



**Agilent N1231A PCI Three-Axis Laser Board**  
**User's Guide**



**Agilent Technologies**



---

# User's Guide

This guide describes how to use the Agilent N1231A PCI Three-Axis Laser Board. The information in this guide applies to boards having the serial prefix number listed below, unless accompanied by a "Manual Updating Changes" package indicating otherwise.

**SERIAL PREFIX NUMBER: US4150**

---

Agilent N1231A PCI Three-Axis  
Laser Board

©Copyright Agilent Technologies, Inc. 2002

All Rights Reserved.  
Reproduction, adaptation, or translations without prior written permission is prohibited, except as allowed under the copyright laws.

Printed: May 2002

Printed in USA

**Manual part number  
N1231-90002**

## Certification and Warranty

### Certification

Agilent Technologies certifies that this product met its published specification at the time of shipment from the factory. Agilent further certifies that its calibration measurements are traceable to the United States National Institute of Standards and Technology (formerly National Bureau of Standards), to the extent allowed by the Institute's calibration facility, and to the calibration facilities of other International Standards Organization members.

### Warranty

Agilent warrants Agilent hardware, accessories and supplies against defects in materials and workmanship for a period of three years from date of shipment. If Agilent receives notice of such defects during the warranty period, Agilent will, at its option, either repair or replace products which prove to be defective. Replacement products may be either new or like-new.

Agilent warrants that Agilent software will not fail to execute its programming instructions, for the period specified above, due to defects in material and workmanship when properly installed and used. If Agilent receives notice of such defects during the warranty period, Agilent will replace software media which does not execute its programming instructions due to such defects.

*For detailed warranty information, see back matter.*

*For detailed software information, see the US Software Limited Warranty on the following page.*

## Safety Considerations General

This product and related documentation must be reviewed for familiarization with these safety markings and instructions before operation.

### Before Cleaning

Disconnect the product from operating power before cleaning.

### Warning Symbols That May Be Used In This Book



Instruction manual symbol; the product will be marked with this symbol when it is necessary for the user to refer to the instruction manual.



Indicates hazardous voltages.



Indicates earth (ground) terminal.



or



Indicates terminal is connected to chassis when such connection is not apparent.



Indicates Alternating current.



Indicates Direct current.

## Safety Considerations (contd)

### WARNING

**BODILY INJURY OR DEATH MAY RESULT FROM FAILURE TO HEED A WARNING. DO NOT PROCEED BEYOND A WARNING UNTIL THE INDICATED CONDITIONS ARE FULLY UNDERSTOOD AND MET.**

### CAUTION

Damage to equipment, or incorrect measurement data, may result from failure to heed a caution. Do not proceed beyond a **CAUTION** until the indicated conditions are fully understood and met.

### Safety Earth Ground

An uninterruptible safety earth ground must be maintained from the mains power source to the product's ground circuitry.

### WARNING

**ANY INTERRUPTION OF THE PROTECTIVE GROUNDING CONDUCTOR (INSIDE OR OUTSIDE THE PRODUCT'S CIRCUITRY) OR DISCONNECTING THE PROTECTIVE EARTH TERMINAL WILL CAUSE A POTENTIAL SHOCK HAZARD THAT COULD RESULT IN PERSONAL INJURY. (GROUNDING ONE CONDUCTOR OF A TWO CONDUCTOR OUTLET IS NOT SUFFICIENT PROTECTION.) WHENEVER IT IS LIKELY THAT THE PROTECTION HAS BEEN IMPAIRED, THE PRODUCT MUST BE MADE INOPERATIVE AND BE SECURED AGAINST ANY UNINTENDED OPERATION.**

*For additional safety and acoustic noise information, see back matter.*

**ATTENTION: USE OF THE SOFTWARE PROVIDED FOR USE WITH THE AGILENT N1231A LASER MEASUREMENT SYSTEM IS SUBJECT TO THE SOFTWARE LIMITED WARRANTY STATEMENT BELOW. BY INSTALLING THE SOFTWARE, YOU ARE AGREEING TO BE BOUND BY THESE TERMS. IF YOU DO NOT AGREE TO ALL OF THESE TERMS, DO NOT INSTALL THE SOFTWARE ON YOUR SYSTEM.**

## **AGILENT TECHNOLOGIES SOFTWARE PRODUCT LIMITED WARRANTY STATEMENT**

### **Software Limited Warranty**

Agilent warrants to you that the Software will not fail to execute its programming instructions after the date of purchase for a period of one year from the date of installation due to defects in material and workmanship when properly installed and used. Agilent further warrants that proprietary Agilent Software will operate in substantial conformance to any documentation describing its use, provided that certain Software (i.e., Example.cpp) is provided merely to illustrate applications or functions and not to perform such functions. Agilent does not warrant that the operation of the Software will be uninterrupted or error free.

If Agilent receives notice of such defects during the warranty period, Agilent will replace the Software that does not execute its programming instructions due to such defects or does not conform to applicable documentation. If Agilent is unable, within a reasonable time, to replace any Software to a condition as warranted, you will be entitled to a refund of the purchase price upon return of the Software or certification of its destruction, including any copies.

### **Exclusions**

This limited warranty does not apply to defects resulting from (a) improper or inadequate maintenance, (b) unauthorized modification or misuse, or (c) operation outside of the published environmental specifications for Software or the System on which it is intended to run.

### **Disclaimer**

TO THE EXTENT ALLOWED BY LOCAL LAW, THE ABOVE WARRANTIES ARE EXCLUSIVE AND NO OTHER WARRANTY OR CONDITION, WHETHER WRITTEN OR ORAL, IS EXPRESSED OR IMPLIED. AGILENT SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OR CONDITIONS OF MERCHANTABILITY, SATISFACTORY QUALITY, AND FITNESS FOR A PARTICULAR PURPOSE. Some countries, states or provinces do not allow limitations on the duration of an implied warranty, so the above limitation or exclusion might not apply to you. This warranty gives you specific legal rights and you might also have other rights that vary from country to country, state to state, or province to province.

### **Limitation of Liability**

TO THE EXTENT ALLOWED BY LOCAL LAW, THE REMEDIES IN THIS WARRANTY STATEMENT ARE YOUR SOLE AND EXCLUSIVE REMEDIES. EXCEPT TO THE EXTENT PROHIBITED BY LOCAL LAW, IN NO EVENT WILL AGILENT OR ITS SUBSIDIARIES, AFFILIATES OR SUPPLIERS BE LIABLE FOR DIRECT, SPECIAL, INCIDENTAL, CONSEQUENTIAL OR OTHER DAMAGES (INCLUDING LOST PROFIT, LOST DATA, OR DOWNTIME COSTS), ARISING OUT OF THE USE, INABILITY TO USE, OR THE RESULTS OF USE OF THE SOFTWARE, WHETHER BASED IN WARRANTY, CONTRACT, TORT OR OTHER LEGAL THEORY, AND WHETHER OR NOT ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Your use of the Software is entirely at your own risk. Some countries, states and provinces do not allow the exclusion or limitation of liability for incidental or consequential damages, so the above limitation may not apply to you.

**NOTE:** EXCEPT TO THE EXTENT ALLOWED BY LOCAL LAW, THESE WARRANTY TERMS DO NOT EXCLUDE, RESTRICT OR MODIFY, AND ARE IN ADDITION TO, THE MANDATORY STATUTORY RIGHTS APPLICABLE TO THE LICENSE OF THE SOFTWARE TO YOU.

\*\*\*

---



---

# Contents

## In This Guide

**Guide Organization**    xi

**Description of the Agilent N1231A Three-Axis Laser Board**    xii

**Option**    xii

## 1 Getting Started

**Agilent N1231A Connector Panel at a Glance**    1-2

**Agilent N1231A Board at a Glance**    1-3

**Installation**    1-4

    Installing the Agilent N1231A board    1-4

    Installing the Agilent N1231A API Library and Monitor program    1-4

        To Install the N1231A API Library    1-4

        To Install the N1231A Monitor Program    1-5

**Operational Verification of the Board, API Library, and Monitor Application**    1-5

**Product Support**    1-9

## 2 Features and Functions

**Chapter Contents**    2-2

**Input and Output Connectors**    2-3

    Interface signals overview    2-3

    Agilent N1231A connector panel    2-3

    Axis 1, 2, and 3 PMAC compatible connectors    2-5

        Axis 1, 2, and 3 Position Data output    2-5

        Axis 1, 2, and 3 Position Data Clock input/output    2-5

**PC Backplane Inputs / Outputs**    2-6

    PCI Bus Inputs    2-6

    PCI Bus Outputs    2-6

        Axis 1, Axis 2, Axis 3 Position and Velocity Data    2-6

        Position system status and errors    2-6

**Error Detection**    2-7

**Cables**    2-7

    PMAC compatible ribbon cable connector pin assignments    2-7

## **3 Programming the Registers**

### **Chapter Contents 3-2**

#### **Memory Map 3-3**

#### **Bitmaps of Registers 3-4**

Bitmap A 3-4

Bitmap B 3-4

Bitmap C 3-4

Bitmap D 3-5

Bitmap E 3-6

Bitmap F 3-6

Bitmap G 3-7

Bitmap H 3-7

Bitmap I 3-7

#### **Velocity Register 3-8**

Velocity equation 3-8

Error calculation 3-8

## **4 System Configuration**

### **Chapter Contents 4-2**

#### **System Overview 4-2**

Laser head and receivers 4-3

Environmental sensors 4-3

Delta Tau<sup>®</sup> boards (PMAC and ACC-14D) 4-4

Typical 3-axis laser/PMAC system configuration using the Agilent 10780C/F receivers 4-4

How the boards are connected 4-6

Typical 3-axis laser/PMAC system configuration using the Agilent E1708A or E1709A receivers 4-6

Setting the jumpers on the ACC-14D ISA boards 4-8

## **5 Functional Block Description**

### **Chapter Contents 5-2**

#### **How It Works 5-3**

Position Measurement Block 5-3

#### **Accuracy & Resolution 5-4**

Axis 1, 2, and 3 position resolution 5-4

Fixed delay 5-4

Wavelength compensation 5-4



## 6 Specifications

### Chapter Contents 6-2

#### General Specifications 6-3

Measurement Range and Resolution 6-3

Maximum Axis Velocity (slew rate) 6-3

Internal Data Update Rate 6-3

#### Electrical Specifications 6-4

Receiver Signal Inputs 6-4

Hardware Position Outputs 6-4

Signal Levels 6-4

Data Age Relative to OCLK 6-4

Data Age Relative to ICLK 6-4

Data Format 6-4

Maximum Data Rate 6-4

PCI Bus I/O Characteristics 6-4

PCI Compliance 6-4

Position and Velocity Data Age relative to the rising edge of the PCI bus Clock on the bus write cycle requesting a Sample 6-4

Position Data Format 6-4

Velocity Data Format 6-5

Theoretical Maximum Data Rates 6-5

Electrostatic Discharge (ESD) 6-5

Power Requirements 6-5

Cooling requirements 6-5

Operating Ambient Temperature 6-5

Air Flow 6-5

Timing data 6-6

PMAC handshake 6-6

Minimum time between rising edges of OCLK 6-6

#### Mechanical Specifications 6-6

## A Downloading Firmware

### Overview A-2

### Downloading New Firmware A-2

## B N1231A.h and N1231A\_reg.h C/C++ Programming Information

### Appendix Contents B-2

### The C/C++ N1231A.h Header File B-2

### The C/C++ N1231A\_reg.h Header File B-22

## **C CD Readme File**

### **General C-2**

Description of setup programs C-2

N1231A API OEM version C-2

N1231A API Development version C-2

N1231A Monitor C-2

**Installing the OEM Version of the N1231A API Library from the CD C-3**

**Installing the Development Version of the N1231A API Library from the CD C-3**

**Installing the N1231A Monitor Program from the CD C-4**

## **D Agilent N1231A API Readme File**

**D-2**

## **E Filter Settings in Bitmap D**

**Introduction E-2**

**Frequency Response E-3**

**Temporal Response E-7**

## **Index**

---

# List of Figures

Figure 1-1.	Start/Programs pop-up menu	1-6
Figure 1-2.	Example program	1-6
Figure 1-3.	Example program after pressing 7 key	1-7
Figure 1-4.	Monitor screen after opening the application	1-8
Figure 2-1.	Agilent N1231A board	2-3
Figure 2-2.	N1231A Connector Panel	2-4
Figure 4-1.	Example of the N1231A and PMAC system with 10780C/F receivers	4-5
Figure 4-2.	Example of the N1231A and PMAC system with E1708A or E1709A receivers	4-7
Figure 5-1.	Agilent N1231A board overall block diagram	5-3
Figure 6-1.	Agilent N1231A physical dimensions	6-7
Figure A-1.	WinZip Self-Extractor dialog box	A-2
Figure A-2.	DOS window	A-3
Figure A-3.	DOS window — READY TO DOWNLOAD IO CARD	A-3

## List of Figures

---

# List of Tables

Table 1-1.	Exchange Assembly	1-9
Table 2-1.	Ribbon Cable connector pin assignments	2-7
Table 3-1.	Memory Map	3-3
Table 3-2.	Control Setup Bits for Axis 3 and Interrupt (Read/Write Offset 0x0010 from PCIBAR2)	3-4
Table 3-3.	Error Status Bits for Axis 3 and Ref (Read/Write Offset 0x0012 from PCIBAR2)	3-4
Table 3-4.	Interrupt Mask Bits (Read/Write Offset 0x0016 from PCIBAR2)	3-4
Table 3-5.	Filter Settings (Read/Write Offset 0x001C from PCIBAR2)	3-5
Table 3-6.	Kv and Kp digital filters	3-5
Table 3-7.	Sample Ready Bits (Read/Write Offset 0x0020 from PCIBAR2)	3-6
Table 3-8.	Sample and Preset Bits (Write Offset 0x0020 from PCIBAR2)	3-6
Table 3-9.	Control Setup Bits for Axis 2 and Axis 1 (Read/Write Offset 0x0110 from PCIBAR2)	3-7
Table 3-10.	Error Status Bits for X1 and X2 (Read/Write Offset 0x0112 from PCIBAR2)	3-7
Table 3-11.	Board and JTAG Read Register (Read Offset 0x0000 from PCIBAR3)	3-7
Table 5-1.	Fixed errors (can be compensated)	5-4
Table 6-1.	Resolution and range	6-3
Table 6-2.	Maximum Axis Velocity (slew rate)	6-3
Table 6-3.	Theoretical Minimum Data Collection Times	6-5



---

# In This Guide

This preface contains the following information:

- Guide Organization page xi
- Description of the Agilent N1231A Three-Axis Laser Board page xii
- Option page xii

---

## Guide Organization

### Table of Contents

#### **In This Guide (this preface) —**

Introduces you to this guide and briefly describes the Agilent N1231A PCI Three-Axis Laser Board.

#### **Chapter 1 Getting Started —**

Provides a visual introduction to the Agilent N1231A board and an overview of its connectors. Board and software application installation, and operation verification instructions are also provided.

#### **Chapter 2 Features and Functions —**

Provides information on Agilent N1231A's features and functions, such as input and output connectors and signals, PMAC-compatible connectors (Axis 1, 2, and 3) pin-assignments, and error detection.

#### **Chapter 3 Programming the Registers —**

Describes the Agilent N1231A register contents in detail.

#### **Chapter 4 System Configuration —**

Provides an overview of the Agilent N1231A board's use with the Delta Tau<sup>®</sup> PMAC servo controller, describing the elements that comprise a laser measurement system that uses the Agilent N1231A board and PMAC servo controller. Also, sample system configurations are illustrated.

#### **Chapter 5 Functional Block Description —**

Provides descriptions and functional block diagrams of the circuitry.

#### **Chapter 6 Specifications —**

Lists the Agilent N1231A specifications and characteristics.

**Appendix A Downloading Firmware —**

Provides information on how to download new firmware to the Agilent N1231A board.

**Appendix B N1231A.h and N1231A\_reg.h C/C++ Programming Information**

**Appendix C CD Readme File**

**Appendix D Agilent N1231A API Readme File**

**Appendix E Filter Settings in Bitmap D**

**Index**

---

## Description of the Agilent N1231A Three-Axis Laser Board

The Agilent Technologies N1231A is a three-axis laser board optimized for connection to a PMAC motion control system from Delta Tau<sup>®</sup>. It is a full size, Universal (3.3V and 5.0V signaling compatibility), 32-bit, 33 MHz, PCI Rev. 2.2 compliant card for use in PC-compatible controllers as part of an Agilent laser interferometry position measurement system.

The primary purpose of the Agilent N1231A is to report the position of the stage via three generic measurement channels: 1, 2, and 3.

The Agilent N1231A board uses +5V, +12V, and –12V power from the backplane of the PC-compatible controller.

The Agilent N1231A board may be used with the Agilent 5517A/B/C/D Laser Head. In most cases, a +15V power supply, such as the Agilent 10884A Power Supply, is required to power the laser head and receivers.

---

## Option

The Agilent N1231A-200 Application Library and Monitor Application Compact Disk (Agilent part number N1231-13601) is available to order.



---

# 1

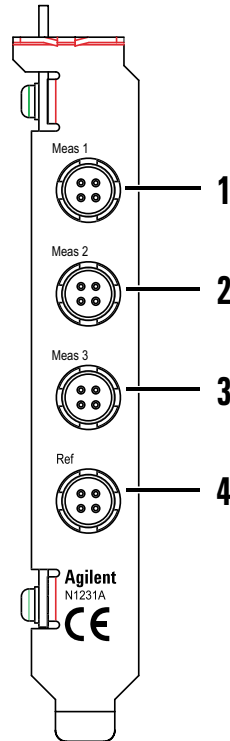
---

## Getting Started

Visual introduction of board, installation of board, installation of API library and monitor software, and operational verification

---

## Agilent N1231A Connector Panel at a Glance



### 1 Meas 1

4-pin connector for connecting the measurements signals from the receiver of Axis 1 to the N1231A.

### 2 Meas 2

4-pin connector for connecting the measurements signals from the receiver of Axis 2 to the N1231A.

### 3 Meas 3

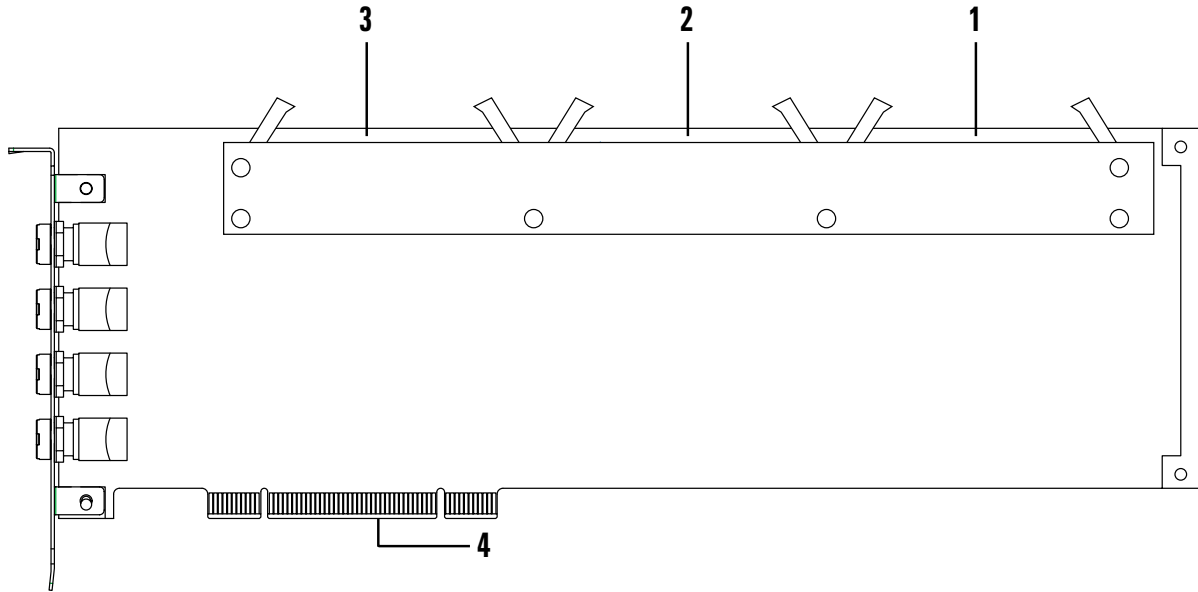
4-pin connector for connecting the measurements signals from the receiver of Axis 3 to the N1231A.

### 4 Ref

4-pin connector for connecting the reference signal from the laser head to the N1231A.

---

## Agilent N1231A Board at a Glance



### 1 Axis 1

50-contact, 3M N3423-5502 or equivalent ribbon cable connector for connecting Axis 1, TTL-level position data and clock outputs to the user's system.

### 2 Axis 2

50-contact, 3M N3423-5502 or equivalent ribbon cable connector for connecting Axis 2, TTL-level position data and clock outputs to the user's system.

### 3 Axis 3

50-contact, 3M N3423-5502 or equivalent ribbon cable connector for connecting Axis 3, TTL-level position data and clock outputs to the user's system.

### 4 PCI Interface

Edge connector for connecting to the backplane of the PC-compatible controller. This connector allows communication between the N1231A board and a software application (N1231A Monitor software application) running on the controller.

## Installation

This section provides information to help you install the Agilent N1231A PCI Three-Axis Laser Board, the API library, and the Monitor software application.

---

**CAUTION**

---

The Agilent N1231A board contains components that may be damaged by electrostatic discharge (ESD). Do not handle the Agilent N1231A board or any of its components without taking adequate measures to prevent damage due to electrostatic discharge (ESD).

### Installing the Agilent N1231A board

The Agilent N1231A board can be installed in any full-length PCI I/O backplane slot as shown in figures 4-1 and 4-2 in Chapter 4, “System Configuration,” in this guide.

For specific details about installing a board in your PC-compatible controller’s backplane, refer to documentation for the controller.

### Installing the Agilent N1231A API Library and Monitor program

#### *To Install the N1231A API Library*

- 1 Exit from all applications.
- 2 Insert the CD in an appropriate drive.
- 3 Navigate to the directory: “\N1231A API Development”.
- 4 Run “Setup.exe”.
- 5 Follow the instructions on the screen.

When asked to choose a Setup Type, the default choice of “Typical” is recommended. Other choices are “Compact” and “Custom.” The setup types are described below:

**Typical** installs all files.

**Compact** installs all files except those in the “User Files” directory.

**Custom** allows selecting the components to be installed.

- 6 If this is the first time the software has been installed, or if the software has been uninstalled prior to this installation, reboot the controller.

## Operational Verification of the Board, API Library, and Monitor Application

- 7 Perform the steps in the following subsection.

### *To Install the N1231A Monitor Program*

---

**NOTE**

---

If the N1231A API Library is not yet installed, follow the instructions above to install it before proceeding with this installation.

---

**NOTE**

---

The readme file, which provides more specifics about the software applications on the CD, has been duplicated in Appendix C, “CD Readme File,” in this guide.

- 1 Exit from all applications.
- 2 Insert the CD in an appropriate drive.
- 3 Navigate to the directory: “\N1231A Monitor”.
- 4 Run “Setup.exe”.
- 5 Follow the instructions on the screen.
- 6 After completing the installation, perform the procedure in “Operational Verification of the Board, API Library, and Monitor Application” on page 1-5.

---

## Operational Verification of the Board, API Library, and Monitor Application

- 1 Press **Start** button, located at the bottom left of the task bar in the Start menu window as shown in Figure 1-1.
- 2 Select **Programs**, then select **Agilent N1231A API** from the pop-up menu.

The **Agilent N1231A API** pop-up menu is displayed as shown in Figure 1-1.

## Operational Verification of the Board, API Library, and Monitor Application

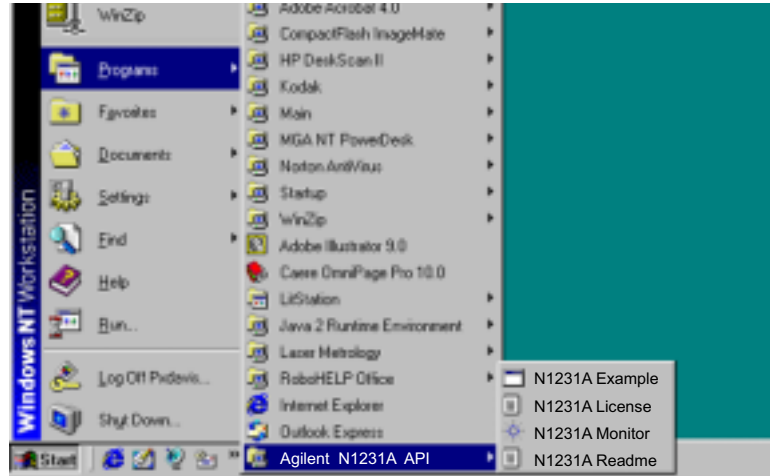


Figure 1-1. Start/Programs pop-up menu

- 3 Click on the **N1231A Example** icon (shown in Figure 1-1) to open the example program.

The example program is displayed as shown in Figure 1-2.

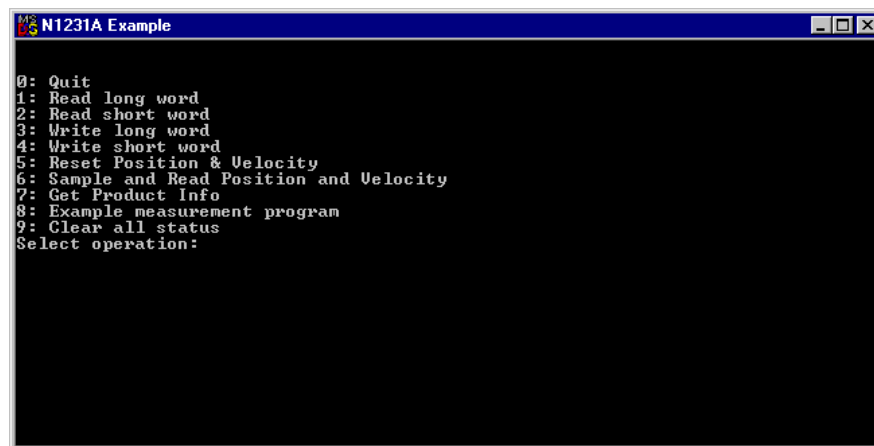
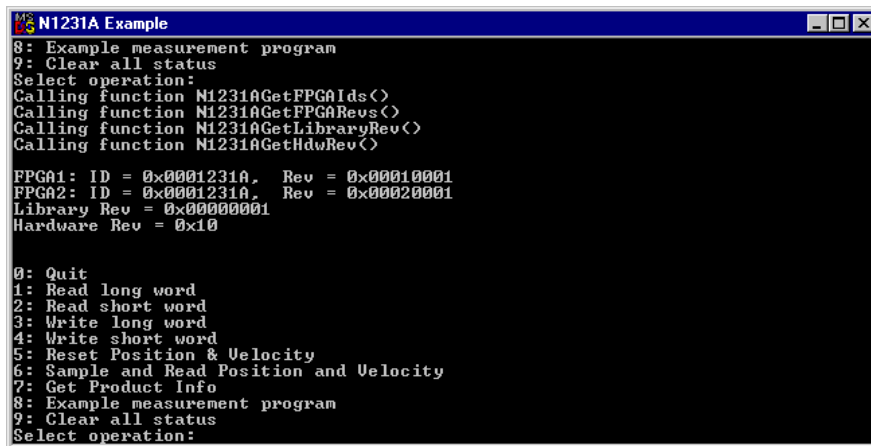


Figure 1-2. Example program

- 4 On your controller keyboard, press “7”.

## Operational Verification of the Board, API Library, and Monitor Application



```

N1231A Example
8: Example measurement program
9: Clear all status
Select operation:
Calling function N1231AGetFPGAIds()
Calling function N1231AGetFPGARevs()
Calling function N1231AGetLibraryRev()
Calling function N1231AGetHwRev()

FPGA1: ID = 0x0001231A, Rev = 0x00010001
FPGA2: ID = 0x0001231A, Rev = 0x00020001
Library Rev = 0x00000001
Hardware Rev = 0x10

0: Quit
1: Read long word
2: Read short word
3: Write long word
4: Write short word
5: Reset Position & Velocity
6: Sample and Read Position and Velocity
7: Get Product Info
8: Example measurement program
9: Clear all status
Select operation:

```

Figure 1-3. Example program after pressing 7 key

- 5 Verify that your results look similar to Figure 1-3.

That is, the display should indicate:


- the calling functions,
- the FPGAs' (1 and 2) identification and revision numbers,
- the Library and Hardware revision numbers, and
- a list of commands.

---

### NOTE

---

The various revision numbers may differ from those shown in Figure 1-3.

- 6 On your controller keyboard, press "0" to exit this program.
- 7 From the **Agilent N1231A API** pop-up menu, click on the  **N1231A Monitor** icon to open the Monitor application.

With no input signals connected to the Agilent N1231A board, the Agilent N1231A Monitor screen should appear as shown in Figure 1-4.

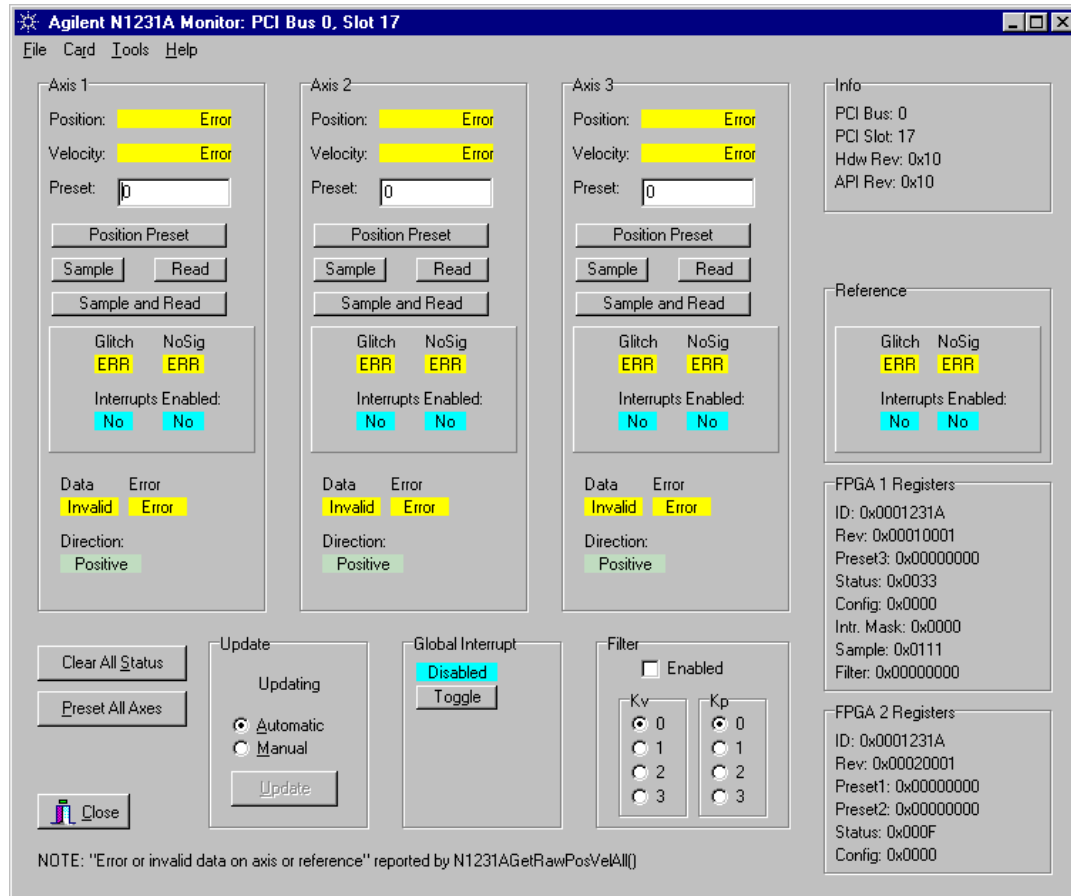
---

### NOTE

---

The PCI bus and slot number may differ from those shown in Figure 1-4.

## Operational Verification of the Board, API Library, and Monitor Application



**Figure 1-4. Monitor screen after opening the application**

- 8** Verify that the following information is displayed in your Monitor screen:
  - Under **Info** section, should have Hdw Rev. and API Rev numbers.
  - Under **FPGA 1 Registers** section, the following should be indicated:
 

ID: 0x0001231A  
Rev: 0x0010001\*
  - Under **FPGA 2 Registers** section, the following should be indicated:
 

ID: 0x0001231A  
Rev: 0x0020001\*

\* This revision number will change when there are FPGA revision changes.



## Product Support

- 9** If all of the items listed in step 8 are displayed in the Monitor screen, your board and software application are installed and working correctly.

To exit this application, click on “Close” button in lower left corner of the application’s screen, or the close (X) button in the upper-right corner of the window.

This completes the N1231A board and Monitor application operational verification.

If the board fails, proceed to the following section.

---

## Product Support

If your board fails the operational verification, contact Agilent Call Center. Telephone numbers of various call centers are provided on the “Service and Support” page at the back of this guide.

Table 1-1 lists the part number for ordering an exchange assembly to replace a defective assembly.

The exchange assembly is factory repaired and tested and is available on a trade-in basis; therefore, the defective assembly must be returned to Agilent for credit.

**Table 1-1. Exchange Assembly**

Assembly	Exchange Assembly Agilent Part Number
Agilent N1231A	N1231-69101





---

## Chapter Contents

You will find that this chapter makes it easy to look up details about a particular feature or function of the Agilent N1231A.

This chapter is organized as follows:

- Input and Output Connectors page 2-3
  - Interface signals overview page 2-3
  - Agilent N1231A connector panel page 2-3
  - Axis 1, 2, and 3 PMAC compatible connectors page 2-5
- PC Backplane Inputs / Outputs page 2-6
  - PCI Bus Inputs page 2-6
  - PCI Bus Outputs page 2-6
- Error Detection page 2-7
- Cables page 2-7
  - PMAC compatible ribbon cable connector pin assignments page 2-7

## Input and Output Connectors

### Interface signals overview

All signals are fed to and from the Agilent N1231A board via connectors on its bracket (or connector panel), the PCI bus, or Axis 1, Axis 2, and Axis 3 ribbon cable connectors as shown in Figure 2-1.

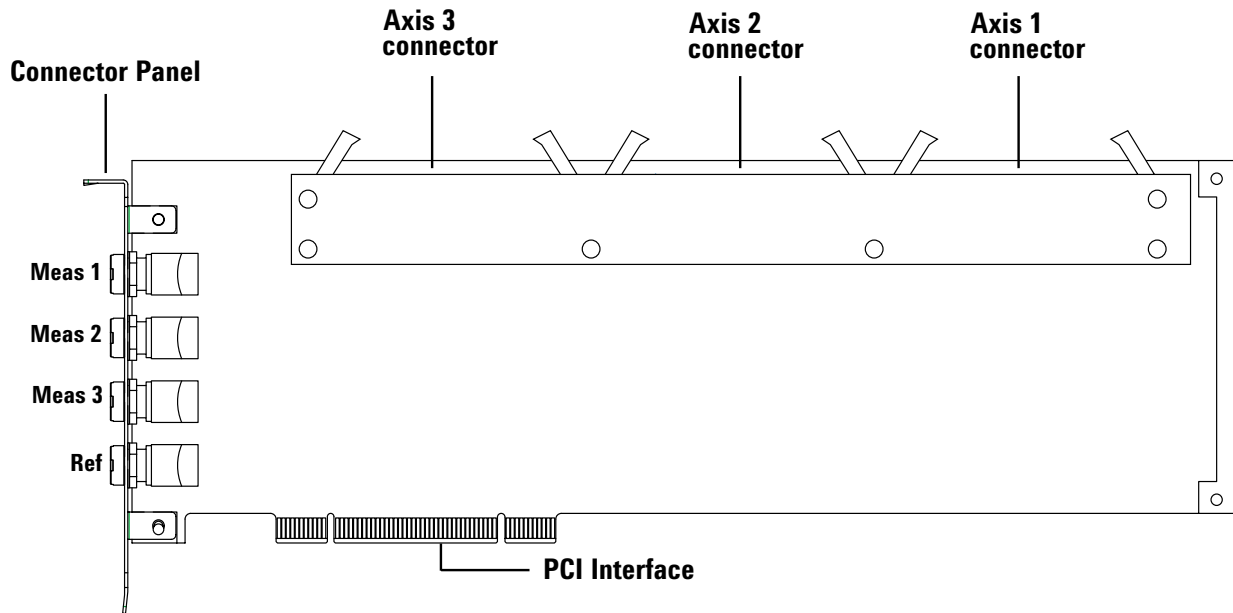


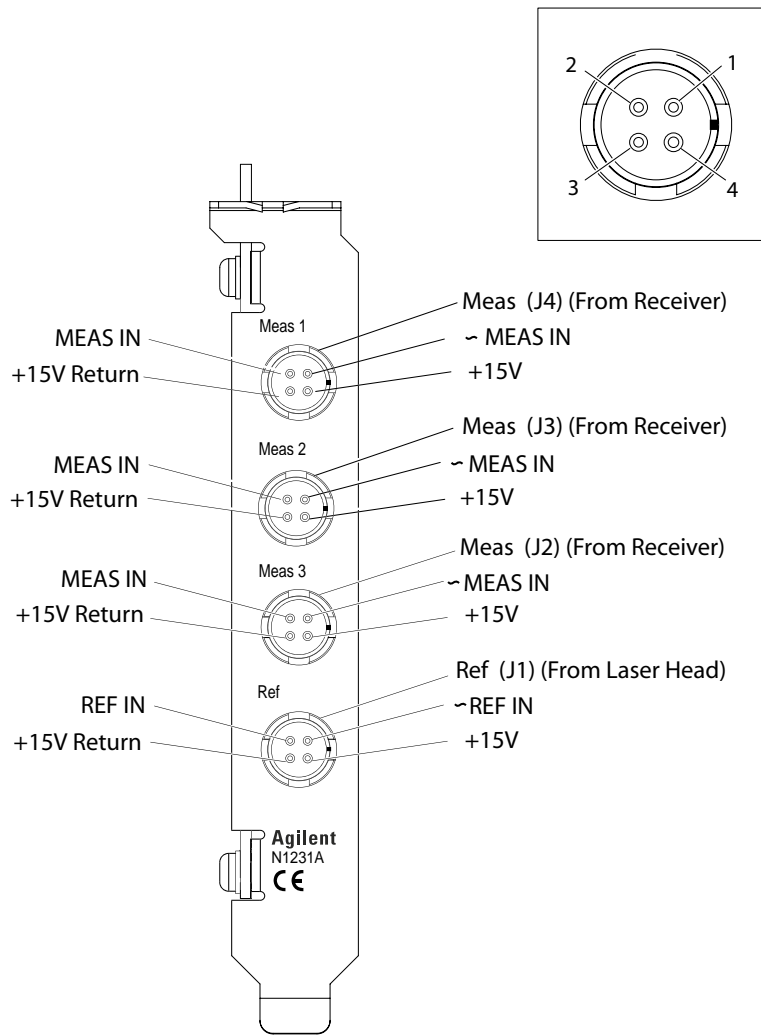
Figure 2-1. Agilent N1231A board

### Agilent N1231A connector panel

As shown in Figure 2-2, the N1231A contains four 4-pin connectors on its panel for connecting the measurements signals from the Axis 1, Axis 2, and Axis 3 receivers, and reference signal from the laser head. The connectors are described as follows:

- Meas 1— Axis 1 Measurement Signal (from receiver)
- Meas 2— Axis 2 Measurement Signal (from receiver)
- Meas 3— Axis 3 Measurement Signal (from receiver)
- Ref—Reference Signal (from Laser Head)

## Input and Output Connectors

**Figure 2-2. N1231A Connector Panel**

As shown in Figure 2-2, on each connector two contacts (1, 2) are used for the differential axis input signals. The other two contacts (3, 4) are bussed between the connectors and are used to distribute +15V power (and its return) to the receivers. Receiver power is normally supplied via the Reference Signal cable and then distributed to the other receivers in the system via these bussed contacts. The N1231A does not supply this +15V power.

## Input and Output Connectors

The Meas 1, 2, and 3 signals from the Agilent receivers (10780C/F, E1708A, or E1709A) are connected to the Agilent N1231A panel via the Agilent 10880A/B/C ((5/10/20 meters) Receiver Cable, or the Agilent N1250A/B (5/10 meters) High Performance Receiver Cable. *Agilent recommends using high performance cables and receivers (Agilent E1708A or E1709A) for slew rates greater than 500 mm/s (plane mirror), or when dynamic measurements are being made.*

The reference signal from the laser head is connected to the Agilent N1231A panel via the Agilent 10881A/B/C (3/7/20 meters) Laser Head Cable, or the Agilent N1251A/B (3/7 meters) High Performance Laser Head Cable. See figures 4-1 and 4-2 in Chapter 4, “System Configuration,” of this guide.

See *Agilent Laser and Optics User's Manual* (Agilent part number 05517-90045) for more specifics on receiver and laser head cables.

### **Axis 1, 2, and 3 PMAC compatible connectors**

The three PMAC compatible connectors (Axis 1, Axis 2, and Axis 3) provide their respective axis's position data and clock outputs to the user's system via a flat-ribbon cable. The ribbon cable connects directly to a Delta Tau<sup>®</sup> ACC-14D when used with a PMAC servo control system as shown in figures 4-1 and 4-2 in Chapter 4, “System Configuration,” of this guide. See Table 2-1 for the PMAC ribbon cable's pin assignments.

#### ***Axis 1, 2, and 3 Position Data output***

This position data is a 24-bit, TTL-level output, which has a resolution of approximately 0.3 nm in a plane mirror system. The output is comprised of the 24 low-order bits of the 32-bit position data.

#### ***Axis 1, 2, and 3 Position Data Clock input/output***

The Agilent N1231A board implements PMAC “Method 4” Latching/Handshaking, where a high OCLK input signal instructs the board to latch data. It then responds with a high ICLK output signal to indicate that the data has been latched (see PMAC Handshake in Chapter 6, “Specifications”).

## PC Backplane Inputs / Outputs

The PC-compatible controller's backplane I/O (or PCI bus) is connected to the Agilent N1231A board via the board's **PCI Interface** edge connector. As shown in Figure 5-1 of Chapter 5, this connection allows the board to receive power, control, and data inputs as well as output position, velocity, status, and interrupts to the controller.

In general, communication between the user's application program running on the controller and the N1231A board is over the PCI bus.

The following subsections lists characteristics of the input and output signals on this bus.

### PCI Bus Inputs

Besides supplying +5V and  $\pm 12$ V power to the board, the PCI interface consists of many registers that can be written. Chapter 3, "Programming the Registers," covers all the registers in detail, but the following data items can be set for each axis:

- Preset position value
- Stage direction
- Filter setup
- IRQ setup
- Sample/Preset control

### PCI Bus Outputs

#### ***Axis 1, Axis 2, Axis 3 Position and Velocity Data***

- data rates over 200k samples per second (one axis, using a 500 MHz Pentium III class computer — see Chapter 6, "Specifications," for maximum theoretical rates)
- Not interrupt driven

#### ***Position system status and errors***

- Available by reading status register
- Interrupts can be generated on error conditions

See the following section for more specifics on status and error detection.



---

## Error Detection

Position system status and errors are available by reading the status registers (see Chapter 3, “Programming the Registers” in this guide). The Agilent N1231A can be programmed to generate a PCI interrupt when any of these errors occurs.

---

## Cables

### PMAC compatible ribbon cable connector pin assignments

This 50-pin flat-ribbon cable connects to the PMAC ACC-14D board (same configuration for all three ribbon cable connectors: Axis 1, Axis 2, and Axis 3).

**Table 2-1. Ribbon Cable connector pin assignments (See Note 1 below)**

Pin	Description	Pin	Description
1	P23	2	Ground
3	P22	4	Ground
5	P21	6	Ground
7	P20	8	Ground
9	P19	10	Ground
11	P18	12	Ground
13	P17	14	Ground
15	P16	16	Ground
17	P15	18	Ground
19	P14	20	Ground
21	P13	22	Ground
23	P12	24	Ground
25	P11	26	Ground
27	P10	28	Ground
29	P9	30	Ground
31	P8	32	Ground
33	P7	34	Ground
35	P6	36	Ground
37	P5	38	Ground
39	P4	40	Ground
41	P3	42	Ground
43	P2	44	Ground
45	P1	46	ICLK output (see Note 2)
47	P0	48	OCLK input (see Note 3)
49	NC	50	Ground

Note 1: Same pin assignments for Axis 1, Axis 2, and Axis 3 connectors.

Note 2: High ICLK means data has been latched by the N1231A.

Note 3: High OCLK instructs N1231A to latch data.

Note 4: PMAC uses Pin 44 for error output (not provided by N1231A).

Note 5: PMAC uses Pin 49 for I/O Pull-ups (not provided by N1231A).

**Cables**



---

## Chapter Contents

This chapter provides memory map and registers information.

This chapter is organized as follows:

- Memory Map page 3-3
- Bitmaps of Registers page 3-4
  - Bitmap A page 3-4
  - Bitmap B page 3-4
  - Bitmap C page 3-4
  - Bitmap D page 3-5
  - Bitmap E page 3-6
  - Bitmap F page 3-6
  - Bitmap G page 3-7
  - Bitmap H page 3-7
  - Bitmap I page 3-7
- Velocity Register page 3-8
  - Velocity equation page 3-8
  - Error calculation page 3-8

# Memory Map

Table 3-1. Memory Map

Offset from PCIBAR2	Bits	Read	Write
0x0000	32	Position (Axis 3)*	Reserved
0x0004	32	Reserved	Reserved
0x0008	32	Velocity (Axis 3)**	Reserved
0x000C	32	Reserved	Reserved
0x0010	16	Control Setup bits (Axis 3/Intr): Bitmap A	Control Setup bits (Axis 3/Intr): Bitmap A
0x0012	16	Error Status bits (Axis 3/REF): Bitmap B	Clear Error bits (Axis 3/REF): Bitmap B
0x0014	16	Reserved	Reserved
0x0016	16	Interrupt mask bits: Bitmap C	Interrupt mask bits: Bitmap C
0x0018	32	Preset position (Axis 3)	Preset position (Axis 3)
0x001C	16	Filter settings: Bitmap D	Filter settings: Bitmap D
0x001E	16	Reserved	Reserved
0x0020	16	Sample Ready bits: Bitmap E	Sample & Preset bits: Bitmap F
0x0022	16	Reserved	Reserved
0x0024	32	Reserved	Reserved
0x0028	32	Reserved	Reserved
0x002C	32	FPGA 1 Rev 0x0001xyyy	Reserved
0x0030	32	FPGA 1 Product ID 0x0001231A	Reserved
0x0100	32	Position (Axis 2)*	Reserved
0x0104	32	Reserved	Reserved
0x0108	32	Velocity (Axis 2)**	Reserved
0x010C	32	Reserved	Reserved
0x0110	16	Control Setup bits (Axis 2/Axis 1): Bitmap G	Control Setup bits (Axis 2/Axis 1): Bitmap G
0x0112	16	Error Status bits (Axis 2/Axis 1): Bitmap H	Clear Error bits (Axis 2/Axis 1): Bitmap H
0x0114	16	Reserved	Reserved
0x0116	16	Reserved	Reserved
0x0118	32	Preset position (Axis 2)	Preset position (Axis 2)
0x011C	32	Preset position (Axis 1)	Preset position (Axis 1)
0x0120	32	Position (Axis 1)*	Reserved
0x0124	32	Reserved	Reserved
0x0128	32	Velocity (Axis 1)**	Reserved
0x012C	32	FPGA 2 Rev 0x0002xyyy	Reserved
0x0130	32	FPGA 2 Product ID 0x0001231A	Reserved

\* Position and Velocity registers changed only by a software sample request or a software Preset request. See Sample register.

\*\*The value indicated in the Velocity register can have an error of up to  $\pm 25$  ppm over the rated temperature range of the N1231A

## Bitmaps of Registers

Position and Velocity registers are changed only by a software sample request or a software Preset request. (See Bitmap F).

### Bitmap A

**Table 3-2. Control Setup Bits for Axis 3 and Interrupt (Read/Write Offset 0x0010 from PCIBAR2)**

Bit	15	14	13	12	11	10	9	8
Function	Reserved	Reserved	Reserved	Global IRQ Enable	Reserved	Reserved	Reserved	Reserved

Bit	7	6	5	4	3	2	1	0
Function	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Axis 3 Direction	Reserved

### Bitmap B

**Table 3-3. Error Status Bits for Axis 3 and Ref (Read/Write Offset 0x0012 from PCIBAR2)**

Bit	15	14	13	12	11	10	9	8
Function	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved

Bit	7	6	5	4	3	2	1	0
Function	Reserved	Reserved	Reference Glitch	Reference No Signal	Reserved	Reserved	Axis 3 Glitch	Axis 3 No Signal

Error Status bits remain set until cleared by writing “1” to each error bit in same register. Also clears IRQ if no other enabled IRQ error conditions exist.

### Bitmap C

**Table 3-4. Interrupt Mask Bits (Read/Write Offset 0x0016 from PCIBAR2)**

Bit	15	14	13	12	11	10	9	8
Function	Reserved	Reserved	Reference Glitch	Reference No Signal	Reserved	Reserved	Axis 1 Glitch	Axis 1 No Signal

Bit	7	6	5	4	3	2	1	0
Function	Reserved	Reserved	Axis 3 Glitch	Axis 3 No Signal	Reserved	Reserved	Axis 2 Glitch	Axis 2 No Signal

**Bitmaps of Registers**

Set bit to “1” in Interrupt Mask to enable an IRQ to be generated if corresponding error occurs. Must also have Global IRQ Enable bit set

Error status bit will be set if the error occurs, whether or not these mask bits are set.

**Bitmap D****Table 3-5. Filter Settings (Read/Write Offset 0x001C from PCIBAR2)**

Bit	15	14	13	12	11	10	9	8
Function	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved

Bit	7	6	5	4	3	2	1	0
Function	Reserved	Reserved	Reserved	Filter Enabled	Filter Kv		Filter Kp	

**Table 3-6. Kv and Kp digital filters**

Kv	Kp	Glitch Threshold			Following Error (for PM optics)
		Linear	PM	Hi. Res.	
0 ( $2^{-16}$ )	0	400 g	200 g	100 g	1.6 nm/g
	1	400 g	200 g	100 g	1.6 nm/g
	2	400 g	200 g	100 g	1.6 nm/g
	3	400 g	200 g	100 g	1.6 nm/g
1 ( $2^{-15}$ )	0	800 g	400 g	200 g	0.8 nm/g
	1	800 g	400 g	200 g	0.8 nm/g
	2	800 g	400 g	200 g	0.8 nm/g
	3	800 g	400 g	200 g	0.8 nm/g
2 ( $2^{-14}$ )	0	1600 g	800 g	400 g	0.4 nm/g
	1	1600 g	800 g	400 g	0.4 nm/g
	2	1600 g	800 g	400 g	0.4 nm/g
	3	1600 g	800 g	400 g	0.4 nm/g
3 ( $2^{-13}$ )	0	3200 g	1600 g	800 g	0.2 nm/g
	1	3200 g	1600 g	800 g	0.2 nm/g
	2	3200 g	1600 g	800 g	0.2 nm/g
	3	3200 g	1600 g	800 g	0.2 nm/g

Refer to Appendix E, “Filter Settings in Bitmap D,” in this guide for details.

**Bitmaps of Registers****Bitmap E****Table 3-7. Sample Ready Bits (Read/Write Offset 0x0020 from PCIBAR2)**

Bit	15	14	13	12	11	10	9	8
Function	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Axis 1 Valid Pos/Vel	Axis 1 Error

Bit	7	6	5	4	3	2	1	0
Function	Reserved	Reserved	Axis 3 Valid Pos/Vel	Axis 3 Error	Reserved	Reserved	Axis 2 Valid Pos/Vel	Axis 2 Error

Valid Pos/Vel indicates that a software sample is valid and available to read in position or velocity register. “Error” indicates that an error has occurred either on that axis or on the reference. If an error occurs *after* a valid sample has been taken, the position and velocity registers will still contain the most recent samples, but both Valid and Error bits will be set. The Error bit for this axis is cleared after all errors (not just masked errors) are cleared in Error Status register for this axis AND a Preset operation is performed.

An error on the Reference will also set both Axis 1 Error, Axis 2 Error, and Axis 3 Error. (In reality, Sample Pos and Sample Vel may indeed update their position and velocity registers even if error conditions continue to exist, but Valid Pos and Valid Vel bits will not be set until after a Preset operation is performed.)

**Bitmap F****Table 3-8. Sample and Preset Bits (Write Offset 0x0020 from PCIBAR2)**

Bit	15	14	13	12	11	10	9	8
Function	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Axis 1 Sample Pos/Vel	Axis 1 Preset

Bit	7	6	5	4	3	2	1	0
Function	Reserved	Reserved	Axis 3 Sample Pos/Vel	Axis 3 Preset	Reserved	Reserved	Axis 2 Sample Pos/Vel	Axis 2 Preset

Set bits initiate software samples or presets of the specified axes. Presets are required to clear the axis error bit and activate the corresponding Valid Pos/Vel bits (when an axis is Preset, its Valid Pos/Vel bit is enabled and the Preset Position value is written into the Position register and 0 into the Velocity register). If Preset AND Sample bits are both set for any axis, the sample bit is ignored and will be cleared.



**Bitmaps of Registers****Bitmap G****Table 3-9. Control Setup Bits for Axis 2 and Axis 1 (Read/Write Offset 0x0110 from PCIBAR2)**

Bit	15	14	13	12	11	10	9	8
Function	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved

Bit	7	6	5	4	3	2	1	0
Function	Reserved	Reserved	Reserved	Reserved	Reserved	Axis 1 Direction	Axis 2 Direction	Reserved

**Bitmap H****Table 3-10. Error Status Bits for X1 and X2 (Read/Write Offset 0x0112 from PCIBAR2)**

Bit	15	14	13	12	11	10	9	8
Function	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved

Bit	7	6	5	4	3	2	1	0
Function	Reserved	Reserved	Reserved	Reserved	Axis 1 Glitch	Axis 1 No Signal	Axis 2 Glitch	Axis 2 No Signal

Error Status bits remain set until cleared by writing “1” to each error bit in same register. Also clears IRQ if no other enabled IRQ error conditions exist.

**Bitmap I****Table 3-11. Board and JTAG Read Register (Read Offset 0x0000 from PCIBAR3)**

Bit	15	14	13	12	11	10	9	8
Function	Board Hardware Rev							

Bit	7	6	5	4	3	2	1	0
Function	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	FPGA Done	Reserved

---

## Velocity Register

The Velocity register is a 32-bit value register.

### Velocity equation

Bit 0 (or LSB, Least Significant Bit) of the Velocity register, uncompensated is:

$$(20 \times 10^6 \text{ Hz} \div 2^{22}) \times \lambda \div \text{Fold Factor}$$

In systems with Plane Mirror Optics the Fold Factor is 4; therefore, the equation simplifies to:

$$20 \times 10^6 \text{ Hz} \times \lambda \div 2^{24}$$

If  $\lambda$  equals 632.99 nm, then

bit 0 (LSB) of the Velocity register = 754.6 nm/second

### Error calculation

The value indicated in the Velocity register can have an error of up to  $\pm 25$  ppm over the rated temperature range of the Agilent N1231A.

The  $20 \times 10^6$  (20 MHz) is the temperature sensitive component of the velocity equation.



## Chapter Contents

This chapter describes how the Agilent N1231A PCI Three-Axis Board and PMAC ISA Board are installed and used in the Agilent laser measurement system.

This chapter is organized as follows:

- System Overview page 4-2
    - Laser head and receivers page 4-3
    - Environmental sensors page 4-3
    - Delta Tau<sup>®</sup> boards (PMAC and ACC-14D) page 4-4
    - Typical 3-axis laser/PMAC system configuration using the Agilent 10780C/F receivers page 4-4
    - How the boards are connected page 4-6
    - Typical 3-axis laser/PMAC system configuration using the Agilent E1708A or E1709A receivers page 4-6
    - Setting the jumpers on the ACC-14D ISA boards page 4-8
- 

## System Overview

The Agilent N1231A PCI Three-Axis Laser Board's hardware outputs are optimized for use with the Delta Tau<sup>®</sup> PMAC servo controller (requires two Delta Tau<sup>®</sup> ACC-14D boards). These boards, along with an Agilent 10886A PC Compensation Board, the Agilent laser measurement system, and a computer comprise a system that ensures the accuracy of a machine's motion and positioning. Controlled through your PC-compatible controller (with Windows NT 4.0<sup>®</sup> installed), the system is able to collect measurement data from, and control the position of, three axes of motion.

The basic function of the Agilent N1231A board is to compare the phase of the reference and measurement signals entering on the four rear-panel connectors. The processed signals are then available as raw position and velocity values via the three hardware output connectors and the PCI interface.

The two boards from Delta Tau perform the servo control function with the ACC-14D board providing the interface to the parallel position word generated by the Agilent N1231A board.

## System Overview

The overall system accuracy is dependent on knowing the wavelength of the laser light, which is a function of the air pressure, temperature, humidity, and composition. The Agilent 10886A PC Compensation Board can be set up to measure all but gas composition.

The rest of the laser measurement system consists of a laser head, measurement optics (not discussed in this user's guide), measurement receivers, and various cables.

### Laser head and receivers

There are two signals involved in position calculation, Reference and Measurement. The Reference signal generally comes from the laser head itself and is the signal to which the Measurement signal is compared for phase changes. The Measurement signal comes from a receiver (either Agilent 10780C/F or E1708A or E1709A), which converts the light emanating from a measurement optic into an electrical signal. This Measurement signal changes in phase with respect to the Reference signal as the stage moves. Using Plane Mirror optics, the phase of the electrical signal changes 360 degrees for every 1/4th of the wavelength-of-light (approximately 633 nanometers), or about 158 nm. This phase change is independent of the laser head's split frequency.

See the *Agilent Laser and Optics User's Manual* (Agilent part number 05517-90045) for a complete discussion of the laser heads, receivers and all measurement optics.

### Environmental sensors

The Agilent 10886A PC Compensation Board and associated sensors (Agilent 10751C/D Air Sensor and Agilent 10757D/E/F Material Temperature Sensor) compensate for environmental conditions (air temperature, air pressure, air humidity, and stage temperature) in the work area of the laser measurement system.

When installed and operating in the PC-compatible controller, the Agilent 10886A provides pressure, temperature, and humidity values that are converted by supplied software into a compensation number or correction factor. Some code must be written that sets up the board, requests this compensation number, and sends it to the correct register of the PMAC Board. The Agilent supplied software consists of a driver library implemented as a DLL. It is available for Windows NT 4.0.

## System Overview

### Delta Tau<sup>®</sup> boards (PMAC and ACC-14D)

The PMAC Board controls multiple axes, sending signals to the servomotors (through an amplifier) based on the position feedback device and the instantaneous desired location. The ACC-14D interface board allows the Agilent laser measurement system to be the position feedback device with no custom interface circuits required. From the PMAC's perspective, the Agilent laser measurement system is a position feedback encoder.

Note that the ACC-14D board is only capable of 48 bits total. Since the Agilent N1231A outputs a 24-bit word, only two axes can go into each ACC-14D board. Thus, a second ACC-14D Board is needed for the third axis as shown in figures 4-1 and 4-2. See the Delta Tau manuals for full details on proper configuration and connection of the ACC-14D boards to the PMAC controller board.

### Typical 3-axis laser/PMAC system configuration using the Agilent 10780C/F receivers

Figure 4-1 shows a system that uses the Agilent N1231A PCI Three-Axis Laser Board, PMAC system, and the Agilent 10780C/F receivers. It consists of the items listed below.

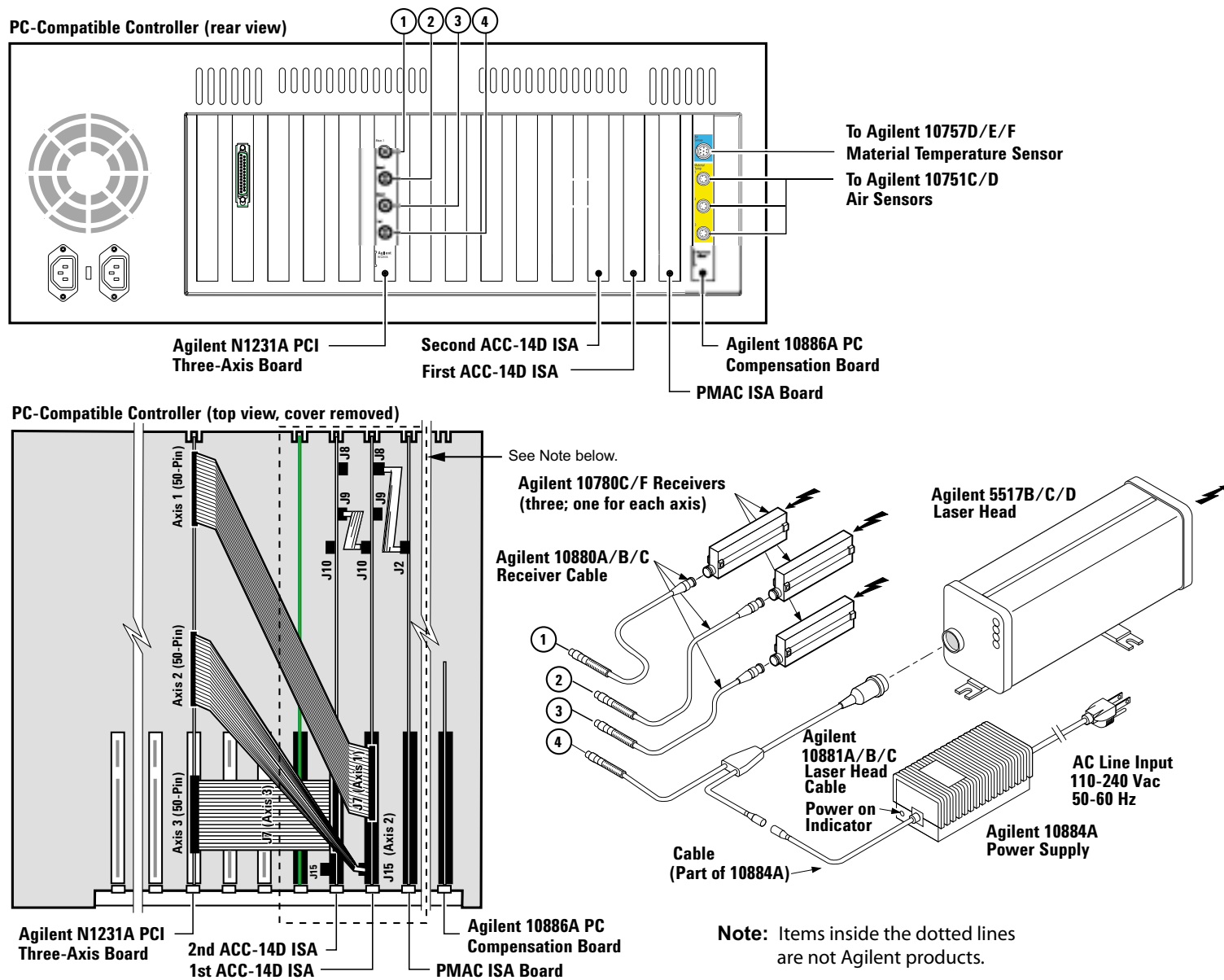
#### Agilent products:

- Agilent N1231A PCI Three-Axis Laser Board
- Agilent 5517B/C/D Laser Head
- Agilent 10881A/B/C Laser Head Cable
- Agilent 10884A Power Supply
- Agilent 10780C/F (three needed; one for each axis)
- Agilent 10880A/B/C Receiver Cable (three needed; one for each Agilent 10780C/F Receiver)
- Agilent 10886A PC Compensation Board
- Agilent 10751C/D Air Sensor
- Agilent 10757D/E/F Material Temperature Sensor

#### Delta Tau products:

- PMAC board
- ACC-14D interface board (ISA bus, two required for a 3-axis system)

Figure 4-1. Example of the N1231A and PMAC system with 10780C/F receivers



## System Overview

### How the boards are connected

As shown in figures 4-1 and 4-2, the Agilent N1231A board's Axis 1 and Axis 2 24-bit position values are sent to the PMAC board via the 50-pin flat-ribbon cable from J8 of the *first* ACC-14D board to J2 of the PMAC board.

The Agilent N1231A board's Axis 3 24-bit position value is sent to the PMAC board via the 50-pin flat-ribbon cable from J9 of the *second* ACC-14D board, to J10 and J8 of the *first* ACC-14D board, and finally to J2 of the PMAC board.

### Typical 3-axis laser/PMAC system configuration using the Agilent E1708A or E1709A receivers

Figure 4-2 shows a system that uses the Agilent N1231A PCI Three-Axis Laser Board, the PMAC system, and the Agilent E1708A or E1709A receivers. It consists of the items listed below.

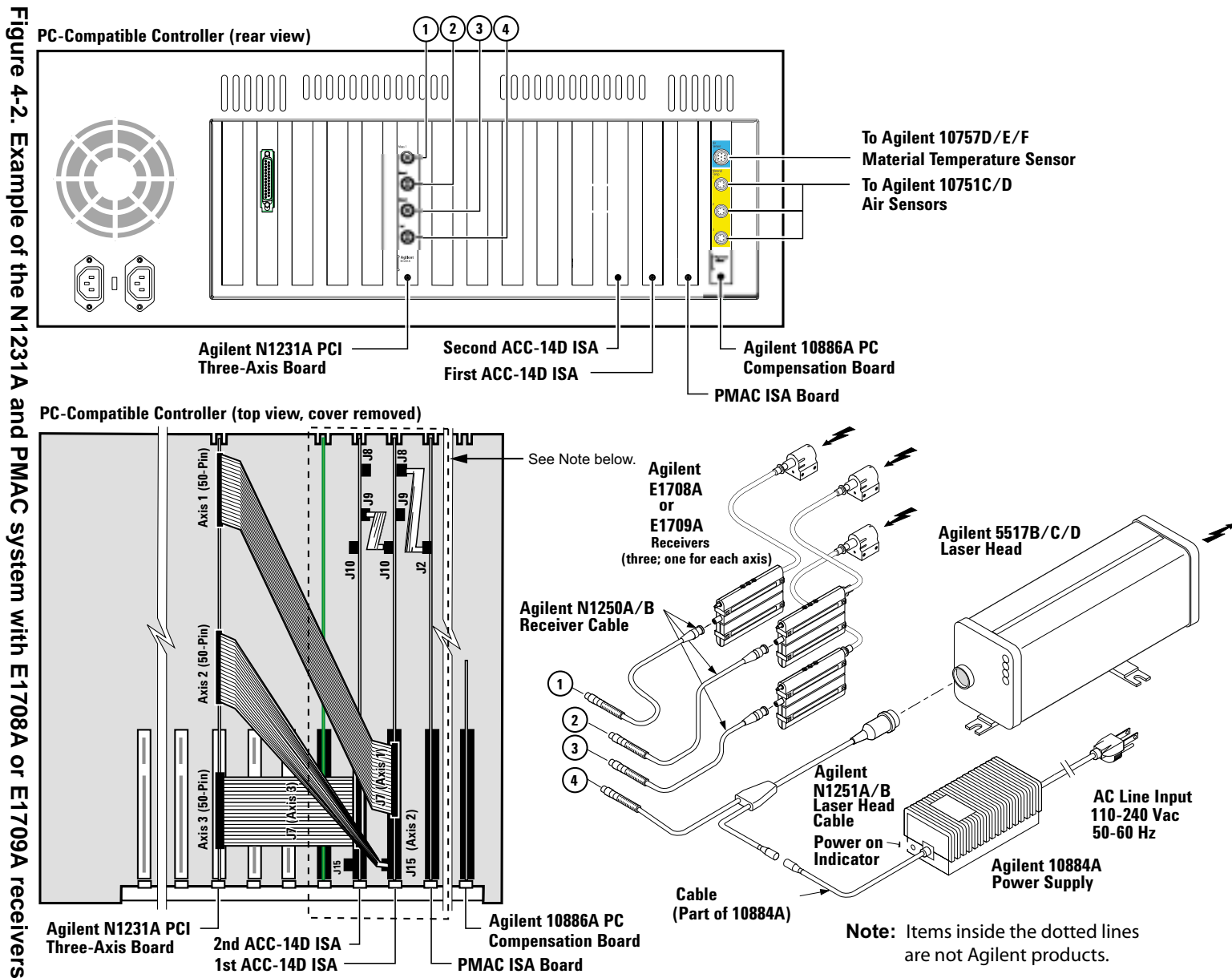
#### Agilent products:

- Agilent N1231A PCI Three-Axis Laser Board
- Agilent 5517B/C/D Laser Head
- Agilent N1251A/B High Performance Laser Head Cable
- Agilent 10884A Power Supply
- Agilent E1708A or E1709A (three needed; one for each axis)
- Agilent N1250A/B High Performance Receiver Cable (three needed; one for each Agilent E1708A or E1709A Receiver)
- Agilent 10886A PC Compensation Board
- Agilent 10751C/D Air Sensor
- Agilent 10757D/E/F Material Temperature Sensor (up to three can be used)

#### Delta Tau products:

- PMAC board
- ACC-14D interface board (ISA bus, two required for a 3-axis system)





## System Overview

### Setting the jumpers on the ACC-14D ISA boards

For the ACC-14D ISA boards to translate the 24-bit word from the Agilent N1231A, the jumpers on the ACC-14D ISA boards must be set as follows:

Acc14D/V option3

E2,E1,E11,E10,E41,E3: 1-2

E31,E30,E29,E38,E39,E40: don't matter

E27,E25,E23,E32,E34,E36: don't matter

E28,E26,E24,E332,E354,E376: don't matter

Method4 setting (High OCLK means latch and high ICLK means latched)

E5,E6 ON

E8,E9 OFF

E21,E22 1-2

No error signal provided

E4 OFF

E7 OFF

\*\*E12-E20 depends on the system:

*For Acc14 which plugs into PMAC*

E20 OFF

*For daisy chained Acc 14*

E20 ON

*First card*

E12 ON, E13-E17 OFF

*Second card*

E13 ON, E12 OFF, E14-E17 OFF

---

#### NOTE

The above jumper settings information is provided based on the current ACC-14D ISA board. This jumper information was provided December 2001.

---



---

## Chapter Contents

This chapter provides a functional block description of the circuitry in the Agilent N1231A Board.

This chapter is organized as follows:

- How It Works page 5-3
  - Position Measurement Block page 5-3
- Accuracy & Resolution page 5-4
  - Axis 1, 2, and 3 position resolution page 5-4
  - Fixed delay page 5-4
  - Wavelength compensation page 5-4

## How It Works

Figure 5-1 shows the Agilent N1231A functional block diagram.

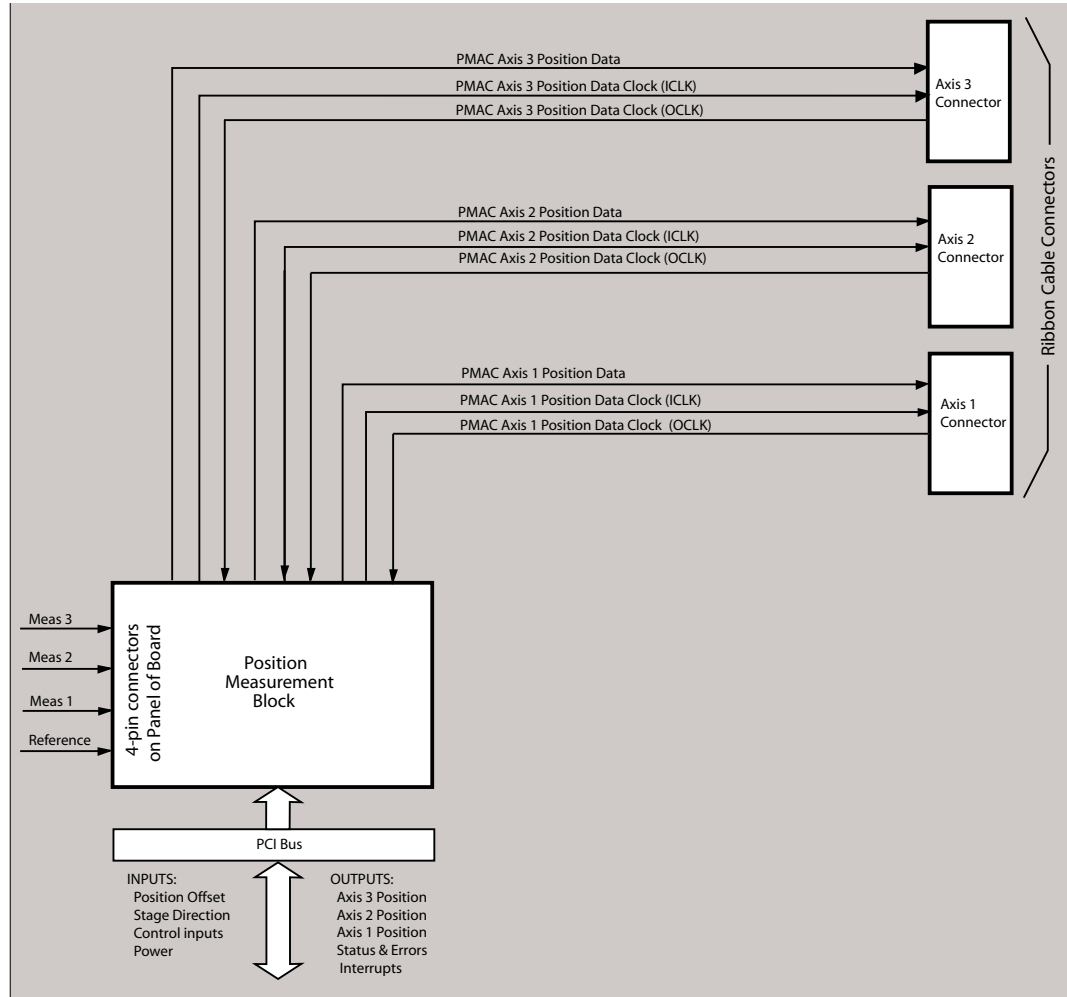


Figure 5-1. Agilent N1231A board overall block diagram

### Position Measurement Block

The Position Measurement Block provides basic functionality similar to that of other Agilent Laser Axis Board. It receives measurement and reference signals from receivers in the interferometry system, monitors the relative phase of these signals, and produces an uncompensated position number. Using plane mirror optics (4-pass), each change in relative phase of 360 degrees corresponds to a change in stage position of  $\lambda/4$ , where  $\lambda$  is the wavelength of the laser light (approximately 632.99 nm). Using interpolation techniques, the Position Measurement Block divides this by an additional factor of 512 yielding an overall resolution of  $\lambda/2048$ , or approximately 0.3 nm.

---

## Accuracy & Resolution

### Axis 1, 2, and 3 position resolution

One system accuracy component is the measurement resolution of the Axis 1, Axis 2, and Axis 3 position data. This resolution is approximately 0.3 nm when using plane mirror optics, which translates to  $\pm 0.15$  nm for system repeatability and accuracy computations (see *Agilent Laser and Optics User's Manual* (Agilent part number 05517-90045) for a more detailed discussion of system repeatability accuracy).

### Fixed delay

Fixed delays in the system result in an offset of the measured value from the true value (i.e., the measured value gives the position of the system at some fixed time in the past). In this system, this Fixed Data Age is approximately 0.6  $\mu$ s. The effect of this delay is shown in Table 5-1, but it is not included in the accuracy calculations, above, because its effect can be compensated (assuming the velocity is constant). Compensation for fixed delays must be handled in the user-supplied software.

**Table 5-1. Fixed errors (can be compensated)**

Velocity	Fixed Data Age (approximate)	Fixed Error caused by Fixed Data Age
20 mm/s	0.6 $\mu$ s	12 nm
25 mm/s	0.6 $\mu$ s	15 nm
36 mm/s	0.6 $\mu$ s	21.6 nm
50 mm/s	0.6 $\mu$ s	30 nm
80 mm/s	0.6 $\mu$ s	48 nm
123 mm/s	0.6 $\mu$ s	73.8 nm

### Wavelength compensation

Measurement accuracy is dependent on accurately knowing the wavelength of the laser light. This varies with the environment (primarily temperature and pressure). The N1231A assumes that this wavelength compensation is handled in the user-supplied software.



---

## Chapter Contents

This chapter provides specifications and operating characteristics for the Agilent N1231 PCI Three-Axis Laser Board.

The information in this chapter is organized as follows:

- General Specifications page 6-3
  - Measurement Range and Resolution page 6-3
  - Maximum Axis Velocity (slew rate) page 6-3
  - Internal Data Update Rate page 6-3
- Electrical Specifications page 6-4
  - Receiver Signal Inputs page 6-4
  - Hardware Position Outputs page 6-4
  - PCI Bus I/O Characteristics page 6-4
  - Electrostatic Discharge (ESD) page 6-5
  - Power Requirements page 6-5
  - Cooling requirements page 6-5
  - Timing data page 6-6
- Mechanical Specifications page 6-6



## General Specifications

### Measurement Range and Resolution

Table 6-1 lists laser optics and their corresponding resolution and range.

**Table 6-1. Resolution and range**

Optics	Resolution	Range*	
		Using PMAC compatible connector (24 bits)	Using PCI connector (32 bits)
Linear or Single Beam	$\lambda$ /1024 or 0.6 nm	± 5 mm	± 1.3 m
Plane Mirror	$\lambda$ /2048 or 0.3 nm	± 2.5 mm	± 0.6 m
High Resolution	$\lambda$ /4096 or 0.15 nm	± 1.2 mm	± 0.3 m

\*The “Range” shown in the above table is based on the number of position bits read via the PMAC compatible connector or over the PCI bus. Software in the PMAC or PC may be able to extend this range indefinitely by detecting and tracking overflow of the value read.

### Maximum Axis Velocity (slew rate)

Table 6-2 lists the Maximum Axis Velocity of different Agilent Laser Head models (5517A, 5517B, 5517C, 5517D).

**Table 6-2. Maximum Axis Velocity (slew rate)**

Optics	Agilent 5517A	Agilent 5517B	Agilent 5517C	Agilent 5517D
Linear or Single Beam	406 mm (16 in)/sec	508 mm (20 in)/sec	711 mm (28 in)/sec	1000 mm (39 in)/sec
Plane Mirror	203 mm (8 in)/sec	254 mm (10 in)/sec	356 mm (14 in)/sec	500 mm (20 in)/sec
High Resolution	102 mm (4 in)/sec	127 mm (5 in)/sec	178 mm (7 in)/sec	250 mm (10 in)/sec

The Agilent N1231A board will accept future Agilent laser head upgrades up to 5600 mm/sec with linear optics, 2800 mm/sec with plane mirror optics, and 1400 mm/sec with high-resolution optics.

### Internal Data Update Rate

The board calculates new position and velocity values for each axis every 50 ns.

## Electrical Specifications

The electrical specifications describe the electrical characteristics and properties of inputs and outputs for the Agilent N1231A PCI Three-Axis Laser Board.

### Receiver Signal Inputs

Differential TTL, 200 kHz to 39.8 MHz

### Hardware Position Outputs

#### Signal Levels

All signals on the PMAC compatible connectors are LVTTTL level signals capable of driving 12 mA at 3.3 volts and sinking 12 mA at 0.4 volts.

#### Data Age Relative to OCLK

520 ns  $\pm$ 25 ns (ambiguity due to OCLK sync to internal 20 MHz clock)

#### Data Age Relative to ICLK

845 ns  $\pm$ 1 ns (over full temp and voltage range)

#### Data Format

24-bit binary with LSB equal to resolution shown in Table 6-1.

#### Maximum Data Rate

4 MHz (see Timing data)

### PCI Bus I/O Characteristics

#### PCI Compliance

All signals on the PCI connector conform to rev 2.2 of the PCI bus specification. As such they operate at 33 MHz and either 3.3 or 5.0 volts. The board supports 16- and 32-bit data transfers and can generate an IRQ.

#### Position and Velocity Data Age relative to the rising edge of the PCI bus Clock on the bus write cycle requesting a Sample

197 ns  $\pm$ 12.5 ns

#### Position Data Format

32-bit binary word with LSB equal to resolution shown in Table 6-1.

### **Velocity Data Format**

32-bit binary word with LSB equal to  $78125 / 32 \times$  (resolution, in nm, shown in Table 6-1) nm per second.

### **Theoretical Maximum Data Rates**

Take the reciprocal of times listed in Table 6-3 (Times assume infinitely fast computer using 0 time to process the data).

**Table 6-3. Theoretical Minimum Data Collection Times\***

Number of Axis	Position or Velocity	Position and Velocity
1 Axis	2.5 $\mu$ s	3.5 $\mu$ s
2 Axes	3.5 $\mu$ s	5.5 $\mu$ s
3 Axes	4.5 $\mu$ s	7.5 $\mu$ s

\* These time values represent only what is required by the N1231A board to handle read and write requests. They do not include any time required by the user's program, the API library, the drivers, the operation system, or any other software components.

### **Electrostatic Discharge (ESD)**

When the product is tested with 8kV air discharge or 4kV contact discharge in accordance with IEC 61000-4-2, errors that will affect the measurement data may occur during the disturbance. The axis board is designed to sense such incidents and to stop further measurements. A reset via software must be performed on the axis board to discard the erroneous data.

### **Power Requirements**

- +5 Vdc  $\pm 0.25$ V /  $-0.125$  V at approximately 2.8 Amps
- +12 Vdc  $\pm 0.5$ V at approximately 50 milliamps.
- -12 Vdc  $\pm 0.5$ V at approximately 50 milliamps.

### **Cooling requirements**

#### **Operating Ambient Temperature**

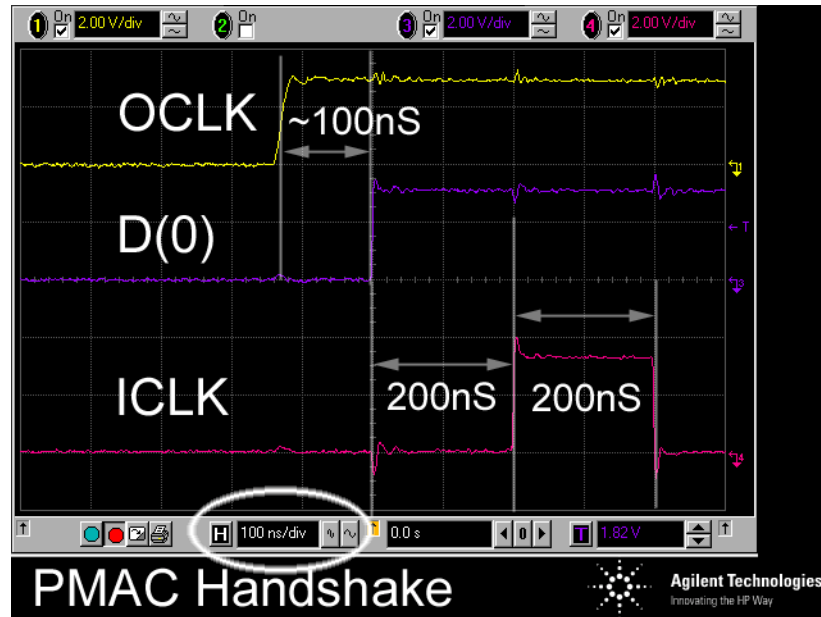
0° to 40° C (32° to 104° F)

#### **Air Flow**

Greater than 20 linear meters/minute, parallel to PCI connector.

## Timing data

### PMAC handshake



#### **Minimum time between rising edges of OCLK**

250 ns (shorter times will update the output data but will prevent ICLK from returning low. Rising edges between the leading rising edge and the data being updated will be ignored).

---

## Mechanical Specifications

The mechanical specifications describe physical dimensions of the Agilent N1231A PCI Dual Laser Axis Board.

See Figure 6-1 for dimensions of the Agilent N1231A.

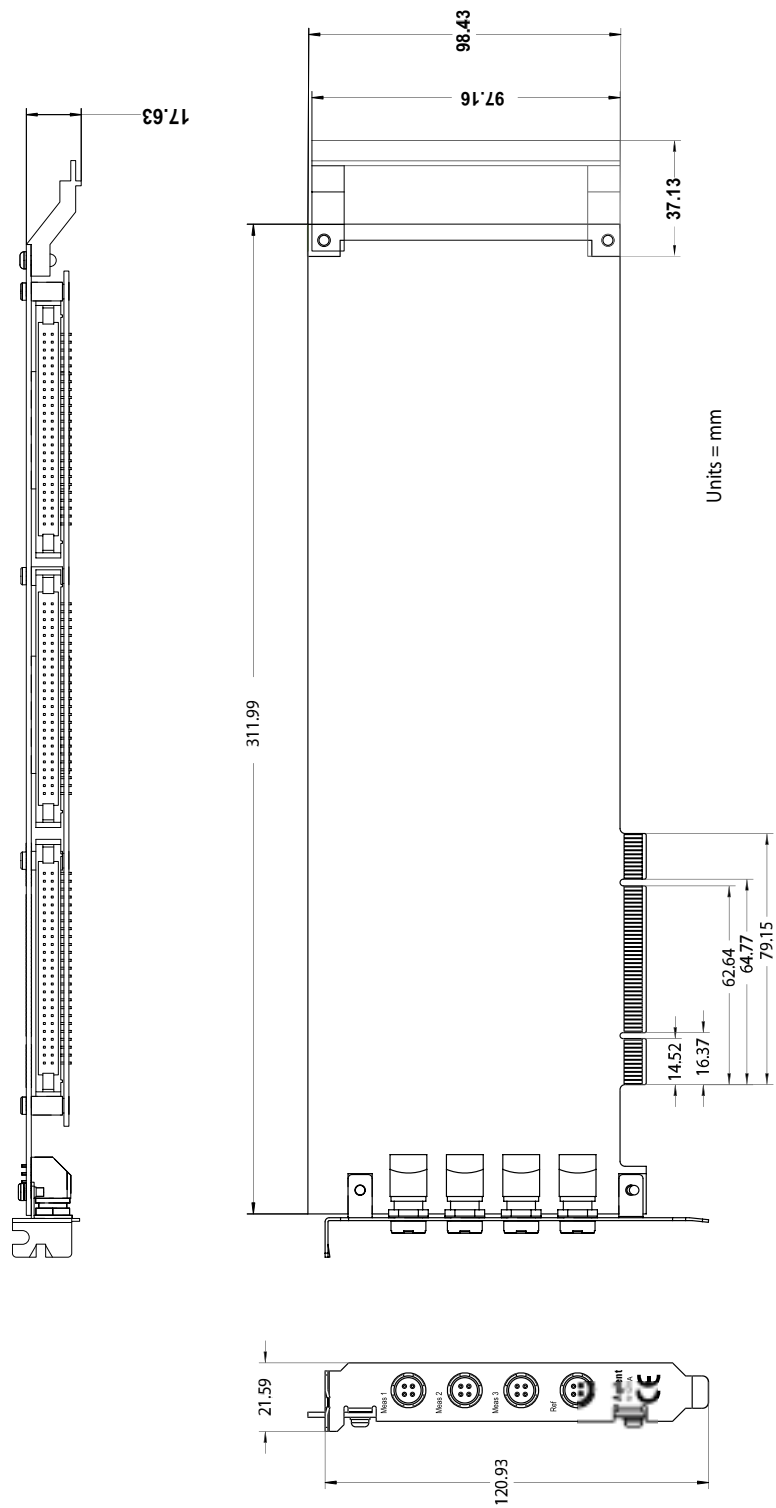


Figure 6-1. Agilent N1231A physical dimensions



---

A

---

Downloading Firmware

**Overview**

---

## Overview

If Agilent determines that an update is needed for the Agilent N1231A board's FPGA code, it will be provided as software to be used under a Microsoft<sup>™</sup> NT operating system to directly download FPGA firmware into the EEPROMs via the PCI bus. This is provided in the form of executable files with the FPGA code embedded in the files.

To download firmware, the Agilent N1231A board must be installed in a system running Microsoft NT.

---

## Downloading New Firmware

- 1 Locate the new version of firmware that you want to install.

It should have a name similar to \* **n1231a\_fpga\_rev1.exe**.

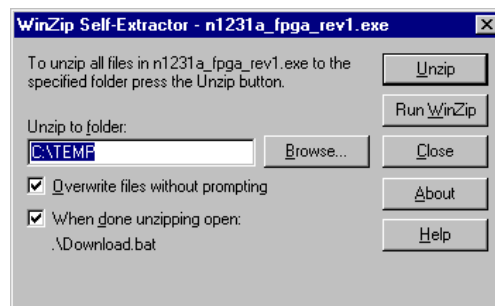
This firmware may be in an E-mail message or on a LAN.

- 2 If the firmware is attached to an E-mail message, double-click the \* **n1231a\_fpga\_rev1.exe** file, then open it.

OR

If the firmware is on a LAN, copy the \* **n1231a\_fpga\_rev1.exe** file to a drive on your PC-compatible controller, then Double-click it.

A dialog box similar to Figure A-1 appears.



**Figure A-1. WinZip Self-Extractor dialog box**

- 3 Click **Unzip** button.
- 4 Four files will be unzipped as will be indicated by a dialog box; click **OK** in the dialog box.

A DOS window appears as shown in Figure A-2.



## Downloading New Firmware

```

C:\WINDOWS\System32\CMD.exe
XSUF Download Program
XSUF file = n1231a_fpga_rev1.xsuf

Checking integrity of xsuf file: Passed

Found 1 card in system.

Card : Bus : Slot : FPGA1 ID : FPGA1 Rev : FPGA2 ID : FPGA2 Rev
=====
1. : 0 : 17 : 0x0001231a : 0x00010001 : 0x0001231a : 0x00020001

Enter number of card for download (or 0 to exit):

```

**Figure A-2. DOS window**

As indicated in the DOS window:

- the integrity of the xsuf file “passed”,
  - one N1231A board found in system (and board location in PC is indicated), and
  - FPGA1 and FPGA2 revisions are indicated.
- 5 Select to which N1231A board you want to download the new firmware. In this case, there is only one card installed and found as indicated in Figure A-2; therefore, press “1” key on your controller.

A DOS window asking you if you are “READY TO DOWNLOAD TO CARD” appears as shown in Figure A-3.

```

C:\WINDOWS\System32\CMD.exe
XSUF Download Program
XSUF file = n1231a_fpga_rev1.xsuf

Checking integrity of xsuf file: Passed

Found 1 card in system.

Card : Bus : Slot : FPGA1 ID : FPGA1 Rev : FPGA2 ID : FPGA2 Rev
=====
1. : 0 : 17 : 0x0001231a : 0x00010001 : 0x0001231a : 0x00020001

Enter number of card for download (or 0 to exit): 1

READY TO DOWNLOAD TO CARD:
  Bus 0, Slot 17
  Hardware Revision: 0x10
  Current FPGA1 ID: 0x0001231a, Revision: 0x00010001
  Current FPGA2 ID: 0x0001231a, Revision: 0x00020001
Do you wish to continue? <y or n>:

```

**Figure A-3. DOS window — READY TO DOWNLOAD IO CARD**

- 6 Type “Y” for yes.

## Downloading New Firmware

The downloading starts and its progress is displayed (by percentage of completeness) in the DOS window.

- 7 When downloading is completed, you can press “yes” to view the FPGA information and then press any key to exit.

This completes the download of the new firmware to the N1231A board.

- 8 See “Operational Verification of the Board, API Library, and Monitor Application” in Chapter 1, “Getting Started,” in this guide to check that the code has been installed and is operating correctly.

---

# B

---

N1231A.h and N1231A\_reg.h C/C++  
Programming Information

## Appendix Contents

---

# Appendix Contents

This appendix contains the following programming information:

- The C/C++ N1231A.h Header File page B-2
- The C/C++ N1231A\_reg.h Header File page B-22

### Note

The DLL is compiled to use the C calling convention with 32-bit (4-byte) data alignment.

---

## The C/C++ N1231A.h Header File

## Appendix B N1231A.h and N1231A\_reg.h C/C++ Programming Information

### The C/C++ N1231A.h Header File

```
/*
 * Copyright 2002 by Agilent Technologies
 *
 * File Name: N1231A.h
 *
 * Header file for the API library for the
 * Agilent Technologies N1231A PCI Three-Axis Board
 * designed for use with an Agilent Technologies
 * laser interferometry position measurement systems.
 *
 * See also file N1231A_reg.h
 */

#ifndef N1231A_H
#define N1231A_H

#include "N1231A_reg.h"

/*
 * *****
 * C/C++ Compatibility
 * *****
 */
#ifdef __cplusplus
extern "C" {
#endif

/*
 * *****
 * Export when building DLL, Import in client applications
 * *****
 */
#ifndef USE_STATIC // Should not be defined if used with N1231A.dll
#define EXPORT // Do nothing
#else
#ifdef MAKE_DLL
#define EXPORT __declspec(dllexport)
#else
#define EXPORT __declspec(dllimport)
#endif
#endif

/*
 * *****
 * DEFINITIONS OF TERMS :
 *
 * Raw Interferometer Units:
 *   Position and velocity values used by these
 *   functions are expressed in Raw Interferometer Units.
 *
 * For Position Information:
 *   Raw Interferometer Position Unit
 *   = Lambda / (512 * FoldFactor)
 *
 * For Velocity Information:
 *   Raw Interferometer Velocity Unit
 *   = (Lambda * 20E6 Hz) / (2^22 * FoldFactor)
 *
 * Where:
 *   Lambda = Wavelength of Light
 */
```

## Appendix B N1231A.h and N1231A\_reg.h C/C++ Programming Information

### The C/C++ N1231A.h Header File

```
*      FoldFactor =
*      2 for a Linear Optics System,
*      4 for a Plane Mirror Optics System, or
*      8 for a High Resolution Optics System.
*
*****
*****
*/

/*
*****
*****
* TABLE OF CONTENTS
*****
*****
*
* =====
* Typedefs, Constants, Enums, Structures
* =====
*
* Used for opening device:
*      N1231A_HANDLE
*      N1231A_IGNORE_FIELD
*      N1231A_LOCATION
*
* Used for interrupts
*      N1231A_EVT_HANDLE;
*
* Used for error returns:
*      N1231A_RETURN
*
* Used for axis identification
*      N1231A_AXIS
*
* =====
* High Level Functions
* =====
*
* N1231AOpen()
* N1231ADefaultDevice()
* N1231AClose()
* N1231AFind()
* N1231APresetRawAll()
* N1231APresetRaw()
* N1231AGetRawPosVelAll()
* N1231AGetRawPosVel()
* N1231AClearStatusBits()
* N1231AClearStatusAll()
* N1231AGetStatus()
* N1231ASetInterruptMask()
* N1231AGetInterruptMask()
* N1231ASetConfig()
* N1231AGetConfig()
* N1231ASetFilter()
* N1231AGetFilter()
* N1231AGetFPGARevs()
* N1231AGetFPGAIds()
* N1231AGetHdwRev()
* N1231AGetLibraryRev()
* N1231AGetErrStr()
*
```

**The C/C++ N1231A.h Header File**

```
* =====
* Interrupt Related Functions
* =====
*
* N1231APciInterruptEnable()
* N1231APciInterruptAttach()
*
* =====
* Lower Level Functions
* =====
*
* N1231ASamplePosVel()
* N1231AReadRawPos()
* N1231AReadRawVel()
* N1231AWriteRawPresetRegister()
* N1231AGetRawPresetRegister()
*
* =====
* Register Functions (for test use only)
* =====
*
* N1231AWriteRegisterLong()
* N1231AReadRegisterLong()
* N1231AWriteRegisterWord()
* N1231AReadRegisterWord()
*
*/

/*
*****
*****
* DECLARATIONS
*****
*****
*/

/*
*****
*****
* Typedefs, Constants, Enums, Structures
*****
*****
*/

/*
* Used for opening device
*/

typedef void *N1231A_HANDLE;

#define N1231A_IGNORE_FIELD 0xffffffff

typedef struct
{
    /*
    * Set following fields to identify device to open.
    * Values set to IGNORE_FIELD (see above) will not be used
    * when searching for a device, but will be filled in
    * by the search or open function once a device is found.
    */
    unsigned long    BusNumber;
    unsigned long    SlotNumber;
} N1231A_LOCATION;
```

## Appendix B N1231A.h and N1231A\_reg.h C/C++ Programming Information

### The C/C++ N1231A.h Header File

```
/*
 * Used for interrupts
 */

typedef void *N1231A_EVT_HANDLE;

/*
 * Used for error returns
 */
typedef short N1231A_RETURN;
enum E_N1231A_RETURN
{
    N1231A_SUCCESS                = 0,
    N1231A_ERR_HANDLE = 16, /* Leave a space for bitmap-type errors */
    N1231A_ERR_PARAM,
    N1231A_ERR_DEVICE,
    N1231A_ERR_DRIVER,
    N1231A_ERR_MEMORY,           /* Insufficient mem for device */
    N1231A_ERR_REG,              /* Invalid register address */
    N1231A_ERR_HEAP,             /* No space on heap */
    N1231A_ERR_HEAP_LOCKED, /* Should never occur */
    N1231A_ERR_BAD_AXIS,         /* Axis does not exist */
    N1231A_ERR_AXIS,            /* Axis or reference has status error or
invalid data */
    N1231A_ERR_UNKNOWN1,        /* Should never occur */
    N1231A_ERR_UNKNOWN2,        /* Should never occur */
    N1231A_ERR_UNKNOWN3,        /* Should never occur */
    N1231A_ERR_UNKNOWN4,        /* Should never occur */
    N1231A_ERR_UNKNOWN5,        /* Should never occur */
    N1231A_ERR_UNKNOWN6,        /* Should never occur */
    N1231A_ERR_MAX_EVENTS
};

/*
 * Used for axis identification
 */
typedef short N1231A_AXIS;
enum E_N1231A_AXIS
{
    AXIS_1,
    AXIS_2,
    AXIS_3
};

/*
*****
*****
 * FUNCTION DECLARATIONS
*****
*****
 */

/*
 * =====
 * High Level Functions
 * =====
 */

/*****
 */
```



**The C/C++ N1231A.h Header File**

```
EXPORT N1231A_RETURN
N1231AOpen(N1231A_LOCATION      *pDevice,
           N1231A_HANDLE        *pN1231AHandle,
           unsigned long         *pProductId);

/*
 *   Opens N1231A-type device specified by structure *pDevice.
 *
 *   If pDevice points to NULL, then this function will open the
 *   first N1231A-type device that is found.
 *
 *   If pDevice points to a structure of type N1231A_LOCATION
 *   then this function will open the first N1231A-type device
 *   matching the information in pDevice. Upon successful
 *   completion of this function any incomplete fields in
 *   *pDevice will have been filled in (to show the bus and
 *   slot of the device opened).
 *
 *   An N1231A-type device is any PCI device with the Agilent
 *   Vendor ID (0x15bc) and Device ID = 0x0a00. Future Agilent
 *   axis cards may use these same IDs, but are expected to be
 *   distinguished by the ProductId (see below).
 *
 *   Parameter *pN1231AHandle returns a handle required by other
 *   functions to reference the opened device.
 *
 *   If pProductId is not NULL, then *pProductId will be set
 *   to the ID code of FPGA1. For an N1231A card this value should
 *   be 0x0001231A. If other axis cards are present, this code can be
 *   used to verify that the correct card model was opened.
 *   NOTE THAT *pProductId is used only to return a value -- its
 *   value upon entry to this function is ignored.
 *
 *   Before exiting from the application which called this function,
 *   a call should be made to N1231AClose() to release the handle and
 *   its associated memory block.
 *
 *   The function N1231ADefaultDevice() can be called to initialize
 *   pDevice to find the first N1231A. Calling N1231AOpen() using
 *   this default pDevice is the same as calling it with a pointer
 *   to NULL, except that supplying a pointer to the N1231A_LOCATION
 *   structure allows return of information on the device found.
 *
 *   The return code specifies the success or failure of this function,
 *   as follows:
 *       N1231A_SUCCESS:           The function returned successfully.
 *       N1231A_ERR_PARAM: A passed parameter is NULL.
 *       N1231A_ERR_DEVICE:       No device found.
 *       N1231A_ERR_DRIVER:       No device driver installed.
 *       N1231A_ERR_MEMORY:       Insufficient memory.
 *       N1231A_ERR_HEAP:         No space on heap.
 */

/*****
 */
EXPORT void
N1231ADefaultDevice(N1231A_LOCATION *pDevice);
/*
 *   Fills structure *pDevice with default values.
 *   Using these default values with N1231AOpen() will cause the
 *   first N1231A in the system to be opened. If there is only
 *   one N1231A in the system, this provides a simple method
```

## Appendix B N1231A.h and N1231A\_reg.h C/C++ Programming Information

### The C/C++ N1231A.h Header File

```
*      for opening the device.  If there are multiple N1231A cards
*      then after calling this function it will be necessary to
*      modify one or more fields to specify the particular card
*      to be opened.
*
*  EXAMPLE:
*  pDevice->BusNumber  = IGNORE_FIELD;
*  pDevice->SlotNumber = IGNORE_FIELD;
*/

/*****
*/
EXPORT N1231A_RETURN
N1231AClose(N1231A_HANDLE      *pN1231AHandle);
/*
*      Closes the N1231A card specified by *pN1231AHandle.
*      If successful, *pN1231AHandle is set to NULL.
*
*      The return code specifies the success or failure of this function,
*      as follows:
*
*          N1231A_SUCCESS:      The function returned successfully
*          N1231A_PARAM:        A passed parameter is NULL.
*          N1231A_ERR_HANDLE:   N1231AHandle invalid
*          N1231A_ERR_DEVICE:   Device not closed properly.
*          N1231A_ERR_HEAP:     Error accessing heap.
*/

/*****
*/
EXPORT N1231A_RETURN
N1231AFind(const N1231A_LOCATION  *pDevice,
            unsigned int           *pNumFound,
            N1231A_LOCATION        *pDeviceArray,
            unsigned int           numMax);
/*
*      Determines the number of N1231A devices in the system
*      and (optionally) builds an array of information on
*      these devices.
*
*      If numMax is zero and pDeviceArray is NULL, then reports
*      total number of N1231A devices matching the pDevice description.
*
*      Otherwise, builds an array (at pDeviceArray) of the first numMax
*      devices found.
*
*  PARAMETERS:
*      *pDevice:
*          Pointer to structure of type N1231A_LOCATION.
*          If NULL, search will find all N1231A devices.
*          If non-NULL, search will be limited to devices
*          matching the specified criteria.
*
*      *pNumFound
*          Returns number of devices found.
*          If numMax is zero and pDeviceArray is NULL then
*          this will be the total number of devices
*          matching the description.
*          Otherwise, this number will not exceed numMax.
*
*      *pDeviceArray
*          Pointer to array of structures of type N1231A_LOCATION
*          used to return information on found devices.  If NULL,
*          then numMax must be zero.
```

## Appendix B N1231A.h and N1231A\_reg.h C/C++ Programming Information

### The C/C++ N1231A.h Header File

```
*          *numMax
*          Maximum number of devices to be reported.  The array
*          at pDeviceArray must be at least this large.
*          If pDeviceArray is NULL, then numMax must be zero.
*
*  ERROR CODES:
*  The return code specifies the success of failure of this
function,
*  as follows:
*      N1231A_SUCCESS:      The function returned successfully.
*      N1231A_ERR_PARAM: A passed parameter is NULL.
*      N1231A_ERR_DEVICE:   No device found.
*      N1231A_ERR_DRIVER:   No device driver installed.
*      N1231A_ERR_MEMORY:   Insufficient memory.
*      In case of error, *pNumFound will be set to zero.
*/

/*****
*/
EXPORT N1231A_RETURN
N1231APresetRawAll(N1231A_HANDLE      N1231AHandle,
                  long                preset1,
                  long                preset2,
                  long                preset3,
                  unsigned short *pStatus);

/*
*  For the card specified by N1231AHandle, attempts to
*  perform the following:
*
*  a) Reset all errors on axis 1, 2, and 3 and the Ref axes.
*  b) Load the Preset Position registers for each axis
*     with preset1, preset2, and preset3.
*  c) Transfer these Preset Position values to the Position registers
*     (i.e., perform a Position Preset operation).
*  d) Set *pStatus to show contents of the status registers.
*
*  The preset values are in raw interferometer
*  units (see Definitions of Terms near top of this file).
*
*  Return value:
*      N1231A_SUCCESS:      Success.
*      N1231A_ERR_PARAM: A passed parameter is NULL.
*      N1231A_ERR_HANDLE:   N1231AHandle invalid.
*      N1231A_ERR_AXIS:    An axis still shows a status error.
*/

/*****
*/
EXPORT N1231A_RETURN
N1231APresetRaw(N1231A_HANDLE N1231AHandle,
                N1231A_AXIS   axis,
                long          preset,
                unsigned short *pStatus);

/*
*  For the card specified by N1231AHandle and the axis specified
*  by axis, attempts to perform the following:
*
*  a) Load the Preset Position register with preset,
*  b) Reset all errors on the specified axis and the ref axis.
*  c) Transfer the Preset Position value to the Position register
*     (i.e., perform a Position Preset operation).
```

## Appendix B N1231A.h and N1231A\_reg.h C/C++ Programming Information

### The C/C++ N1231A.h Header File

```
*      d) Set *pStatus to show contents of the status registers.
*
*      The preset value is in raw interferometer
*      units (see Definitions of Terms near top of this file).
*
*      Return value:
*          N1231A_SUCCESS:          Success.
*          N1231A_ERR_PARAM: A passed parameter is NULL.
*          N1231A_ERR_HANDLE:      N1231AHandle invalid.
*          N1231A_ERR_AXIS:  The axis still shows a status error.
*/

/*****
*/
EXPORT N1231A_RETURN
N1231AGetRawPosVelAll(N1231A_HANDLE      N1231AHandle,
                     long                *pPosition1,
                     long                *pVelocity1,
                     long                *pPosition2,
                     long                *pVelocity2,
                     long                *pPosition3,
                     long                *pVelocity3,
                     unsigned short      *pValid);
/*
*      For the card specified by N1231AHandle:
*
*      a) Samples position and velocity on axis 1, 2, and 3.
*      b) Reads position and velocity values on axis 1, 2, and 3
*          into the corresponding parameters.
*
*      If an error has occurred on an axis, the position and
*      velocity parameters for that axis will be left unchanged.
*
*      The contents of the sample register are returned in *pValid.
*      Therefore, if N1231A_ERR_AXIS is returned, the *pValid
*      parameter can be used to determine which axis were invalid.
*      The constants:
*          N1231A_VALID_1
*          N1231A_VALID_2
*          N1231A_VALID_3
*      can be used to assist in analyzing *pValid.
*
*      The position and velocity values are in raw interferometer
*      units (see Definitions of Terms near top of this file).
*
*      Return value:
*          N1231A_SUCCESS:          Success.
*          N1231A_ERR_PARAM: A passed parameter is NULL.
*          N1231A_ERR_HANDLE:      N1231AHandle invalid.
*          N1231A_ERR_AXIS:  Data was not valid on at least one axis
*/

/*****
*/
EXPORT N1231A_RETURN
N1231AGetRawPosVel(N1231A_HANDLE      N1231AHandle,
                   N1231A_AXIS        axis,
                   long                *pPosition,
                   long                *pVelocity);
/*
```

**The C/C++ N1231A.h Header File**

```
*      For the card specified by N1231AHandle and the axis specified
*      by axis, attempts to perform the following:
*
*      a) Samples position and velocity on the specified axis.
*      b) Reads position and velocity values on the specified axis
*         into the corresponding parameters.
*
*      If an error has occurred on the specified axis, the position and
*      velocity parameters will be left unchanged.
*
*      The position and velocity values are in raw interferometer
*      units (see Definitions of Terms near top of this file).
*
*      Return value:
*          N1231A_SUCCESS:          Success.
*          N1231A_ERR_PARAM: A passed parameter is NULL.
*          N1231A_ERR_HANDLE:      N1231AHandle invalid.
*          N1231A_ERR_AXIS:  Data invalid on specified axis.
*/

/*****
*/
EXPORT N1231A_RETURN
N1231AClearStatusBits(N1231A_HANDLE      N1231AHandle,
                     unsigned short      resetBits,
                     unsigned short      *pStatus);

/*
*      For the card specified by N1231AHandle:
*
*      a) Attempts to clear specific bits (resetBits) in status registers
*      b) Sets *pStatus to the new values of the status registers
*
*      NOTE: Although two separate hardware registers are used for
*      status, this function packs them into a single *pStatus register.
*
*      The following constants can be used to clear and
*      interpret the bits in *pStatus:
*
*          N1231A_NO_SIG_1
*          N1231A_GLITCH_1
*          N1231A_NO_SIG_2
*          N1231A_GLITCH_2
*          N1231A_NO_SIG_3
*          N1231A_GLITCH_3
*          N1231A_NO_SIG_REF
*          N1231A_GLITCH_REF
*
*      Return value:
*          N1231A_SUCCESS:          Success.
*          N1231A_ERR_PARAM: A passed parameter is NULL.
*          N1231A_ERR_HANDLE:      N1231AHandle invalid.
*/

/*****
*/
EXPORT N1231A_RETURN
N1231AClearStatusAll(N1231A_HANDLE      N1231AHandle,
                    unsigned short      *pStatus);

/*
*      For the card specified by N1231AHandle:
*
*      .
```

## Appendix B N1231A.h and N1231A\_reg.h C/C++ Programming Information

### The C/C++ N1231A.h Header File

```
* a) Attempts to clear the status registers
* b) Sets *pStatus to the show the new values of the status
   registers
*
* NOTE: Although two separate hardware registers are used for
* status, this function packs them into a single *pStatus register.
*
* The following constants can be used to interpret the
* bits in *pStatus:
*
*     N1231A_NO_SIG_1
*     N1231A_GLITCH_1
*     N1231A_NO_SIG_2
*     N1231A_GLITCH_2
*     N1231A_NO_SIG_3
*     N1231A_GLITCH_3
*     N1231A_NO_SIG_REF
*     N1231A_GLITCH_REF
*
* Return value:
*     N1231A_SUCCESS:      Success.
*     N1231A_ERR_PARAM: A passed parameter is NULL.
*     N1231A_ERR_HANDLE:  N1231AHandle invalid.
*/

/*****
*/
EXPORT N1231A_RETURN
N1231AGetStatus(N1231A_HANDLE N1231AHandle,
               unsigned short *pStatus);

/*
* For the card specified by N1231AHandle:
*
* a) Sets *pStatus to show the contents of the status registers.
*
* NOTE: Although two separate hardware registers are used for
* status, this function packs them into a single *pStatus register.
*
* The following constants can be used to interpret the
* bits in *pStatus:
*
*     N1231A_NO_SIG_1
*     N1231A_GLITCH_1
*     N1231A_NO_SIG_2
*     N1231A_GLITCH_2
*     N1231A_NO_SIG_3
*     N1231A_GLITCH_3
*     N1231A_NO_SIG_REF
*     N1231A_GLITCH_REF
*
* Return value:
*     N1231A_SUCCESS:      Success.
*     N1231A_ERR_PARAM: A passed parameter is NULL.
*     N1231A_ERR_HANDLE:  N1231AHandle invalid.
*/

/*****
*/
EXPORT N1231A_RETURN
N1231ASetInterruptMask(N1231A_HANDLE N1231AHandle,
                      unsigned short intrMask);
```

**The C/C++ N1231A.h Header File**

```
/*
 *   For the card specified by N1231AHandle:
 *
 *   a) Sets the interrupt mask register.
 *
 * The following constants can be used for IntrMask:
 *       N1231A_INTR_NO_SIG_1
 *       N1231A_INTR_GLITCH_1
 *       N1231A_INTR_NO_SIG_2
 *       N1231A_INTR_GLITCH_2
 *       N1231A_INTR_NO_SIG_3
 *       N1231A_INTR_GLITCH_3
 *       N1231A_INTR_NO_SIG_REF
 *       N1231A_INTR_GLITCH_REF
 *
 * Return value:
 *       N1231A_SUCCESS:          Success.
 *       N1231A_ERR_PARAM: A passed parameter is NULL.
 *       N1231A_ERR_HANDLE:      N1231AHandle invalid.
 */

/*****
 */
EXPORT N1231A_RETURN
N1231AGetInterruptMask(N1231A_HANDLE      N1231AHandle,
                      unsigned short *pIntrMask);

/*
 *   For the card specified by N1231AHandle:
 *
 *   a) Sets *pIntrMask to show contents of the
 *       interrupt mask register.
 *
 * The following constants can be used to interpret bits in
 * pIntrMask:
 *       N1231A_INTR_NO_SIG_1
 *       N1231A_INTR_GLITCH_1
 *       N1231A_INTR_NO_SIG_2
 *       N1231A_INTR_GLITCH_2
 *       N1231A_INTR_NO_SIG_3
 *       N1231A_INTR_GLITCH_3
 *       N1231A_INTR_NO_SIG_REF
 *       N1231A_INTR_GLITCH_REF
 *
 * Return value:
 *       N1231A_SUCCESS:          Success.
 *       N1231A_ERR_PARAM: A passed parameter is NULL.
 *       N1231A_ERR_HANDLE:      N1231AHandle invalid.
 */

/*****
 */
EXPORT N1231A_RETURN
N1231ASetConfig(N1231A_HANDLE N1231AHandle,
                unsigned short config);

/*
 *   For the card specified by N1231AHandle:
 *
 *   a) Sets the configuration registers.
 */
```

## Appendix B N1231A.h and N1231A\_reg.h C/C++ Programming Information

### The C/C++ N1231A.h Header File

```
* NOTE: Although two separate hardware registers are used for
* config, this function packs them into a single config register.
*
* The following constants can be used for config:
*     N1231A_DIR_MINUS_1
*     N1231A_DIR_MINUS_2
*     N1231A_DIR_MINUS_3
*     N1231A_IRQ_ENB
*
* Return value:
*     N1231A_SUCCESS:          Success.
*     N1231A_ERR_PARAM: A passed parameter is NULL.
*     N1231A_ERR_HANDLE:      N1231AHandle invalid.
*/

/*****
*/
EXPORT N1231A_RETURN
N1231AGetConfig(N1231A_HANDLE N1231AHandle,
               unsigned short *pConfig);
/*
* For the card specified by N1231AHandle:
*
* a) Sets *pConfig to show contents of the
*     configuration registers.
*
* NOTE: Although two separate hardware registers are used for
* config, this function packs them into a single *pconfig register.
*
* The same constants shown under N1231ASetConfig()
* can be used to interpret the bits in the word.
*
* Return value:
*     N1231A_SUCCESS:          Success.
*     N1231A_ERR_PARAM: A passed parameter is NULL.
*     N1231A_ERR_HANDLE:      N1231AHandle invalid.
*/

/*****
*/
EXPORT N1231A_RETURN
N1231ASetFilter(N1231A_HANDLE N1231AHandle,
               unsigned short filter);
/*
* For the card specified by N1231AHandle:
*
* a) Sets the filter register.
*
* The following constants can be used for filter:
*
*     N1231A_FILTER_ENB
*
*     N1231A_KP0
*     N1231A_KP1
*     N1231A_KP2
*     N1231A_KP3
*
*     N1231A_KV0
*     N1231A_KV1
*     N1231A_KV2
*     N1231A_KV3
```



## Appendix B N1231A.h and N1231A\_reg.h C/C++ Programming Information

### The C/C++ N1231A.h Header File

```
*
*   Return value:
*       N1231A_SUCCESS:          Success.
*       N1231A_ERR_PARAM: A passed parameter is NULL.
*       N1231A_ERR_HANDLE:      N1231AHandle invalid.
*/

/*****
*/
EXPORT N1231A_RETURN
N1231AGetFilter(N1231A_HANDLE N1231AHandle,
               unsigned short *pFilter);
/*
*   For the card specified by N1231AHandle:
*
*   a) Sets *pFilter to show contents of the filter register.
*
*   The same constants shown under N1231ASetFilter()
*   can be used to interpret the bits in the word.
*
*   Return value:
*       N1231A_SUCCESS:          Success.
*       N1231A_ERR_PARAM: A passed parameter is NULL.
*       N1231A_ERR_HANDLE:      N1231AHandle invalid.
*/

/*****
*/
EXPORT N1231A_RETURN
N1231AGetFPGARevs(N1231A_HANDLE N1231AHandle,
                 unsigned long *pFPGA1Rev,
                 unsigned long *pFPGA2Rev);
/*
*   For the card specified by N1231AHandle:
*
*   a) Sets *pFPGA1Rev and *pFPGA2Rev to show the
*       revision codes of the two FPGAs.
*
*   Return value:
*       N1231A_SUCCESS:          Success.
*       N1231A_ERR_PARAM: A passed parameter is NULL.
*       N1231A_ERR_HANDLE:      N1231AHandle invalid.
*/

/*****
*/
EXPORT N1231A_RETURN
N1231AGetFPGAIds(N1231A_HANDLE N1231AHandle,
                unsigned long *pFPGA1Id,
                unsigned long *pFPGA2Id);
/*
*   For the card specified by N1231AHandle:
*
*   a) Sets *pFPGA1Id and *pFPGA2Id to show the
*       ID codes of the two FPGAs.
*
*   Return value:
*       N1231A_SUCCESS:          Success.
*       N1231A_ERR_PARAM: A passed parameter is NULL.
*       N1231A_ERR_HANDLE:      N1231AHandle invalid.
*/
```

## Appendix B N1231A.h and N1231A\_reg.h C/C++ Programming Information

### The C/C++ N1231A.h Header File

```

/*****
*/
EXPORT N1231A_RETURN
N1231AGetHdwRev(N1231A_HANDLE N1231AHandle,
                unsigned short *pHdwRev);
/*
 *   For the card specified by N1231AHandle:
 *
 *   a) Reads the Hardware Revision code.
 *
 *   Return value:
 *       N1231A_SUCCESS:          Success.
 *       N1231A_ERR_PARAM: A passed parameter is NULL.
 *       N1231A_ERR_HANDLE:      N1231AHandle invalid.
 */

/*****
*/
EXPORT unsigned int
N1231AGetLibraryRev(void);
/*
 *   Returns the library revision number.
 */

/*****
*/
EXPORT char*
N1231AGetErrStr(N1231A_RETURN err);
/*
 *   Returns a string describing the error conditions
 *   related to error parameter "err".
 */

/*****
*/
/*
 *   =====
 *   * Interrupt Related Functions
 *   =====
 */

/*****
*/
EXPORT N1231A_RETURN
N1231APciInterruptEnable(N1231A_HANDLE N1231AHandle,
                        int enable);
/*
 *   For the card specified by N1231AHandle:
 *
 *   Enables interrupts if parameter enable is non-zero.
 *   Disables interrupts if parameter enable is zero.
 *
 *   NOTE: See also the IRQ_ENB bit in the configuration
 *          register and the Interrupt Mask bits.
 *
 *   The Interrupt Mask bits determine whether
 *   individual status bits can generate interrupts.
 */

```

**The C/C++ N1231A.h Header File**

```
*          The IRQ_ENB bit in the configuration register
*          controls whether interrupt conditions can be
*          passed to the PCI interface.  It can be used to
*          temporarily inhibit generation of interrupts.
*
*          This function (N1231APciInterruptEnable) is used
*          to turn on (or off) PCI interrupt handling
*          so that it can respond to the card conditions.
*
*          Return value:
*          N1231A_SUCCESS:          Success.
*          N1231A_ERR_PARAM: A passed parameter is NULL.
*          N1231A_ERR_HANDLE:      N1231AHandle invalid.
*/

/*****
*/
EXPORT N1231A_RETURN
N1231APciInterruptAttach(N1231A_HANDLE      N1231AHandle,
                        N1231A_EVT_HANDLE
                        *eventHandle);
/*
*   For the card specified by N1231AHandle:
*
*   Returns an *eventHandle to a Windows Event Object that can be used
by
*   a Windows Wait Function (for example, WaitForSingleObject()).  The
*   Event Object will be set to a signalled state when the N1231A card
*   generates an interrupt.
*/

/*****
*/
/*
*   =====
*   Lower Level Functions
*   =====
*/

/*****
*/
EXPORT N1231A_RETURN
N1231ASamplePosVel(N1231A_HANDLE      N1231AHandle,
                  N1231A_AXIS          axis);
/*
*   For the card specified by N1231AHandle and the axis specified
*   by axis:
*
*   a) Samples position and velocity.
*
*   Return value:
*   N1231A_SUCCESS:          Success.
*   N1231A_ERR_PARAM: A passed parameter is NULL.
*   N1231A_ERR_HANDLE:      N1231AHandle invalid.
*/

/*****
*/
EXPORT N1231A_RETURN
N1231AReadRawPos(N1231A_HANDLE      N1231AHandle,
                 N1231A_AXIS          axis,
```

## Appendix B N1231A.h and N1231A\_reg.h C/C++ Programming Information

### The C/C++ N1231A.h Header File

```
                                long                *pPosition);
/*
 *   For the card specified by N1231AHandle and the axis specified
 *   by axis:
 *
 *   a) Reads position value.
 *
 *   IMPORTANT NOTE: This function reads the sample that was taken
 *   on the last "sample" command. It does not cause a new sample
 *   to be latched. Therefore, repeated calls to this function will
 *   return the same value unless a new sample command is issued.
 *   See also: N1231ASamplePosVel() and N1231AGetRawPosVel().
 *
 *   If an error has occurred on the axis, *pPosition
 *   will be left unchanged.
 *
 *   The position value is in raw interferometer
 *   units (see Definitions of Terms near top of this file).
 *
 *   Return value:
 *       N1231A_SUCCESS:          Success.
 *       N1231A_ERR_PARAM: A passed parameter is NULL.
 *       N1231A_ERR_HANDLE:      N1231AHandle invalid.
 *       N1231A_ERR_AXIS         Error on the axis or reference.
 */

/*****
 */
EXPORT N1231A_RETURN
N1231AReadRawVel(N1231A_HANDLE      N1231AHandle,
                 N1231A_AXIS        axis,
                 long                *pVelocity);
/*
 *   For the card specified by N1231AHandle and the axis specified
 *   by axis:
 *
 *   a) Reads velocity value.
 *
 *   IMPORTANT NOTE: This function reads the sample that was taken
 *   on the last "sample" command. It does not cause a new sample
 *   to be latched. Therefore, repeated calls to this function will
 *   return the same value unless a new sample command is issued.
 *   See also: N1231ASamplePosVel() and N1231AGetRawPosVel().
 *
 *   If an error has occurred on the axis, *pVelocity
 *   will be left unchanged.
 *
 *   The velocity value is in raw interferometer
 *   units (see Definitions of Terms near top of this file).
 *
 *   Return value:
 *       N1231A_SUCCESS:          Success.
 *       N1231A_ERR_PARAM: A passed parameter is NULL.
 *       N1231A_ERR_HANDLE:      N1231AHandle invalid.
 *       N1231A_ERR_AXIS:       Error on axis or reference.
 */

/*****
 */
EXPORT N1231A_RETURN
N1231AWriteRawPresetRegister(N1231A_HANDLE      N1231AHandle,
```

## Appendix B N1231A.h and N1231A\_reg.h C/C++ Programming Information

### The C/C++ N1231A.h Header File

```

                                N1231A_AXIS  axis,
                                long
    presetValue);
/*
 *   For the card specified by N1231AHandle and the axis specified
 *   by axis:
 *
 *   a) Writes presetValue to the preset position register for the
 *       axis.
 *
 *   IMPORTANT NOTE:
 *       Does not transfer this value into the position register.
 *       See also: N1231APresetRaw().
 *
 *       The preset value is in raw interferometer
 *       units (see Definitions of Terms near top of this file).
 *
 *   Return value:
 *       N1231A_SUCCESS:          Success.
 *       N1231A_ERR_PARAM: A passed parameter is NULL.
 *       N1231A_ERR_HANDLE:      N1231AHandle invalid.
 */
/*****
 */
EXPORT N1231A_RETURN
N1231AGetRawPresetRegister(N1231A_HANDLE  N1231AHandle,
                           N1231A_AXIS    axis,
                           long
                           *pPresetValue);
/*
 *   For the card specified by N1231AHandle and the axis specified
 *   by axis:
 *
 *   a) Sets *pPresetValue to the contents of the
 *       preset position register for the axis.
 *
 *       The preset value is in raw interferometer
 *       units (see Definitions of Terms near top of this file).
 *
 *   Return value:
 *       N1231A_SUCCESS:          Success.
 *       N1231A_ERR_PARAM: A passed parameter is NULL.
 *       N1231A_ERR_HANDLE:      N1231AHandle invalid.
 */
/*****
 */
/*
 *   =====
 *   Register Functions (for test use only)
 *   =====
 */
/*****
 */
EXPORT N1231A_RETURN
N1231AWriteRegisterLong(N1231A_HANDLE  N1231AHandle,
                        unsigned int    N1231ARegister,
                        long             value);
```

## Appendix B N1231A.h and N1231A\_reg.h C/C++ Programming Information

### The C/C++ N1231A.h Header File

```
/*
 * For the card specified by N1231AHandle:
 *
 * a) Writes 32-bit value to the specified register.
 *
 * The following constants can be used
 * for the parameter N1231ARegister:
 *
 *     N1231A_OFST_PRESET_1
 *     N1231A_OFST_PRESET_2
 *     N1231A_OFST_PRESET_3
 *
 * Return value:
 *     N1231A_SUCCESS:      Success.
 *     N1231A_ERR_PARAM: A passed parameter is NULL.
 *     N1231A_ERR_HANDLE:  N1231AHandle invalid.
 *     N1231A_ERR_REG:     Invalid register for long write.
 */

/*****
 */
EXPORT N1231A_RETURN
N1231AReadRegisterLong(N1231A_HANDLE      N1231AHandle,
                      unsigned int        N1231ARegister,
                      long                *value);

/*
 * For the card specified by N1231AHandle:
 *
 * a) Reads 32-bit value from the specified register.
 *
 * The following constants can be used
 * for the parameter N1231ARegister:
 *
 *     N1231A_OFST_POS_1
 *     N1231A_OFST_POS_2
 *     N1231A_OFST_POS_3
 *     N1231A_OFST_VEL_1
 *     N1231A_OFST_VEL_2
 *     N1231A_OFST_VEL_3
 *     N1231A_OFST_PRESET_1
 *     N1231A_OFST_PRESET_2
 *     N1231A_OFST_PRESET_3
 *     N1231A_OFST_FPGA1_REV
 *     N1231A_OFST_FPGA2_REV
 *     N1231A_OFST_FPGA1_ID
 *     N1231A_OFST_FPGA2_ID
 *
 * Return value:
 *     N1231A_SUCCESS:      Success.
 *     N1231A_ERR_PARAM: A passed parameter is NULL.
 *     N1231A_ERR_HANDLE:  N1231AHandle invalid.
 *     N1231A_ERR_REG:     Invalid register for long read.
 */

/*****
 */
EXPORT N1231A_RETURN
N1231AWriteRegisterWord(N1231A_HANDLE      N1231AHandle,
                       unsigned int        N1231ARegister,
                       short               value);

/*
```

**The C/C++ N1231A.h Header File**

```
*      For the card specified by N1231AHandle:
*
*      a) Writes 16-bit value to the specified register.
*
*      The following constants can be used
*      for the parameter N1231ARegister:
*
*          N1231A_OFST_CONFIG_12
*          N1231A_OFST_CONFIG_3I
*          N1231A_OFST_STATUS_12
*          N1231A_OFST_STATUS_3R
*          N1231A_OFST_INTR_MASK
*          N1231A_OFST_FILTER
*          N1231A_OFST_SAMPLE
*          N1231A_OFST_PRESET
*
*      Return value:
*          N1231A_SUCCESS:          Success.
*          N1231A_ERR_PARAM: A passed parameter is NULL.
*          N1231A_ERR_HANDLE:      N1231AHandle invalid.
*          N1231A_ERR_REG:         Invalid register for word write.
*/

/*****
*/
EXPORT N1231A_RETURN
N1231AReadRegisterWord(N1231A_HANDLE      N1231AHandle,
                      unsigned int        N1231ARegister,
                      short                *value);

/*
*      For the card specified by N1231AHandle:
*
*      a) Reads 16-bit value from the specified register.
*
*      The following constants can be used
*      for the parameter N1231ARegister:
*
*          N1231A_OFST_CONFIG_12
*          N1231A_OFST_CONFIG_3I
*          N1231A_OFST_STATUS_12
*          N1231A_OFST_STATUS_3R
*          N1231A_OFST_INTR_MASK
*          N1231A_OFST_FILTER
*          N1231A_OFST_SAMPLE
*          N1231A_OFST_PRESET
*
*      Return value:
*          N1231A_SUCCESS:          Success.
*          N1231A_ERR_PARAM: A passed parameter is NULL.
*          N1231A_ERR_HANDLE:      N1231AHandle invalid.
*          N1231A_ERR_REG:         Invalid register for word read.
*/

#ifdef __cplusplus
}
#endif

#endif /* IFNDEF N1231A_H */
```

**The C/C++ N1231A\_reg.h Header File**

---

## The C/C++ N1231A\_reg.h Header File



## Appendix B N1231A.h and N1231A\_reg.h C/C++ Programming Information

### The C/C++ N1231A\_reg.h Header File

```
/*
 * Copyright 2002 by Agilent Technologies
 *
 * File Name: N1231A_reg.h
 *
 * Header file providing register map for the
 * Agilent Technologies N1231A PCI Three-Axis Board
 * designed for use with an Agilent Technologies
 * laser interferometry position measurement system.
 */
#ifndef N1231A_REG_H
#define N1231A_REG_H

/*
*****
*****

*****
*****
*
* REGISTER MAP OF N1231A
*

*****
*****
* Used for direct register access.
* These constants represent the offset from PCI BAR 2.
*
* If using the N1231A API library, these constants are used with
* functions:
*     N1231AReadRegisterLong()
*     N1231AWriteRegisterLong()
*     N1231AReadRegisterWord()
*     N1231AWriteRegisterWord()
*/
#define N1231A_OFST_POS_1          0x0120      /* long read */
#define N1231A_OFST_POS_2          0x0100      /* long read */
#define N1231A_OFST_POS_3          0x0000      /* long read */
#define N1231A_OFST_VEL_1          0x0128      /* long read */
#define N1231A_OFST_VEL_2          0x0108      /* long read */
#define N1231A_OFST_VEL_3          0x0008      /* long read */
#define N1231A_OFST_PRESET_1        0x011c      /* long read/write */
#define N1231A_OFST_PRESET_2        0x0118      /* long read/write */
#define N1231A_OFST_PRESET_3        0x0018      /* long read/write */
#define N1231A_OFST_CONFIG_12       0x0110      /* word read/write */
#define N1231A_OFST_CONFIG_3I       0x0010      /* word read/write */
#define N1231A_OFST_STATUS_12       0x0112      /* word read/write */
#define N1231A_OFST_STATUS_3R       0x0012      /* word read/write */
#define N1231A_OFST_INTR_MASK       0x0016      /* word read/write */
#define N1231A_OFST_FILTER          0x001c      /* word read/write */
/*
#define N1231A_OFST_SAMPLE          0x0020      /* word read/write */
*/
#define N1231A_OFST_PRESET          0x0020      /* word read/write */
/*
#define N1231A_OFST_FPGA1_REV        0x002c      /* long read */
#define N1231A_OFST_FPGA2_REV        0x012c      /* long read */
#define N1231A_OFST_FPGA1_ID         0x0030      /* long read */
#define N1231A_OFST_FPGA2_ID         0x0130      /* long read */
*/
```

## Appendix B N1231A.h and N1231A\_reg.h C/C++ Programming Information

### The C/C++ N1231A\_reg.h Header File

```
/*
*****
*****
*****
*****
*
* BIT MAPS
*
*****
*****
*****
*****
*/

/*
*****
*****
* Used for status
*
* IMPORTANT NOTE: THESE CONSTANTS SHOULD ***ONLY*** BE USED IN
* CONJUNCTION WITH THE FOLLOWING API LIBRARY FUNCTIONS:
*
*      N1231AClearStatusBits()
*      N1231AClearStatusAll()
*      N1231AGetStatus()
*      N1231APresetRawAll()
*      N1231APresetRaw()
*
* The above functions pack the axis 1, axis 2, axis 3 and Ref status
* bits into one 16-bit word. These constants can be used for
* interpreting and clearing these packed bits.
*/
#define N1231A_NO_SIG_1      0x0004
#define N1231A_GLITCH_1     0x0008
#define N1231A_NO_SIG_2     0x0001
#define N1231A_GLITCH_2     0x0002
#define N1231A_NO_SIG_3     0x0100
#define N1231A_GLITCH_3     0x0200
#define N1231A_NO_SIG_REF   0x1000
#define N1231A_GLITCH_REF   0x2000

/*
*****
*****
* Used for status ONLY WITH DIRECT REGISTER ACCESS
*
* IMPORTANT NOTE:
* The following constants are provided for use ***ONLY*** in
* conjunction with direct access to the hardware registers.
* Direct access is provided by the library functions:
*      ReadRegisterWord()
*      WriteRegisterWord()
* They SHOULD NOT be used with any other library functions, as
* these other functions expect a packed status word.
*/
```

## Appendix B N1231A.h and N1231A\_reg.h C/C++ Programming Information

### The C/C++ N1231A\_reg.h Header File

```
#define N1231A_REG_NO_SIG_1      0x0004      // At offset
N1231A_STATUS_12 (see note above)
#define N1231A_REG_GLITCH_1     0x0008      // At offset
N1231A_STATUS_12 (see note above)
#define N1231A_REG_NO_SIG_2     0x0001      // At offset
N1231A_STATUS_12 (see note above)
#define N1231A_REG_GLITCH_2     0x0002      // At offset
N1231A_STATUS_12 (see note above)

#define N1231A_REG_NO_SIG_3     0x0001      // At offset
N1231A_STATUS_3R (see note above)
#define N1231A_REG_GLITCH_3     0x0002      // At offset
N1231A_STATUS_3R (see note above)
#define N1231A_REG_NO_SIG_REF 0x0010      // At offset N1231A_STATUS_3R
(see note above)
#define N1231A_REG_GLITCH_REF 0x0020      // At offset N1231A_STATUS_3R
(see note above)

/*
*****
*****
* Used for configuration
*
* IMPORTANT NOTE: THESE CONSTANTS SHOULD ***ONLY*** BE USED IN
* CONJUNCTION WITH THE FOLLOWING API LIBRARY FUNCTIONS:
*
*      N1231ASetConfig()
*      N1231AGetConfig()
*
* The above functions pack the axis 1, axis 2, axis 3, and IRQ
* control bits into one 16-bit word. These constants can be used
* for setting and interpreting these packed bits.
*/
#define N1231A_DIR_MINUS_1      0x0004
#define N1231A_DIR_MINUS_2      0x0002
#define N1231A_DIR_MINUS_3      0x0200
#define N1231A_IRQ_ENB          0x1000

/*
*****
*****
* Used for configuration ONLY WITH DIRECT REGISTER ACCESS
*
* IMPORTANT NOTE:
* The following constants are provided for use ***ONLY*** in
* conjunction with direct access to the hardware registers,
* Direct access is provided by the library functions:
*      ReadRegisterWord()
*      WriteRegisterWord()
* They SHOULD NOT be used with any other library functions, as
* these other functions expect a packed status word.
*/
#define N1231A_REG_DIR_MINUS_1   0x0004      // At offset
N1231A_CONFIG_12 (see note above)
#define N1231A_REG_DIR_MINUS_2   0x0002      // At offset
N1231A_CONFIG_12 (see note above)

#define N1231A_REG_DIR_MINUS_3   0x0002      // At offset
N1231A_CONFIG_3I (see note above)
```

## Appendix B N1231A.h and N1231A\_reg.h C/C++ Programming Information

### The C/C++ N1231A\_reg.h Header File

```
#define N1231A_REG_IRQ_ENB          0x1000      // At offset
N1231A_CONFIG_3I (see note above)

/*

*****
* Used for Interrupt Mask
*
*   These constants can be used in conjunction with the following
*   API Library Functions:
*
*       N1231ASetInterruptMask()
*       N1231AGetInterruptMask()
*
*   These can also be used in conjunction with direct register access
*   at address offset N1231A_INTR_MASK
*/
#define N1231A_INTR_NO_SIG_1        0x0100
#define N1231A_INTR_GLITCH_1        0x0200
#define N1231A_INTR_NO_SIG_2        0x0001
#define N1231A_INTR_GLITCH_2        0x0002
#define N1231A_INTR_NO_SIG_3        0x0010
#define N1231A_INTR_GLITCH_3        0x0020
#define N1231A_INTR_NO_SIG_REF      0x1000
#define N1231A_INTR_GLITCH_REF      0x2000

/*

*****
* Used for Filter
*
*   These constants can be used in conjunction with the following
*   API Library Functions:
*
*       N1231ASetFilter()
*       N1231AGetFilter()
*
*   These can also be used in conjunction with direct register access
*   at address offset N1231A_FILTER
*/
#define N1231A_FILTER_ENB           0X0010

#define N1231A_KP0                   0x0000
#define N1231A_KP1                   0x0001
#define N1231A_KP2                   0x0002
#define N1231A_KP3                   0x0003

#define N1231A_KV0                   0x0000
#define N1231A_KV1                   0x0004
#define N1231A_KV2                   0x0008
#define N1231A_KV3                   0x000C

/*
```

## Appendix B N1231A.h and N1231A\_reg.h C/C++ Programming Information

### The C/C++ N1231A\_reg.h Header File

```
/*
*****
* Used for reading Sample Register
*
* These constants can be used in conjunction with direct register
access
* reads at address offset N1231A_SAMPLE
*/
#define N1231A_ERR_1          0x0100
#define N1231A_VALID_1       0x0200
#define N1231A_ERR_2          0x0001
#define N1231A_VALID_2       0x0002
#define N1231A_ERR_3          0x0010
#define N1231A_VALID_3       0x0020

/*
*****
* Used for writing Sample Register
*
* These constants can be used in conjunction with direct register
access
* writes at address offset N1231A_SAMPLE
*/
#define N1231A_PRESET_1       0x0100
#define N1231A_SAMPLE_1      0x0200
#define N1231A_PRESET_2       0x0001
#define N1231A_SAMPLE_2      0x0002
#define N1231A_PRESET_3       0x0010
#define N1231A_SAMPLE_3      0x0020

#endif /* ifndef N1231A_REG_H */
```

**The C/C++ N1231A\_reg.h Header File**

---

# C

---

## CD Readme File

Software installation instructions

## General

---

# General

The Agilent Technologies N1231A is a three-axis laser board for use as part of an Agilent laser interferometry position measurement system.

This CD contains three setup programs to install software for use in conjunction with the N1231A. This software is intended for use on a Microsoft™ Windows® NT 4.0 machine.

## Description of setup programs

### ***N1231A API OEM version***

- Installs the device drivers and the API Library DLL, and modifies the Registry.
- Does not install any standalone executables.
- Intended for installation of the library on an end-user machine where the executable program is supplied by a separate setup program.

### ***N1231A API Development version***

Includes the files and Registry entries installed by the OEM version, and also installs additional files useful by those developing software using the API library.

### ***N1231A Monitor***

- Installs the N1231A Monitor program which provides a graphical interface to the N1231A card.
- The N1231A API (either the OEM version or the Development version) should be installed prior to installing the N1231A Monitor.



---

## Installing the OEM Version of the N1231A API Library from the CD

- 1 Exit from all applications.
- 2 Insert the CD in an appropriate drive.
- 3 Navigate to the directory: “\N1231A API OEM”.
- 4 Run “Setup.exe”.
- 5 Follow the instructions on the screen.
- 6 When asked to choose a Setup Type, the default choice of “OEM” is recommended.
- 7 If this is the first time the software has been installed, or if the software has been uninstalled prior to this installation, reboot the computer.

---

## Installing the Development Version of the N1231A API Library from the CD

- 1 Exit from all applications.
- 2 Insert the CD in an appropriate drive.
- 3 Navigate to the directory: “\N1231A API Development”.
- 4 Run “Setup.exe”.
- 5 Follow the instructions on the screen.

When asked to choose a Setup Type, the default choice of “Typical” is recommended. Other choices are “Compact” and “Custom.” The setup types are described below:

**Typical** installs all files.

**Compact** installs all files except those in the “User Files” directory.

**Custom** allows selecting the components to be installed.

## Installing the N1231A Monitor Program from the CD

- 6** If this is the first time the software has been installed, or if the software has been uninstalled prior to this installation, reboot the computer.

---

## Installing the N1231A Monitor Program from the CD

---

**NOTE**

If the N1231A API Library is not yet installed, follow the instructions above to install it before proceeding with this installation (either the OEM or development version may be used).

- 1** Exit from all applications.
- 2** Insert the CD in an appropriate drive.
- 3** Navigate to the directory: “\N1231A Monitor”.
- 4** Run “Setup.exe”.
- 5** Follow the instructions on the screen.

---

D

---

Agilent N1231A API Readme File

## **Agilent Technologies N1231A API Readme File**

=====

Version 1.1

=====

### **REBOOTING COMPUTER:**

The setup program installs a driver file, Pci9030.sys. If this is the first installation of this software, or if the driver file was removed during a previous uninstall, then it will be necessary to reboot the computer after the installation.

### **TYPICAL SETUP:**

The “Typical” setup installs the DLL files for accessing the N1231A into the windows system directory. The PCI driver file for the N1231A is installed into the “Drivers” directory under the windows system directory.

An example program is installed in the default location of:

C:\Program Files\Agilent\N1231A API\

Files needed for code development (including copies of the DLL and driver file) are installed in the default location of:

C:\Program Files\Agilent\N1231A API\User Files

### **FILES INSTALLED IN THE “USER FILES” DIRECTORY:**

Files developed by PLX Technology, Inc.:

Pci9030.sys    Driver file

PlxApi.dll    Application interface for Pci9030.sys

Files developed by Agilent Technologies:

Example.cpp    Source code for the Example.exe program

N1231A.dll    Application interface for N1231A

N1231A.h    Header file

N1231A\_reg.h    Header file with memory map and bit maps

N1231A.lib    Import Library for use with N1231A.dll

**USE OF ABOVE FILES DURING CODE DEVELOPMENT:**

- N1231A.h should be included in the source code (using “#include N1231A.h”). This file (N1231A.h) will also include the file N1231A\_reg.h.
- N1231A.lib should be available to the linker.
- N1231A.dll and PlxApi.dll must be available at run time. They can be in the same directory as the executable, in the windows directory, in the windows system directory, or in the “path”. Note that the setup program for “N1231A API” places copies of these files in the windows system directory.
- Pci9030.sys is a driver file which must be available at run time. It is installed in the “Drivers” directory under the windows system directory. After this file is installed for the first time the system must be rebooted (with the N1231A card inserted) so windows can recognize this file as a driver.
- Example.cpp is a source code file that can be used as a sample of how the N1231A API functions are used.



---

E

---

Filter Settings in Bitmap D

## Introduction

---

## Introduction

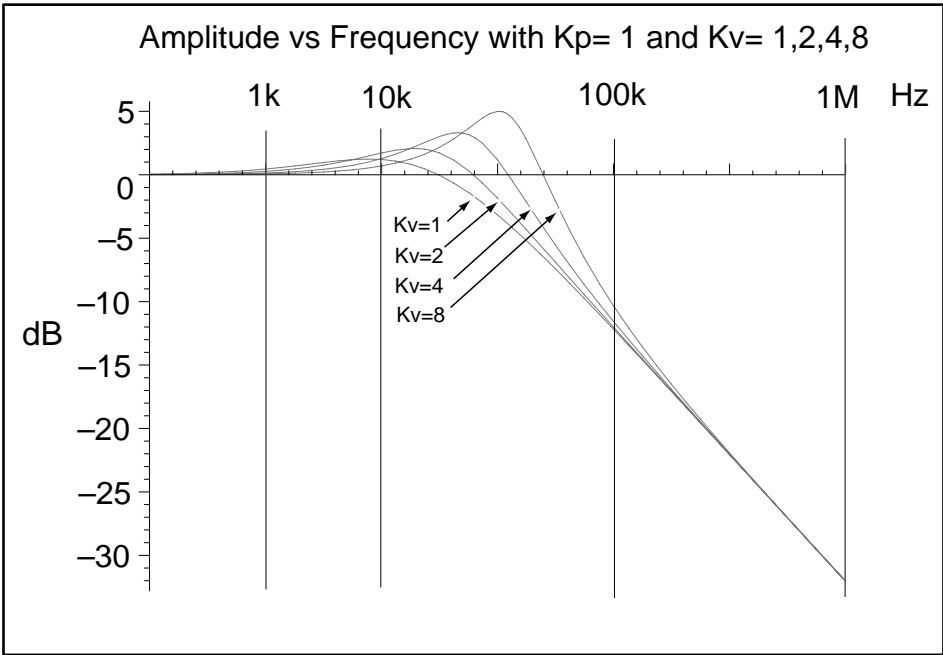
The Agilent N1231A board uses a digital filter to verify signal integrity and to optionally smooth the output data, thus reducing any noise introduced by the digital nature of the resolution extension electronics.

The following graphs show the *frequency* (amplitude and phase) and *temporal* (step and impulse) responses for the various filter settings. Note that the Kp and Kv settings from Bitmap D are the  $\log_2$  of the Kp and Kv values used in the graphs below.

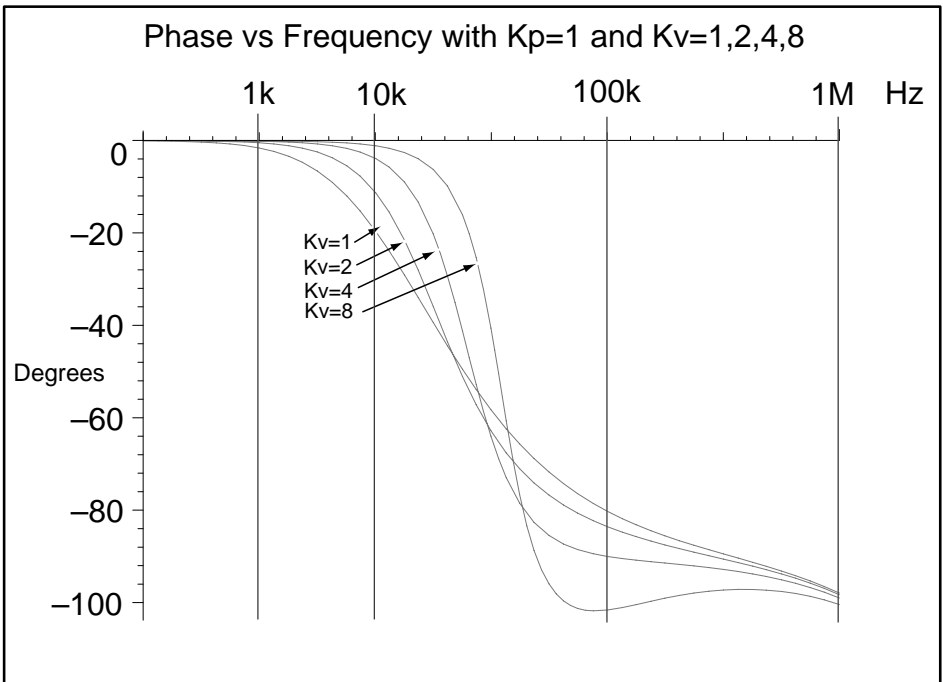


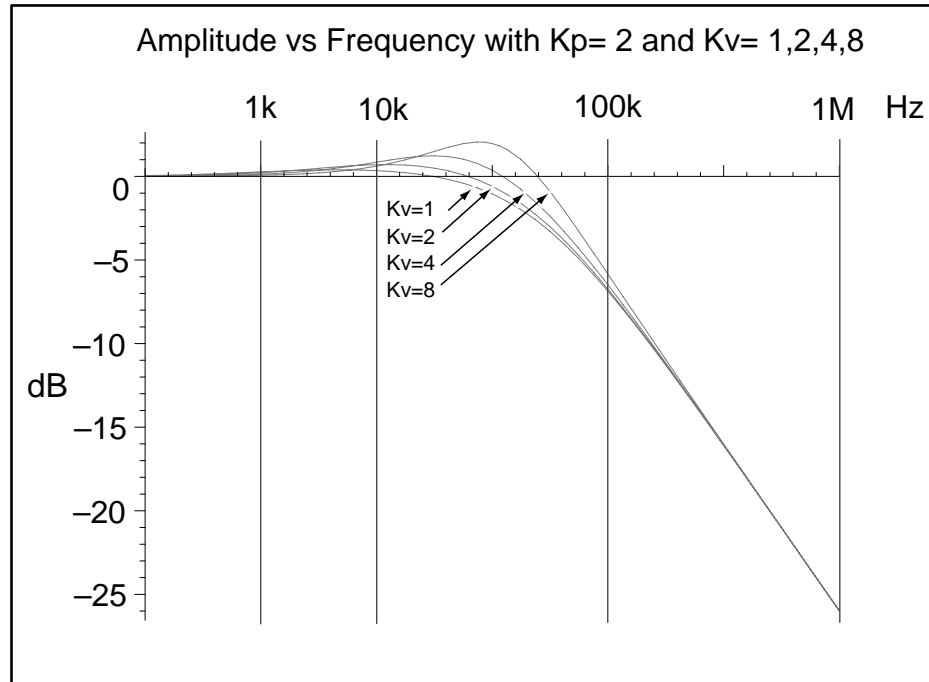
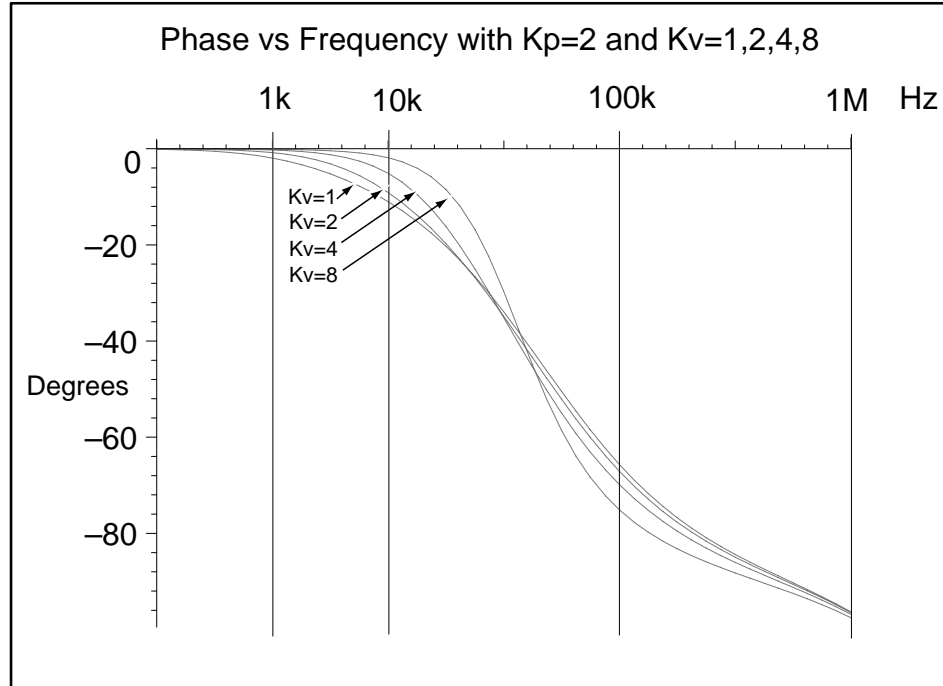
# Frequency Response

Graph 1 — Amplitude:  $K_p = 1$ ,  $K_v = 1, 2, 4, \text{ and } 8$



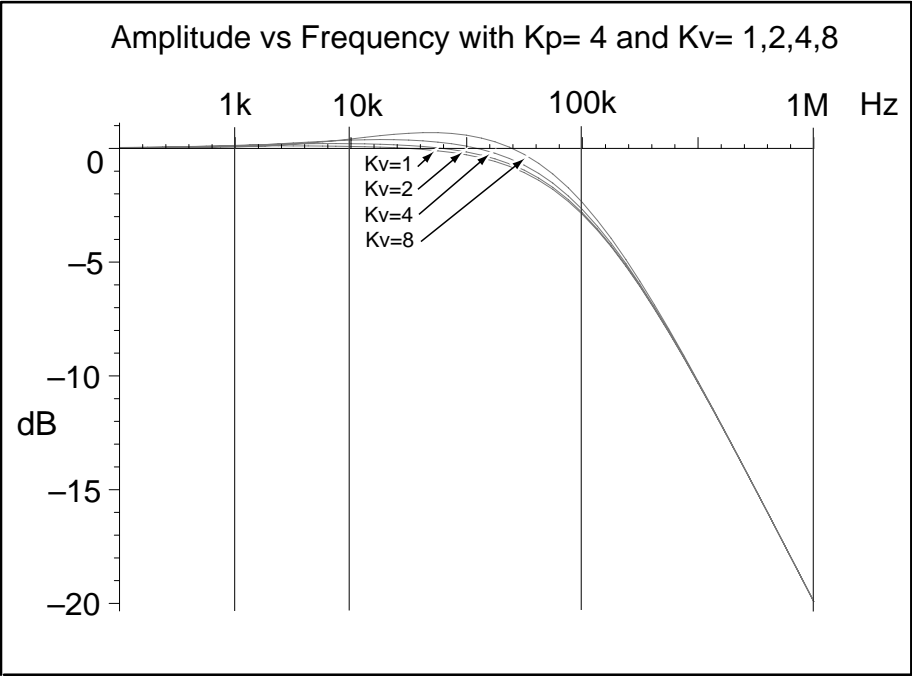
Graph 2 — Phase:  $K_p = 1$ ,  $K_v = 1, 2, 4, \text{ and } 8$



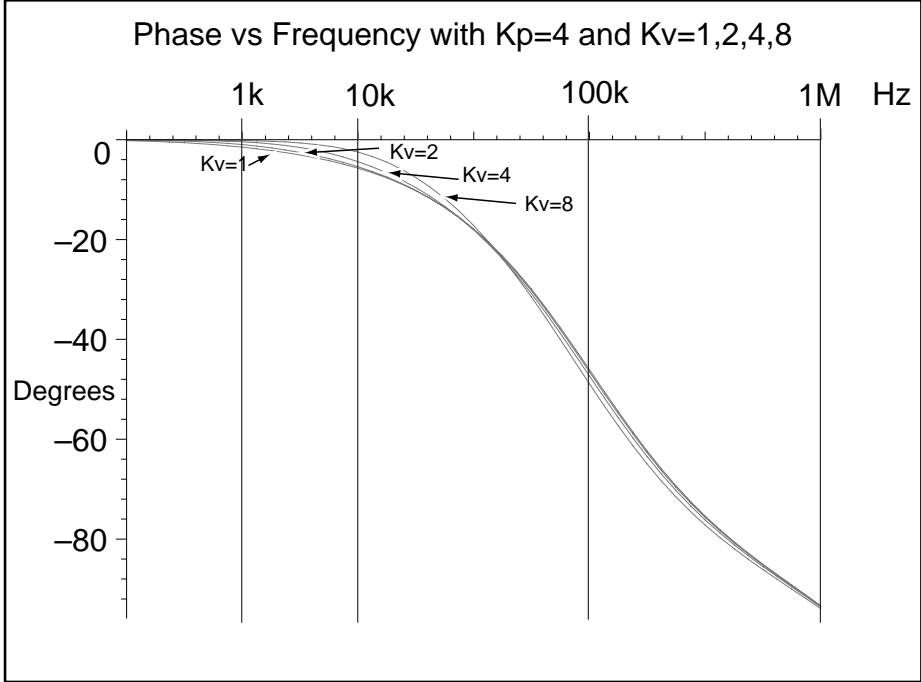
**Frequency Response**Graph 3 — Amplitude:  $K_p = 2$ ,  $K_v = 1, 2, 4$ , and  $8$ Graph 4 — Phase:  $K_p = 2$ ,  $K_v = 1, 2, 4$ , and  $8$ 

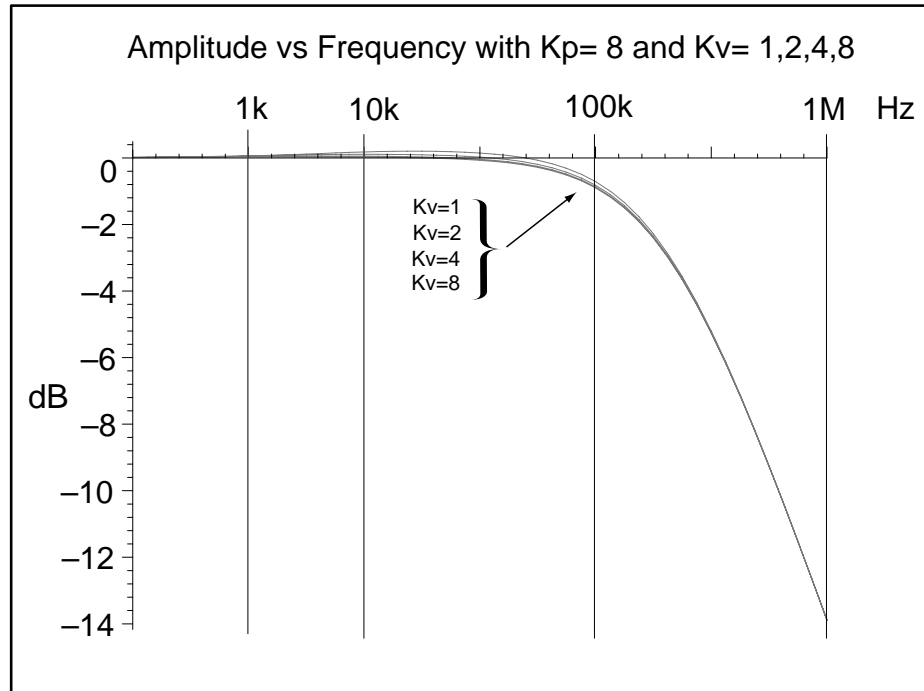
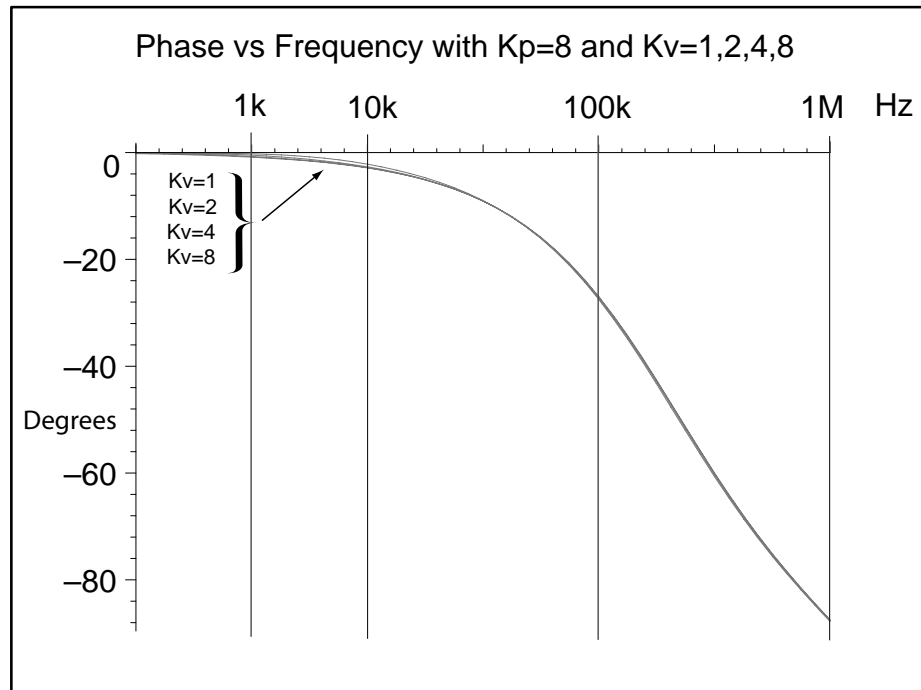
Frequency Response

Graph 5 — Amplitude:  $K_p = 4$ ,  $K_v = 1, 2, 4$ , and  $8$



Graph 6 — Phase:  $K_p = 4$ ,  $K_v = 1, 2, 4$ , and  $8$



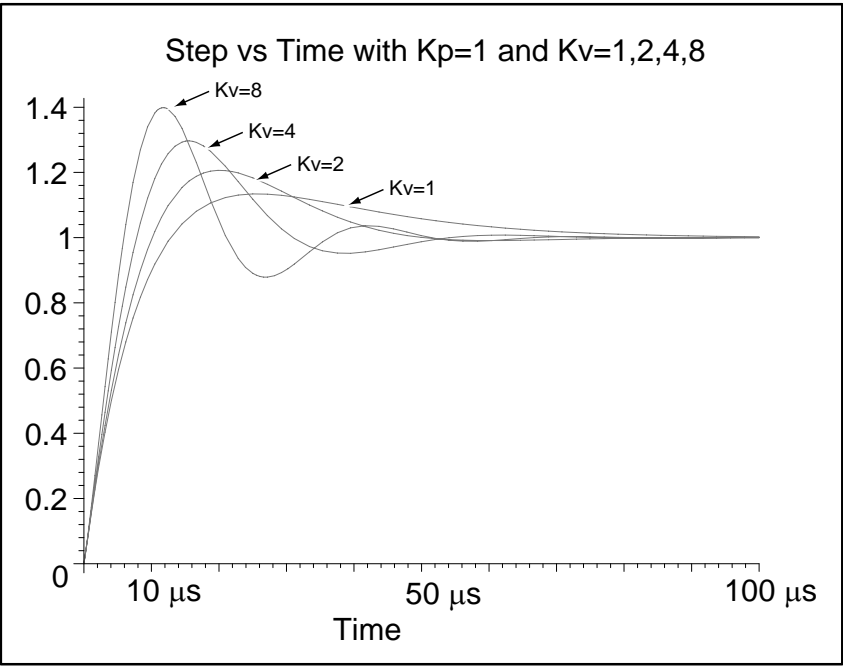
**Frequency Response**Graph 7 — Amplitude:  $K_p = 8$ ,  $K_v = 1, 2, 4$ , and  $8$ Graph 8 — Phase:  $K_p = 8$ ,  $K_v = 1, 2, 4$ , and  $8$ 

Temporal Response

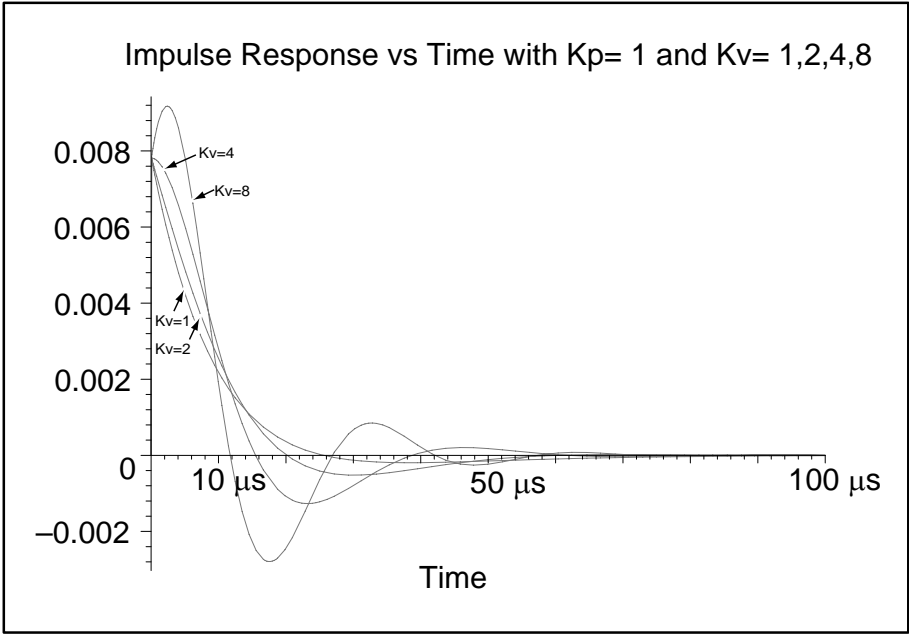
# Temporal Response

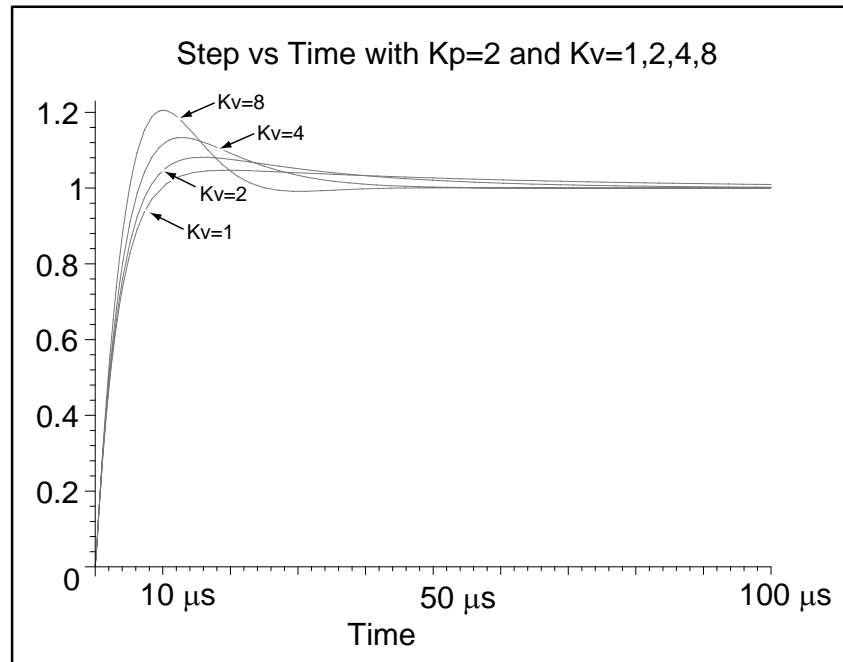
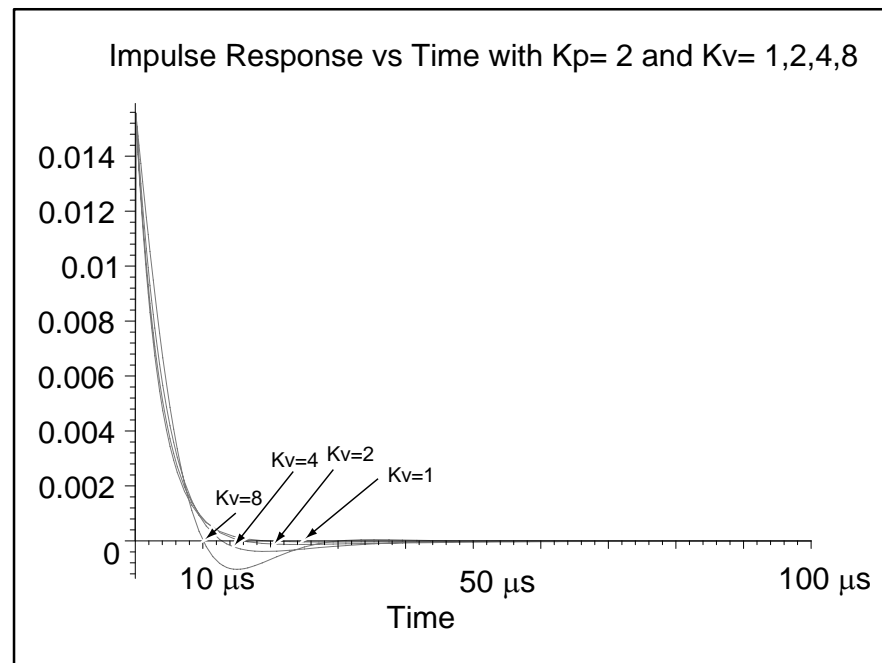
The temporal response graphs show the filter response vs sample periods (n) for a unit step input and an impulse input.

Graph 9 — Step:  $K_p = 1$ ,  $K_v = 1, 2, 4$ , and  $8$



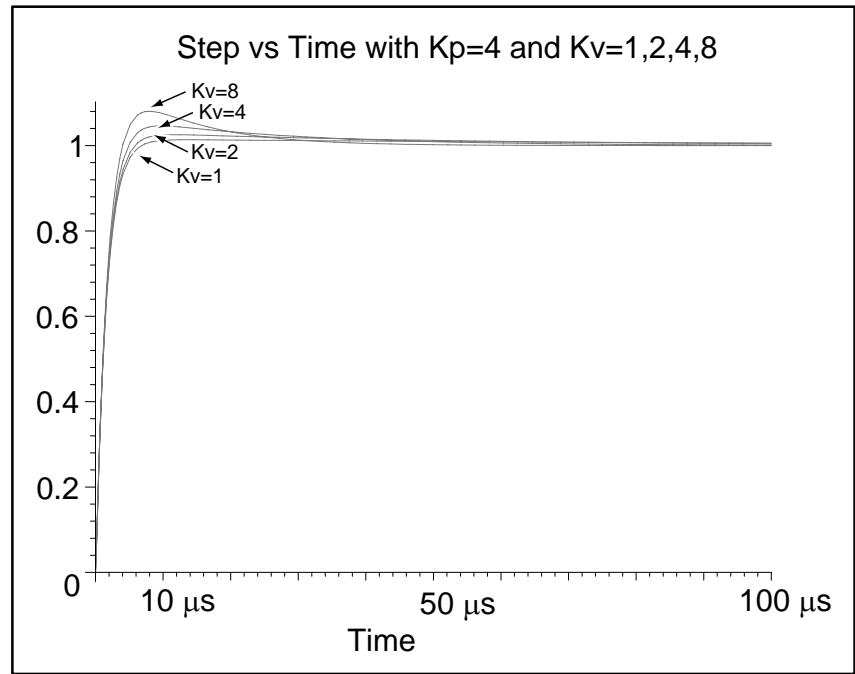
Graph 10 — Impulse:  $K_p = 1$ ,  $K_v = 1, 2, 4$ , and  $8$



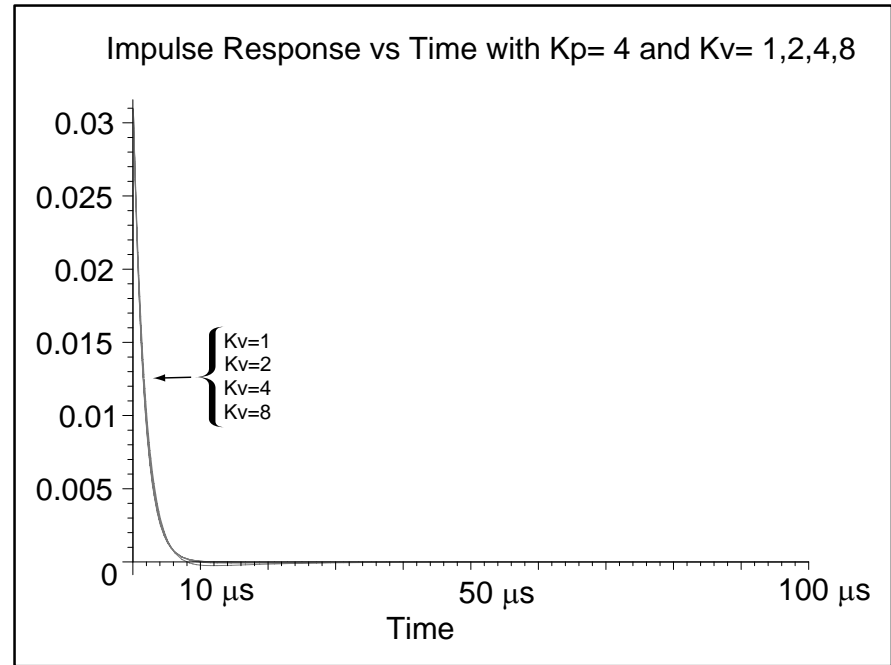
**Temporal Response**Graph 11 — Step:  $K_p = 2$ ,  $K_v = 1, 2, 4$ , and  $8$ Graph 12 — Impulse:  $K_p = 2$ ,  $K_v = 1, 2, 4$ , and  $8$ 

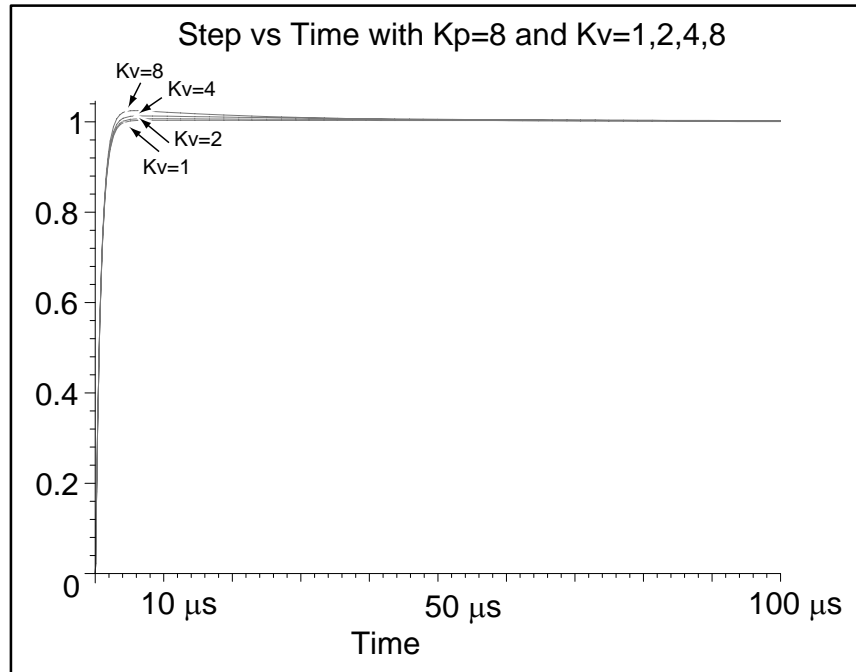
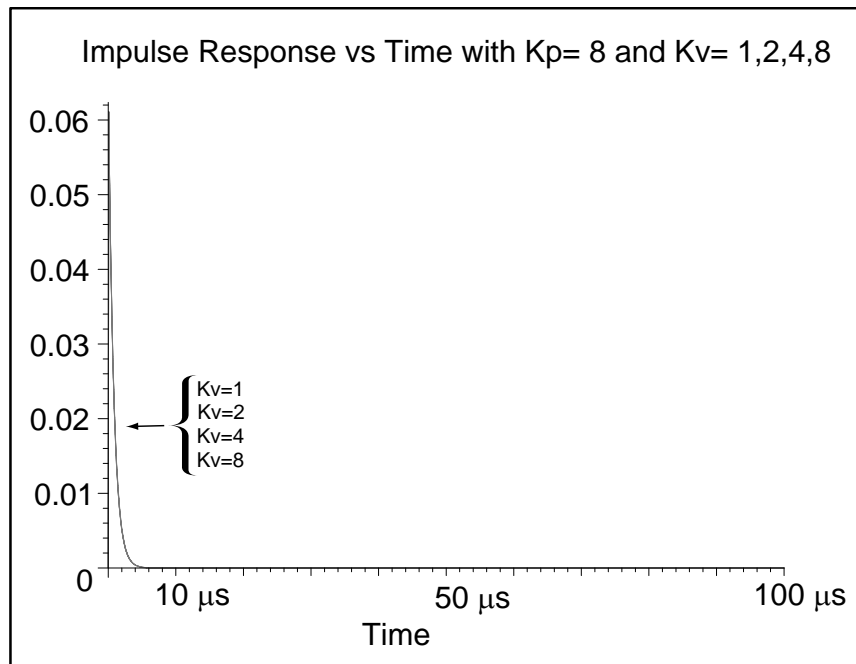
Temporal Response

Graph 13 — Step:  $K_p = 4$ ,  $K_v = 1, 2, 4$ , and  $8$



Graph 14 — Impulse:  $K_p = 4$ ,  $K_v = 1, 2, 4$ , and  $8$



**Temporal Response**Graph 15 — Step:  $K_p = 8$ ,  $K_v = 1, 2, 4$ , and  $8$ Graph 16 — Impulse:  $K_p = 8$ ,  $K_v = 1, 2, 4$ , and  $8$ 



---

# Index

## NUMERICS

10751C/D Air Sensor, 4-3, 4-4, 4-6  
10757D/E/F Material Temperature  
Sensor, 4-3, 4-4, 4-6  
10780C/F Receiver, 2-5, 4-4  
10880A/B/C Receiver Cable, 2-5, 4-4  
10881A/B/C Laser Head Cable, 2-5, 4-4  
10884A Power Supply, 4-4, 4-6  
10886A PC Compensation Board, 4-3,  
4-4, 4-6  
50-pin, 2-7  
50-pin flat-ribbon cable, 2-7  
5517B/C/D Laser Head, 4-4, 4-6

## A

ACC-14D boards, 4-2, 4-4  
ACC-14D ISA boards  
setting jumpers, 4-8  
accuracy and resolution, 5-4  
Agilent 10751C/D Air Sensor, 4-3, 4-4, 4-6  
Agilent 10757D/E/F Material  
Temperature Sensor, 4-3, 4-4, 4-6  
Agilent 10880A/B/C Receiver Cable, 2-5,  
4-4  
Agilent 10881A/B/C Laser Head  
Cable, 2-5  
Agilent 10881A/B/C Laser Head  
Cable, 4-4  
Agilent 10884A Power Supply, 4-4, 4-6  
Agilent 10886A PC Compensation  
Board, 4-3, 4-4, 4-6  
Agilent 5517B/C/D Laser Head, 4-4, 4-6  
Agilent N1231A  
description of, xii  
Agilent N1250A/B High Performance  
Receiver Cable, 2-5, 4-6  
Agilent N1251A/B High Performance  
Laser Head Cable, 2-5, 4-6  
air sensor  
10751C/D, 4-3, 4-4, 4-6  
API Library, 1-4, C-2, C-3  
API Library DLL, C-2  
Axes position resolution, 5-4  
Axis 1 connector, 1-3, 2-5  
Axis 1, 2, and 3 position data  
position data  
Axis 1, 2, and 3, 2-6  
Axis 1,2,3 position data clock input/  
output  
input/output  
Axis 1,2,3 position data clock  
input/output, 2-5  
Axis 1,2,3 position data output

## output

Axis 1,2,3 position data, 2-5  
Axis 2 connector, 2-5  
Axis 2 connector, 1-3  
Axis 3 connector, 2-5  
Axis 3 connector, 1-3

## B

backplane I/O  
PC-compatible controller, 2-6  
Bitmap A, 3-4  
Bitmap B, 3-4  
Bitmap C, 3-4  
Bitmap D, 3-5  
Bitmap E, 3-6  
Bitmap F, 3-6  
Bitmap G, 3-7  
Bitmap H, 3-7  
Bitmap I, 3-7  
board  
N1231A, illustrated, 1-3  
bus I/O  
PC-compatible controller, 2-6

## C

C/C++ programming information, B-2  
cables  
10880A/B/C Receiver Cable, 2-5, 4-4  
10881A/B/C Laser Head Cable, 2-5, 4-4  
N1250A/B High Performance Receiver  
Cable, 2-5, 4-6  
N1251A/B High Performance Laser  
Head Cable, 2-5, 4-6  
PMAC ribbon, 2-7  
CD  
Monitor software, xii  
N1231A, xii  
CD readme file, C-2  
characteristics, 6-2  
compact disk  
Monitor software, xii  
compensation  
wavelength, 5-4  
configuration  
system, 4-1  
connector  
Axis 1, 1-3, 2-5  
Axis 2, 2-5  
Axis 3, 2-5  
Meas 1, 1-2  
Meas 2, 1-2  
Meas 3, 1-2  
PCI, 2-6

- PCI Interface, 1-3
- PMAC compatible, 2-5
- Ref, 1-2
- connector panel, 1-2, 2-3
- controller, PC-compatible, 1-4, 4-2, 4-3
- cooling requirements, 6-5

**D**

- delays
  - fixed, 5-4
- Delta Tau ACC-14D boards, 4-2
- Delta Tau products, 4-4, 4-6
- Delta Tau, PMAC motion control system, xii
- description of the N1231A, xii
- detection of error, 2-7
- diagram
  - timing, 6-6
- digital filters, Kv and Kp, table, 3-5
- dimensions, physical, 6-6
- downloading
  - firmware, A-2

**E**

- E1708A or E1709A receivers, 4-6
- E1708A Receiver, 2-5
- E1709A Receiver, 2-5
- electrical specifications, 6-4
- Electrostatic Discharge, ESD, 6-5
- error
  - Velocity register, 3-3, 3-8
- error and interrupt programming, 2-7
- error calculation, velocity register, 3-8
- error detection, 2-7
- errors, 2-6
- ESD, 6-5
- exchange assembly, 1-9

**F**

- features and functions of N1231A, 2-2
- filter settings in Bitmap D, E-1
- firmware
  - downloading, A-2
- fixed delays, 5-4
- frequency response graphs, E-3
- functional block description
  - functional block, 5-2

**G**

- getting started, 1-1
- graphs
  - frequency response, E-3
  - temporal response, E-7

**H**

- handshake timing diagram

- PMAC, 6-6
- handshaking and latching
  - Method 4, 2-5
- header file
  - C/C++, B-2

**I**

- installation, 1-4
- installing API library and Monitor program, 1-4
- installing board, 1-4
- installing board in a PC-compatible controller, 4-4, 4-5, 4-6, 4-7
- installing software, 1-4
- interface signals, 2-3
- interrupt generations on error conditions, 2-6

**K**

- Kv and Kp digital filters, table, 3-5

**L**

- laser head
  - 5517B/C/D, 4-4, 4-6

**M**

- material temperature sensor
  - 10757D/E/F, 4-3, 4-4, 4-6
- Maximum Axis Velocity of different Agilent Laser Head models, 6-3
- Meas 1 connector, 1-2
- Meas 2 connector, 1-2
- Meas 3 connector, 1-2
- mechanical specifications, 6-6
- Memory Map, 3-3
- memory map, 3-2
- Method 4 Latching/Handshaking, 2-5
- monitor application installation from CD
  - installing Monitor program from CD, 1-4

**N**

- N1231A API Readme File, D-2
- N1231A board, illustrated, 1-3
- N1231A specifications and characteristics, 6-2
- N1231A-200 CD, xii
- N1250A/B High Performance Receiver
  - Cable, 2-5, 4-6
- N1251A/B High Performance Laser Head
  - Cable, 2-5, 4-6

**O**

- operational verification, 1-5
- option
  - N1231A-200 CD, xii
- overview of system, xii, 4-2

**P**

panel  
     at a glance, 1-2  
     connectors, 1-2  
 panel of N1231A, 2-3  
 panel of N1231A, illustrated, 1-2  
 pc board, illustrated, 1-3  
 PC-compatible controller, 1-4, 4-2, 4-3  
 PC-compatible controller backplane, 2-6  
 PCI bus, 2-6  
 PCI Bus outputs  
     output  
         PCI Bus, 2-6  
 PCI edge connector, 1-3  
 PCI Interface, 1-3  
 PMAC Board, 4-4  
 PMAC board, 4-4  
 PMAC compatible connectors, 2-5  
 PMAC Handshake timing diagram, 6-6  
 PMAC servo controller, 4-2  
 Position Measurement Block, 5-3  
 position resolution  
     Axis 1, Axis 2, and Axis 3, 5-4  
 position system status and errors, 2-6  
 power requirements, 6-5  
 power supply, 4-4, 4-6  
 preface, xi  
 preparing for use, 1-4  
 product support, 1-9  
 programming information, 3-2, B-2

**R**

readme file  
     API, D-2  
     CD, software installation  
         instructions, C-2  
 receiver  
     E1708A, 4-3  
 receiver cables, 2-5  
 receivers  
     10780C/F, 2-5, 4-3  
     E1708A, 2-5, 4-3  
     E1709A, 2-5, 4-3  
 Ref connector, 1-2  
 Registers  
     Bitmap A, 3-4  
     Bitmap B, 3-4  
     Bitmap C, 3-4  
     Bitmap D, 3-5  
     Bitmap E, 3-6  
     Bitmap F, 3-6  
     Bitmap G, 3-7  
     Bitmap H, 3-7  
     Bitmap I, 3-7  
 registers information, 3-2  
 resolution and accuracy, 5-4

resolution and range, 6-3  
 ribbon cable, 2-7

**S**

servo controller, PMAC, 4-2  
 setting jumpers on ACC-14D board, 4-8  
 signals  
     interface, 2-3  
 slew rate, maximum, 6-3  
 specifications  
     electrical, 6-4  
     general, 6-2  
     mechanical, 6-6  
 switches  
     address, illustrated, 1-3  
     S1 and S2, 1-3  
 system configuration, 4-1  
 system overview, xii, 4-2

**T**

temporal response graphs, E-7  
 theory of operation, 5-2  
 timing diagram, 6-6

**U**

use  
     preparing, 1-4

**V**

Velocity register error, 3-3, 3-8  
 velocity, maximum, 6-3  
 verification installation of board and  
     monitor application, 1-5

**W**

waveform  
     timing, 6-6  
 wavelength compensation, 5-4



## Service and Support

### Contacting Agilent Technologies:

For more information about Agilent test and measurement products, applications, and services, visit our web site at <http://www.agilent.com>.

### Technical Support:

If you need technical assistance with an Agilent test and measurement product or application, you can find a list of local service representatives on the web site at <http://www.agilent.com>. If you do not have access to the Internet, one of the following centers can direct you to your nearest representative:

#### Asia Pacific:

Hong Kong  
**Tel: 800 930 871**

Indonesia  
**Tel: (62-21) 380-0902**

Japan  
**Tel: 0120-421-345**

Korea  
**Tel: 82-2-2004-5004**

Malaysia  
**Tel: 1-800-828-848**

Philippines  
**Tel: 1-800-828-848**

Singapore  
**Tel: 1800-375-8100**

#### Canada:

**Tel: 877-894-4414**

#### Europe:

France  
**Tel: +33 0825 010 700**

Germany  
**Tel: 49 (01805) 24 6333**

Ireland  
**Tel: +353 1890 924 204**

Netherlands  
**Tel: 31 (0) 20 547 2111**

Scotland  
**Tel: 07004-666666**

UK  
**Tel: 07004-666666**

#### United States:

Test & Measurement Call Center  
Englewood, CO, U.S.A.

**Tel: (800) 452-4844** (Toll free in US)  
Fax: (303) 662-3726



Agilent Technologies

## DECLARATION OF CONFORMITY

According to ISO/IEC Guide 22 and CEN/CENELEC EN 45014



**Manufacturer's Name:** Agilent Technologies, Incorporated  
**Manufacturer's Address:** Santa Clara Site  
5301 Stevens Creek Blvd  
Santa Clara, California 95051

### Declares, that the product

**Product Name:** PCI Three-Axis Laser Board  
**Model Number:** N1231A  
**Product Options:** *This declaration covers all options of the above product.*

### Conforms with the following European Directives:

*The product herewith complies with the requirements of the Low Voltage Directive 73/23/EEC and the EMC Directive 89/336/EEC (including 93/68/EEC) and carries the CE Marking accordingly.*

EMC	Standard	Limit
	IEC 61326-1:1997+A1:1998 / EN 61326-1:1997+A1:1998 CISPR 11:1990 / EN 55011:1991 IEC 61000-4-2:1995+A1:1998 / EN 61000-4-2:1995 IEC 61000-4-3:1995 / EN 61000-4-3:1995 IEC 61000-4-4:1995 / EN 61000-4-4:1995 IEC 61000-4-5:1995 / EN 61000-4-5:1995 IEC 61000-4-6:1996 / EN 61000-4-6:1996 IEC 61000-4-11:1994 / EN 61000-4-11:1994	Group 1 Class A <sup>[1]</sup> 4kV CD, 8kV AD 3 V/m, 80-1000 MHz 0.5kV signal lines, 1kV power lines 0.5 kV line-line, 1 kV line-ground 3V, 0.15-80 MHz 1 cycle, 100%
	Canada: ICES-001:1998 Australia/New Zealand: AS/NZS 2064.1	
<b>Safety</b>	IEC 61010-1:1990+A1:1992+A2:1995 / EN 61010-1:1993+A2:1995	

### Supplemental Information:

<sup>[1]</sup> *The product was tested in a typical configuration with Agilent Technologies test systems.*

August 16, 2001  
Date

  
Art Nanawa, Regulations Manager

For further information, please contact your local Agilent Technologies sales office, agent or distributor.  
Authorized EU-representative: Agilent Technologies Deutschland GmbH, Herrenberger Straße 130, D 71034 Böblingen, Germany

*Continued from front matter. . .*

#### **Warranty (contd)**

Agilent does not warrant that the operation of Agilent products will be uninterrupted or error free. If Agilent is unable, within a reasonable time, to repair or replace any product to a condition as warranted, customer will be entitled to a refund of the purchase price upon prompt return of the product.

Agilent products may contain remanufactured parts equivalent to new in performance or may have been subjected to incidental use.

The warranty period begins on the date of delivery or on the date of installation if installed by Agilent. If customer schedules or delays Agilent installation more than 30 days after delivery, warranty begins on the 31st day from delivery.

Warranty does not apply to defects resulting from  
(a) improper or inadequate maintenance or calibration,  
(b) software, interfacing, parts or supplies not supplied by Agilent,  
(c) unauthorized modification or misuse,  
(d) operation outside of the published environmental specifications for the product, or (e) improper site preparation or maintenance.

TO THE EXTENT ALLOWED BY LOCAL LAW, THE ABOVE WARRANTIES ARE EXCLUSIVE AND NO OTHER WARRANTY OR CONDITION, WHETHER WRITTEN OR ORAL, IS EXPRESSED OR IMPLIED AND AGILENT SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OR CONDITIONS OF MERCHANTABILITY, SATISFACTORY QUALITY, AND FITNESS FOR A PARTICULAR PURPOSE.

Agilent will be liable for damage to tangible property per incident up to the greater of \$300,000 or the actual amount paid for the product

that is the subject of the claim, and for damages for bodily injury or death, to the extent that all such damages are determined by a court of competent jurisdiction to have been directly caused by a defective Agilent product.

TO THE EXTENT ALLOWED BY LOCAL LAW, THE REMEDIES IN THIS WARRANTY STATEMENT ARE CUSTOMER'S SOLE AND EXCLUSIVE REMEDIES. EXCEPT AS INDICATED ABOVE, IN NO EVENT WILL AGILENT OR ITS SUPPLIERS BE LIABLE FOR LOSS OF DATA OR FOR DIRECT, SPECIAL, INCIDENTAL, CONSEQUENTIAL (INCLUDING LOST PROFIT OR DATA), OR OTHER DAMAGE, WHETHER BASED IN CONTRACT, TORT, OR OTHERWISE.

For consumer transactions in Australia and New Zealand: the warranty terms contained in this statement, except to the extent lawfully permitted, do not exclude, restrict or modify and are in addition to the mandatory statutory rights applicable to the sale of this product to you.

#### **Assistance**

Product maintenance agreements and other customer assistance agreements are available for Agilent products.

For any assistance, contact your nearest Agilent Sales and Service Office.

#### **Safety Considerations (contd)**

##### **WARNING**

**INSTRUCTIONS FOR ADJUSTMENTS WHILE COVERS ARE REMOVED AND FOR SERVICING ARE FOR USE BY SERVICE-TRAINED PERSONNEL ONLY. TO AVOID DANGEROUS ELECTRIC SHOCK, DO NOT PERFORM SUCH ADJUSTMENTS OR SERVICING UNLESS QUALIFIED TO DO SO.**

#### **Acoustic Noise Emissions**

LpA<47 dB at operator position, at normal operation, tested per EN 27779. All data are the results from type test.

#### **Geräuschemission**

LpA<47 dB am Arbeitsplatz, normaler Betrieb, geprüft nach EN 27779. Die Angaben beruhen auf Ergebnissen von Typenprüfungen.



**Agilent Technologies**



Manual Part Number N1231-90002

Printed in U.S.A, MAY 2002