

Any hardware setting changes must be matched by making appropriate changes to the NI-488.2 handler. To accomplish this, you must run the configuration utility, `IBCONF`, and edit the board parameters (such as the base I/O address or interrupt line) that you have changed. Refer to Chapter 2 of the *NI-488.2 MS-DOS Software Reference Manual* for instructions on running `IBCONF`.

Appendix B
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
Netherlands	03480-33466	03480-30673
Norway	32-848400	32-848600
Spain	(91) 640 0085	(91) 640 0533
Sweden	08-730 49 70	08-730 43 70
Switzerland	056/20 51 51	056/27 00 25
U.K.	0635 523545	0635 523154

Technical Support Form

Photocopy this form and update it each time you make changes to your software or hardware, and use the completed copy of this form as a reference for your current configuration. Completing this form accurately before contacting National Instruments for technical support helps our applications engineers answer your questions more efficiently.

If you are using any National Instruments hardware or software products related to this problem, include the configuration forms from their user manuals. Include 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 _____

(continues)

National Instruments software product _____

Version _____

Configuration _____

The problem is _____

_____

_____

_____

_____

_____

_____

_____

List any error messages _____

_____

_____

_____

_____

_____

_____

_____

The following steps will reproduce the problem _____

_____

_____

_____

_____

_____

_____

GPIB-PCII/IIA Hardware and Software Configuration Form

Record the settings and revisions of your hardware and software on the line located to the right of each item. Complete this form each time you revise your software or hardware configuration, and use this form as a reference for your current configuration. Completing this form accurately before contacting National Instruments for technical support helps our applications engineers answer your questions more efficiently.

National Instruments Products

GPIB-PCII/IIA Hardware

- GPIB-PC Mode of GPIB-PCII/IIA _____
- 7210/9914 Mode of GPIB-PCII/IIA _____
- Interrupt Level of GPIB-PCII/IIA _____
- DMA Channel of GPIB-PCII/IIA _____
- Base I/O Address of GPIB-PCII/IIA _____
- Shield Ground Configuration of GPIB-PCII/IIA _____

NI-488.2 Software

- NI-488.2 Software Revision Number on Disk _____
  (Disk Label: *NI-488.2 Distribution Disk for GPIB-PCII/IIA MS-DOS/Windows Handler, BASICA, QuickBASIC, BASIC, C & Universal Interfaces*)
- Application Programming Language (BASICA, QuickBASIC, C, Pascal, and so on) _____
- Programming Language Interface Revision _____

(continues)

- GPIB-PC Handler Type set in `IBCONF` _____
- Interrupt Level set in `IBCONF` _____
- DMA Channel set in `IBCONF` _____
- Base I/O Address set in `IBCONF` _____

Other Products

- Computer Make and Model _____
- Microprocessor _____
- Clock Frequency _____
- Type of Monitor Card Installed _____
- DOS Version _____
- Other Boards in System _____
- Base I/O Address of Other Boards _____
- DMA Channels of Other Boards _____
- Interrupt Level of Other Boards _____

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-PCII/IIA and the NI-488.2™ Software for MS-DOS**

Edition Date: **June 1992**

Part Number: **320320-01**

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