Read Me First: 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/NI-VISA features.

Contents

What Do You Have?

Your kit contains hardware, software, and documentation. Any optional equipment or software should be present in the kit also.

April 2003
321847F-01

NATIONAL INSTRUMENTS

Hardware

Your hardware includes a VXIpc embedded controller, which you install in your VXI mainframe, and the following accessories (as appropriate for the model of your VXIpc):

- 8 in. serial conversion cable
- AT-PS/2 cable
- 6 in. IEEE-1284 parallel conversion cable
- 2 m GPIB conversion cable
- Electromagnetic emissions kit

Software

Your VXIpc kit contains the following software components:

- Host-Side Software for VXIpc with VxWorks (Microsoft Windows Installation)
- Target-Side Software for VXIpc with VxWorks

For your convenience, NI-VXI/NI-VISA, GPIB interface software, and a VxWorks boot ROM 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 also can 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 770/870B Series User Manual* (in VXIpc 770/870B Series kits), *VXIpc 870 Series User Manual* (in VXIpc 870 kits), *VXIpc 800 Series User Manual* (in VXIpc 800 kits), or *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—Adobe Acrobat portable document format (PDF) files. This online documentation is installed by the *Host-Side Software for VXIpc with VxWorks* setup software and is intended for use on the Microsoft Windows operating systems. You can view these manuals online, navigate through them, and print them from your computer using the Adobe Acrobat Reader.

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.
- The *NI-VXI Text Utilities Reference Manual* describes in detail the interactive text utilities for the NI-VXI software.
- 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-488.2 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 HTML file in your Board Support Package (BSP). The files you use to access the online manuals are `manuals-vxi.html`, `manuals-visa.html`, and `manuals-gpib.html`. See step 3 in the *Where Do You Start?* section for more information on the BSP.

Where Do You Start?

1. Compare your kit contents with the description in the *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 Embedded 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 with*

VxWorks installation disk to install the software. This disk is intended for Windows *host* machines. For a first time installation, we recommend you install all 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-488.2.

Note The VXIpc BSP designated **vxipc** requires Tornado 2.0.2. For Tornado 2.2, use the VXIpc BSP designated **vxipc2**. If you have an earlier 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 GPIB interface software already installed, along with a working `bootrom.sys` file to help you get started.
   - If you need to reinstall this software for any reason, use the disks labeled *Target-Side Software for VXIpc with VxWorks*. You also can refer to the *Reinstalling the NI-VXI Software* section in Chapter 1, *Introduction*, of your getting started manual.
   - A working `bootrom.sys` is included 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.
   - 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.
   - 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-488.2 board-level API.

- 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. 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 where you installed the software for VxWorks on your host machine. There are separate `README` files for several of the components.

   You can also reference the National Instruments Web sites `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 NI-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, TCP/IP, 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 and Serial portions of the API and is compatible with VISA programs written for those interfaces on 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, Solaris, Linux, and Macintosh, if the system-level drivers are available for that platform.

NI-VXI/NI-VISA Release Notes

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

Supported Application Development Environments

This release of NI-VXI/NI-VISA for VxWorks supports the Tornado 2.2 gcc compiler, version 2.9, with the **vxipc2** BSP. The release also includes the older Tornado 2.0.2 **vxipc** BSP which works with the Tornado 2.0.2 gcc compiler, version 2.7. Other Application Development Environments (ADEs) or a later version of this ADE also may 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 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 will not work. To instruct the library to search for the NI-VISA files in a different location, use the `putenv` syntax described above; for example, to use a floppy disk for your `VXIpnp` directory, your function syntax would be `putenv("VXIPNPPATH=/fd0/vxipnp/")`.

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 with VxWorks.

High-Level VXIbus Access Functions

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

- Make sure your buffers are 32-bit aligned.
- Transfer 32-bit data whenever possible.
- Use VXI block access privileges to significantly improve performance to devices that can accept block transfers.

Interrupts in NI-VXI for VxWorks

To provide real-time performance, NI-VXI and NI-VISA event handlers in VxWorks behave differently than those in other operating systems. When you install a handler for a VXI event such as an interrupt, trigger, or signal, and the appropriate event occurs, the interrupt service routine (ISR) executes your code. The context for the ISR is independent of any task and has certain restrictions regarding what operating system functions may be used because, for the duration of the ISR, no normal task will execute. For example, you may not attempt to acquire a semaphore (`semTake`) in the ISR. Refer to your VxWorks reference manuals for details on functions which may not be executed in the ISR. If your application requires a function with this restriction, or performs other complex or time-consuming operations, you may want to consider using a signal or a semaphore to alert a "worker" task which will respond to the interrupt outside of the ISR. The worker task can be given high priority, if needed, to ensure optimal performance.

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 can be

configured to 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. If you disable the system IRQ line by setting it to zero, you can configure asynchronous events to be reported back to the controller by mapping each specific event (trigger or interrupt) individually. 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 also can 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.