VXIpc™ Embedded Controller for VxWorks

This document contains information to help you understand the components of your kit, determine where to start setting up your kit, and learn about the NI-VXI/VISA features.

Contents

What Do You Have? 2
- Hardware 2
- Software 2
- Printed Documentation 2
- Online Documentation 3

Where Do You Start? 3
What Is NI-VXI? 5
What Is VISA? 5
NI-VXI/VISA Release Notes 6
- Supported Application Development Environments 6
- The NI-VXI Directory 6
- The VXIpnp Directory 6
- Features and Terminology 7
  - Window Mapping 7
  - MITE DMA 9
  - Shared Memory 10
  - Remote Controllers 10
- Enhancements to the NI-VXI Software 11
  - Compatibility 11
  - System Configuration Functions 11
  - Low-Level VXIbus Access Functions 11
  - High-Level VXIbus Access Functions 11
- NI-VISA for VxWorks 12

MITE™, NI-488™, National Instruments™, ni.com™, NI-VISA™, NI-VXI™, VXIpc™ are trademarks of National Instruments Corporation. Product and company names are trademarks or trade names of their respective companies.

 © Copyright 1997, 1999 National Instruments Corp. All rights reserved.

What Do You Have?

Your kit contains hardware, software, and documentation. You may also have ordered some optional equipment or software.

Hardware

Your hardware includes either the VXIpc 800 Series or the VXIpc 700 Series controller, which you install in your VXI mainframe. You also receive the following accessories:

- 8 in. serial conversion cable (available for the VXIpc 800 Series only)
- Electromagnetic emissions kit
- AT-PS/2 cable

Software

Your VXIpc kit contains the following software components:

- Host-Side Software for VXIpc for VxWorks (Microsoft Windows NT/98 Installation)
- NI-VXI for VXIpc 800/700 Series for VxWorks (target software)
- NI-VISA for VXIpc 800/700 Series for VxWorks (target software)
- NI-488 for VXIpc 800/700 Series for VxWorks (target software)

For your convenience, NI-VXI/VISA, NI-488 and a VxWorks bootroom loader (`bootrom.sys`) are already installed on the hard drive of your VXIpc embedded controller.

Printed Documentation

Aside from this document, your kit contains the following printed manuals:

- *Getting Started with Your VXIpc Controller for VxWorks* contains an overview of the VXIpc hardware and the NI-VXI software, guides you through setting up your kit, and helps you get started with application development. You can also use this manual as a reference for the hardware and software default settings, to find the answers for commonly asked questions, and for information on reinstalling the software, if necessary.
- The *VXIpc 800 Series User Manual* (in VXIpc 800 kits) or the *VXIpc 700 Series User Manual* (in VXIpc 700 kits) contains information on configuring and installing your VXIpc controller. You may not need this manual if you are performing a simple hardware installation that uses the standard default settings. However, you should keep this manual in case you decide to try a different switch

or jumper setting at a later time. The user manual also contains information on BIOS, the front panel and connectors, the LEDs, and system resources.

Online Documentation

Your kit includes links to online manuals that are not in printed form, but in electronic form: Adobe Acrobat version 3.0 portable document format (PDF) files. This online documentation is installed by the *Host-Side Software for VXIpc for VxWorks* setup software and is intended for use on the Microsoft Windows NT/98/95 operating systems. You can view these manuals online, navigate through them, and print them from your computer using the Adobe Acrobat Reader 3.0 (or later version).

The National Instruments manuals in PDF format are as follows:

- The *NI-VXI User Manual* describes the features and concepts of the NI-VXI software and explains how to use NI-VXI in your application.
- The *NI-VXI Programmer Reference Manual* describes in detail the VXI/VME function calls in the C/C++ and BASIC languages.
- *NI-VXI Text Utilities Manual*
- The *NI-VISA User Manual* describes how to program using VISA.
- The *NI-VISA Programmer Reference Manual* describes in detail the attributes, events, and operations you use in NI-VISA.
- The *GPIB Reference Manual for VXIpc Embedded Controllers and VxWorks* contains information about using NI-488 on your VXIpc controller with the VxWorks operating system.

To access these manuals, the NI-VXI and NI-VISA function references, and the ReadMe files, follow the links in the appropriate `manuals.html` file in your Board Support Package (BSP). The files you use to access the online manuals are `manuals-vxi.html`, or `manuals-visa.html`, or `manuals-gpib.html`. See Step 3 in the *Where Do You Start?* section of this manual for more information on the BSP.

Where Do You Start?

1. Compare your kit contents with the description in the preceding *What Do You Have?* section. Contact National Instruments regarding any discrepancies.
2. VxWorks uses a *target* and *host* machine. If you are unfamiliar with this concept, refer to the *Developing for VxWorks* section in Chapter 1, *Introduction*, of *Getting Started with Your VXIpc Controller for VxWorks* (hereafter referred to as your getting started manual).

3. The following steps assume that Tornado, the VxWorks host-side development environment, is already installed on your host computer. Run the Setup program from the *Host-Side Software for VXIpc for VxWorks* installation disk to install the software. This disk is intended for Windows NT/98 *host* machines. For a first time installation, we recommend that you install all of the software, which contains the VxWorks Board Support Package (BSP), the VXI/VME and GPIB driver component software, NI-VISA support for VxWorks, online help, examples, and header files for NI-VXI, NI-VISA, and NI-488.

Caution The current version of the VXIpc BSP requires Tornado II. If you have an older version of Tornado, contact Wind River for an upgrade.

4. Refer to the *Using the VXIpc Board Support Package to Create the VxWorks Operating System* section in Chapter 1, *Introduction*, of your getting started manual. This section contains information about building the VxWorks operating system, which will run on the VXIpc, using the VXIpc Board Support Package (BSP). This section also contains information about configuring the VXIpc BSP.
5. On the *target* side, your VXIpc should have the NI-VXI, NI-VISA, and NI-488 software already installed, along with a working `bootrom.sys` file to help you get started.
   a. If you need to reinstall the NI-VXI software for any reason, use the disks labeled *Target-Side Software for VXIpc for VxWorks*. You can also refer to the *Reinstalling the NI-VXI Software* section in Chapter 1, *Introduction*, of your getting started manual.
   b. A working `bootrom.sys` is included with the host-side software and on the target-side software disks, should it become necessary to reinstall that component.
6. Most VXI/VME features on the VXIpc controller are configurable through software. Use the vxitedit configuration utility in the NI-VXI software if you want to change any of the options from their default settings. Run Resman to set up your VXI system and victext to verify the system configuration.
7. After you finish setting up your system, refer to the following documents to learn how to use VXI/VME or GPIB.
   a. Refer to Chapter 4, *Developing Your Application*, in your getting started manual to learn how to use your VXI/VME system and ensure it is operating properly.
   b. Refer to Chapter 2, *Developing Your Application with the VxWorks Drivers*, in the *GPIB Reference Manual for VXIpc Embedded Controllers and VxWorks* to learn how to use the

VxWorks GPIB drivers and how to develop a GPIB application using the NI-488 API.

c. Refer to Chapter 3, *VISA Overview*, in the *NI-VISA User Manual* for information about programming with VISA. Use the Acrobat Reader to view this manual online.

8. Please refer to the `README.TXT` files for important information that may affect your application program, including known issues and software corrections with this release and additional information relevant to API development.

   You can access the `README.TXT` files in the directory where you installed the software for VxWorks on your host machine.

   You can also reference the National Instruments sites `http://www.ni.com` or `ftp.ni.com` for driver updates, examples, and product news.

What Is NI-VXI?

The NI-VXI system-level software is the driver that controls your VXIpc and VXI/VME system. NI-VXI includes a Resource Manager, an interactive configuration program, libraries of software routines for test and measurement programming, and an interactive control program. You can use this software to seamlessly program multiple-mainframe configurations and ensure software compatibility across a variety of controller platforms.

What Is VISA?

VISA is a standard I/O Application Programming Interface (API) for instrumentation programming. In VxWorks, VISA by itself does not provide any instrumentation programming functionality. VISA is a high-level API that calls into system-level drivers. As an example, the NI-VISA implementation of VISA uses the NI-VXI system-level driver for National Instruments VXI controllers.

In its full implementation, VISA can control VXI, PXI, GPIB, or Serial instruments, making the appropriate driver calls depending on the type of instrument being used. VISA uses the same operations to communicate with instruments regardless of the interface type. For example, the VISA command to write an ASCII string to a message-based instrument is the same whether the instrument is Serial, GPIB, or VXI. As a result, VISA gives you interface independence. This makes it easier to switch bus interfaces and means that users who must program instruments for multiple interfaces need learn only one API. This release of NI-VISA for VxWorks

provides the VXI portion of the API, and is compatible with the VXI portions of VISA programs written for other platforms.

Another advantage of VISA is that it is an object-oriented API that will easily adapt to new instrumentation interfaces as they evolve, making application migration to the new interfaces easy.

Because VISA is the industry standard for developing instrument drivers, most instrument drivers currently written by National Instruments use VISA and therefore support VxWorks, Windows 3.*x*, Windows NT/98/95, Solaris 1, Solaris 2, HP-UX, and Macintosh, if the system-level drivers are available for that platform.

NI-VXI/VISA Release Notes

This section describes the new utilities and features in this release of NI-VXI for VxWorks.

Supported Application Development Environments

This release of NI-VXI/VISA for VxWorks supports the Tornado gcc compiler version 2.7. Although NI-VXI and NI-VISA have been tested and found to work with this Application Development Environment (ADE), other ADEs or a later version of this ADE may also work.

The NI-VXI Directory

The `nivxi` directory on your VXIpc hard drive contains important files that describe the configuration of your hardware and software. By default, the library searches for these files in the directory `/ide0/nivxi`, which corresponds to `C:\NIVXI` in MS-DOS and means that the `nivxi` directory must be at the root level of the hard drive. If, for some reason, you must move or rename the directory, use the environmental variable `NIVXIPATH` to inform the library of the new location. The VxWorks method for setting an environment variable uses the `putenv("variableName=value")` function. For example, to instruct the library to search for the NI-VXI files in a subdirectory of your NI-VXI directory, your function syntax would be `putenv("NIVXIPATH=/ide0/nivxi/vxworks")`.

The VXIpnp Directory

The `vxipnp` directory on your VXIpc hard drive, like the `nivxi` directory, contains important information used by NI-VISA. If the files in that location (`/ide0/vxipnp`) are not available, VISA uses its default configuration options.

Features and Terminology

NI-VXI/VISA exploits the MITE ASIC to provide a rich implementation of the VXI interface, especially in regard to the following:

- Window mapping
- MITE DMA
- Shared memory
- Remote controllers

Window Mapping

The MITE architecture allows much more flexibility in low-level mapping of VXI address spaces. In particular, the CPU interface of the MITE has windows that can be dynamically resized and relocated from CPU space to VXI space. The low-level functions have new extensions that reflect this feature. Refer to the NI-VXI online help or the *NI-VXI Programmer Reference Manual* for information about these functions in NI-VXI. The NI-VISA online help and the *NI-VISA Programmer Reference Manual* cover this information for NI-VISA applications. Use the Acrobat Reader 3.0 to view and navigate through these manuals.

As in earlier versions of the drivers, the functions `MapVXIAddress()` and `viMapAddress()` check whether a window that can be shared already maps to the desired address space and location. If so, they return a pointer to that window. If the desired space is not already mapped, they set up a new MITE window to the VXI address and return a pointer to the new window.

The `MapVXIAddressSize()` function is the standard mechanism for specifying how large a window the driver should map on a call to `MapVXIAddress()`. The default size of a mapped window when using NI-VXI is 64 KB. In VISA, you specify the window size directly in `viMapAddress()`.

The success of this allocation depends on the availability of three factors:

- Address space in the user window
- Number of MITE windows
- Memory for allocating data structures for the map

The MITE has two windows that must be set up in the VxWorks operating system. The driver window has a fixed size and, unless you change the hardware in your VXIpc, you probably should not alter this window. The user window defaults to 64 KB.

If you need a different user window size, perform the following steps:

1. Change the **user window size** field in the vxitedit Bus Configuration Editor.
2. Reboot the target to allow the changes to take effect.

Address Space

The VXIpc controller can decode any 32-bit address on the PCI bus as a VXI cycle, giving 4 GB of addressability, which can be used for windows on the VXIpc. The operating system or computer architecture may limit which addresses can be assigned to the VXIpc.

To change the address space, use vxitedit to edit the **user window size** field in the **VXIpc Bus Configuration Editor**. This setting limits the total amount of memory you can map with `MapVXIAddress()` or `viMapAddress()`. If the user window is disabled, the `MapVXIAddress()` function returns `NO_HARDWARE_SUPPORT (-1)`.

The *NI-VXI Programmer Reference Manual* implies that the error code `MAP_TIMEOUT (-8)` is returned when the window is in use. Because the MITE-based products have multiple hardware windows of variable size, the meaning of this error has been modified. `MapVXIAddress()` now returns the error code `MAP_TIMEOUT (-8)` whenever there are not enough resources to map the window.

For example, if you use `MapVXIAddressSize()` and `MapVXIAddress()` to request a 1 MB window to A32 space and you request a user window of only 64 KB, `MapVXIAddress()` returns the error code `MAP_TIMEOUT` because there are not enough resources to complete the request.

Number of MITE Windows

The MITE has eight CPU windows. NI-VXI uses four of these windows, leaving four for user applications.

Memory for Allocating Data Structures

You need to have sufficient memory available to set up the necessary page tables. If you request a very large window—hundreds of megabytes, for example—you may run out of memory.

MITE DMA

The MITE has two DMA channels to improve the throughput of block transfers to and from the VXI system. The DMA channels can use various high-speed bus protocols, such as the following:

- VXI block
- Burst mode (on the PCI bus)
- VME64 (on the VXI bus)

The DMA channels can transfer data between a VXI device and local memory, or between VXI devices. The DMA channel can handle contiguous or noncontiguous local memory. If it is handling noncontiguous memory, it can perform scatter-gather operations on the noncontiguous memory.

The `VXImove()` and `viMove`*`XX`*`()` functions automatically use appropriate bus protocols and transfer types to efficiently perform the data transfer specified in the function. You can also configure the NI-VXI/VISA software to use DMA channels for particular types of operations and to designate what protocols the channel should use. In addition, you can programmatically control which protocols to use in NI-VXI. See the NI-VXI online help or the *NI-VXI Programmer Reference Manual* for complete descriptions of `VXImove()` and other high-level functions. Notice that previously written NI-VXI and NI-VISA code use the DMA capabilities of the MITE without modification.

To take full advantage of the throughput of the DMA channels, you should perform 32-bit transfers in which both the source and the destination are longword aligned. If you need to transfer character data between devices of different byte orders—for example, between a big-endian device and an Intel 80*x*86-based VXIpc—transfer the data as longwords but adjust the byte-ordering parameters in `VXImove()` to get the correct data in the most efficient manner.

Examples:

```
/* Transferring 32-bit data to a big-endian A32 device */
VXImove(0x0, userBuffer, 0x3, deviceOffset, numDataPoints, 4);
/* Transferring 8-bit data to a big-endian A32 device */
VXImove(0x80, userBuffer, 0x3, deviceOffset, numDataPoints / 4, 4);
```

Shared Memory

The **Logical Address Editor** includes options you can use to share memory on your computer. Consult the NI-VXI online help or Chapter 3, *NI-VXI Configuration Utility*, of *Getting Started with Your VXIpc Controller for VxWorks* for more information. You can use the `VXImemAlloc()` function to access shared memory on your computer.

Remote Controllers

This section applies to multiframe systems where you use interrupts, triggers, or utility bus signals.

Remote controllers, when configured to detect asynchronous events such as a VXI interrupt or VXI trigger, need to inform the local controller that such an asynchronous event has occurred. The remote controllers report these events back to the local controller via a VXI IRQ line. This IRQ line is called the *system IRQ line*. You can use vxitedit to select which VXI interrupt line the remote controller uses to report remote events to the local controller. Map the system IRQ line back to the local controller to receive remote controller interrupts. The Resource Manager performs this mapping automatically in the parent-side VXI-MXI-2 controllers, but not in other mainframe extenders. You can map interrupts through vxitedit, or with the `MapVXIint()` function, which is described in the NI-VXI online help or the *NI-VXI Programmer Reference Manual*.

The system IRQ line is treated differently than other IRQ lines used by NI-VXI:

- The Resource Manager (Logical Address 0) always acknowledges the system IRQ line.
- You cannot disable the system IRQ line on the Resource Manager. Calling `DisableVXIint()` on the system IRQ line does *not* disable it.
- Devices other than remote controllers can also interrupt on the system IRQ line, provided the device at Logical Address 0 is the handler for the interrupt.
- National Instruments recommends that you do *not* route the system IRQ line to the signal queue. Because the system IRQ line cannot be disabled, this routing could lose interrupts.

Passing the value –1 as the logical address of a controller in NI-VXI causes NI-VXI to select the *local* controller. Notice that on external controllers such as the PCI-MXI-2, –1 refers to the first *remote* controller in your system. This is to maintain compatibility with older systems where the external controller needed an extender to assert and receive interrupts.

Enhancements to the NI-VXI Software

The following sections describe the additional options beyond what is documented in the *NI-VXI User Manual* and the *NI-VXI Programmer Reference Manual*.

Compatibility

NI-VXI applications that follow the guidelines documented in the *NI-VXI User Manual* will work with NI-VXI for the VXIpc embedded controllers.

System Configuration Functions

The `InitVXIlibrary()` function has a new return value of `INIT_RET_OK_RMERROR (2)`. If this value is returned, it means "The NI-VXI library successfully initialized, but the Resource Manager has not been run successfully." Always run the Resource Manager before using the NI-VXI library.

Low-Level VXIbus Access Functions

Do not make any assumptions about the size and features of a window returned from `MapVXIAddress()`. Use `GetWindowRange()` to determine the size of a window.

The 32-bit value returned from `GetContext()` and passed to `SetContext()` has a new format. Applications that set the context bits directly for use in `SetContext()` may not be compatible with the new format for context. Because the MITE allows more flexible window mapping, extra bits have been added to this field to reflect these new features. Do not manipulate the context bits directly.

High-Level VXIbus Access Functions

For best performance, keep the following in mind when using `VXImove()`:

- Make sure your buffers are 32-bit aligned.
- Transfer 32-bit data whenever possible.
- Using VXI block access privileges significantly improves performance to devices that can accept block transfers.
- `VXImove()` must lock the user buffer in memory on virtual memory systems, so locking the buffer yourself optimizes `VXImove()`.
- Because `VXImove()` must build a scatter-gather list for the user buffer on paged memory systems, using a contiguous buffer optimizes `VXImove()`.

`VXImemAlloc()` returns 32-bit aligned, page-locked, contiguous buffers, which work efficiently with `VXImove()`, but only if the function returns `MEM_OK (0)`. A status of `MEM_OK_USE_MEMCOPY (1)` means this buffer cannot be used directly with `VXImove()`.

`VXImove()` can also move blocks of data to and from a single VXI address. This is commonly referred to as *FIFO mode*. Refer to the *NI-VXI Programmer Reference Manual* for more information.

NI-VISA for VxWorks

NI-VISA Version 1.3 for VxWorks currently supports only the VXI interface.

321847C-01 Dec99