Document Type: Tutorial
NI Supported: Yes
Publish Date: Apr 16, 2009

Development of PXI Modules

Overview

This paper discusses development of custom PXI modules. First, the main features of the CompactPCI and PXI specifications are discussed. These specifications define the fundamental implementation and functionality of a PXI module. Second, an overview and comparison of three PXI Prototyping boards on the market is presented, all of which provide an excellent starting point for PXI module development. Third, the PXI software framework and process for developing PXI drivers is presented.

Table of Contents

Overview

PXI (PCI eXtensions for Instrumentation) defines a compact modular PC platform for industrial instrumentation. PXI leverages the PCI (Peripheral Component Interconnect) Bus, which is the de facto standard driving today's desktop computer software and hardware designs. Because PCI is so widespread and commonly used, efforts have been made to compliment the PCI specification for particular applications. To extend the capabilities of the PCI bus, the CompactPCI specification was created. It defines a rugged form factor for PCI that offers superior mechanical integrity and easy installation and removal of hardware components. PXI uses the standard architecture from the PCI specification and the CompactPCI specification for mechanical integrity and modularity to create a rugged, modular, computer-based measurement and control system. PXI supports the mechanical, electrical, and software features tailored to industrial instrumentation, data acquisition, and automation applications. PXI products offer higher and more carefully defined levels of environmental performance required by the vibration, shock, temperature, and humidity extremes of industrial environments. It also adds mandatory environmental testing and active cooling to the CompactPCI mechanical specification to ease system integration and ensure multi-vendor interoperability. Now, as the PXI specification grows in popularity, users are looking to create their own modules, which they can design with their particular applications in mind.

PXI and CompactPCI Breadboards

Some developers may prefer to build their PXI module from scratch. In this case, the main reference is the CompactPCI specification, which is built on the PCI specification but uses a different form factor and allows for eight cards per bus segment instead of four. It offers the form that can fit into rugged rack mount systems, while maintaining the performance of PCI and its compatibility with operating systems and applications software. CompactPCI features 33 and 66 MHz performance, 32-bit and 64-bit data transfers, and cards available in 3U and 6U sizes. The requirements of the CompactPCI specification must be observed when making a PXI device because the PXI specification adheres to the CompactPCI specification as well. The complete CompactPCI specification can be ordered from the PCI Industrial Computer's Manufacturing Group (PICMG) at www.picmg.org.

Software Implementation using VISA

The electrical requirements for CompactPCI board design must follow the PCI specification with several additional requirements. These additional requirements are separated into two groups, those requirements for 33 MHz operation, and those for 66 MHz operation.

The requirements for 33 MHz operation start with decoupling requirements because each CompactPCI board must have adequate decoupling for the board's intended application. The minimum decoupling requirements that should be used on a CompactPCI board are given in Table 2 of the CompactPCI specification.

Additional signals are defined in the electrical requirements of the CompactPCI specification, but not all of these are applicable in board design. They include Push Button Reset, Power Supply Status, System Slot Identification, Legacy IDE Interrupts, System Enumeration, Geographic Addressing, and System Management Bus. For more specific details on these signal lines, see section 3.2.7 of the CompactPCI specification.

There are varying requirements for stub termination, depending on the particular signal in question. There is a set of signals that must include a 10 stub termination resistor, which includes AD0-AD31, C/BE0#-C/BE3#, PAR, FRAME#, IRDY#, TRDY#, STOP#, LOCK#, IDSEL, DEVSEL#, PERR#, SERR#, and RST#. A second set of signals must be terminated if they are used by the board. They are INTA#, INTB#, INTC3, INTD#, AD32-AD63, C/BE4#-C/BE7#, REQ64#, ACK64#, and PAR64. Three particular signals do not require termination: CLK, REQ#, and GNT#. Stub termination is important because it minimizes the effect of the stub on each board to the backplane. The resistor used for termination must be placed within 15.2 mm of the connector pin for the signal and its length must be included in the overall allowable length of the trace. The overall signal length must be no greater than 63.5 mm, measured from the connector pin through the stub to the PCI device pin. Each PCI signal is allowed a maximum of one PCI load. Exceeding this will violate the CompactPCI specification. The PCI clock-signal length on peripheral boards must have a length of 63.5 mm 2.54 mm, and it may only drive one load on the board. The required characteristic impedance range for the signal traces is given in Table 4 of the CompactPCI specification.

A low impedance return path for logic ground between the board and the backplane must be provided, therefore the J1 and J2 connectors must load a shield at row F on the board. The IEC-61076 connector has a Z-row shield option, which is not required by the CompactPCI specification. This shield must not be loaded if it extends into the interboard separation plane, thereby violating the specification.

Creating a board that operates at 66 MHz requires following all design rules for a 33 MHz board, except where the rules for a 66 MHz board specifically addresses a design characteristic. The only major design difference for peripheral boards is in the PCI clock signal. The length must be 63.5 mm 1mm, and it must still drive only one load on the board. In order to equalize delays, inner-layer stripline construction must be used for all clock lines.

The CompactPCI specification recommends that CompactPCI boards use metalized shell front panel connectors for EMI/RMI protection. It also recommends connecting the shell electrically to the front panel through a low impedance path. The specification requires the front panel be connected to frame ground while being isolated from logic ground. It also recommends that a mechanism be provided to connect frame ground to logic ground through a low impedance path.

Summary

CompactPCI boards are available in two sizes, 3U and 6U. 3U will be the focus of this discussion. The dimensions of a 3U board are 100 mm by 160 mm, with the PCB measuring 1.6 0.2 mm thick, and a 2mm connector for interfacing to the CompactPCI bus. This connector is split into two sections, J1 and J2. J1 is used for 32-bit applications, and J2 may be used for 64-bit PCI signaling or rear-panel I/O. Rear-panel I/O is not used in PXI applications.

CompactPCI boards must be constructed to allow for safe electrostatic discharge (ESD). This is accomplished by using an ESD strip and an ESD clip in the board construction. All boards must have an ESD strip that complies with IEEE 1101.10 and IEEE 1101.11. The minimum requirement for ESD protection is to have ESD strips located on both sides of the bottom edge of the board. ESD strips are segmented into three sections to achieve a controlled discharge. The first section is for discharging static built up from the user or the board, the second segment is for discharging the board's ground planes, and the third is for discharging static from the backplane as the board is connected. The third segment connects directly to the front panel. There are additional ways of reducing ESD, such as protective clothing, and precautions should be observed according to the electrostatic content of the environment. An ESD clip is used to distribute ESD energy into the chassis as the board is inserted. It must be located in the lower card guide 35 mm 5 mm from the front attachment plane.

The regulations in IEEE 1101.1 and IEEE 1101.10 are used in the CompactPCI specification for the rules governing top and bottom connector shields, component outline, and warpage. The top connector shield must be installed to meet electrical requirements, while the bottom connector shield may be installed, so long as it does not extend into the interboard separation plane. The interboard separation plane boundaries also limit the total length of the component height/lead lengths and warpage. Their sum cannot exceed the boundaries.

Some boards may utilize backside components or through-hole pins. A board that uses either of these features should be equipped with a protective solder side cover. The cover is meant to protect components during installation or removal of the board from the chassis. The specification recommends using IEEE 1101.10 for guidance in the placement of mounting holes for the protective cover.

CompactPCI boards use front panels that are consistent with Eurocard packaging. The regulations for front panel design are laid out in IEEE 1101.1, which describes flat panels, and IEEE 1101.10, which describes EMC panels. Either panel type may be used, but the EMC panels are preferred. The front panel for 3U boards also must be equipped with an ejector/injector handle, as specified by IEEE 1101.10. This handle must be located on the bottom of the front panel. There also must be proper labels on the front panel, which include the CompactPCI logo and the proper compatibility glyph. Peripheral boards are labeled with a circular compatibility glyph.

The backplane connectors on a CompactPCI card are designated from bottom to top and numbered 1 to 5. The board connectors are designated with a "J" prefix, while the corresponding backplane connectors are designated with a "P" prefix. Therefore according to this labeling, the lowest connector on a CompactPCI board is labeled as the J1 connector. The connectors are of the 2 mm hard metric type, which is defined in IEC 61076-4-101. There are three possible housing types for the connectors, Type A, Type B, and Type AB. The differences in the three types exist in the inclusion of alignment features and coding keys. Type A connectors have both, Type B connectors have neither, and Type AB has only alignment features. Any of the three can be used for front panel I/O. The specification also discusses connector tail lengths, but these are only necessary for rear panel I/O and are outside the scope of this paper.

For CompactPCI boards of size 3U, only the J1 and J2 connectors are used, so our discussion will be limited to these. The J1 connector is used for the 32-bit PCI signals, and must be included on all CompactPCI boards. The J2 connector is for 64-bit PCI transfers, and is optional on CompactPCI boards. The keying for the J2 connector is given in IEEE 1101.10. The pin assignments for both connectors are given in the CompactPCI specification in Table 13. The BRSVP*xxx* signals given in the Table must be bussed between connectors. They have been reserved for future definition. The RSV signals given in the Table 13 must be non-bussed between connectors, and they are also reserved for future definition.

CompactPCI cards can operate on either 3.3 V or 5 V signaling voltage, and some boards can be designed to handle both. CompactPCI boards must be keyed with a colored connector that corresponds to its signaling voltage. Cards operating on 3.3 V will use a cadmium yellow connector, and boards operating on 5 V will be keyed with brilliant blue. Boards that can operate with either voltage will have no color key. Additional keying details can be found in *PICMG 2.10, Keying of CompactPCI Boards and Backplanes specification*.

For further details, the complete CompactPCI specification can be obtained from the PCI Industrial Computers Manufacturing Group (PICMG) at www.picmg.org.
The PXI specification expands on the design rules established in the CompactPCI specification, and is available from the PXI Systems Alliance (PXISA) at www.pxisa.org The J1 connector has the exact same signal mapping for both specifications, and all signals necessary for 64-bit operations are included on the J2 connector. The differences in the two specifications lie in the signals on the J2 connector that are used for user-specified purposes in the CompactPCI specification. For PXI, these lines are used for the trigger bus lines, the reference clock, the local bus lines, and the star trigger lines.

The PXI trigger bus consists of eight signals, PXI_TRIG[0:7], which allow for intermodule communication and synchronization. These lines are available for trigger or clock transmission, and several standard triggering protocols are defined to ease interoperability. These protocols are not meant to limit the use of the trigger lines, though, as they can also be used as general purpose intermodule bussed lines, which would use other manufacturer-defined protocols. For instruments that integrate the use of triggers in a PXI system, care must be taken in which trigger lines are used. The PXI trigger lines are not open collectors. This means that multiple modules cannot drive them at the same time. Double-driving a trigger line has the potential to cause damage to the PXI trigger components. For the latest available solutions for trigger implementation, contact National Instruments (www.ni.com) or the PXI Systems Alliance (www.pxisa.org) for the latest specification.

The PXI_CLK10 signal is the common reference clock provided by the PXI backplane for the purpose of synchronizing multiple modules in the PXI system. The signal is a 10 MHz clock that the backplane provides independently to each peripheral slot. The variable frequency of the PXI_CLK10 signal limits its usefulness in synchronization applications, but it can be ideal for qualifying trigger protocols because of its low skew qualities.

The PXI local bus is another P2 implementation unique to the PXI specification. It consists of a bus, 13 lines wide, that is daisy-chained between adjacent modules. The functionality of the local bus is typically user-defined, ranging from passing analog signals from one module to another to high speed data transfer. Any use of the local bus will not degrade the PXI bandwidth, as the 13 lines are separate from the normal data transfer pathways. There are several cases where the use of the local bus is specified, which are applicable to star trigger modules and modules in the rightmost slot of the chassis. The left local bus lines of the star trigger module are used for star trigger implementation. The right local bus of the module used in the rightmost slot in the chassis would be used for chassis-specific applications, such as extending the local bus to another chassis. In both these cases, a local bus does not exist opposite the specified lines, which is why their use must be described. All other local bus lines are undefined, for use as the developer sees fit.

The star trigger bus is another addition to the P2 functionality in the PXI specification. Each module in a PXI system has an equal-length star trigger line running to it, and they originate from the left local bus of the second slot in a chassis. In order for these lines to be utilized, a star trigger module must be used in the second slot of the chassis that can take advantage of the star trigger bus functionality. There are 13 star trigger lines that run from the second slot, which are enough lines to provide signals to two PCI bus segments. Particular uses for the star trigger lines have yet to be defined, and the specification does not require that a star trigger module even be present in a PXI system. Possible star trigger applications could include monitoring a trigger from peripheral slots, triggering multiple modules independently, or routing triggers between slots. One star trigger slot pin is dedicated to allow an external 10 MHz frequency standard to be routed as PXI_CLK10.

For further details, the PXI specification can be downloaded from the PXI Systems Alliance (PXISA) at www.pxisa.org.
There are ways to build a PXI module other than starting from scratch with the CompactPCI and PXI Specifications. Commercially available breadboards are ready to be developed for a PXI system with the requirements of the two specifications already in place. There are several breadboard options available from different manufacturers, and they differ from one another in cost, features, and software compatibility. For this discussion, we studied the features of three of the available breadboards, the PXIT PX2000-107, the Chroma 52901, and the PLDA CPCI10K-PROD. These three boards are not the only available options on the market, but they illustrate some differences in the available breadboards.

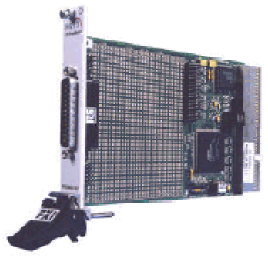
Figure 1. PXIT PX2000-107 (wwwpxit.com)

For basic PXI module development, the PXIT PX2000-107 represents a low-end solution for compliance with both specifications. At $758, the PX2000-107 is the most cost effective of the three boards, and also has the largest prototyping area at 12 in2. Its available I/O connector is the 25-pin D-type. Since it is a basic model, it does not come standard with a mezzanine board. It also has the smallest number of device drivers supported, and no Complex Programmable Logical Device (CPLD). Supported software platforms include NI-VISA, LabVIEW, Visual Basic, and Visual C++.

Figure 2. Chroma 52901 (www.chromaate.com)

The board from Chroma, the 52901, is intermediate in both price and features of the three boards we evaluated. It has a smaller prototyping area than the PX2000-107 at 5 in2, but it does come standard with a mezzanine board that provides an additional 9in2 of development area. It supports the same driver packages as the PX2000-107, and adds IVI and Borland C++ to the list. Additionally, it is compatible with the industrial standard language VHDL (VHSIC Hardware Description Language). Using VHDL, applications for the 52901 can be developed stand alone, without plugging it into a chassis. Chroma provides several examples on this implementation procedure in the 52901 package. The 52901 also has CPLD support, and comes standard with 44-pin D-type connector. The 52901 is priced at $2399.

The third board we looked at is the CPCI10K-PROD, made by PLDA. It is considered more of a high-end model, and comes at a price of $3190. One advantage it has over the other two boards is that it can perform 64-bit and 66MHz operations, while the other two are limited to 32-bit and 33 MHz performance. It also comes standard with on board SRAM, which can be useful for applications requiring large buffers. The motherboard has no prototyping space available for development, but optional mezzanine boards can be used for development area. These mezzanine boards must be purchased at an additional cost. The compatible software list is very limited, as Visual C++ is the only major supported platform. The standard connector is an 80-pin SCSI type I/O connector.

Any of these boards work in a PXI system and can save a tremendous amount of development time. All three comply with both the CompactPCI and PXI specifications, and drivers can be developed with VISA, even if the board does not already have an NI-VISA driver, (iit would just require more driver development time). The appropriate board depends on the end-use of the board, along with the available expertise for implementing the application. The following table allows for a side-by-side comparison of these three models.

Table 1. Breadboard Model Comparison

Product Number	**Chroma 52901**	**PXIT PX20000-107**	**PLDA CPCI10K-PROD**
Dimension	1-Slot, 3U PCI/cPCI	1-Slot, 3U PCI/cPCI	1-Slot, 3U PCI/cPCI
PCI Performance	32Bit/33MHz	32Bit/33MHz	32,64Bit / 33,66MHz (Depends on IP core buy and download)
Prototyping Area	5” inch^2	12” inch^2	0
PXI compliant	PXI 1.0	PXI 1.0	PXI 1.0

Product Number	**Chroma 52901**	**PXIT PX20000-107**	**PLDA CPCI10K-PROD**
Programmable Logical Device Gate Count	▪ Altera ACEX EP1K100 CPLD	▪ No	▪ Altera FLEX 10K100 CPLD ▪ (Same grade of ACEX1K100)
I/O Connector	▪ 44-pin D-Type	▪ 25-pin D-Type	▪ 80-pin SCSI type
Mezzanine Connector	▪ One 40-pin ▪ ANSI/VITA 12-1996 M-Module compatible connector	▪ No	▪ Four ▪ IEEE 1386 ▪ PMC connectors
Device Driver	▪ IVI ▪ NI-VISA ▪ LabVIEW ▪ VisualBasic ▪ VC/C++ ▪ BC/C++ ▪ For Windows 95/98/ME/NT/2000	▪ NI-VISA ▪ LabVIEW ▪ VisualBasic ▪ VisualC/C++	▪ Jungo’s Windriver driver ▪ Supply VC++ source files for ▪ PCI Manager ▪ Configuration Manager ▪ Memory Manager ▪ Interrupt Manager ▪ DMA Manager ▪ PLDA Windriver API sources & Windriver includes files ▪ for Windows 9x/NT/2000
Mezzanine Board Prototype Area	▪ 9” inch^2	▪ No	▪ ~5” inch^2 ▪ (Optional)
Others	▪ Example LED example daughter board and example program to learn Chroma 52901 ▪ 3 VHDL examples for training purpose		▪ CPLD has 10K30, 10K50, 10K100 and 10K200 options ▪ On board SRAM 512Kx48 bits, 512Kx64 bits and 512Kx128 bits
List Price	US $2,399	US $758	US $3,190

The PXI specification also requires that all PXI devices must have drivers available that are compatible with Windows operating systems. A very efficient way to develop a driver for your new PXI instrument is to use NI-VISA, the software layer that acts as a device driver for PXI devices, along with VXI, serial, and GPIB. NI-VISA can help cut down on programming time significantly because it eliminates the need to understand low-level Windows programming. Programs written in NI-VISA are also compatible across all Windows platforms and with LabVIEW Real-Time, eliminating the necessary programming for cross-platform compatibility. NI-VISA even comes with an interactive utility called the PXI Driver Development Wizard to assist in setting up the Windows setup (.inf) file for your instrument, and configuring how interrupts from the PXI device are handled.

The steps for configuring NI-VISA to recognize your PXI device are made relatively easy by following the PXI Driver Development Wizard. Using the Wizard, you can set up a .inf file, which will contain the necessary information for Windows to recognize NI-VISA as the driver for your PXI device. The first thing the PXI Driver Development Wizard will need is the PXI Manufacturer ID and PXI Model Code. Next, select whether or not NI-VISA will handle PCI interrupts for this device (this step is optional). Both interrupt detection handling and interrupt removal handling will need to be configured in this step. The Wizard will finish up by generating the necessary output files based on the information entered.

To verify this installation, power down the system and install the card. The card should be properly recognized as Windows boots up. You may also want to check in Device Manager, if available, to see that the device has been successfully recognized without issue. You may need to point Windows to the .inf file created by the Wizard. If this all looks fine, launch Measurement & Automation Explorer (MAX) and press F5 to refresh the instruments displayed. Your instrument should now be listed under Devices and Interfaces. At this point you should be able to see the VISA resource string that will apply to your instrument. The resource string is based on the current PCI logical address for the device. This string will be used for opening sessions to the instrument in LabVIEW and LabWindows. You also can find the resource string by launching VISA Interactive Control (VISAIC)which is also used to test I/O operations before implementing them in LabVIEW or LabWindows.

You should now be ready to write a program for your instrument using either LabVIEW or LabWindows. VISA programs constructed with either software package will follow the same basic structure, having an Open section, an I/O section, and a Close section. The Open section will consist of opening a VISA session to the Default Resource Manager (RM) and then a session to the PXI instrument. Both of these sessions must be opened in LabWindows, while in LabVIEW both are handled by using the VISA Open VI. The new PXI instrument is specified for the Open functions by using the VISA resource string, found in VISAIC or MAX, that corresponds to the instrument in the function. An example of a VISA resource string is PXI0::4::INSTR. The chassis number and slot number can be used in the VISA resource string to differentiate between cards as well. In order for slot identification to work properly, though, the proper pxisys.ini file must be present in the system. Slot numbers can be used with the VISA Find Resource function as one of the criteria to search for.

Now, with a session open to the new instrument, I/O functions can be performed. This can be done by several communication methods: register-based commands, attributes, and event handling. Register-based I/O is used to access the configuration registers and Base Address Registers (BAR) on the PXI device. Register accesses are performed using VISA In and VISA Out functions. You also can control the instrument through attributes, using Get Attribute and Set Attribute functions. The formats of all these functions are similar using either LabVIEW or LabWindows, with similar parameters needed for the functions to work properly.

Performing event handling I/O can be a bit trickier, and the handling options will differ slightly depending on if you are using LabVIEW or LabWindows. Event handling is used to react appropriately to PCI interrupts produced by the instrument. The first method for handling events is through using a queue to store interrupts as they occur, and then to monitor the queue periodically and react to the interrupts. Both LabVIEW and LabWindows can use queues to handle events. The other method for event handling is through using callbacks, which can only be done in LabWindows. A callback function is never called by the main program, but instead by the VISA driver when a particular event occurs since the application does not need to check a queue to determine if an interrupt has occurred. This allows for a different mechanism to handle interrupts than the queuing method, but either can be used depending on the organization of the code. There are several important functions in LabVIEW and LabWindows for setting up event handling. They include functions to enable event handling, to instruct the program to wait on the event, and to disable event handling.

The proper way to wrap up a VISA program is to use a VISA Close function to close the session to the PXI instrument. If you are programming in LabWindows, a Close Default RM function is necessary as well. It is important to close these VISA sessions, as open sessions will use up resources on your computer.

The following chart shows some common VISA functions, both in LabWindows and LabVIEW formats. For more information, please see the *NI-VISA Programmer Reference Manual*.

LabWindows	**LabVIEW**
ViOpen	VISA Open
ViClose	VISA Close
ViWrite	VISA Write
ViRead	VISA Read
viIn8	VISA In 8
viOut8	VISA Out 8

You may find that you want to use your new PXI instrument with more than one particular computer, and to do this, it will be necessary to distribute the .inf file created by the PXI Driver Development Wizard. This distribution will require multiple VISA licenses, which can be purchased by contacting National Instruments. A convenient way to distribute the necessary setup files for your new device is to use the installer package that can be created using LabWindows/CVI. It can be used to distribute VXI*plug&play* or IVI instrument drivers. The CVI Win32 installation package creation is launched from the **Build»Create Distribution Package** menu option. The installer generates several files, including a setup.exe file. All these files must be included to a target machine for proper installation. If the PXI device was installed before the driver was installed, it may be necessary to remove it from Device Manager, where it will probably be listed as an Unknown Device. After the removal, scan for new hardware changes and the device should be detected properly now that the driver has been installed.

For more detailed information on using NI-VISA to set up a driver for your new PXI device, please see the National Instruments white paper Using the VISA Driver Development Wizard and NI-VISA to Develop a PXI/PCI Driver in Windows. Additional information about VISA, as well as PXI interfaces to VISA, is available in the NI-VISA User Manual.
For developers who want to create their own PXI modules, there are several possible starting points, as we have discussed. For those who want to start from scratch, the CompactPCI and PXI specifications can provide proper guidance in designing a board that complies with both. Some developers may wish to delve into their particular design instead of wading through the details of the specification requirements, so they should look at the PXI-ready breadboards that are commercially available. We looked over three boards, from PXIT, Chroma, and PLDA, which offered a look at different options and price ranges available. Whichever route a developer goes in designing their board, they will also need to design software that will allow their operating system to utilize their board. We have recommended NI-VISA as our driver development environment of choice, and we have detailed the various features of NI-VISA. Armed with this information, a developer can make better-informed choices when undertaking the design of a PXI board.