PXI

NI-Sync User Manual

NATIONAL INSTRUMENTS™

November 2003 Edition
Part Number 370926A-01

Worldwide Technical Support and Product Information

ni.com

National Instruments Corporate Headquarters

11500 North Mopac Expressway   Austin, Texas 78759-3504   USA   Tel: 512 683 0100

Worldwide Offices

Australia 1800 300 800, Austria 43 0 662 45 79 90 0, Belgium 32 0 2 757 00 20, Brazil 55 11 3262 3599, Canada (Calgary) 403 274 9391, Canada (Ottawa) 613 233 5949, Canada (Québec) 450 510 3055, Canada (Toronto) 905 785 0085, Canada (Vancouver) 514 685 7530, China 86 21 6555 7838, Czech Republic 420 224 235 774, Denmark 45 45 76 26 00, Finland 385 0 9 725 725 11, France 33 0 1 48 14 24 24, Germany 49 0 89 741 31 30, Greece 30 2 10 42 96 427, India 91 80 51190000, Israel 972 0 3 6393737, Italy 39 02 413091, Japan 81 3 5472 2970, Korea 82 02 3451 3400, Malaysia 603 9131 0918, Mexico 001 800 010 0793, Netherlands 31 0 348 433 466, New Zealand 0800 553 322, Norway 47 0 66 90 76 60, Poland 48 22 3390150, Portugal 351 210 311 210, Russia 7 095 783 68 51, Singapore 65 6226 5886, Slovenia 386 3 425 4200, South Africa 27 0 11 805 8197, Spain 34 91 640 0085, Sweden 46 0 8 587 895 00, Switzerland 41 56 200 51 51, Taiwan 886 2 2528 7227, Thailand 662 992 7519, United Kingdom 44 0 1635 523545

For further support information, refer to the *Technical Support and Professional Services* appendix. To comment on the documentation, send email to techpubs@ni.com.

Important Information

Warranty

The media on which you receive National Instruments software are warranted not to fail to execute programming instructions, due to defects in materials and workmanship, for a period of 90 days from date of shipment, as evidenced by receipts or other documentation. National Instruments will, at its option, repair or replace software media that do not execute programming instructions if National Instruments receives notice of such defects during the warranty period. National Instruments does not warrant that the operation of the software shall be uninterrupted or error free.

A Return Material Authorization (RMA) number must be obtained from the factory and clearly marked on the outside of the package before any equipment will be accepted for warranty work. National Instruments will pay the shipping costs of returning to the owner parts which are covered by warranty.

National Instruments believes that the information in this document is accurate. The document has been carefully reviewed for technical accuracy. In the event that technical or typographical errors exist, National Instruments reserves the right to make changes to subsequent editions of this document without prior notice to holders of this edition. The reader should consult National Instruments if errors are suspected. In no event shall National Instruments be liable for any damages arising out of or related to this document or the information contained in it.

EXCEPT AS SPECIFIED HEREIN, NATIONAL INSTRUMENTS MAKES NO WARRANTIES, EXPRESS OR IMPLIED, AND SPECIFICALLY DISCLAIMS ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. CUSTOMER'S RIGHT TO RECOVER DAMAGES CAUSED BY FAULT OR NEGLIGENCE ON THE PART OF NATIONAL INSTRUMENTS SHALL BE LIMITED TO THE AMOUNT THERETOFORE PAID BY THE CUSTOMER. NATIONAL INSTRUMENTS WILL NOT BE LIABLE FOR DAMAGES RESULTING FROM LOSS OF DATA, PROFITS, USE OF PRODUCTS, OR INCIDENTAL OR CONSEQUENTIAL DAMAGES, EVEN IF ADVISED OF THE POSSIBILITY THEREOF. This limitation of the liability of National Instruments will apply regardless of the form of action, whether in contract or tort, including negligence. Any action against National Instruments must be brought within one year after the cause of action accrues. National Instruments shall not be liable for any delay in performance due to causes beyond its reasonable control. The warranty provided herein does not cover damages, defects, malfunctions, or service failures caused by owner's failure to follow the National Instruments installation, operation, or maintenance instructions; owner's modification of the product; owner's abuse, misuse, or negligent acts; and power failure or surges, fire, flood, accident, actions of third parties, or other events outside reasonable control.

Copyright

Trademarks

CVI™, IVI™, LabVIEW™, MXI™, National Instruments™, NI™, ni.com™, NI-DAQ™, NI-FGEN™, NI-Motion™, NI-VISA™, NI-VXI™, and RTSI™ are trademarks of National Instruments Corporation.

Product and company names mentioned herein are trademarks or trade names of their respective companies.

Patents

For patents covering National Instruments products, refer to the appropriate location: **Help»Patents** in your software, the `patents.txt` file on your CD, or `ni.com/patents`.

WARNING REGARDING USE OF NATIONAL INSTRUMENTS PRODUCTS

(1) NATIONAL INSTRUMENTS PRODUCTS ARE NOT DESIGNED WITH COMPONENTS AND TESTING FOR A LEVEL OF RELIABILITY SUITABLE FOR USE IN OR IN CONNECTION WITH SURGICAL IMPLANTS OR AS CRITICAL COMPONENTS IN ANY LIFE SUPPORT SYSTEMS WHOSE FAILURE TO PERFORM CAN REASONABLY BE EXPECTED TO CAUSE SIGNIFICANT INJURY TO A HUMAN.

(2) IN ANY APPLICATION, INCLUDING THE ABOVE, RELIABILITY OF OPERATION OF THE SOFTWARE PRODUCTS CAN BE IMPAIRED BY ADVERSE FACTORS, INCLUDING BUT NOT LIMITED TO FLUCTUATIONS IN ELECTRICAL POWER SUPPLY, COMPUTER HARDWARE MALFUNCTIONS, COMPUTER OPERATING SYSTEM SOFTWARE FITNESS, FITNESS OF COMPILERS AND DEVELOPMENT SOFTWARE USED TO DEVELOP AN APPLICATION, INSTALLATION ERRORS, SOFTWARE AND HARDWARE COMPATIBILITY PROBLEMS, MALFUNCTIONS OR FAILURES OF ELECTRONIC MONITORING OR CONTROL DEVICES, TRANSIENT FAILURES OF ELECTRONIC SYSTEMS (HARDWARE AND/OR SOFTWARE), UNANTICIPATED USES OR MISUSES, OR ERRORS ON THE PART OF THE USER OR APPLICATIONS DESIGNER (ADVERSE FACTORS SUCH AS THESE ARE HEREAFTER COLLECTIVELY TERMED "SYSTEM FAILURES"). ANY APPLICATION WHERE A SYSTEM FAILURE WOULD CREATE A RISK OF HARM TO PROPERTY OR PERSONS (INCLUDING THE RISK OF BODILY INJURY AND DEATH) SHOULD NOT BE RELIANT SOLELY UPON ONE FORM OF ELECTRONIC SYSTEM DUE TO THE RISK OF SYSTEM FAILURE. TO AVOID DAMAGE, INJURY, OR DEATH, THE USER OR APPLICATION DESIGNER MUST TAKE REASONABLY PRUDENT STEPS TO PROTECT AGAINST SYSTEM FAILURES, INCLUDING BUT NOT LIMITED TO BACK-UP OR SHUT DOWN MECHANISMS. BECAUSE EACH END-USER SYSTEM IS CUSTOMIZED AND DIFFERS FROM NATIONAL INSTRUMENTS' TESTING PLATFORMS AND BECAUSE A USER OR APPLICATION DESIGNER MAY USE NATIONAL INSTRUMENTS PRODUCTS IN COMBINATION WITH OTHER PRODUCTS IN A MANNER NOT EVALUATED OR CONTEMPLATED BY NATIONAL INSTRUMENTS, THE USER OR APPLICATION DESIGNER IS ULTIMATELY RESPONSIBLE FOR VERIFYING AND VALIDATING THE SUITABILITY OF NATIONAL INSTRUMENTS PRODUCTS WHENEVER NATIONAL INSTRUMENTS PRODUCTS ARE INCORPORATED IN A SYSTEM OR APPLICATION, INCLUDING, WITHOUT LIMITATION, THE APPROPRIATE DESIGN, PROCESS AND SAFETY LEVEL OF SUCH SYSTEM OR APPLICATION.

Contents

About This Manual

Chapter 1
Introduction, Installation, and Configuration

Chapter 2
Building and Programming Applications

Chapter 3 Synchronizing Multiple NI PXI-4472 Modules

Chapter 4
Synchronizing Multiple NI PXI-5112 Modules

Chapter 5
Synchronizing Multiple NI PXI-6115 Modules

Chapter 6
Synchronizing Multiple NI MIO Modules

Chapter 7
Synchronizing Multiple NI PXI-5411 Modules

Appendix A
Technical Support and Professional Services

Glossary

Index

About This Manual

The *NI-Sync User Manual* is for users of the NI-Sync driver software, an application programming interface (API) for controlling NI PXI-665*x* timing modules. This manual describes the fundamentals of developing applications with NI-Sync. In addition, this manual includes examples for using NI-Sync with specific measurement hardware.

Conventions

The following conventions appear in this manual:

<> Angle brackets that contain numbers separated by an ellipsis represent a range of values associated with a bit or signal name—for example, DIO<3..0>.

» The » symbol leads you through nested menu items and dialog box options to a final action. The sequence **File»Page Setup»Options** directs you to pull down the **File** menu, select the **Page Setup** item, and select **Options** from the last dialog box.

♦ The ♦ symbol indicates that the following text applies only to a specific product, a specific operating system, or a specific software version.

✎ This icon denotes a note, which alerts you to important information.

⚠ This icon denotes a caution, which advises you of precautions to take to avoid injury, data loss, or a system crash.

bold Bold text denotes items that you must select or click in the software, such as menu items and dialog box options. Bold text also denotes parameter names.

italic Italic text denotes variables, emphasis, a cross reference, or an introduction to a key concept. This font also denotes text that is a placeholder for a word or value that you must supply.

`monospace` Text in this font denotes text or characters that you should enter from the keyboard, sections of code, programming examples, and syntax examples. This font is also used for the proper names of disk drives, paths, directories, programs, subprograms, subroutines, device names, functions, operations, variables, filenames and extensions, and code excerpts.

Related Documentation

The following documents contain information that you might find helpful as you read this manual:

- *PICMG 2.0 R3.0*, *CompactPCI Core Specification*, available from PICMG, available from `www.picmg.org`
- *PXI Specification*, Revision 2.1, available from `www.pxisa.org`
- *NI PXI-665x User Manual*, available from `ni.com/manuals`
- *Getting Started with Multi-Chassis Synchronization Using the NI PXI-665x*, available from `ni.com/manuals`

1

Introduction, Installation, and Configuration

This chapter provides an overview of the NI-Sync driver software and explains how to install and configure NI-Sync for use with NI PXI-665*x* timing modules.

About the NI-Sync Driver Software

NI-Sync is a library of VIs and functions for controlling NI PXI-665*x* timing modules. Using NI-Sync, you can configure all aspects of timing and synchronization for NI PXI-665*x* devices, including sharing trigger signals and clocks in one or more chassis. You can use NI-Sync in conjunction with other measurements software, such as NI-DAQmx or Traditional NI-DAQ, to create advanced, high-channel-count measurements that span multiple PXI chassis.

Introduction

The NI-Sync driver software includes the following:

- NI-Sync instrument driver API and device driver
- Hardware-specific example software, illustrating multichassis synchronization for the following measurement devices:
  - NI PXI-4472 modules
  - NI PXI-5112 modules
  - NI PXI-6115/6120 modules
  - NI PXI-5411 modules

When developing your application, refer to Chapter 2, *Building and Programming Applications*, for information about creating an application with your specific application development environment (ADE). Also, refer to the appropriate hardware-specific chapter in this manual for specific examples of using NI-Sync with your application.

Operating System Support

NI-Sync supports Windows 2000/XP and LabVIEW RT.

Application Software and Programming Language Support

Table 1-1 lists the application software versions that NI-Sync supports. If you are not using National Instruments application software, refer to Table 1-2.

Table 1-1. National Instruments Application Software Support

NI Application Software	Versions NI-Sync Supports
LabVIEW	6.1 or later
LabVIEW RT Module	7.0 or later
LabWindows™/CVI™	5.5 or later

Table 1-2 lists additional programming languages supported by NI-Sync.

Table 1-2. Additional Programming Language Support

Programming Language	Versions NI-Sync Supports
ANSI C	✓
Microsoft Visual C++	5.0 or later

Device Support

NI-Sync supports the NI PXI-665*x* family of timing modules.

Installing the Software

The software package that ships with the NI PXI-665*x* provides the following items:

- NI-Sync driver software
- LabVIEW example code
- LabWindows/CVI example code
- *NI PXI-665x User Manual*

Complete the following steps to install your NI-Sync software:

1. Insert the NI-Sync CD into the CD-ROM drive of your computer.
2. Run the `Setup.exe` program to install the NI-Sync software on your system.

Several high-level examples are provided to give you a starting point in using the NI PXI-665*x* to synchronize data acquisition across multiple PXI chassis with various National Instruments PXI modules.

Note Be sure to install the NI-Sync software *before* installing the NI PXI-665*x* hardware.

Device and System Configuration

Before you begin using your NI PXI-665*x* timing module, you must ensure that your PXI system software is configured properly. NI-Sync uses PXI configuration information to enable features such as chassis identification, slot identification, and trigger terminal reservation. This configuration information is enabled by identifying PXI system components in Measurement & Automation Explorer.

Using Measurement & Automation Explorer

Measurement & Automation Explorer (MAX) is a Windows-based application that you use to configure and view National Instruments device settings under Windows operating systems.

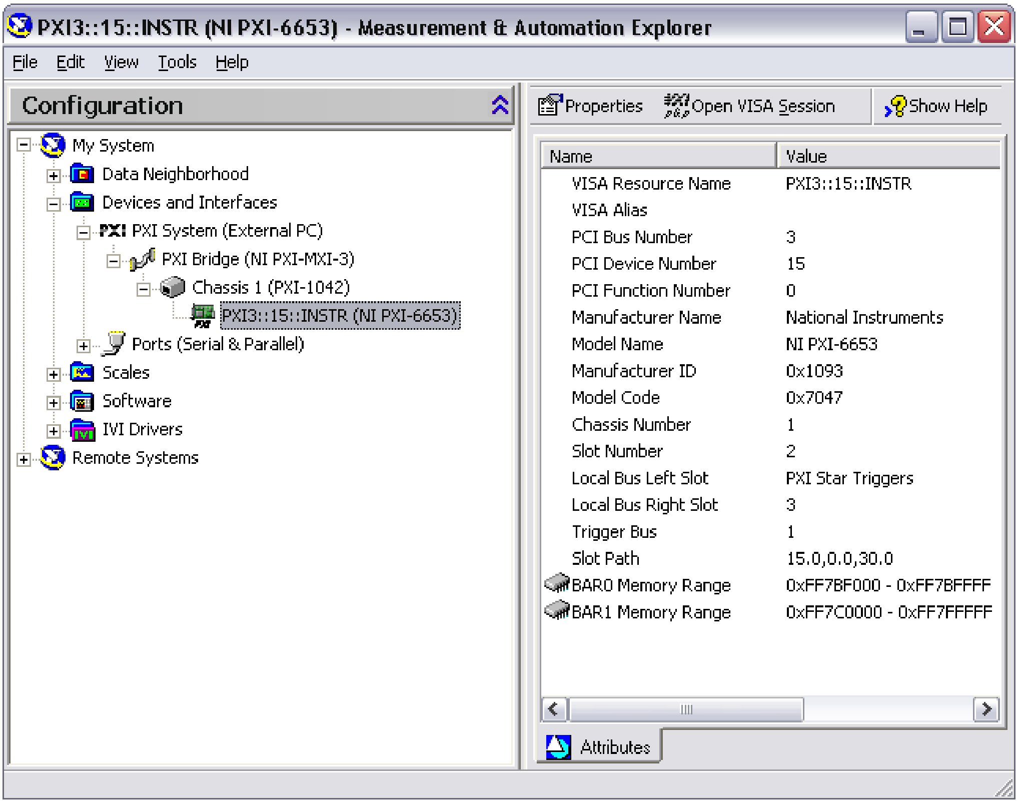
Identifying Your PXI System

Double-click the **Measurement & Automation Explorer** icon on your desktop to run MAX. In the Configuration pane, select **My System»Devices and Interfaces»PXI System**. Using the PXI System view, you can identify the controller and chassis in your PXI system. Refer to the PXI System context help for detailed instructions on identifying your PXI system components.

Locating Your NI PXI-665*x* Modules

Once you have identified your PXI system components, you can locate your NI PXI-665*x* modules by browsing the PXI System view (**My System»Devices and Interfaces»PXI System**). Instances of the NI PXI-665*x* timing module are displayed under their corresponding chassis. By selecting an instance of the NI PXI-665*x*, you can view its attributes, including VISA Resource Name and PXI slot number. Refer to Figure 1-1 for an example of the type of device information available in MAX.

Note Viewing NI PXI-665*x* device information in MAX is useful for obtaining the VISA Resource Name for an instance of the NI PXI-665*x*. The VISA Resource Name is used to create a session to a device using the NI-Sync API. Refer to Chapter 2, *Building and Programming Applications*, for detailed information about device initialization.



Name	Value
VISA Resource Name	PXI3::15::INSTR
VISA Alias	
PCI Bus Number	3
PCI Device Number	15
PCI Function Number	0
Manufacturer Name	National Instruments
Model Name	NI PXI-6653
Manufacturer ID	0x1093
Model Code	0x7047
Chassis Number	1
Slot Number	2
Local Bus Left Slot	PXI Star Triggers
Local Bus Right Slot	3
Trigger Bus	1
Slot Path	15.0,0.0,30.0
BAR0 Memory Range	0xFF7BF000 - 0xFF7BFFFF
BAR1 Memory Range	0xFF7C0000 - 0xFF7FFFFF

Figure 1-1. NI PXI-665*x* Device Information in MAX

2

Building and Programming Applications

This chapter describes the fundamentals of building and programming NI-Sync applications for LabVIEW, LabWindows/CVI, and Microsoft Visual C++.

The NI-Sync Instrument Driver

The NI-Sync driver software includes an instrument driver API for configuring attributes and programming the features of NI PXI-665*x* timing modules. The NI-Sync instrument driver function library is a C DLL. This DLL should be linked using the appropriate import library for your application development environment.

The following sections provide guidelines for creating applications that use the NI-Sync driver software.

Note If you are not using one of the tools listed, refer to your development tool reference manual for details on creating applications that call C DLLs.

Creating a Windows Application Using LabVIEW

This section assumes that you are using LabVIEW 6.1 or later to manage your code development and that you are familiar with the LabVIEW environment basics.

Developing an NI-Sync Application

To develop an NI-Sync application with LabVIEW, complete the following steps:

1. Open an existing or new LabVIEW VI.
2. From the Function Palette, locate the NI-Sync VIs at **Instrument I/O»Instrument Drivers»NI-Sync**.
3. Select the VIs you want to use and drop them on the block diagram to build your application.

Example Programs

You can find LabVIEW example programs from the Windows **Start** menu at **Programs»National Instruments»NI-Sync»Examples**. The examples are organized by LabVIEW version number and measurement hardware.

Creating a Windows Application Using LabWindows/CVI

This section assumes that you are using LabWindows/CVI 5.5 or later to manage your code development and that you are familiar with the LabWindows/CVI environment.

Developing an NI-Sync Application

To develop an NI-Sync application with LabWindows/CVI, complete the following steps:

1. Open an existing or new project file.
2. Load the NI-Sync function panel at `\VXIpnp\winnt\niSync`.
3. Use the function panel to navigate the function hierarchy and generate function calls with the proper syntax and variable values.

Example Programs

You can find LabWindows/CVI example programs from the Windows **Start** menu at **Start»Programs»National Instruments» NI-Sync»Examples»CVI Examples**. The examples are organized by measurement hardware.

Creating a Windows Application Using Microsoft Visual C++

This section assumes that you are using the Microsoft Visual C++ (MSVC) ADE to manage your code development and that you are familiar with the MSVC environment.

Developing an NI-Sync Application

To develop an NI-Sync application with MSVC, complete the following steps:

1. Open an existing or new Visual C++ project to manage your application code.
2. Create files of type `.c` (C source code) or `.cpp` (C++ source code) and add them to the project. Make sure to include the NI-Sync header file in each source file:

   ```
   #include "niSync.h"
   ```

3. Specify the directory that contains the NI-Sync header file under the **Preprocessor»Additional include directories** settings in your compiler. For MSVC 5.0/6.0, this setting is found under **Project» Settings»C/C++**. The NI-Sync header file is in the `\VXIpnp\winnt\include` directory.
4. Add the NI-Sync import library `niSync.lib` to the project under the **Link»General»Object/Library Modules** setting. The NI-Sync import library is in the `\VXIpnp\winnt\lib\msc` folder.
5. Add NI-Sync function calls to your application.
6. Build your application.

Example Programs

You can find C-based example programs from the Windows **Start** menu at **Start»Programs»National Instruments»NI-Sync»Examples»CVI Examples**. The examples are organized by measurement hardware.

Note While the C-based examples are written specifically for use with LabWindows/CVI, they can be adapted for use with MSVC and other C/C++ compilers.

Special Considerations

When developing applications with MSVC, observe the following special considerations.

- **String Passing**—To pass strings as arguments to functions, pass a pointer to the first element of the character array. Be sure the string is null terminated.

NI-Sync Programming Flow

Figure 2-1 shows the basic programming flow of typical NI-Sync applications. NI-Sync VIs and functions are organized under the Initialize, Configure Hardware, Connect Terminals, Disconnect Terminals, and Close categories to assist you in understanding where you should call a function or VI in your applications. Functions and VIs that do not fall into the programming flow categories are considered Advanced or Utility functions that perform various tasks such as resetting the NI PXI-665*x* module, returning the revision number of the NI-Sync instrument driver and instrument firmware, and other functions.

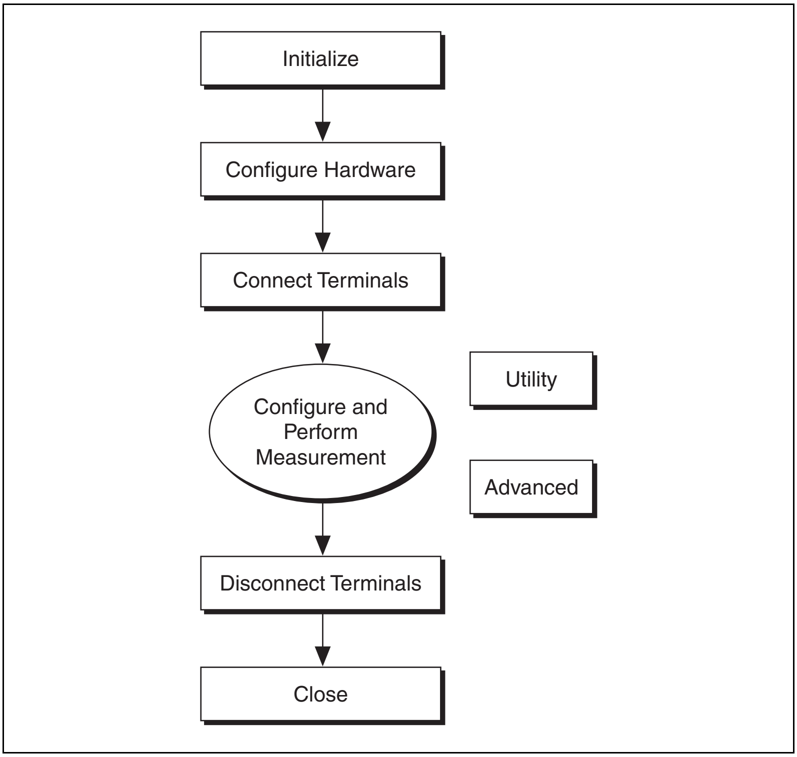


Figure 2-1. Basic Programming Flow of an NI-Sync Application

Initialize

For any application you write, you must first open a session to establish communication with the NI PXI-665*x* timing module using the **Initialize** VI or function.

LabVIEW VI	C Function
niSync Initialize	`niSync_init`

In addition to establishing a session with the timing module, **niSync Initialize** can reset the device to a known state and verify that the NI-Sync instrument driver is valid for a particular instrument. The **Initialize** VI or function returns a ViSession handle you can use to identify the instrument in all subsequent NI-Sync calls.

Note The **Initialize** VI and function take the VISA Resource Name corresponding to an NI PXI-665*x* module and use this information to locate the instrument and create a session to it. You can obtain the VISA Resource Name for an instance of the NI PXI-665*x* using MAX. Refer to Chapter 1, *Introduction, Installation, and Configuration*, for an example of using MAX with the NI PXI-665*x*.

The **Initialize** VI and function create a new instrument session. Repeated calls to **Initialize** for the same resource name will return the same session. You can use this session in multiple program threads.

Configure Hardware

Use Configuration VIs and functions to adjust settings of the timing and synchronization features of the NI PXI-665*x* module, including DAC input threshold voltage levels, DDS frequency, synchronization clock sources, and other settings and features needed for timing operations.

NI PXI-665*x* attributes are configured using a LabVIEW property node or the `niSync_SetAttribute` and `niSync_GetAttribute` functions.

Accessing Attributes

In LabVIEW, you can find NI PXI-665*x* attributes in the **niSync** property node. To access these attributes, complete the following steps:

1. Open a VI.
2. Make sure you are viewing the block diagram. Navigate the **niSync** palette at **Instrument I/O»Instrument Drivers»NI-Sync** and drag the property node to the diagram.

3. Left-click the property node and select the attribute you want to use.
4. To configure additional attributes, resize the property node.

In C, attributes are accessed with the `niSync_SetAttribute`… and `niSync_GetAttribute`… functions. These functions correspond to a particular data type. For example, to set the PFI0 DAC voltage level (type ViReal64), use `niSync_SetAttributeViReal64`.

Refer to the *NI-Sync API Reference* for a complete list of attributes.

Connect Terminals

After you have configured the NI PXI-665*x* timing module, you can route signals between terminals using the Connect Terminals functions. Connecting terminals forms the core of typical NI-Sync applications. Source and destination terminals can be connected using a variety of mechanisms. NI-Sync considers three types of terminals—clock terminals, trigger terminals, and software trigger terminals.

Clock Terminals

Clock terminals include terminals associated with the 10 MHz PXI reference clock (PXI_Clk10). Clock terminal connections are used to route clock signals between the backplane and front panel of the NI PXI-665*x* module. Refer to the *NI PXI-665x User Manual* for a complete discussion of clock terminals.

Clock terminal connections have a variety of uses, including:

- Multichassis PXI_Clk10 synchronization
- PXI_Clk10 replacement with a high-precision onboard oscillator

Clock terminal connections are characterized by source and destination terminals.

The following VI and function deal with clock terminal connections.

LabVIEW VI	C Function
niSync Connect Clock Terminals	`niSync_ConnectClkTerminals`

Trigger Terminals

Trigger terminals include terminals associated with hardware trigger lines. Trigger terminals can also carry clocks, but they are not associated with any specific clock signal. Trigger terminals include the PXI trigger lines (PXI_Trig[0:7]), the PXI star triggers (PXI_Star[0:12]), and the NI PXI-665*x* front panel PFI lines (PFI[0:5]). Refer to the *NI PXI-665x User Manual* for a complete discussion of trigger terminals.

You can use trigger terminals to route single digital pulses between chassis. In addition, trigger terminals can carry and distribute clock signals. Typical uses of trigger terminals include:

- Sharing a trigger signal to start data acquisition between multiple chassis
- Sharing a "sync pulse" to align common clocks on multiple chassis
- Distributing high-speed clock signals (typically on PXI_Star terminals)

Note Some trigger types are reservable. That is, they can be connected to only a single source at a time. For example, PXI_Trig terminals use TTL drivers and should not be driven by multiple signal sources. To solve this problem, NI driver software supports reservation of PXI_Trig terminals so that only one source is active on a destination at any given time. This reservation software integrates with other NI measurements software to prevent multiple sources for a single PXI_Trig terminal.

Trigger terminal connections are characterized by a source terminal, destination terminal, and synchronization clock. In addition, trigger terminal signals can be inverted or synchronized to the rising or falling edge of the specified synchronization clock. Trigger signals can also be routed asynchronously.

The following VI and function deal with trigger terminal connections.

LabVIEW VI	C Function
niSync Connect Trigger Terminals	`niSync_ConnectTrigTerminals`

Software Trigger Terminals

Software trigger terminals include those terminals associated with software-initiated trigger pulses. Currently, the NI PXI-665*x* timing module includes a single software trigger terminal, referred to as the Global Software Trigger. The Global Software Trigger terminal can be connected to any other trigger terminal (PXI_Trig, PXI_Star, and PFI). When the software trigger terminal is connected, a pulse can be sent to all connected destination terminals simultaneously.

Typical uses of the Global Software Trigger include:

- Generating a trigger signal to start data acquisition
- Generating a "sync pulse" to align common clocks on multiple chassis
- Resetting clocks (or divided clocks) to synchronize clock generation across multiple NI PXI-665*x* modules

Software trigger terminal connections are characterized by a source terminal (always set to the Global Software Trigger), a destination terminal (any valid trigger terminal destination), and a synchronization clock. In addition, the software trigger signal can be inverted, synchronized to the rising or falling edge of the specified synchronization clock, or delayed by an integer multiple of the synchronization clock period. Refer to the *NI PXI-665x User Manual* for a complete discussion of the Global Software Trigger.

The following VIs and functions deal with software trigger terminal connections.

LabVIEW VIs	C Functions
niSync Connect Software Trigger	`niSync_ConnectSWTrigToTerminal`
niSync Send Software Trigger	`niSync_SendSoftwareTrigger`

Configure and Perform Measurement

After making terminal connections, you are ready to perform your measurement. Taking a measurement is an application-specific operation that typically involves the use of a Measurements API such as NI-DAQmx, Traditional NI-DAQ, NI-Scope, NI-FGEN, and so on. If you are synchronizing measurements using multiple NI PXI-4472, NI PXI-5112, NI PXI-6115/6120, or NI PXI-5411 modules, refer to the appropriate chapter of this manual for detailed examples of using your measurement device with NI-Sync.

Note If you are not using measurement devices discussed in this manual, refer to the reference manuals for your measurements hardware for specific instructions on configuring and performing a measurement.

Disconnect Terminals

Once a measurement has been performed, connected terminals should be disconnected. This returns the PXI system to its premeasurement state and avoids disrupting the activity of other timing and synchronization applications. Terminals are disconnected by supplying the connected source and destination terminals to Disconnect VIs or functions.

Terminals are disconnected according to their type.

Clock Terminals

Use the following VI or function to disconnect clock terminals.

LabVIEW VI	C Function
niSync Disconnect Clock Terminals	`niSync_DisconnectClkTerminals`

Trigger Terminals

Use the following VI or function to disconnect trigger terminals.

LabVIEW VI	C Function
niSync Disconnect Trigger Terminals	`niSync_DisconnectTrigTerminals`

Software Trigger Terminals

Use the following VI or function to disconnect software trigger terminals.

LabVIEW VI	C Function
niSync Disconnect Software Trigger	`niSync_DisconnectSWTrigFromTerminal`

Note A special terminal value exists for disconnecting multiple terminals from a source or destination terminal. Use the AllConnected terminal (`NISYNC_VAL_ALL_CONNECTED`) to disconnect multiple sources or destinations. If this value is supplied as the source and destination terminal, all connections of the specified terminal type are disconnected.

Note In addition to the explicit disconnect VIs and functions, you can use **niSync Reset** to disconnect all connected terminals.

Close

When your program finishes, terminate the session with the **Close** VI or function.

LabVIEW VI	C Function
niSync Close	`niSync_close`

The **Close** VI or function is essential for deallocating memory and freeing other operating system resources. Every session you initialize must be closed, even if an error occurs during program execution.

While debugging your application, it is possible to abort the application without calling **Close**. While aborting execution should not cause problems, it is not recommended for terminating your application.

Note Calling **Close** will *not* disconnect terminals that were connected while a session is open. Terminals must be explicitly disconnected using Disconnect VIs/functions or by resetting the module.

Utility

In addition to resource and terminal connection management, NI-Sync includes several Utility VIs and functions for performing tasks such as resetting the NI PXI-665*x*, converting error codes to messages, and obtaining information about existing terminal connections.

Terminal Connection Information

NI-Sync includes VIs and functions for obtaining information about terminal connections.

LabVIEW VIs	C Functions
niSync Get Clock Connection Info	`niSync_GetClkTerminalConnectionInfo`
niSync Get Trigger Connection Info	`niSync_GetTrigTerminalConnectionInfo`
niSync Get Software Trigger Connection Info	`niSync_GetSWTrigConnectionInfo`

Using these VIs and functions, you can determine if a given destination terminal is connected. You also can determine the connection parameters, such as synchronization clock, clock edge, and signal inversion.

Instrument Driver Utility Functions

In addition to terminal connection information, NI-Sync supports the standard set of instrument driver utility functions.

LabVIEW VIs	C Functions
niSync Reset	`niSync_reset`
niSync Self-Test	`niSync_self_test`
niSync Revision Query	`niSync_revision_query`
niSync Error Message	`niSync_error_message`

Refer to the *NI-Sync API Reference* for details regarding these functions.

Advanced

NI-Sync also includes advanced features, including frequency measurement and FPGA reconfiguration.

Frequency Measurement

NI-Sync can measure the frequency of a signal at the PFI0 terminal of an NI PXI-665*x* module. The following VI and function support this operation.

LabVIEW VI	C Function
niSync Measure Frequency	`niSync_MeasureFrequency`

Frequency measurement is useful for verifying that clock signals are properly connected. For example, a clock signal connected to PXI_Star3 could be measured by connecting the PXI_Star3 terminal to the PFI0 terminal and using the **Measure Frequency** VI or function.

FPGA Reconfiguration

The NI PXI-665*x* module includes a field programmable gate array (FPGA) for managing the device's timing and synchronization features. For a detailed block diagram of the NI PXI-665*x*, refer to the *NI PXI-665x User Manual*. In some special cases, you may want to program the FPGA with an alternate bitstream file. The following VI and function support this operation.

LabVIEW VI	C Function
niSync Configure FPGA	`niSync_ConfigureFPGA`

⚠ **Caution** FPGA Reconfiguration is a sensitive operation that can damage your NI PXI-665*x* module. Do *not* use this operation unless you are absolutely sure about what you are doing.

3

Synchronizing Multiple NI PXI-4472 Modules

The NI PXI-665*x* Timing and Synchronization Module enables you to pass PXI timing and triggering signals between two or more PXI chassis. This module is especially useful for synchronizing acquisitions involving large numbers of NI PXI-4472 Dynamic Signal Acquisition modules. Tight multichassis synchronization guarantees simultaneous sampling and minimal interchannel phase mismatch in high-channel-count systems.

This chapter describes four typical hardware configurations where the NI PXI-665*x* enables synchronization for NI PXI-4472 Dynamic Signal Acquisition (DSA) modules in one or more PXI chassis. The necessary hardware, signal connections, and example code are described for each configuration.

NI highly recommends that you thoroughly review the *Theory of Operation: Synchronizing Data Acquisition Across Multiple NI PXI-4472 Modules* section before running the application examples or starting your application development. The main goal of this section is to familiarize you with the NI PXI-4472 synchronization signals and with how the NI PXI-665*x* hardware can extend this synchronization beyond a single PXI chassis. After familiarizing yourself with the concepts described here, move on to the LabVIEW or LabWindows/CVI example programs before beginning your software development.

The *NI PXI-665x User Manual*, NI PXI-4472 documentation, and remote chassis link (such as MXI-3) documentation might be useful to you as you read this document. You can download these documents from `ni.com/manuals`.

Theory of Operation: Synchronizing Data Acquisition Across Multiple NI PXI-4472 Modules

The following sections provide low-level background information about the electrical signals needed to synchronize NI PXI-4472 modules. The *Synchronizing Signals for Multiple NI PXI-4472 Modules in a Single*

Chassis section describes the signals required to synchronize an acquisition when all NI PXI-4472 modules reside in the same PXI chassis. No NI PXI-665*x* is required in this case. The *Using the NI PXI-665x to Route Synchronization Signals Between Multiple Chassis* section discusses synchronization where NI PXI-4472 modules are physically located in two or more PXI chassis. In a multichassis system, two or more NI PXI-665*x* modules generate and route the necessary signals to synchronize all NI PXI-4472 channels in the system. The NI-Sync software package includes application examples that address three multichassis cases as well as a single-chassis case.

Synchronizing Signals for Multiple NI PXI-4472 Modules in a Single Chassis

This section describes the synchronization between NI PXI-4472 modules housed in a single PXI chassis. This section introduces three signals: the oversample clock, the SYNC pulse, and the acquisition start trigger. The *Using the NI PXI-665x to Route Synchronization Signals Between Multiple Chassis* section frequently refers to these signals.

Delta-Sigma ADCs and the Oversample Clock

The 24-bit A/D converters (ADCs) employed on the NI PXI-4472 belong to a class of components called *delta-sigma* (or ΔΣ) ADCs. The advantages of delta-sigma components as compared to other digitizers include high dynamic range, excellent linearity, and digital filtering to remove aliased frequency components from the data.

Most ADCs, including the successive approximation ADCs used in many data acquisition (DAQ) devices, are timed by a *sample clock*. This clock is simply a digital pulse train that drives the acquisition. In most cases, a rising edge on the sample clock signal starts a conversion. When an ADC is timed by a sample clock, the acquisition rate is equal to the frequency of the sample clock. For example, a 10 kHz sample clock produces a 10 kS/s acquisition rate.

One distinguishing feature of delta-sigma converters, including those on the NI PXI-4472, is that they use an *oversample clock* to drive the conversion. As the name implies, the physical frequency of the oversample clock signal is greater than the sample rate. When a single NI PXI-4472 acquires data, the high-frequency oversample clock is locally generated by a Direct Digital Synthesis (DDS) chip on the device. If two or more NI PXI-4472 modules are run in a single PXI chassis, they must share the oversample clock to provide a tightly synchronized acquisition.

When sharing this clock, one module becomes the "clock master," and the other modules become the "clock slaves." The clock master module must be placed in Slot 2 of the PXI chassis. The clock master uses the DDS chip to produce the oversample clock and drive its own acquisition. The clock master also exports the oversample clock to the PXI star trigger bus. Figure 3-1 shows a LabVIEW code snippet that illustrates how to export the oversample clock to a clock slave module in PXI Slots 3 and 4.

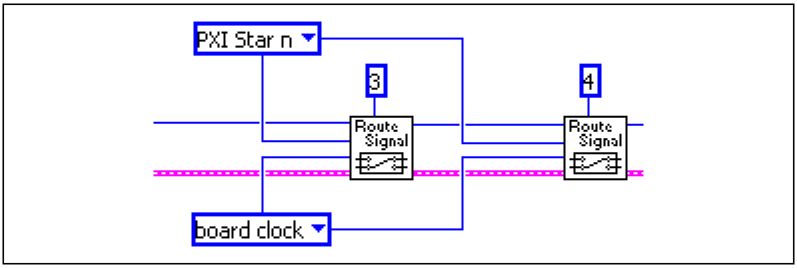


Figure 3-1. Exporting the Oversample Clock to PXI_Star

You now order each clock slave device to import the oversample clock from PXI_Star to drive its ADCs. Figure 3-2 shows the LabVIEW code snippet for this operation.

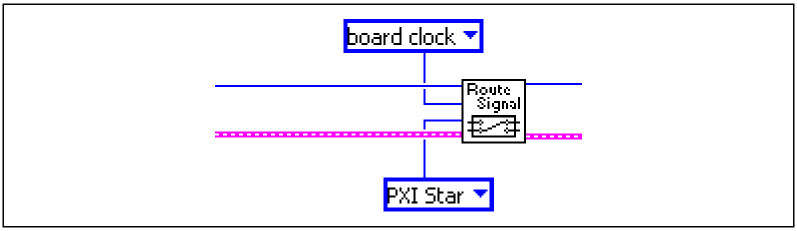


Figure 3-2. Importing the Oversample Clock to Drive Slave ADCs

On the NI PXI-4472, the ratio between the oversample (f_{os}) clock and the sample rate (f_s) can have one of two possible values, depending on the sample rate. Table 3-1 shows the possible values.

Table 3-1. Relationship Between the Sample Rate and Oversample Clock

Sample Rate	Oversample Clock
$f_s \leq 51.2$ kS/s	$f_{os} = 256 \times f_s$
$f_s > 51.2$ kS/s	$f_{os} = 128 \times f_s$

The highest possible oversample frequency for the NI PXI-4472 occurs at either f_s = 51.2 kS/s or f_s = 102.4 kS/s. In both cases, the oversample rate is slightly faster than 13.1 MHz.

Most delta-sigma converters, including those on the NI PXI-4472, require a very steady frequency for the oversample clock. The DDS chip on the NI PXI-4472 has good frequency characteristics and fulfills this need. However, arbitrarily changing frequencies, such as those from tachometers, generally do not work well with delta-sigma ADCs. For this reason, the NI PXI-4472 and other National Instruments DSA products do not support external clocking from arbitrary signal sources. This constraint also applies when you use one or more NI PXI-665*x* modules for synchronization with NI PXI-4472 devices.

The NI PXI-4472 SYNC Pulse

As discussed in the *Delta-Sigma ADCs and the Oversample Clock* section, the oversample clock is many times faster than the actual acquisition rate of the NI PXI-4472. Using the concepts described in that section, you can share the oversample clock between two or more NI PXI-4472 modules. This sharing guarantees that both modules sample at the same frequency and that there will be no drift between their acquisitions. However, oversample clock sharing does not guarantee that the samples on both modules are acquired at exactly the same time. Figure 3-3 illustrates the oversample pulse trains on two NI PXI-4472s sampling at 102.4 kS/s. In this example, there is a delay between acquisition samples of five oversample intervals, or about 350 ns. This delay may be any value up to a whole sample interval, which is about 10 µs at this acquisition rate.

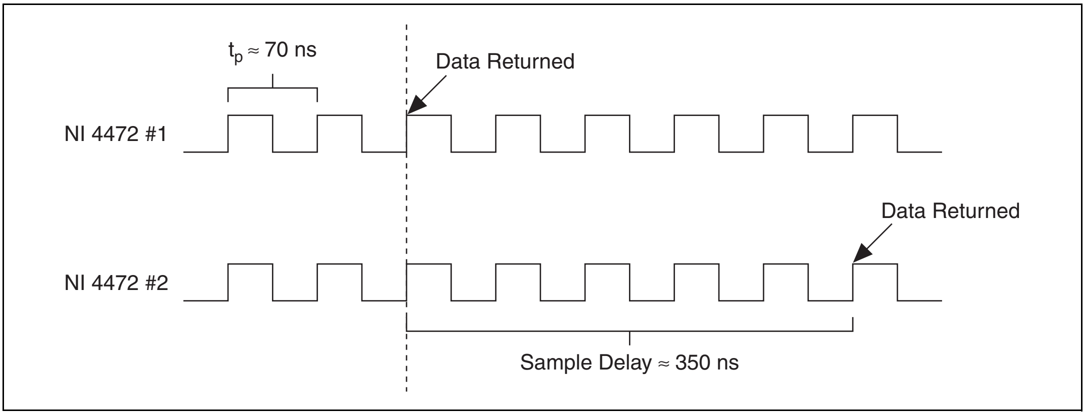


Figure 3-3. Sample Delay Between NI PXI-4472 Modules After Receiving a Shared Oversample Clock

The solution to the clock-delay issue is to configure the master device to issue a SYNC pulse before the acquisition. The clock master sends a single active-low, or inverted, pulse on the PXI_Trig5 line, the dedicated line for the SYNC pulse. The ADCs in the clock master and clock slaves receive this pulse nearly simultaneously. The SYNC pulse forces all the ADCs to a reset state, emptying their digital filters and synchronizing their clock dividers. After exiting the reset state, all NI PXI-4472 modules in the chassis run at the same frequency and have minimal phase difference between sample clocks. Figure 3-4 illustrates how this technique minimizes the sample delay.

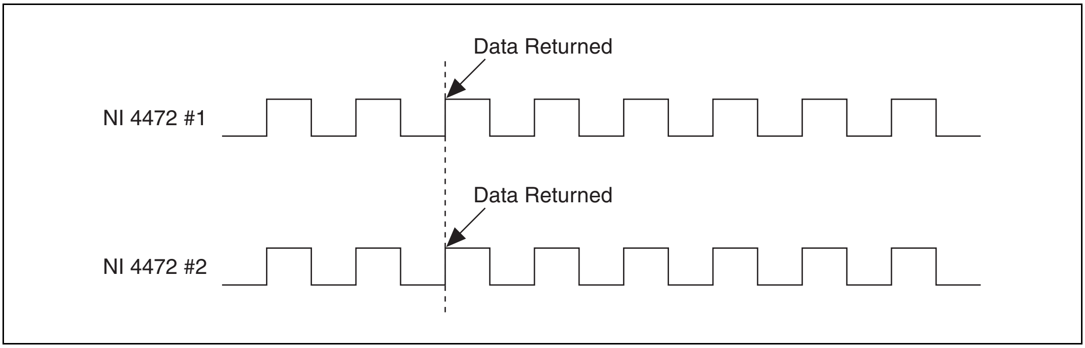


Figure 3-4. NI PXI-4472 Modules with Shared Oversample Clock Using the SYNC Pulse

The NI PXI-4472 cannot acquire data while it is in the reset state that occurs immediately after it receives the SYNC pulse. During this period, the ADC returns only zeros. The length of the reset period T_{reset} (in samples) is a function of sample rate. Table 3-2 shows the possible lengths of the reset period.

Table 3-2. Reset Period as a Function of Sample Rate

Sample Rate	Reset Period (T_{reset})
$f_s \leq 51.2$ kS/s	T_{reset} = 8,960 samples
$f_s > 51.2$ kS/s	T_{reset} = 17,920 samples

When all NI PXI-4472 modules are housed in a single chassis, the SYNC pulse is automatically handled by NI-DAQ, the NI driver software for DAQ hardware. Therefore, you do not need to make an explicit software call to generate or route the SYNC pulse. However, you must make specific calls to the NI PXI-665*x* modules in all chassis to properly generate and route the SYNC pulse when configuring a multichassis system. These calls are discussed in the *Using the NI PXI-665x to Route Synchronization Signals Between Multiple Chassis* section.

The Acquisition Start Trigger

After sharing the oversampling clock and issuing the SYNC pulse, the ADCs on every NI PXI-4472 in the system run in lock-step. At this point, the only remaining task is to synchronize the beginning of the data acquisition on each NI PXI-4472.

In the *Sharing the Oversample Clock Between Chassis* and *Sharing the SYNC Pulse Between Chassis* sections, clock master and clock slave modules were discussed. Because the oversample clock is carried on the PXI star trigger lines, the clock master module must always reside in Slot 2. This concept of master and slave devices is also useful when discussing the acquisition start trigger. Exactly one module should be the "trigger master," while all the other NI PXI-4472 modules are "trigger slaves." However, the trigger master can reside in any PXI slot; it is not limited to Slot 2. Therefore, the clock master and trigger master are not necessarily the same module.

First, configure all trigger slave NI PXI-4472 modules to start their acquisitions on a digital trigger from a PXI_Trig line. You can choose PXI_Trig<0..4> for the start trigger; the choice of PXI_Trig0 line for this example is arbitrary. Figure 3-5 shows how to wire a trigger cluster to the **AI Start** VI to accomplish this task.

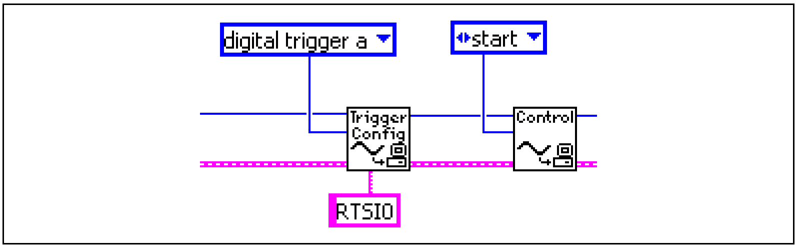


Figure 3-5. Starting the Acquisition on a Trigger Slave

Next, direct the trigger master to export a pulse to PXI_Trig0 when it begins acquiring. This pulse, in turn, triggers all the slaves. Figure 3-6 shows the LabVIEW code snippet used to configure the trigger master.

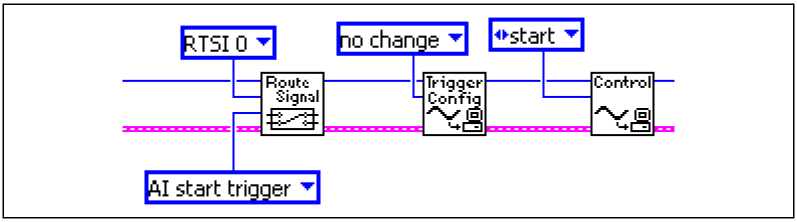


Figure 3-6. Routing the Start Trigger and Starting the Acquisition on the Trigger Master

The code in Figure 3-6 uses immediate software triggering for the trigger master NI PXI-4472. However, you could easily change the triggering parameters on the trigger master **AI Start** VI to use external digital or analog triggering.

Using the NI PXI-665*x* to Route Synchronization Signals Between Multiple Chassis

The *Synchronizing Signals for Multiple NI PXI-4472 Modules in a Single Chassis* section discussed why the oversample clock, SYNC pulse, and acquisition start trigger are necessary and addressed the software calls needed to share these signals between NI PXI-4472 modules residing in a single PXI chassis. This section explains how the NI PXI-665*x* enables you to expand the channel count of your system by sharing these same three signals among multiple chassis.

Sharing the Oversample Clock Between Chassis

In a single-chassis synchronized DSA system, the clock master NI PXI-4472 generates the oversample clock and distributes it to the clock slave modules. When the DSA system consists of multiple PXI chassis, the NI PXI-665*x* in the master chassis generates the oversample clock and distributes it to all the NI PXI-4472 modules. Thus, every DSA device in a multichassis system is a clock slave; you do not define a clock master NI PXI-4472.

First, consider a system with either two or three chassis where each chassis contains an NI PXI-665*x* in Slot 2 and several NI PXI-4472 modules. One NI PXI-665*x* is defined as the “NI PXI-665*x* master.” This module uses its DDS chip to generate the appropriate oversample clock. The NI PXI-665*x* master is also responsible for distributing the oversample clock to all the other “slave” chassis. Each NI PXI-665*x* (both master and slave) must route the oversample clock to the PXI backplane so that it can drive the ADCs on the NI PXI-4472 modules housed in the chassis. Refer to Figure 3-17 for an illustration of the hardware setup for this configuration.

In this diagram, the NI PXI-665*x* in the top chassis is the master, while the NI PXI-665*x* modules in the other two chassis are slaves. The master connects to both slaves. Notice that the master includes one cable that "loops back" to itself. This loopback guarantees that the oversample pulse train travels over a similar path before reaching the PXI star trigger bus on both backplanes. As the pulse travels along the coaxial cable and through the circuitry on the NI PXI-665*x*, it creates a significant propagation delay. The loopback minimizes this timing difference and ensures symmetry between the paths to the backplane of each chassis.

The code snippet shown in Figure 3-7 programs the master NI PXI-665*x* to generate a frequency of 13.1072 MHz, the oversample frequency corresponding to an NI PXI-4472 sampling rate of either 51.2 kS/s or 102.4 kS/s. The code then routes the signal to three front panel pins: PFI 1, PFI 2, and PFI 4. PFI 1 is connected to PFI 0, while PFI 2 and PFI 4 connect to the NI PXI-665*x* modules in the second and third chassis, respectively. *PFI* stands for Programmable Function Interface. These pins are discussed in more detail in the *NI PXI-665x User Manual*.

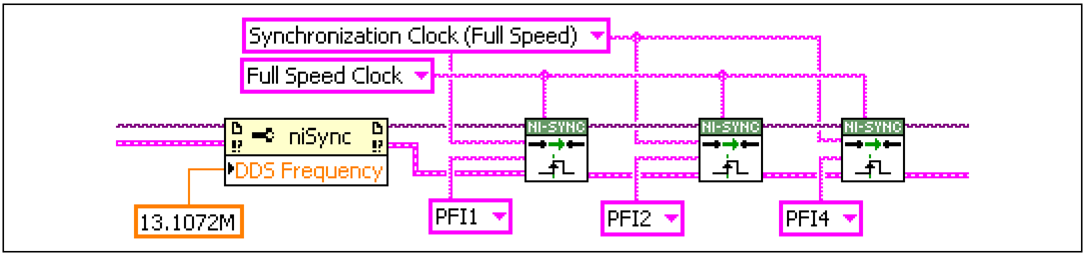


Figure 3-7. Generating and Routing the Oversample Frequency

Next, both the master and slave NI PXI-665*x* modules must import the oversample clock signal on PFI 0 and route it to the appropriate PXI star trigger lines. This routing is an asynchronous operation. *Asynchronous routing* means that the signal follows a short path from the PFI input to the PXI star trigger bus and that the NI PXI-665*x* hardware does not actively adjust the phase or timing of the signal. The code shown in Figure 3-8 routes the oversample clock from PFI 0 to NI PXI-4472 boards in Slots 3 through 5.

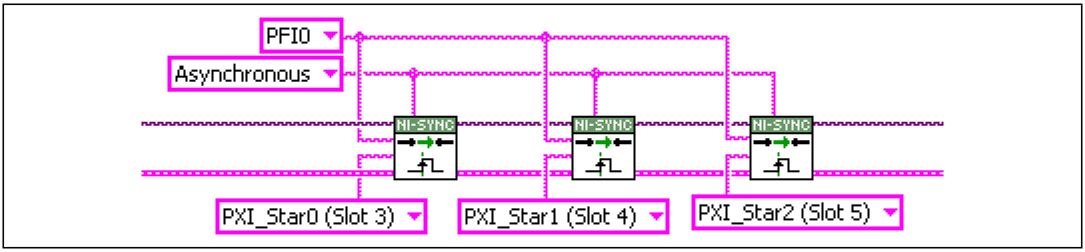


Figure 3-8. Routing the Oversample Clock Using the PXI Star Trigger Bus

All NI PXI-4472 modules in the system then import this oversample clock from their local PXI star lines. The LabVIEW code snippet used to import the clock is shown in Figure 3-9.

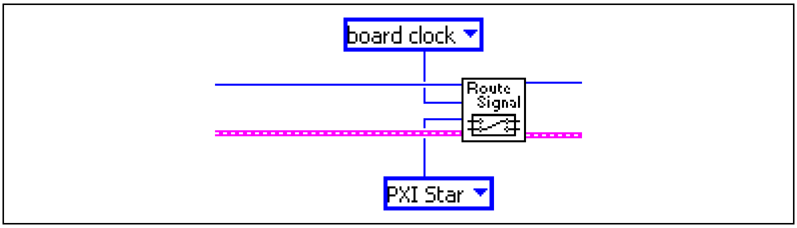


Figure 3-9. Importing the Oversample Clock from PXI_Star

Sharing the SYNC Pulse Between Chassis

The SYNC pulse is an active-low signal on PXI_Trig5 that resets the internal clock dividers in the NI PXI-4472 ADCs and minimizes phase mismatch across devices. When all NI PXI-4472 modules are in a single chassis, NI-DAQ automatically handles the SYNC signal. In a multichassis DSA system; however, you must include code to explicitly handle the SYNC pulse using the NI PXI-665*x*.

The master NI PXI-665*x* is responsible for issuing the SYNC. This signal is an active-low pulse, so the NI PXI-4472 ADCs reset when they receive a falling edge on PXI_Trig5.

All the generation and routing of the SYNC is performed synchronously with the oversample clock. This synchronous routing means that the edges of the SYNC pulse precisely line up with the edges of the oversample clock. Synchronous routing allows the NI PXI-665*x* in the master chassis to issue the SYNC directly to PXI_Trig5 without requiring a loopback scheme like that of the oversample clock. Instead, the master NI PXI-665*x* actively delays the pulse based on the cable length between chassis. The code snippet in Figure 3-10 prepares the master to issue the SYNC pulse to PFI3 and PFI5 (physically cabled to PFI 1 on the two slave chassis) and to the local PXI_Trig5 line. The code includes delay on the master to compensate for propagation delay through 5 m of coaxial cable.

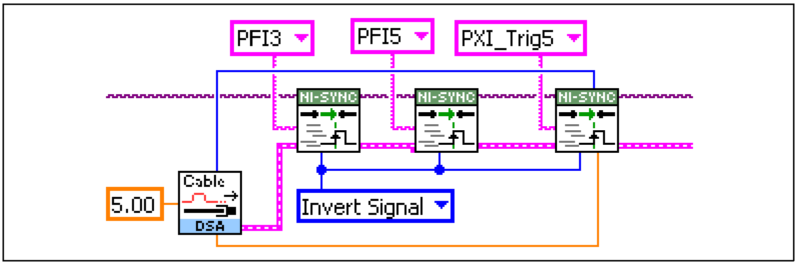


Figure 3-10. Compensating for Propagation Delay on the SYNC Pulse

Each slave NI PXI-665*x* then routes the incoming SYNC pulse on PFI1 to the PXI_Trig5 line as shown in Figure 3-11.

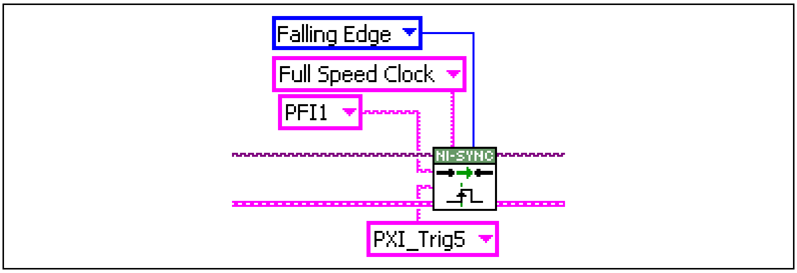


Figure 3-11. Routing the SYNC Pulse

Note The previous code snippets illustrate how to set up the routing for the NI PXI-4472 SYNC pulse. They guarantee that when the pulse is generated, it propagates with minimal skew to all the NI PXI-4472 modules in the system. A subsequent subVI that is not shown in the code snippets actually issues the SYNC pulse.

Sharing the Acquisition Start Trigger Between Chassis

The last signal you must pass between chassis is the start trigger, used to initiate the NI PXI-4472 acquisition. Unlike the oversample clock and the SYNC pulse, the trigger is actually generated by an NI PXI-4472, not by an NI PXI-665*x*. The trigger master (the NI PXI-4472 that generates the trigger signal) can be any NI PXI-4472 residing in the master chassis. As with the single-chassis case, this trigger master can initiate a system acquisition based on an immediate software trigger or an external digital or analog trigger.

The start trigger is passed asynchronously. Asynchronous routing means there is no compensation for cable propagation time between the master and slave chassis. The trigger signal is automatically resynchronized to the sample clock (as opposed to the faster oversample clock) when it is received on each NI PXI-4472 device. For cables shorter than about 30 m, the trigger signal arrives within the same sample interval on both chassis, guaranteeing tight synchronization. However, the trigger signal arrives one sample *later* on the slave chassis for cables longer than about 30 m. There are comments highlighting this effect in the DSA example programs. If you do use long cables, the easiest solution is simply to acquire one more sample on the master chassis than on the slaves and to discard the first sample on the master. This "software data realignment" compensates for the asynchronous trigger delay. If you notice a significant phase mismatch between the same signal acquired on each chassis with long cables, you should try to use the data realignment technique to correct the issue. This behavior generally is easily noticeable if it is present, usually on the order of several degrees. For example, consider the case of a 1 kHz tone acquired at a sampling rate of 100 kS/s. A mismatch of exactly one sample corresponds to a time difference of 10 µS, or 1% of a sine tone cycle. Because 360 degrees comprise a full cycle, this error would show up as 3.6 degrees. Note that the trigger delay never produces an error greater than one sample interval. Thus, a 60 or 70 m cable still produces a single sample mismatch, just as would a 30 m cable.

As in the single-chassis case, the trigger master NI PXI-4472 exports the start trigger signal to PXI_Trig0 (here again, the choice of PXI_Trig line for passing the trigger is arbitrary). This trigger starts the acquisition on all NI PXI-4472 modules in the master chassis. The NI PXI-665*x* in the master chassis is then responsible for asynchronously routing the start trigger from PXI_Trig0 to its front panel pins PFI 3 and PFI 5. These same pins were earlier used to propagate the SYNC pulse. Figure 3-12 shows the LabVIEW code snippet used to export the start trigger signal.

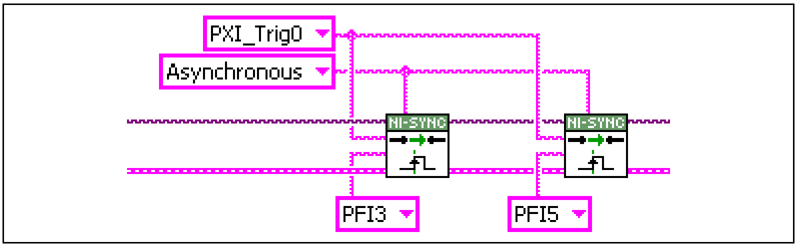


Figure 3-12. Asynchronously Routing the Start Trigger to the PFI Pins

Each slave NI PXI-665*x* then routes the incoming pulse on PFI1 to PXI_Trig0. Again, this operation is performed asynchronously. Figure 3-13 shows the code snippet you need to route the signal.

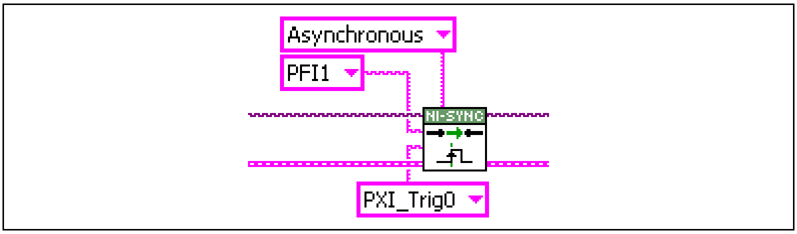


Figure 3-13. Routing the Incoming Start Trigger to PXI_Trig0

With respect to the acquisition triggering, the NI PXI-4472 modules operate exactly as they do in a single-chassis synchronized system. The slaves digitally trigger from PXI_Trig0. Figure 3-14 shows the LabVIEW code snippet that you can use to digitally trigger from PXI_Trig0.

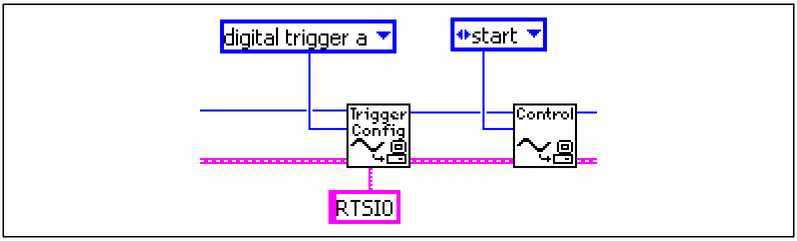


Figure 3-14. Digital Trigger from PXI_Trig0 on Trigger Slave

The master exports its start trigger signal to PXI_Trig0. This operation is shown in Figure 3-15.

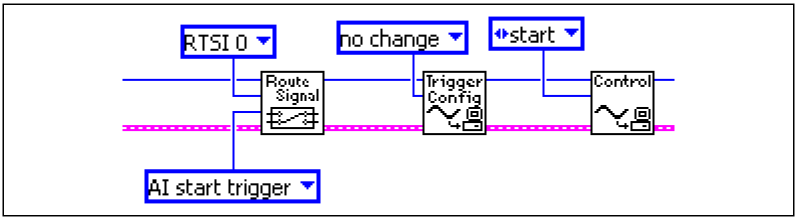


Figure 3-15. Exporting a Start Trigger to PXI_Trig0 on Trigger Master

In this case, the master starts with an immediate software trigger. However, you could easily modify the parameters for the **AI Trigger Config** VI on the master to enable external digital or analog triggering.

Configuring the Chassis

The following sections provide information for configuring the hardware for four use cases:

- *Configuration #1: One Chassis with Synchronized Acquisition and One CPU*
- *Configuration #2: Two or Three Chassis with Synchronized Acquisition and One CPU*
- *Configuration #3: Four or More Chassis with Synchronized Acquisition and One CPU*
- *Configuration #4: Two Chassis with Synchronized Acquisition and Two CPUs*

Determine which configuration you will use and refer to that section for more information.

Note The following system configurations use 1 m SMB-SMB cables for sharing clock and trigger signals between chassis. You can use longer cables if desired; refer to the *NI PXI-665x User Manual* for the maximum cable length specifications. However, to attain the tightest possible synchronization, it is important that you maintain equal length for all cables in the system and compensate for cable propagation delays in your software.

Software compensation for the SYNC pulse is necessary for all cable lengths. Acquisition start trigger compensation is needed for cables longer than about 30 m. The shipping examples for multichassis DSA systems address cable compensation, and it is also described in the *Using the NI PXI-665x to Route Synchronization Signals Between Multiple Chassis* section.

Configuration #1: One Chassis with Synchronized Acquisition and One CPU

This configuration allows synchronization between NI PXI-4472 modules housed in a single PXI chassis. The NI PXI-665*x* in PXI Slot 2 provides the oversample clock and SYNC pulse for the modules. Note that a single chassis application does not actually require an NI PXI-665*x*. The same requirements can be fulfilled (with somewhat simpler software) using only NI PXI-4472 modules in a single chassis. These single-chassis examples are included to provide tutorials or "stepping stones" before proceeding to a multichassis system that requires the NI PXI-665*x* and the NI-Sync driver.

The following LabVIEW and LabWindows/CVI example programs are provided for this configuration.

LabVIEW

- The **niSync_DSA Example Single Chassis** VI—This example illustrates a finite acquisition using an NI PXI-665*x* device to synchronize two or more NI PXI-4472 devices in a single chassis. It uses high-level VIs to control the NI PXI-665*x* and NI PXI-4472 modules.
- The **niSync_DSA Example 1 Chassis [Low Level]** VI—This example illustrates a finite acquisition synchronizing NI PXI-4472 devices in a single chassis. Unlike the other single-chassis LabVIEW example, this program relies on low-level VIs to control the hardware. This produces a larger and more intricate block diagram. The code is less modular and hierarchical than in the high-level LabVIEW examples.

LabWindows/CVI

- **Single Chassis Finite Acq**—This example illustrates a finite acquisition using an NI PXI-665*x* device to synchronize two or more NI PXI-4472 devices in a single chassis. This program does not include a LabWindows/CVI graphical user interface (GUI), so you can quickly adapt it to work with Microsoft Visual C++ or other ANSI C environments.

What You Need to Get Started

To set up and use Configuration #1, you need the following items:

❑ One PXI chassis

❑ CPU

The CPU can be an embedded PXI controller running Windows 2000/XP, or it can be a desktop computer running Windows 2000/XP with an installed remote chassis link (such as MXI-3). The embedded controller or remote link module should be installed in Slot 1 of the chassis.

❑ One NI PXI-6653 installed in Slot 2 of the PXI chassis

❑ Two or more NI PXI-4472 modules. Place the NI PXI-4472 modules in any available PXI slot numbered 3 to 15.

Note Do not install the NI PXI-4472 in Slots 16, 17, or 18 of an 18-slot chassis. You cannot route the oversample clock to those slots.

❑ One 1 m SMB-200 cable

❑ One 1 kHz sine source (function generator), for testing phase match between chassis.

Note The function generator is not required, but it is highly recommended for testing purposes.

Connecting the NI PXI-665*x* Device

Use an SMB-200 cable to connect PFI 0 to PFI 1 on the NI PXI-665*x*. This cable loops back to connect two external pins on the same device.

Connecting the NI PXI-4472 Devices

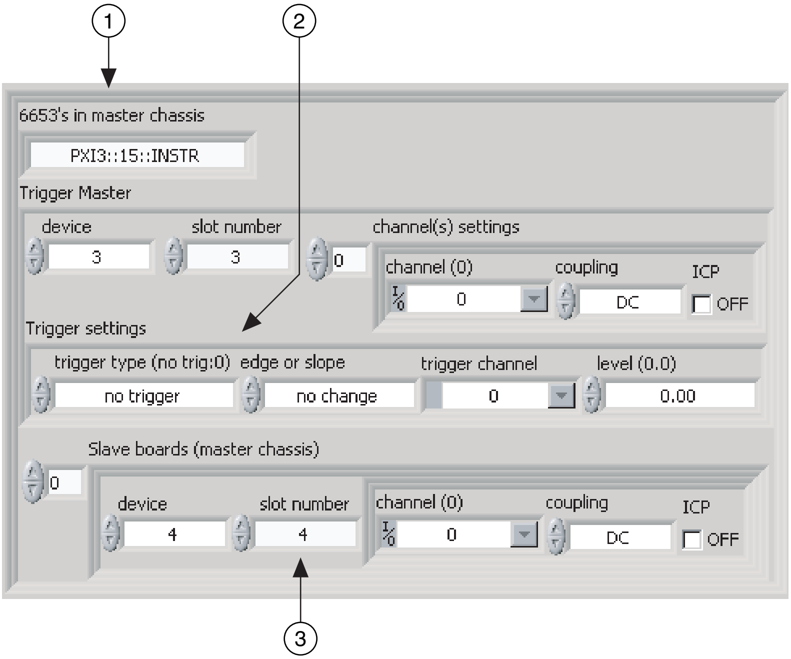
Connect the external sine signal to a single input channel on one NI PXI-4472 in each chassis. The same signal should be fed to each chassis to measure the phase mismatch and verify that the synchronization is within specifications.

Configuring and Running the Software Example

After the hardware has been configured, launch MAX. Verify that all NI PXI-4472 modules are recognized and properly functioning and that the NI PXI-665*x* module is recognized.

Using the niSync_DSA Example Single Chassis VI

Like the other high-level NI-Sync DSA examples, this program relies on modular subVIs to provide a level of abstraction from the driver calls that directly control the NI PXI-665*x* and NI PXI-4472 devices. All the high-level DSA examples share a nested configuration cluster labeled **Systems Settings**. This is a flexible data structure that allows you to precisely describe your hardware configuration before running the example. Figure 3-16 shows the **System Settings** cluster for the **niSync_DSA Example Single Chassis** VI.



6653's in master chassis
PXI3::15::INSTR

Trigger Master
device: 3
slot number: 3
channel(s) settings: 0
channel (0): 0
coupling: DC
ICP: □ OFF

Trigger settings
trigger type (no trig:0): no trigger
edge or slope: no change
trigger channel: 0
level (0.0): 0.00

Slave boards (master chassis): 0
device: 4
slot number: 4
channel (0): 0
coupling: DC
ICP: □ OFF

1 NI PXI-665*x* Master
2 NI PXI-4472 Trigger Master
3 NI PXI-4472 Trigger Slaves

Figure 3-16. System Settings Cluster for the niSync_DSA Example Single Chassis VI

- **NI PXI-665*x* Master**—Enter the VISA Resource Name for the NI PXI-665*x* in the master chassis. The control labeled **6653s in master chassis** is actually an array, although in this particular example the array index control is not shown. The reason for this is that the single-chassis configuration implies only a single NI PXI-665*x* in the system.

- **NI PXI-4472 Trigger Master**—A single NI PXI-4472 is responsible for issuing the acquisition trigger signal to start the data acquisition on every NI PXI-4472 in the system. The trigger master can use any of its three trigger modes (software trigger, external analog trigger, external digital trigger on **EXT Trig** pin).
- **NI PXI-4472 Trigger Slaves**—This is an array of clusters with one element per board. All NI PXI-4472 devices except the one responsible for issuing the trigger are described here.

After filling in the **System Settings** cluster, you should set the desired sampling rate and run the example. To test the quality of the synchronization between NI PXI-4472 devices, connect a common 1 kHz sine tone to the first input channel on the trigger master and the first input channel on the first trigger slave. A phase mismatch result of less than 0.01° indicates excellent synchronization.

Using the niSync_DSA Example 1 Chassis [Low Level] VI

The **niSync_DSA Example 1 Chassis [Low Level]** VI is simpler and less modular than the high-level single-chassis example described above. It uses only a software trigger (the code does not illustrate analog or external digital hardware triggering).

Enter all NI PXI-4472 device numbers in the array labeled **4472 Device Numbers**. Note that the trigger master is defined as the first device number appearing in this array. Next, fill in the PXI slot numbers for the NI PXI-4472 devices (typically, the device numbers equal the slot numbers).

To test the quality of the synchronization between NI PXI-4472 devices, connect a common 1 kHz sine tone to the first input channel on the trigger master and the first input channel on the first trigger slave. A phase mismatch result of less than 0.01° indicates excellent synchronization.

Using niSync_DSA_Example Single Chassis (LabWindows/CVI Example Project)

This simple example is configured to synchronize two NI PXI-4472 devices in one chassis using an NI PXI-665*x* to deliver the oversample clock and SYNC pulse. This program does not include a LabWindows/CVI GUI. Instead, it uses the Standard I/O window for user interaction.

The code contains several hardwired values that may need to be changed to accommodate your particular hardware setup:

- The NI PXI-665*x* VISA Resource Name is set to `PXI3::15::INSTR`.
- The NI PXI-4472 devices are numbered 3 and 4 and reside in PXI Slots 3 and 4.
- Data is acquired from the first channel on each NI PXI-4472. The `numChan` variable controls how many channels (starting with ACH0) should acquire data on each DSA device.
- The sampling rate is 102.4 kS/s.

These constants are defined at the beginning of the main code block and should be changed as needed.

To test the quality of the synchronization between NI PXI-4472 devices, connect a common 1 kHz sine tone to the first input channel on the trigger master and the first input channel on the first trigger slave. A phase mismatch result of less than 0.01° indicates excellent synchronization. When the program is run, the phase mismatch is displayed in the I/O window.

Configuration #2: Two or Three Chassis with Synchronized Acquisition and One CPU

This configuration enables synchronization between NI PXI-4472 modules housed in two or three PXI chassis. It assumes that only one CPU controls the acquisition. The CPU can be either a desktop PCI machine with a MXI-3 link or an embedded PXI controller in the master chassis.

The following LabVIEW and LabWindows/CVI example programs are provided for this configuration.

LabVIEW

- The **niSync_DSA Example 2 or 3 Chassis Finite Acq** VI—This example illustrates a finite acquisition synchronized between two or three chassis. It uses high-level VIs to control the NI PXI-665*x* and NI PXI-4472 modules.
- The **niSync_DSA Example 2 or 3 Chassis Continuous Acq** VI—This example illustrates a continuous acquisition synchronized between two or three chassis. It uses high-level VIs to control the NI PXI-665*x* and NI PXI-4472 modules.
- The **niSync_DSA Example 2 Chassis [Low Level]** VI—This example illustrates a finite acquisition synchronized between two

chassis. Unlike the other examples for this configuration, it relies on low-level VIs to control the hardware. This produces a larger and more intricate block diagram. The code is less modular and hierarchical than the high-level examples.

LabWindows/CVI

- **Dual Chassis Finite Acq**—This example synchronizes NI PXI-4472 modules across two chassis and performs a finite acquisition. You can readily modify it to support three chassis. This program does not include a LabWindows/CVI graphical user interface (GUI), so you can quickly adapt it to work with Microsoft Visual C++ or other ANSI C environments.
- **Dual Chassis Continuous Acq**—This example synchronizes NI PXI-4472 modules across two chassis and performs a continuous acquisition. You can readily modify it to support three chassis. This program does not include a LabWindows/CVI GUI, so you can quickly adapt it to work with Microsoft Visual C++ or other ANSI C environments.
- **Dual Chassis Finite Acq GUI**—This example synchronizes NI PXI-4472 modules across two chassis and performs a finite acquisition. It includes a full LabWindows/CVI GUI.

What You Need to Get Started

To set up and use Configuration #2, you need the following items:

❑ Two or three PXI chassis

Note The chassis containing either the embedded controller or the desktop MXI-3 link is referred to as Chassis #1. The other chassis are referred to as Chassis #2 and Chassis #3 (if a third chassis is present). In the remainder of this section, the number of chassis present is called *N*.

❑ CPU

The CPU can be an embedded PXI controller running Windows 2000/XP, or it can be a desktop computer running Windows 2000/XP with an installed remote chassis link (such as MXI-3). The embedded controller or remote link module should be installed in Slot 1 of Chassis #1.

❑ One NI PXI-665*x* per PXI chassis

Install one NI PXI-665*x* in Slot 2 of each chassis. The board in the first chassis must be an NI PXI-6653.

❑ One or more NI PXI-4472 per PXI chassis

Place the NI PXI-4472 modules in any available PXI slot numbered 3 to 15.

Note Do *not* install the NI PXI-4472 in Slots 16, 17, or 18 of an 18-slot chassis. You cannot route the oversample clock to those slots.

❑ (*N* – 1) SMB-210 cables

❑ One 1-meter SMB-200 cable

❑ (*N* – 1) PXI-to-PXI MXI-3 kits

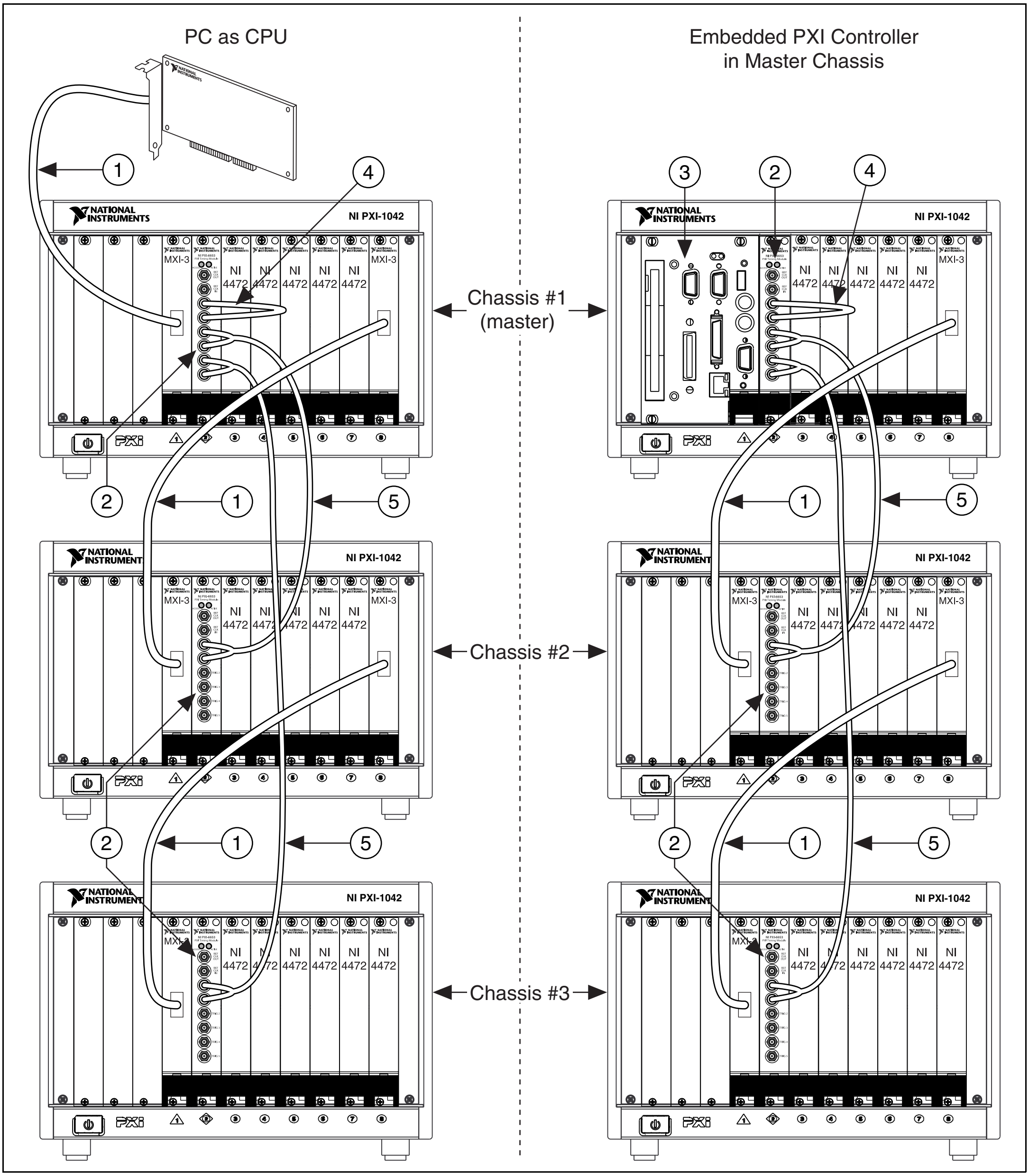
You should install a MXI-3 PXI module in Slot 1 of Chassis #2 (and Chassis #3, in a three-chassis system). You can use either fiber-optic or copper MXI-3 connections.

You also should install MXI-3 PXI modules in any available slot, *except* Slot 2, of Chassis #1 (and Chassis #2), with Slot 8 being the designated slot throughout this document.

❑ One 1 kHz sine source (function generator), for testing phase match between chassis.

Note The function generator is not required, but it is highly recommended for testing purposes.

Refer to Figure 3-17 for an illustration of the two configurations for a three-chassis synchronized acquisition using the NI PXI-665*x* and the NI PXI-4472.



1	MXI-3 Cable	3	Embedded PXI Controller	5	SMB-210 Cable
2	NI PXI-665*x* Module	4	SMB-200 Cable (1 m)		

Figure 3-17. Three Chassis with Synchronized Acquisition (With Either a PC or an Embedded PXI Controller)

Connecting the MXI-3 Devices

Referring to Figure 3-17, complete the following steps to connect the MXI-3 devices:

1. If you are using a desktop computer, install a MXI-3 cable to connect the PCI MXI-3 device to the PXI MXI-3 module in Slot 1 of Chassis #1.
2. Connect the MXI-3 module in Slot 1 of Chassis #2 to the MXI-3 module in Slot 8 of Chassis #1.

♦ Three-chassis system only

3. Connect the MXI-3 module in Slot 1 of Chassis #3 to the MXI-3 device in Slot 8 of Chassis #2.

Refer to the *Set Up Your PXI-MXI-3 System* document, which is included with the MXI-3 kit or available for download from `ni.com/manuals`, for further details.

Connecting the NI PXI-665*x* Devices

1. Use an SMB-200 cable to connect PFI 0 to PFI 1 on the NI PXI-665*x* of Chassis #1. This cable loops back to connect two external pins on the same device.
2. Use an SMB-210 cable to connect PFI 2 and PFI 3 on Chassis #1 to PFI 0 and PFI 1 of Chassis #2, respectively.

♦ Three-chassis system only

3. Use an SMB-210 cable to connect PFI 4 and PFI 5 of Chassis #1 to PFI 0 and PFI 1 of Chassis #3, respectively.

Connecting the NI PXI-4472 Devices

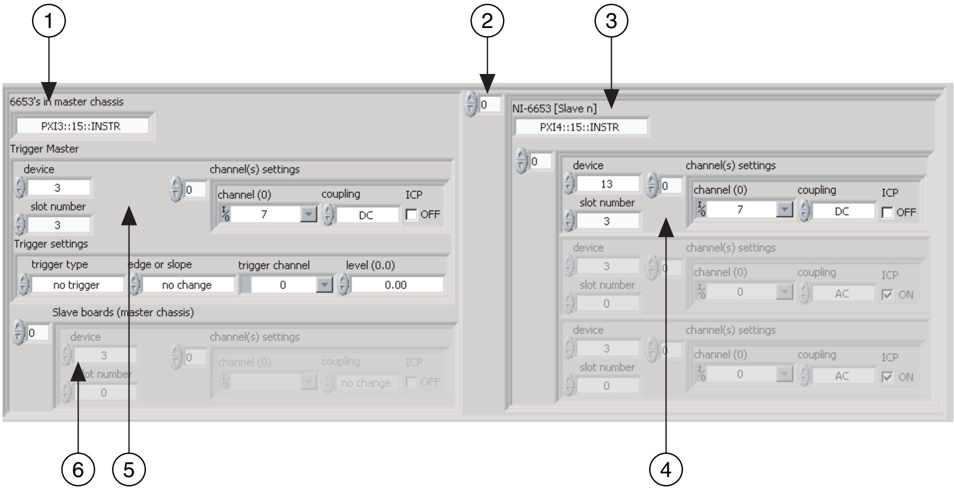
Connect the external sine signal to a single input channel on one NI PXI-4472 in each chassis. The same signal should be fed to each chassis to measure the phase mismatch and to verify that the synchronization is within specifications.

Configuring and Running the Software Example

After configuring the hardware, launch MAX. Verify that all NI PXI-4472 modules are recognized and properly functioning and that the NI PXI-665*x* modules are recognized.

Using the niSync_DSA_Example 2 or 3 Chassis Finite Acq VI and the niSync_DSA_Example 2 or 3 Chassis Continuous Acq VI

These two example programs use similar hardware setups and code; the only major difference between them is that one demonstrates a single-shot finite acquisition while the other shows a continuous double-buffered acquisition. Like the other high-level NI-Sync DSA examples, these programs rely on modular subVIs to provide a level of abstraction from the driver calls that directly control the NI PXI-665*x* and NI PXI-4472 devices. All the high-level DSA examples share a nested configuration cluster labeled **Systems Settings**. This is a flexible data structure that allows you to precisely describe your hardware configuration before running the example. Figure 3-18 shows the **System Settings** cluster for the **niSync_DSA_Example 2 or 3 Chassis Finite Acq** VI and the **niSync_DSA_Example 2 or 3 Chassis Continuous Acq** VI.



6653's in master chassis
PXI3::15::INSTR
Trigger Master
device 3
slot number 3
channel(s) settings 0
channel (0) 7 coupling DC ICP □ OFF
Trigger settings
trigger type no trigger
edge or slope no change
trigger channel 0
level (0.0) 0.00
Slave boards (master chassis) 0
device 3
slot number 0
channel(s) settings 0
channel (0) coupling no change ICP □ OFF

0
NI-6653 [Slave n]
PXI4::15::INSTR
0
device 13
slot number 3
channel(s) settings 0
channel (0) 7 coupling DC ICP □ OFF
device 3
slot number 0
channel(s) settings 0
channel (0) 0 coupling AC ICP ☑ ON
device 3
slot number 0
channel(s) settings 0
channel (0) 0 coupling AC ICP ☑ ON

1	NI PXI-665*x* Master	4	NI PXI-4472 Trigger Slaves in Second/Third Chassis
2	Slave Chassis Array Index	5	NI PXI-4472 Trigger Master
3	NI PXI-665*x* in Slave Chassis	6	NI PXI-4472 Trigger Slaves in First Chassis

Figure 3-18. System Settings Cluster for the niSync_DSA_Example 2 or 3 Chassis Finite Acq VI and the niSync_DSA_Example 2 or 3 Chassis Continuous Acq VI

- **NI PXI-665*x* Master**—Enter the VISA Resource Name for the NI PXI-665*x* in the master chassis. The control labeled **6653s in master chassis** is actually an array. However, in this particular screen the array index control is not shown. The reason for this is that the two/three chassis configuration implies only a single NI PXI-665*x* in the master chassis.

- **Slave Chassis Array Index**—Each slave chassis is described by an element in this array of clusters. This configuration can support one or two slave chassis.
- **NI PXI-665*x* in Slave Chassis**—This control is the VISA Resource Name for the NI PXI-665*x* in each slave chassis. This is *not* an array—the slave chassis should each have only one NI PXI-665*x* in Slot 2.
- **NI PXI-4472 Trigger Slaves in Second/Third Chassis**—This is an array describing each NI PXI-4472 in the slave chassis.
- **NI PXI-4472 Trigger Master**—A single NI PXI-4472 is responsible for issuing the acquisition trigger signal to start the data acquisition on every NI PXI-4472 in the system. This board must reside in the master chassis. The trigger master can use any of its three trigger modes (software trigger, external analog trigger, external digital trigger on **EXT Trig** pin).
- **NI PXI-4472 Trigger Slaves in First Chassis**—This is an array of clusters with one element per board. All NI PXI-4472 devices in the master chassis except the trigger master are described here. The trigger master is addressed in the **NI PXI-4472 Trigger Master** control.

After filling in the **System Settings** cluster, you should set the desired sampling rate and run the example. To test the quality of the synchronization between NI PXI-4472 devices, connect a common 1 kHz sine tone to the first input channel on the trigger master and the first input channel on the first trigger slave. A phase mismatch result of less than 0.01° indicates excellent synchronization.

Using the niSync_DSA Example 2 Chassis [Low Level] VI

The **niSync_DSA Example 2 Chassis [Low Level]** VI is simpler and less modular than the LabVIEW examples described above for synchronizing two or three chassis. This program uses only a software trigger (the code does not illustrate analog or external digital hardware triggering).

Enter all the NI PXI-4472 device numbers in the array labeled **4472 Device Numbers**. Note that the trigger master is defined as the first device number appearing in this array. The trigger master must reside in the first (master) PXI chassis. Next, fill in the PXI slot numbers for the NI PXI-4472 devices (typically, the device numbers equal the slot numbers in the first chassis but are different in the second chassis.).

To test the quality of the synchronization between NI PXI-4472 devices, connect a common 1 kHz sine tone to the first input channel on the trigger master and the first input channel on the first NI-PXI-4472 in the slave chassis. A phase mismatch result of less than 0.01° indicates excellent synchronization.

Using niSync_DSA_Example Dual Chassis Finite and niSync_DSA_Example Dual Chassis Continuous Acq (LabWindows/CVI Example Projects)

These examples are configured to synchronize two NI PXI-4472 devices in two separate PXI chassis. NI PXI-665*x* devices in Slot 2 of each chassis share the oversample clock, SYNC pulse, and acquisition start trigger. These programs do not include a LabWindows/CVI GUI; they use the Standard I/O window for user interaction. The only substantial difference between these programs is that the first demonstrates a single-shot finite acquisition, while the second shows a double-buffered continuous acquisition.

The code contains several hardwired values that may need to be changed to accommodate your particular hardware setup:

- The VISA Resource Name of the NI PXI-665*x* in the master chassis is set to `PXI3::15::INSTR`.
- The VISA Resource Name of the NI PXI-665*x* in the slave chassis is set to `PXI4::15::INSTR`.
- The NI PXI-4472 in the master chassis is device number 3 and resides in Slot 3.
- The NI PXI-4472 in the slave chassis is device number 13 and resides in Slot 3.
- Data is acquired from the first channel on each NI PXI-4472. The `numChan` variable controls how many channels (starting with ACH0) should acquire data on each DSA device.
- The sampling rate is 102.4 kS/s.

These constants are defined at the beginning of the main code block and should be changed as needed. If additional NI PXI-4472 devices are required in either chassis, you can add code for them directly after the existing functions to control the first slave NI PXI-4472. This hardware configuration can support a second slave chassis if needed. To add a second slave chassis, you can add code for its NI PXI-665*x* directly after the existing functions to control the first slave NI PXI-665*x*.

To test the quality of the synchronization between NI PXI-4472 devices, connect a common 1 kHz sine tone to ACH0 on the trigger master and trigger slave. A phase mismatch result of less than 0.01° indicates excellent synchronization. When the program is run, the phase mismatch is displayed in the I/O window.

Using niSync_DSA_Example Dual Chassis Finite GUI (LabWindows/CVI Example Project)

This example is configured to synchronize two NI PXI-4472 devices in two separate PXI chassis. NI PXI-665*x* devices in Slot 2 of each chassis share the oversample clock, SYNC pulse, and acquisition start trigger. This program includes a standard LabWindows/CVI GUI with callback user interface callback functions. If additional NI PXI-4472 devices are required in either chassis, you can add code for them directly after the existing functions to control the first slave NI PXI-4472. Data is acquired from the first channel on each NI PXI-4472. The `numChan` variable controls how many channels (starting with ACH0) should acquire data on each DSA device.

This hardware configuration can support a second slave chassis if needed. To add a second slave chassis, you can add code for its NI PXI-665*x* directly after the existing functions to control the first slave NI PXI-665*x*. After running the program and setting the controls on the GUI to match your hardware configuration, click the **Acquire** button to perform a single-shot acquisition on both NI PXI-665*x* devices.

To test the quality of the synchronization between NI PXI-4472 devices, connect a common 1 kHz sine tone to ACH0 on the trigger master and trigger slave. A phase mismatch result of less than 0.01° indicates excellent synchronization. The phase mismatch between the two DSA boards is displayed on the Phase Mismatch GUI indicator. Click **Acquire** to perform additional single-shot acquisitions as desired.

Configuration #3: Four or More Chassis with Synchronized Acquisition and One CPU

This configuration enables synchronization between NI PXI-4472 modules housed in four or more PXI chassis. Like the previous configuration, this configuration assumes that only one CPU controls the acquisition, so there is no separate embedded PXI controller for each chassis. This CPU may be either a desktop PCI machine with a MXI-3 link or an embedded PXI controller.

A LabVIEW example is provided for this configuration. The example program is called **niSync_DSA_Example N Chassis**. It uses high-level VIs to illustrate a finite acquisition.

What You Need to Get Started

To set up and use Configuration #3, you need the following items:

❑ Four or more PXI chassis

Note The chassis containing either the embedded controller or the desktop MXI-3 link is referred to as Chassis #1. The other chassis are referred to as Chassis #2 through Chassis #*N*. In the remainder of the section, the number of chassis present is called *N*.

❑ CPU

The CPU can be an embedded PXI controller running Windows 2000/XP or a desktop computer running Windows 2000/XP with a remote chassis link (such as MXI-3). Install the embedded controller or remote link module in Slot 1 of Chassis #1.

❑ Several NI PXI-665*x* modules

You will need one module per chassis, plus an additional module for every three chassis after the first three chassis. Table 3-3 shows the relationship between chassis and modules.

Table 3-3. Number of Chassis and Required Number of NI PXI-665*x* Modules

Number of PXI Chassis	Number of NI PXI-665*x* Required
4	5
5	6
6	7
7	9
10	13

Install one NI PXI-665*x* in Slot 2 of each chassis. Install the remaining NI PXI-665*x* modules in the leftmost available slots in the first chassis (Slot 3, Slot 4, and so on).

Note All NI PXI-665*x* boards in the first chassis must be NI PXI-6653 devices.

❑ One or more NI PXI-4472 modules per chassis

Place the NI PXI-4472 modules in any of the available PXI slots numbered 3 through 15.

Note Do *not* install the NI PXI-4472 in Slots 16, 17, or 18 of an 18-slot chassis. You cannot route the oversample clock to those slots.

❑ (*N* – 1) SMB-210 cables

❑ One 1 m SMB-200 cable

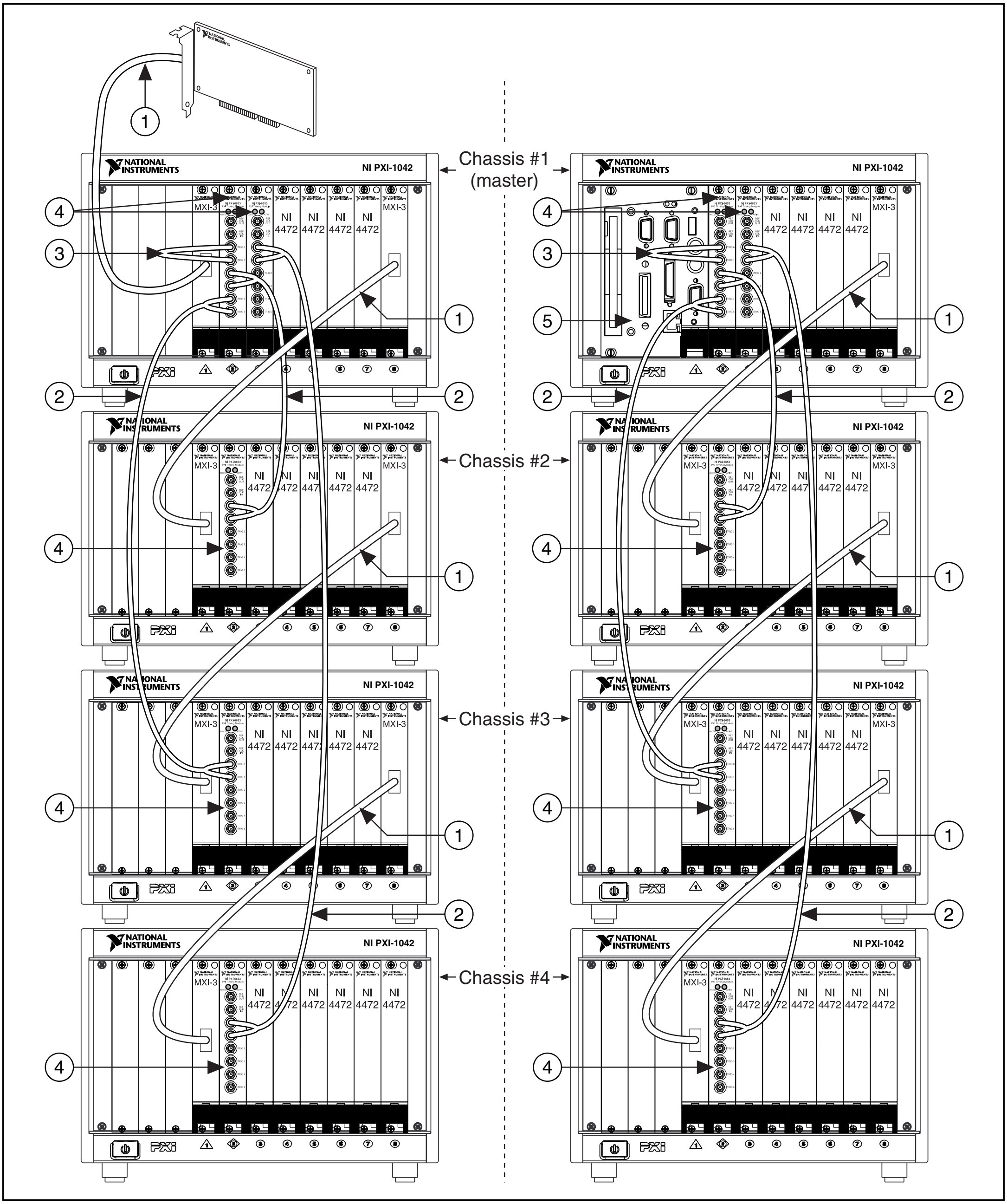
❑ (*N* – 1) PXI-to-PXI MXI-3 kits

Install a MXI-3 PXI device in Slot 1 of each chassis except for Chassis #1. Then install a MXI-3 PXI device in any available slot of Chassis #1 through Chassis (*N* – 1). This slot is designated as Slot 8 for the remainder of this document.

❑ One 1 kHz sine source (function generator), for testing phase mismatch between chassis

Note The function generator is not required, but it is highly recommended for testing purposes.

Refer to Figure 3-19 for an illustration of the two configurations for a four-chassis synchronized acquisition using the NI PXI-665*x* and the NI PXI-4472.



1	MXI-3 Cable	3	SMB-200 Cable (1 m)	5	Embedded PXI Controller
2	SMB-210 Cable	4	NI PXI-665*x* Module		

Figure 3-19. Four Chassis with Synchronized Acquisition

Connecting the MXI-3 Devices

Referring to Figure 3-19, complete the following steps to connect the MXI-3 devices:

1. If you are using a desktop computer, use a MXI-3 cable to connect the PCI MXI-3 device to the PXI MXI-3 module in Slot 1 of Chassis #1.
2. Connect the MXI-3 module in Slot 8 of Chassis #1 to the MXI-3 module in Slot 1 of Chassis #2.
3. Daisy-chain all remaining chassis. Start by connecting the MXI-3 module in Slot 8 of Chassis #2 to the MXI-3 module in Slot 1 of Chassis #3. Continue until each chassis is connected with a MXI-3 cable to Slot 1 of the next chassis.

Refer to the *Set Up Your PXI-MXI-3 System* document, which is included with the MXI-3 kit or available for download from `ni.com/manuals`, for further details.

Connecting the NI PXI-665*x*

1. Use an SMB-200 cable to connect PFI 0 to PFI 1 on the NI PXI-665*x* in Slot 2 of Chassis #1. This cable “loops back” to connect two external pins on the same device.
2. Use an SMB-210 cable to connect PFI 2 and PFI 3 on the NI PXI-665*x* in Slot 2 of Chassis #1 to PFI 0 and PFI 1 of Chassis #2, respectively.
3. Use an SMB-210 cable to connect PFI 4 and PFI 5 on the NI PXI-665*x* in Slot 2 of Chassis #1 to PFI 0 and PFI 1 on Chassis #3, respectively.
4. Use an SMB-210 cable to connect PFI 0 and PFI 1 on the NI PXI-665*x* in Slot 3 of Chassis #1 to PFI 0 and PFI 1 of Chassis #4, respectively.
5. Follow the instruction that applies to your system:

♦ Five or more chassis system only

Use an SMB-210 cable to connect PFI 2 and PFI 3 on the NI PXI-665*x* in Slot 3 of Chassis #1 to PFI 0 and PFI 1 of Chassis #5, respectively.

♦ Six or more chassis system only

Use an SMB-210 cable to connect PFI 4 and PFI 5 on the NI PXI-665*x* in Slot 3 of Chassis #1 to PFI 0 and PFI 1 of Chassis #6, respectively.

♦ Seven or more chassis system only

A seven-chassis system requires three NI PXI-665*x* modules in the first chassis. The previous steps have detailed all cable connections for the first two NI PXI-665*x* modules in the master. The connections for the third NI PXI-665*x* in the master chassis are identical to those for the second, except that they pertain to an additional one to three slave chassis. This configuration can be extended for up to 16 NI PXI-665*x* modules in the master chassis, yielding a potential maximum of 48 chassis containing NI PXI-4472 modules.

Connecting the NI PXI-4472 Devices

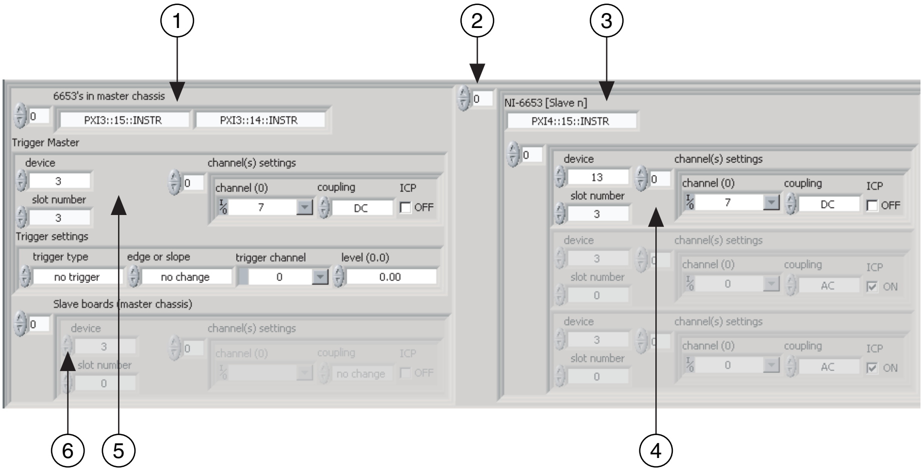
Connect the external sine signal to a single input channel on one NI PXI-4472 in each chassis. The same signal should be fed to each chassis to measure the phase mismatch and to verify that the synchronization is within specifications.

Configuring and Running the Software Example

After configuring the hardware, launch MAX. Verify that all NI PXI-4472 modules are recognized and functioning properly and that the NI PXI-665*x* modules are recognized.

Using the niSync_DSA_Example N Chassis VI

This example uses high-level subVIs to illustrate a finite acquisition synchronizing NI PXI-4472 devices in an arbitrary number of PXI chassis. Like the other high-level NI-Sync DSA examples, these programs rely on modular subVIs to provide a level of abstraction from the driver calls that directly control the NI-PXI 665*x* and NI PXI-4472 devices. All the high-level DSA examples share a nested configuration cluster labeled **Systems Settings**. This is a flexible data structure that allows you to precisely describe your hardware configuration before running the example. The screen short below shows the **System Settings** cluster for the **niSync_DSA_Example N Chassis** VI.



6653's in master chassis: 0 | PXI3::15::INSTR | PXI3::14::INSTR

Trigger Master

device: 3

slot number: 3

channel(s) settings: 0 | channel (0): 7 | coupling: DC | ICP: □ OFF

Trigger settings

trigger type: no trigger | edge or slope: no change | trigger channel: 0 | level (0.0): 0.00

Slave boards (master chassis): 0

device: 3

slot number: 0

channel(s) settings: 0 | channel (0) | coupling: no change | ICP: □ OFF

0

NI-6653 [Slave n]: PXI4::15::INSTR

0

device: 13 | slot number: 3 | channel(s) settings: 0 | channel (0): 7 | coupling: DC | ICP: □ OFF

device: 3 | slot number: 0 | channel(s) settings: 0 | channel (0): 0 | coupling: AC | ICP: ☑ ON

device: 3 | slot number: 0 | channel(s) settings: 0 | channel (0): 0 | coupling: AC | ICP: ☑ ON

1	NI PXI-665*x* Devices in Master Chassis	4	NI PXI-4472 Trigger Slaves in Additional Chassis
2	Slave Chassis Array Index	5	NI PXI-4472 Trigger Master
3	NI PXI-665*x* in Slave Chassis	6	NI PXI-4472 Trigger Slaves in First Chassis

Figure 3-20. System Settings Cluster for the niSync_DSA_Example N Chassis VI

- **NI PXI-665*x* Devices in Master Chassis**—Enter the VISA Resource Names for the NI PXI-665*x* devices in the master chassis. The N chassis configuration requires two or more NI PXI-665*x* devices in the master chassis to connect to three or more slave chassis. Notice that the first element in this array must refer to the NI PXI-665*x* residing in PXI Slot 2 of the master chassis.
- **Slave Chassis Array Index**—Each slave chassis is described by an element in this array of clusters. This configuration can support one or two slave chassis.
- **NI PXI-665*x* in Slave Chassis**—This control is the VISA Resource Name for the NI PXI-665*x* in each slave chassis. This is *not* an array—the slave chassis should each have only one NI PXI-665*x* in Slot 2.
- **NI PXI-4472 Trigger Slaves in Additional Chassis**—This is an array describing each NI PXI-4472 in the slave chassis.

- **NI PXI-4472 Trigger Master**—A single NI PXI-4472 in is responsible for issuing the acquisition trigger signal to start the data acquisition on every NI PXI-4472 in the system. This board must reside in the master chassis. The trigger master can use any of its three trigger modes (software trigger, external analog trigger, external digital trigger on EXT Trig pin).
- **NI PXI-4472 Trigger Slaves in First Chassis**—This is an array of clusters with one element per board. All NI PXI-4472 devices in the master chassis except the trigger master are described here. The trigger master is addressed in the **NI PXI-4472 Trigger Master** control.

After filling in the **System Settings** cluster, you should set the desired sampling rate and run the example. To test the quality of the synchronization between NI PXI-4472 devices, connect a common 1 kHz sine tone to the first input channel on the trigger master and the first input channel on the first trigger slave. A phase mismatch result of less than 0.01° indicates excellent synchronization.

Configuration #4: Two Chassis with Synchronized Acquisition and Two CPUs

This configuration enables synchronization between NI PXI-4472 modules housed in two PXI chassis. It assumes that each PXI chassis is equipped with its own embedded PXI controller. You also can use these examples with two separate desktop PCs, each controlling a single chassis using a remote chassis link (such as MXI-3) instead of embedded PXI controllers.

Three LabVIEW examples are provided for this configuration. All three of the following example VIs are required to use this configuration:

- The **niSync_DSA Example 2 or 3 Dual Chassis Dual Controller [Master]** VI—This example runs on the CPU of the master chassis. It controls the hardware on this chassis to perform a synchronized finite acquisition.
- The **niSync_DSA Example 2 or 3 Dual Chassis Dual Controller [Slave]** VI—This example runs on the CPU of the slave chassis. It controls the hardware on this chassis to perform a synchronized finite acquisition.
- The **niSync_DSA Example 2 or 3 Dual Chassis Dual Controller [Main]** VI—This example coordinates the execution of the slave and master programs. It may be run on the master CPU or on a third machine as desired.

What You Need to Get Started

To set up and use Configuration #4, you need the following items:

❑ Two PXI chassis

❑ Two CPUs

Each PXI chassis should contain an embedded PXI controller running Windows 2000/XP. Alternately, two desktop PCs with remote chassis links (such as MXI-3) running Windows 2000/XP could control the two chassis.

❑ Two NI PXI-665*x* modules

Install one NI PXI-665*x* in Slot 2 of each chassis.

The device installed in the master chassis must be an NI PXI-6653.

❑ One or more NI PXI-4472 modules per chassis

Place the NI PXI-4472 modules in any of the available PXI slots numbered 3 through 15.

Note Do *not* install the NI PXI-4472 in Slots 16, 17, or 18 of an 18-slot chassis. You cannot route the oversample clock to those slots.

❑ Two Ethernet ports

If the PXI controllers or desktop PCs have embedded Ethernet ports, you can use these without additional hardware. If you have controllers that do not contain built-in ports, such as the NI PXI-8170, install a PXI Ethernet card, such as the NI PXI-8231, in an available slot in each chassis.

❑ External Ethernet hub or common LAN

You can use any setup that allows each chassis to communicate with the other chassis over the Ethernet.

❑ Two Ethernet cables

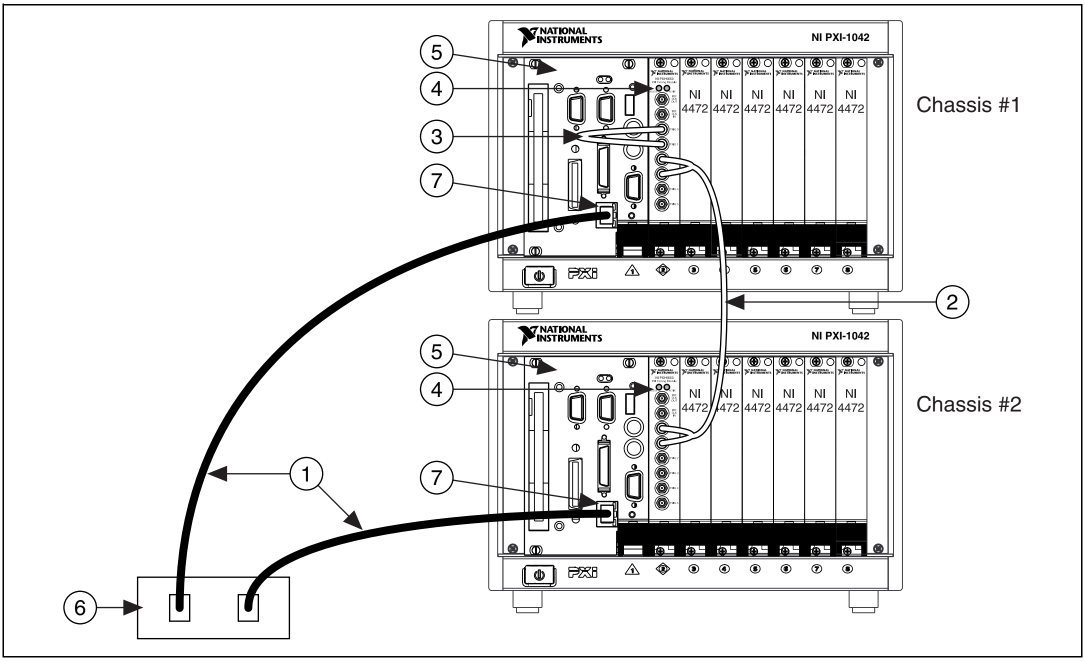
❑ One SMB-210 cable

❑ One 1 m SMB-200 cable

❑ One 1 kHz sine source (function generator) for testing phase mismatch between chassis

Note The function generator is not required, but it is highly recommended for testing purposes.

Refer to Figure 3-21 for an illustration of the hardware configuration for two synchronized chassis, each controlled by its own CPU.



1	Ethernet Cable	5	Embedded PXI Controller
2	SMB-210 Cable	6	External Ethernet Hub or Common LAN
3	SMB-200 Cable (1 m)	7	Ethernet Port
4	NI PXI-665*x* Module		

Figure 3-21. Two Chassis with Synchronized Acquisition and Separate CPUs (Embedded PXI Controllers)

Connecting to Ethernet

All chassis should be connected to a common LAN or Ethernet hub.

Connecting the NI PXI-665*x*

1. Use a 1 m SMB-200 cable to connect PFI 0 to PFI 1 on the NI PXI-665*x* in Slot 2 of Chassis #1. This cable connects two external pins on the same module.
2. Use an SMB-210 cable to connect PFI 2 and PFI 3 on the NI PXI-665*x* in Slot 2 of Chassis #1 to PFI 0 and PFI 1 of Chassis #2, respectively.

Connecting the NI PXI-4472 Modules

Connect the external sine signal to a single input channel on one NI PXI-4472 in each chassis. The same signal should be fed to each chassis to measure the phase mismatch and verify that the synchronization is within specifications.

Configuring and Running the Software Example

1. After the hardware has been configured, launch MAX. Verify that all the NI PXI-4472 modules in the first chassis are recognized and properly functioning and that the NI PXI-665*x* modules are recognized. Repeat this procedure for the second chassis.
2. Check that each controller can recognize and communicate with the others over the Ethernet.

Running the Multi-CPU DSA Examples Programs

Three LabVIEW example programs are included to illustrate synchronizing NI PXI-4472 devices across two PXI chassis, each equipped with its own PXI controller or MXI-3 connection to a PC. The three programs are:

- The **niSync_DSA Example 2 or 3 Dual Chassis Dual Controller [Master]** VI
- The **niSync_DSA Example 2 or 3 Dual Chassis Dual Controller [Slave]** VI
- The **niSync_DSA Example 2 or 3 Dual Chassis Dual Controller [Main]** VI

The master and slave examples must run on the CPU controlling the appropriate chassis. If desired, these programs may be deployed on PXI controllers running LabVIEW RT. The main program may be run on the same computer with the master program. Alternately, it can be used on a third machine that does not actually contain any PXI hardware. If you are deploying the master and slave VIs on RT controllers, NI recommends executing the main VI on a third machine running LabVIEW for Windows.

The main VI is the only program of the three that requires direct interaction from the user. All parameters for acquisitions on the master and the slave are entered on the front panel of the main VI. When the main VI runs, these settings are transferred to the slave and master VIs through VI Server calls by reference. After the master and slave are launched by the main VI, they use a network software handshaking technique to maintain a deterministic order of execution between the steps of hardware synchronization in each VI. A fourth VI, also called by reference from the master and slave, serves as the clearinghouse for communication between the master and slave computers. This VI, the **niSync_UTIL Network Com Engine** VI, resides on the same machine with the main VI. The master and slave use VI Server calls by reference to pass information to and from the engine. These examples are configured to use exactly one master and one slave chassis. However, the network communication scheme illustrated in these examples can support multiple slaves for systems requiring more than two chassis.

As outlined above, the network communication scheme is built around VI Server calls by reference. This requires that the VI Server security settings on the computers running the master, slave, and main VIs must be configured to allow remote calls by reference.

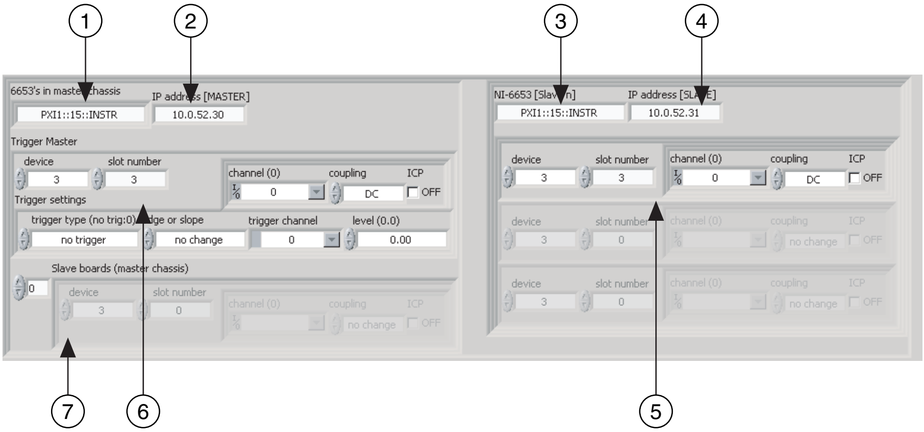
For each machine running LabVIEW for Windows, complete the following steps to verify that the VI Server settings are configured correctly:

1. Go to **Tools»Options»VI Server: Configuration**. Make sure that all six check boxes (TCP/IP, ActiveX, and so on) are checked. This enables VI Server control via TCP/IP.
2. Go to **Tools»Options»VI Server: TCP/IP Access**.
   a. Enter a wildcard (`*`) in the text box.
   b. Click the **Allow Access** button.
   c. Click **Add** and **OK**. This allows any computer to communicate with the local machine via VI Server.
3. Go to **Tools»Options»VI Server: Exported VIs**.
   a. Enter a wildcard (`*`) in the text box.
   b. Click the **Allow Access** button.
   c. Click **Add** and **OK**. This exports all VIs on the local machine for control via VI Server.

If you are running the master and slave VIs under LabVIEW RT, these same settings are accessed from a Windows host machine under **Tools»Network *x.x.x.x* Options**, where ***x.x.x.x*** is the IP address of the RT target.

After you verify the VI Server settings, you must open LabVIEW on the machines running the master and slave VIs. (This step is unnecessary if the master and slave are running on RT targets.)

Although the multichassis example setup uses distinct master, slave, and main VIs, the user is required to interact directly with the main VI only. VI Server handles the rest. Figure 3-22 shows the front panel for the **niSync_DSA Example 2 or 3 Dual Chassis Dual Controller [Main]** VI.



6653's in master chassis: PXI1::15::INSTR
IP address [MASTER]: 10.0.52.30

Trigger Master
device: 3 slot number: 3
channel (0): 0 coupling: DC ICP: □ OFF

Trigger settings
trigger type (no trig:0): no trigger edge or slope: no change trigger channel: 0 level (0.0): 0.00

Slave boards (master chassis): 0
device: 3 slot number: 0
channel (0): coupling: no change ICP: □ OFF

NI-6653 [Slav n]: PXI1::15::INSTR
IP address [SLAVE]: 10.0.52.31

device: 3 slot number: 3 channel (0): 0 coupling: DC ICP: □ OFF
device: 3 slot number: 0 channel (0): coupling: no change ICP: □ OFF
device: 3 slot number: 0 channel (0): coupling: no change ICP: □ OFF

1	NI PXI-665*x* Masters	5	NI PXI-4472 Trigger Slaves in Slave Chassis
2	IP Address for CPU Running Master VI	6	NI PXI-4472 Trigger Master
3	NI PXI-665*x* in Slave Chassis	7	NI PXI-4472 Trigger Slaves in Master
4	IP Address for CPU Running Slave VI		

Figure 3-22. Front Panel for the niSync_DSA Example 2 or 3 Dual Chassis Dual Controller [Main] VI

- **NI PXI-665*x* Masters**—Enter the VISA Resource Name for the NI PXI-665*x* in the master chassis. The control labeled **6653s in master chassis** is actually an array. However, in this particular screen the array index control is not shown. The reason for this is that the two-chassis configuration implies only a single NI PXI-665*x* in the master chassis.

- **IP Address for CPU Running Master VI**—Enter the IP address for the master CPU here. IP information is needed to establish remote VI Server communication between the main, master, and slave VIs.
- **NI PXI-665*x* in Slave Chassis**—This control is the VISA Resource Name for the NI PXI-665*x* in the slave chassis.
- **IP Address for CPU Running Slave VI**—Enter the IP address for the slave CPU here. IP information is needed to establish remote VI Server communication between the main, master, and slave VIs.
- **NI PXI-4472 Trigger Slaves in Slave Chassis**—This is an array describing each NI PXI-4472 in the slave chassis.
- **NI PXI-4472 Trigger Master**—A single NI PXI-4472 in is responsible for issuing the acquisition trigger signal to start the data acquisition on every NI PXI-4472 in the system. This board must reside in the master chassis. The trigger master can use any of its three trigger modes (software trigger, external analog trigger, external digital trigger on **EXT Trig** pin).
- **NI PXI-4472 Trigger Slaves in Master**—This is an array of clusters with one element per board. All NI PXI-4472 devices in the master chassis except the trigger master are described here. The trigger master is addressed in the **NI PXI-4472 Trigger Master** control.

After filling in the **System Settings** cluster, you should set the desired sampling rate and run the example. To test the quality of the synchronization between NI PXI-4472 devices, connect a common 1 kHz sine tone to the first input channel on the trigger master and the first input channel on the first trigger slave. A phase mismatch result of less than 0.01° indicates excellent synchronization.

4

Synchronizing Multiple NI PXI-5112 Modules

NI PXI-5112 Theory of Operation

The NI PXI-5112 100 MHz, 100 MS/s digitizer can use a phase-locked loop to synchronize its 100 MHz sample clock to a 10 MHz reference clock such as PXI_Clk10. The high-precision oscillator on the NI PXI-665*x* can supply this reference frequency to the PXI backplane.

Synchronizing multiple NI PXI-5112 modules requires the following PXI timing and triggering features:

- **PXI_Clk10**—This is a 10 MHz clock with at least 100 ppm accuracy. It is independently distributed to each PXI peripheral slot through equal-length traces, with a skew of less than 1 ns between slots. Multiple devices can use this common timebase for synchronization, so each NI PXI-5112 can phase lock to a clock shared by the entire system.
- **PXI Trigger Bus**—This bus features eight bidirectional lines that link all PXI slots, providing interdevice synchronization and communication. The skew from slot to slot is less than 10 ns in a chassis with eight or fewer slots.
- **PXI Star Trigger**—This special trigger slot provides an independent dedicated bidirectional line for each of up to 13 peripheral slots on a single backplane. All lines are matched in length, which provides a low slot-to-slot skew of less than 1 ns. A star trigger controller plugged into this slot can route triggers and clocks among peripheral slots.

To synchronize multiple NI PXI-5112 modules, your system must be configured as follows:

- All NI PXI-5112 devices must share a common timebase. The NI PXI-5112 modules can achieve this by phase locking all NI PXI-5112 modules in the system to the PXI_Clk10.
- All NI PXI-5112 devices must use a synchronization pulse (sync pulse) to align the clock dividers on each NI PXI-5112. In a single-chassis system, a master NI PXI-5112 device in Slot 2 of the

PXI chassis typically generates the sync pulse. For a multichassis system, this example uses the NI PXI-665*x* to generate this pulse.

- To designate the start of an acquisition, a start trigger signal must be distributed to all devices in the system.

Note When synchronizing multiple NI PXI-5112 devices, you must route the asynchronous trigger and synchronization pulse on different trigger lines.

Using the NI PXI-665*x* to Route Synchronization Signals Between Multiple Chassis

Example Overview

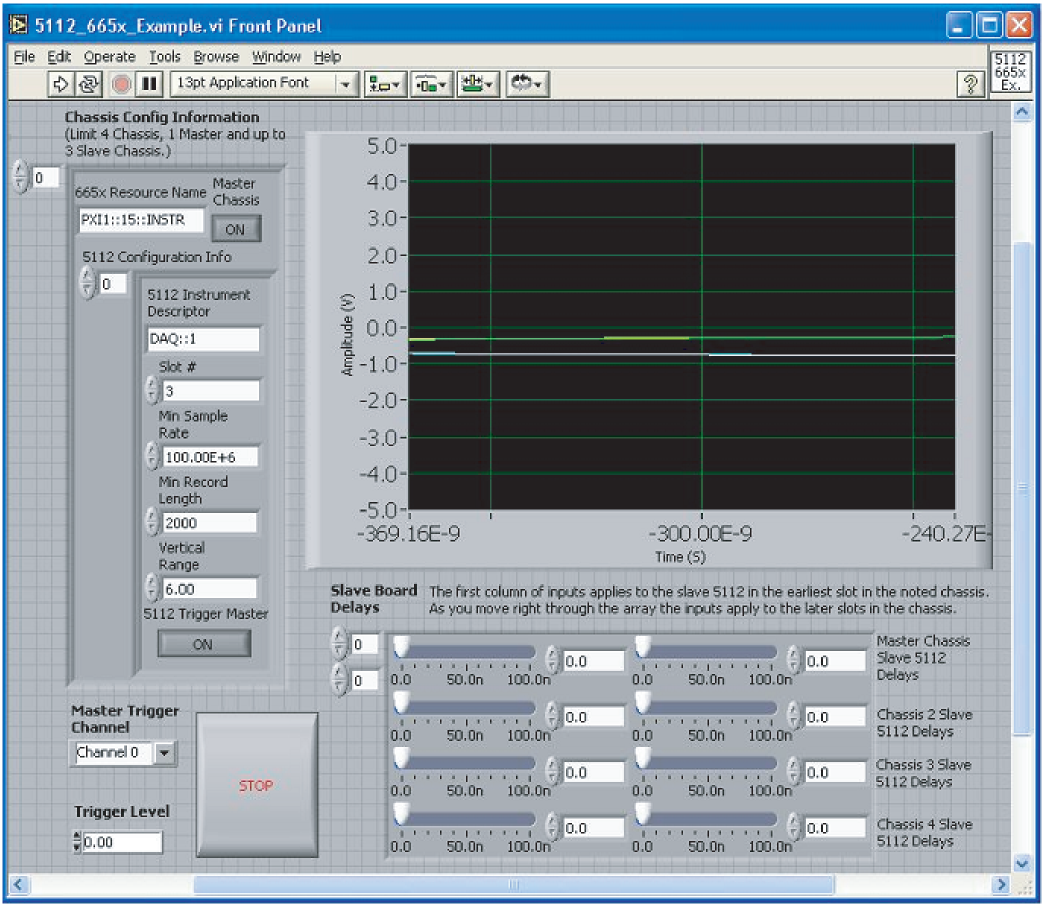
The following section discusses the **5112 665x Example** VI. This example synchronizes up to four chassis containing NI PXI-5112 modules. The NI PXI-665*x* shares the synchronization signals for the NI PXI-5112 described in the previous section. In this example, the high-precision oscillator in the NI PXI-665*x* replaces the PXI_Clk10 backplane clock in the master chassis, as well as the slave chassis. The NI PXI-665*x* master module then uses its software trigger to send the NI PXI-5112 sync pulse to all devices in the system. Finally, the master NI PXI-5112 module triggers from one of its front panel input channels and sends this trigger to the PXI backplane. The NI PXI-665*x* modules then route this signal from the master chassis to all slave chassis and slave NI PXI-5112 modules.

Figure 4-1 shows the front panel for this example VI. The left side of the panel contains a **Chassis Config Information** array, where each element specifies the configuration information for an individual chassis. Within each element, you can specify the acquisition characteristics as well as the synchronization characteristics for a chassis.

The graph on the right displays the various signals acquired from each NI PXI-5112 module. The **Stop** button at the bottom of the front panel stops VI execution. The remaining front panel controls specify the remaining trigger parameters, such as:

- **Master Trigger Channel**—the analog input channel that triggers the master NI PXI-5112 module
- **Trigger Level**—the threshold level of the analog trigger
- **Slave Board Delay**—the phase delays for each slave NI PXI-5112 module in the master and slave chassis

The **Slave Board Delays** control is a two-dimensional array. The first dimension controls which chassis the delays apply to, and the second dimension of the array controls which particular NI PXI-5112 module delay is being configured. For more information about the NI PXI-5112 modules and phase compensation, refer to the *NI-Scope User Manual*.



Chassis Config Information
(Limit 4 Chassis, 1 Master and up to 3 Slave Chassis.)

0

665x Resource Name: PXI1::15::INSTR

Master Chassis: ON

5112 Configuration Info: 0

5112 Instrument Descriptor: DAQ::1

Slot #: 3

Min Sample Rate: 100.00E+6

Min Record Length: 2000

Vertical Range: 6.00

5112 Trigger Master: ON

Master Trigger Channel: Channel 0

Trigger Level: 0.00

Slave Board Delays The first column of inputs applies to the slave 5112 in the earliest slot in the noted chassis. As you move right through the array the inputs apply to the later slots in the chassis.

0
0

0.0	0.0	Master Chassis Slave 5112 Delays
0.0	0.0	Chassis 2 Slave 5112 Delays
0.0	0.0	Chassis 3 Slave 5112 Delays
0.0	0.0	Chassis 4 Slave 5112 Delays

0.0 50.0n 100.0n

Figure 4-1. 5112 665x Example VI Front Panel

What You Need to Get Started

To get started and set up this example, you need the following hardware and software:

❑ PC and remote chassis links (such as MXI-3) or PXI Embedded Controller and Remote chassis link

❑ PXI chassis (up to four chassis may be used)

❑ Master NI PXI-665*x*

❑ Slave NI PXI-665*x* (one per chassis)

❑ NI PXI-5112 modules

❑ SMB cables

❑ BNC cables

❑ Clock splitter (if using more than two chassis)

Setting Up a Multichassis PXI System

The following section describes how to set up a multichassis system and connect the MXI-3 devices:

1. Refer to your remote link (such as MXI-3) documentation for detailed instructions on installation of PCI and PXI remote link modules.
2. If you are using a PC, install a PCI remote link device in your PC. If you are using an embedded controller, install this controller into the first PXI chassis.
3. For each chassis that needs to link to a secondary chassis, you need to install a PXI remote link module in some slot (other than Slot 2). For best performance, if you have a PXI chassis that contains a PCI-PCI bridge, it is best to install the link to secondary chassis in the first bus segment of the PXI chassis.
4. For each secondary chassis, install a PXI remote link module in Slot 1, or the system controller slot, of the PXI chassis.
5. Connect your remote link cable from each PCI or PXI remote link module to its corresponding PXI remote link module on the secondary side.

For additional help, refer to the documentation for the remote links that you are using in your system. If it is a National Instruments remote link such as MXI-3, you can download this documentation from `ni.com/manuals`.

Connecting the NI PXI-665*x* Devices

The following section describes how to properly connect the trigger and synchronization signals between your master and slave NI PXI-665*x* modules.

1. For each chassis in the multichassis system, install an NI PXI-665*x* module in Slot 2 of the chassis. This allows the NI PXI-665*x* module to replace the PXI_Clk10 reference clock with the onboard high-precision oscillator, and allows this module control of the PXI Star Trigger lines.
2. Use an SMB cable to connect Clk Out of the master NI PXI-665*x* module to the Clk In of a slave NI PXI-665*x* module if using a single secondary chassis, or to a clock splitter if using more than one secondary chassis. If using more than one secondary chassis, connect the output from the clock splitter to the Clk In of each slave NI PXI-665x device.
3. Use SMB cables to connect the outputs of the master NI PXI-665*x* module that will carry the sync pulse to each secondary chassis. For this example, the sync pulse is routed out of the even-numbered front panel PFI outputs, PFI 0, PFI 2, and PFI 4. Connect one front panel PFI output for each slave chassis you are supporting to the front panel PFI 0 of each slave NI PXI-665*x* module. As previously noted, you can support up to three slave chassis for a total of four chassis with this example.
4. Use SMB cables to connect the outputs of the master NI PXI-665*x* module that will carry the start trigger to each secondary chassis. For this example, the start trigger is routed out of the odd-numbered front panel PFI outputs, PFI 1, PFI 3, and PFI 5. Connect one front panel PFI output for each slave chassis you are supporting to the front panel PFI 1 of the slave NI PXI-665*x* module for the slave chassis.

Connecting the NI PXI-5112 Devices

The following section describes how to properly connect the signals for your data acquisition to the NI PXI-5112 modules in your system.

1. Connect the signals you are measuring to each NI PXI-5112 module in your system. Each NI PXI-5112 module has two front panel BNC inputs.
2. Be sure to connect a signal to the front panel BNC for the NI PXI-5112 that you designate as the "master" module, and be sure this is the correct input channel that you designate as the "Trigger Channel" in the example VI.

Configuring and Running the Software Example

The following sections describe how to configure and run the **5112 665x Example** VI.

1. Set the configuration information for each chassis. Each **Chassis Config Info** array element describes the configuration parameters for each module in one chassis of the multichassis system. Scroll through the array to enter the information for each chassis, up to four. Designating information in any array element beyond the fourth in the array results in an error, as this example only supports up to four chassis.
    a. Type in the NI PXI-665*x* resource name for each chassis (for example, `PXI1::15::INSTR`). An NI PXI-665*x* module should be installed in Slot 2 of each chassis. You can find these resource names using Measurement & Automation Explorer (MAX).
    b. Be sure that one chassis in your array is configured as the master chassis by clicking the **Master Chassis** button. One and only one chassis must be configured as the master in the array, or the example generates an error. The chassis configured as the master must contain the master NI PXI-5112 module.
    c. Each element of the **Chassis Config Info** array contains an array to hold the information about the NI PXI-5112 devices in that chassis. Populate the instrument descriptor for each NI PXI-5112 module in the chassis (for example, one instrument descriptor in the first chassis may be noted as `DAQ::1`). Use MAX to determine the instrument descriptors for all your NI PXI-5112 modules.
    d. For each NI PXI-5112 module you specify in a chassis, you must also specify the slot number where it resides in the chassis, minimum sample length, minimum record length, vertical range, and whether the device is the NI PXI-5112 trigger master. The

NI PXI-5112 trigger master is configured to output the start trigger when a specific signal level is detected on one of its input channels. There must be one and only one NI PXI-5112 trigger master in your multichassis PXI system. All parameters other than **Slot #** and **5112 Trigger Master** are NI-Scope parameters. For more information about these parameters, refer to the documentation for the NI-Scope software.

e. The last parameters you need to configure are the trigger parameters for the Master PXI-5112 module and the delay parameters for the slave NI PXI-5112 modules. The trigger channel and trigger level denote the analog input channel on the master NI PXI-5112 device that initiates the acquisition for the NI PXI-5112 modules. The trigger level specifies the analog voltage that trips the trigger, causing the master NI PXI-5112 device to send a start trigger to all slave NI PXI-5112 devices in the system. The slave board delays for each slave NI PXI-5112 module allow you to adjust the phase of the acquired signals to account for any cable delays or signal propagation delays in your system.

f. After you make all the signal connections and configure all the devices according to the above steps, you can run the VI. The stop button in the lower left of the front panel allows you to safely stop VI execution.

Example Functionality

This section describes each subVI of the **5112 665x Example** VI, including the functionality of the NI-Sync VIs. A significant amount of NI-Scope programming in this example is not discussed in this manual. Refer to the NI-Scope documentation for more information about programming with NI-Scope.

Initialization and Error Checking

This section describes error checking included in this example to verify that configuration information is correct. This section also describes opening references to and initializing devices in the system. The two subVIs covered in this section are the **5112 665x Check** VI, which verifies that the user has properly configured the example, and the **5112 665x Init** VI, which opens sessions to all devices in the system.

The **5112 665x Check** VI verifies that the user entered the appropriate configuration information for the system. The first configuration rule verified is that there are only four chassis specified in the system. If the user enters information for more than four chassis, this VI generates an error, and the rest of the example VI does not execute. This VI also verifies that the number of masters in the system is correct. It checks that there is at least one and only one master chassis for the system, and at least one and only one master NI PXI-5112 module in the master chassis.

The **5112 665x Init** VI walks through each device in the system and opens a session to this device. This VI calls on the NI-Scope driver to open sessions to the NI PXI-5112 devices. This VI uses the **niSync Initialize** VI shown in Figure 4-2 to open sessions to the NI PXI-665*x* modules in the system. This example specifies that the NI PXI-665*x* modules are reset on initialization. This action clears any previous routes set on the board.

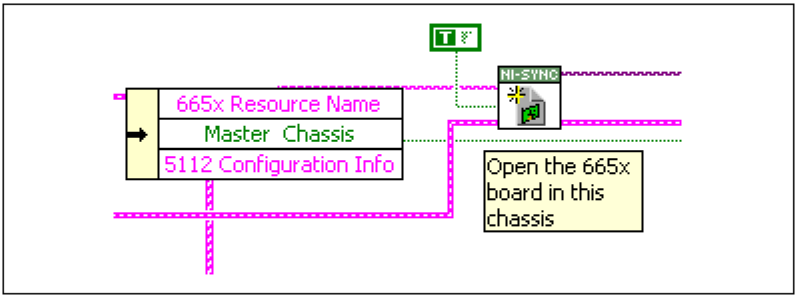


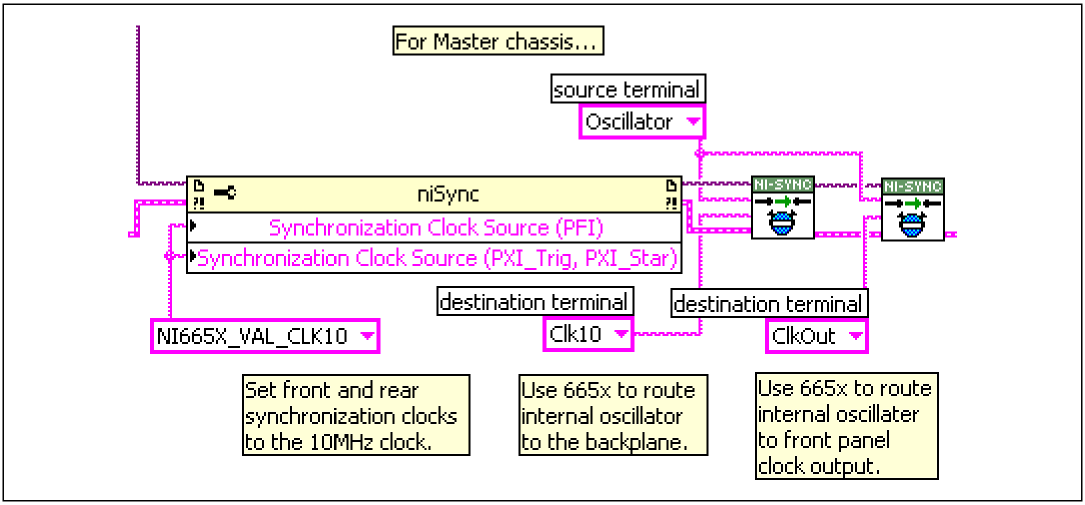
Figure 4-2. niSync Initialize VI Used to Open Sessions to NI PXI-665*x* Devices

If the NI-Scope driver or the NI-Sync driver cannot open a session to their device, this VI passes along the error from their respective drivers, and the example stops execution of any subsequent subVIs.

Configuring the 10 MHz Timebase

This section discusses the configuration of the shared 10 MHz timebase in the system. For the purpose of this example, the master NI PXI-665*x* module in the system uses its high-precision oscillator to replace the PXI_Clk10 signal in each chassis. The clock from the high-precision oscillator is routed to the PXI backplane and also out the front panel Clk Out SMB output to be shared, usually via a clock splitter, with the other chassis in the system. Each NI PXI-665*x* in a slave chassis is configured to input this high-precision oscillator signal into its Clk In front panel SMB input, and route this clock to replace the chassis-supplied PXI_Clk10 signal. This allows a single reference clock to be shared with all devices in the system.

The section of code shown in Figure 4-3 is the clock configuration for the master chassis. The synchronization clock for both the front panel and backplane are set to be the PXI_Clk10 using a property node. Next, the **niSync Connect Clock Terminals** VI is called to route the onboard oscillator to PXI_Clk10 in the master chassis. Finally, the **niSync Connect Clock Terminals** VI is called again to route the same oscillator signal to the front panel Clk Out SMB.



For Master chassis…

source terminal: Oscillator

niSync
Synchronization Clock Source (PFI)
Synchronization Clock Source (PXI_Trig, PXI_Star)

NI665X_VAL_CLK10

destination terminal: Clk10

destination terminal: ClkOut

Set front and rear synchronization clocks to the 10MHz clock.

Use 665x to route internal oscillator to the backplane.

Use 665x to route internal oscillater to front panel clock output.

Figure 4-3. NI-Sync VI Calls to Configure the Master NI PXI-665*x* Clock Outputs

The section of code shown in Figure 4-4 is the clock configuration for the slave chassis. The slave NI PXI-665*x* devices are configured to receive a clock at their Clk In SMB and route this clock to replace PXI_Clk10 in the chassis. The **niSync Connect Clock Terminals** VI routes the signal on ClkIn to the destination PXI_Clk10.

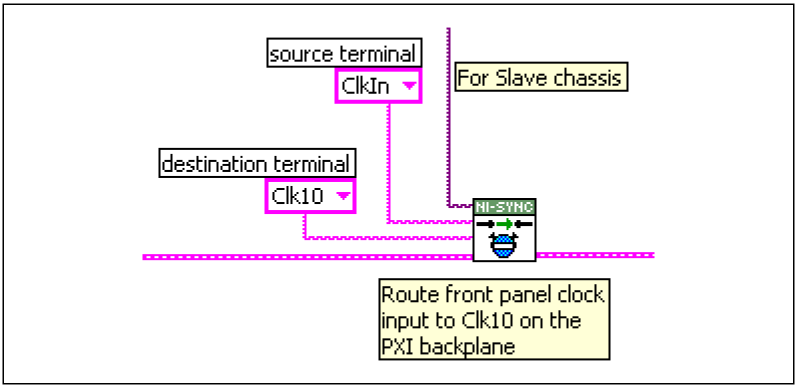
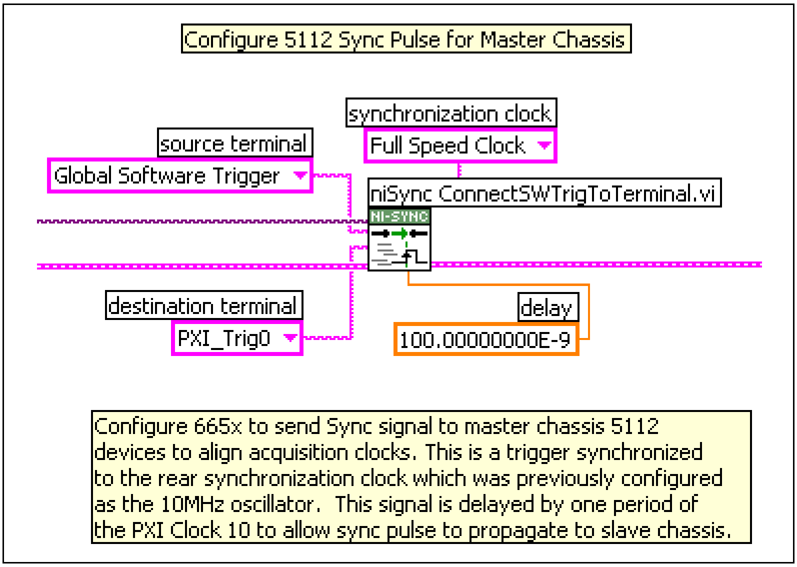


Figure 4-4. Clk10 Routing for Slave NI PXI-665*x* Devices

Routing the Necessary Synchronization Signals

This section discusses the NI PXI-665*x* trigger routing in this example. The NI PXI-665*x* routes two synchronization signals. The first signal is the NI PXI-5112 sync pulse. The master NI PXI-665*x* generates this pulse as a software trigger and passes it to all NI PXI-5112 devices in all chassis. The second signal is the NI PXI-5112 start trigger. The master NI PXI-5112 device generates this trigger and routes it through the NI PXI-665*x* modules to all slave NI PXI-5112 devices in the system.

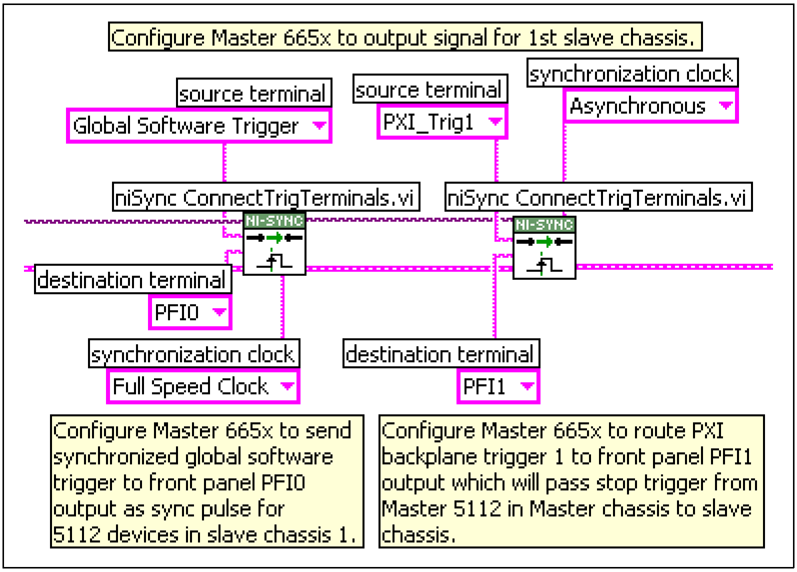
The **665x Route** VI first determines whether the chassis it is configuring is a master or slave chassis. Once it determines the appropriate configuration steps, it makes the routes for the chassis. Figure 4-5 shows the first configuration step if the chassis is a master chassis. This step involves setting up the synchronization signals to be sent to the master chassis. In this case, it is only the SW trigger used by the sync pulse. The **niSync Connect Software Trigger** VI is called to route the SW trigger to PXITrig0, where the master NI PXI-5112 device expects to receive its sync pulse. This pulse is specified to be synchronized to the full speed clock, which was previously assigned as PXI_Clk10.



Configure 665x to send Sync signal to master chassis 5112 devices to align acquisition clocks. This is a trigger synchronized to the rear synchronization clock which was previously configured as the 10MHz oscillator. This signal is delayed by one period of the PXI Clock 10 to allow sync pulse to propagate to slave chassis.

Figure 4-5. Master NI PXI-665*x* Configures Software Trigger Routes for Master Chassis

The next few steps to configure the routes in the master NI PXI-665*x* involve setting up the routes to output the sync pulse and start trigger for each slave chassis. Figure 4-6 shows the code to create these routes for the first slave chassis. These same calls are repeated for each slave chassis, although the specific PFI outputs used change for each chassis. The **niSync Connect Software Trigger to Terminal** VI routes the SW trigger used for the sync pulse to the first slave chassis. The **niSync Connect Trigger to Terminal** VI routes the start trigger generated by the NI PXI-5112 on the backplane to the front panel PFI0 output to be sent to the first slave chassis. This example does not resynchronize the start trigger; therefore, the synchronization clock is left **Asynchronous**.



Configure Master 665x to output signal for 1st slave chassis.

source terminal
Global Software Trigger

source terminal
PXI_Trig1

synchronization clock
Asynchronous

niSync ConnectTrigTerminals.vi

niSync ConnectTrigTerminals.vi

destination terminal
PFI0

synchronization clock
Full Speed Clock

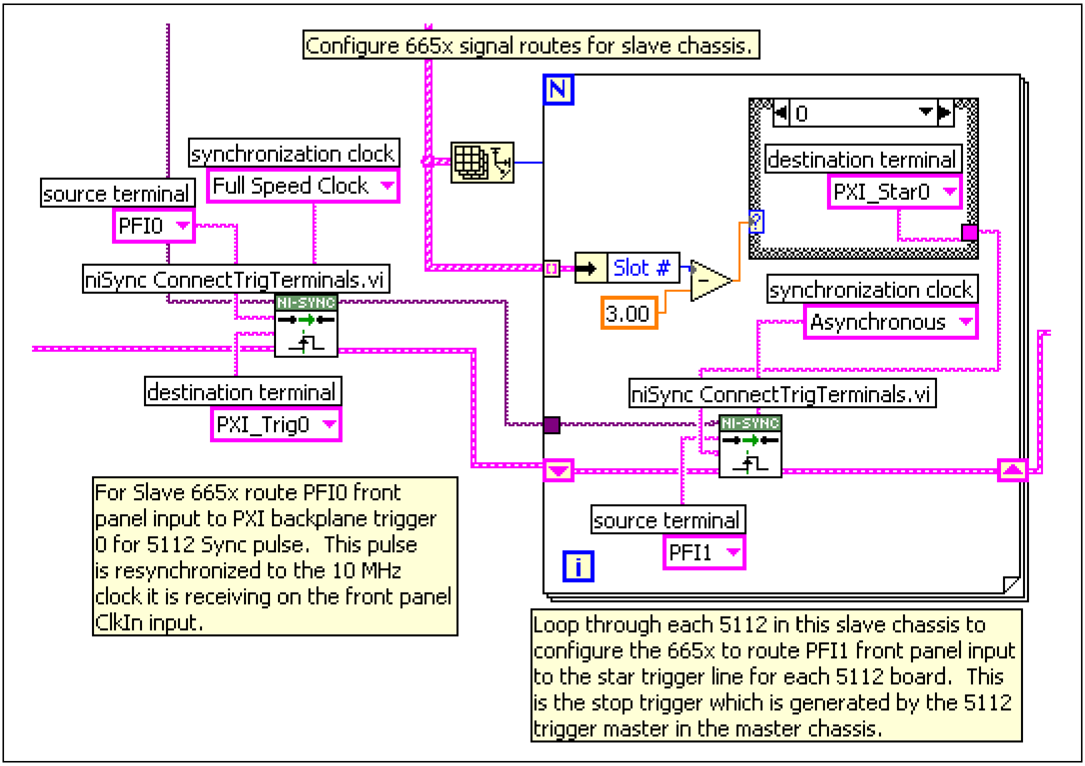
destination terminal
PFI1

Configure Master 665x to send synchronized global software trigger to front panel PFI0 output as sync pulse for 5112 devices in slave chassis 1.

Configure Master 665x to route PXI backplane trigger 1 to front panel PFI1 output which will pass stop trigger from Master 5112 in Master chassis to slave chassis.

Figure 4-6. Master NI PXI-665*x* Configures Software Trigger Routes and Start Trigger Routes for Slave Chassis

The final routing steps are for the NI PXI-665*x* modules in the slave chassis. Each module must input the sync pulse on the PFI0 SMB input and route it to PXI_Trig0. Then each slave NI PXI-665*x* must input the start trigger on the PFI1 SMB input and route this signal to any star trigger line where a device resides. The first step is to call the **niSync Connect Trigger Terminals** VI with a source of **PFI0** and a destination of **PXI_Trig0**. The sync pulse is resynchronized to the synchronization clock, which in our example is the PXI_Clk10. The same VI is called for the next route, but this time with the source and destination of **PFI1** and the appropriate PXI Star Trigger, respectively. This route passes the trigger asynchronously as specified in the call to the VI.



For Slave 665x route PFI0 front panel input to PXI backplane trigger 0 for 5112 Sync pulse. This pulse is resynchronized to the 10 MHz clock it is receiving on the front panel ClkIn input.

Loop through each 5112 in this slave chassis to configure the 665x to route PFI1 front panel input to the star trigger line for each 5112 board. This is the stop trigger which is generated by the 5112 trigger master in the master chassis.

Figure 4-7. Slave NI PXI-665*x* Configures Routing for Sync Pulse and Start Trigger Inputs to Slave Chassis

Configuring the NI PXI-5112 Synchronization and Acquisitions

The next two VIs called in this example deal primarily with setting up the acquisitions on the NI PXI-5112 devices. Refer to the NI-Scope documentation for more detailed information about configuring and using the NI-Scope software.

The first VI, the **5112 Sync Config** VI, sets up all synchronization parameters for the master and slave NI PXI-5112 devices in the system. For this example, all NI PXI-5112 devices receive an external sync pulse on PXI Trigger line 0. The master NI PXI-5112 device generates a start trigger based on receiving a signal of the appropriate level on the channel designated as the Trigger Channel. The master NI PXI-5112 module outputs this trigger onto the PXI Trigger bus, where this trigger is then routed to the rest of the chassis in the system. All slave NI PXI-5112 devices receive an external start trigger. If these slave NI PXI-5112 modules reside as slaves in the master chassis, they receive this trigger on PXI Trigger line 1 on the backplane. If these slave NI PXI-5112 modules reside as slaves in a slave chassis, they receive their start trigger on the PXI Star Trigger for their particular slot.

The second VI, the **5112 Acq Config** VI, sets all acquisition parameters for the NI PXI-5112 devices in the system. This VI sets parameters such as the acquisition rate and sample size. To simplify this example, it abstracts away many of the configuration options the NI-Scope software provides. As previously noted, more detailed information about these capabilities is beyond the scope of this manual and is in the user documentation for your NI-Scope software.

Beginning the Acquisition

The next section discusses initiating the actual acquisition. Two VIs in this example handle this functionality. The first VI, the **5112 Start Acq** VI, initiates all NI PXI-5112 devices in the system. For the master NI PXI-5112, initiating the acquisition means it then waits for the appropriate trigger signal to be input on the specified trigger channel. For slave NI PXI-5112 modules, initiating the acquisition means they wait for a start trigger on the appropriate PXI Trigger Bus or PXI Star Trigger lines.

The second VI involved in beginning the acquisition, the **665x Send Sync** VI, involves firing the NI PXI-665*x* software trigger that all NI PXI-5112 devices utilize for their sync pulse. Figure 4-8 shows the NI-Sync code that fires the software trigger previously configured in the example by calling the **665x Route** VI.

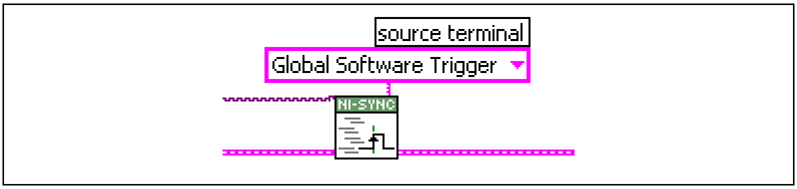
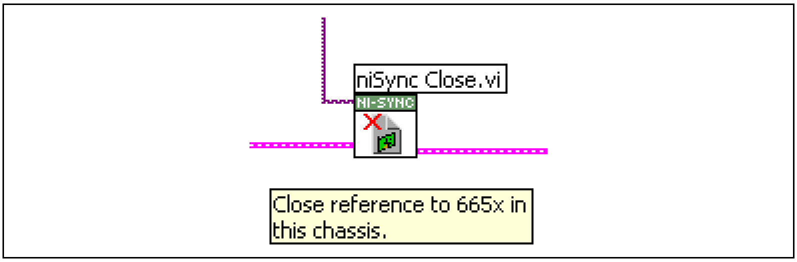


Figure 4-8. Firing the Global Software Trigger on the Master NI PXI-665*x*

Closing and Error Checking

The final functionality in this example is closing references to devices and reporting any errors that occurred during execution. **5112 665x Close** VI closes all references to the NI PXI-5112 and NI PXI-665*x* devices that were opened during initialization. Figure 4-9 shows the call to the **niSync Close** VI, which is called for each reference to an NI PXI-665*x* device opened.



Close reference to 665x in this chassis.

Figure 4-9. Closing Each Reference to an NI PXI-665*x*

The final VI is a standard LabVIEW VI that reports any errors detected during execution. If an error is encountered during execution, no further calls to the driver are executed, and the error location and message is passed to the **Simple Error Handler** VI and reported with a pop-up dialog.

5

Synchronizing Multiple NI PXI-6115 Modules

Note There are two NI PXI-6115 examples included with the NI-Sync software. The example described here uses a single host CPU. The example **6115_665*x*_Network_Example** VI uses the same NI-Sync and NI-DAQ functionality, but adds LabVIEW support for the remote VI calls and message passing that is necessary to make the example run on several different host CPUs coordinated via an Ethernet connection. This functionality is beyond the scope of this manual.

NI PXI-6115 Theory of Operation

The NI PXI-6115 10 MS/s, 12-bit, multifunction DAQ module can use an externally supplied scan clock to synchronize its acquisition. You can use the Direct Digital Synthesis (DDS) clock on the NI PXI-665*x* as a source for this clock.

You can synchronize multiple NI PXI-6115 modules with the following PXI timing and synchronization features:

- **PXI Trigger Bus (RTSI)**—This bus features eight bidirectional lines that link all PXI slots, providing interdevice synchronization and communication. The skew from slot to slot is less than 10 ns in a typical 8-slot chassis.
- **PXI Star Trigger**—This special trigger slot provides an independent dedicated bidirectional line for each of up to 13 peripheral slots on a single backplane. All lines are matched in length, which provides a low slot-to-slot skew of less than 1 ns. A star trigger controller such as the NI PXI-665*x* can be plugged into the PXI star trigger controller slot to route triggers and clocks among peripheral slots.

To synchronize multiple NI PXI-6115 devices, the following must be true:

- All NI PXI-6115 devices must share a common scan clock. The NI PXI-6115 modules can achieve this by using an externally supplied clock. This clock should be distributed over the star trigger lines to minimize the skew between the scan clocks supplied to each module.
- All NI PXI-6115 devices must use a start trigger to signal the start of the acquisition. You can use the PXI Trigger Bus to send a digital trigger.

Using the NI PXI-665*x* to Route Synchronization Signals Between Multiple Chassis

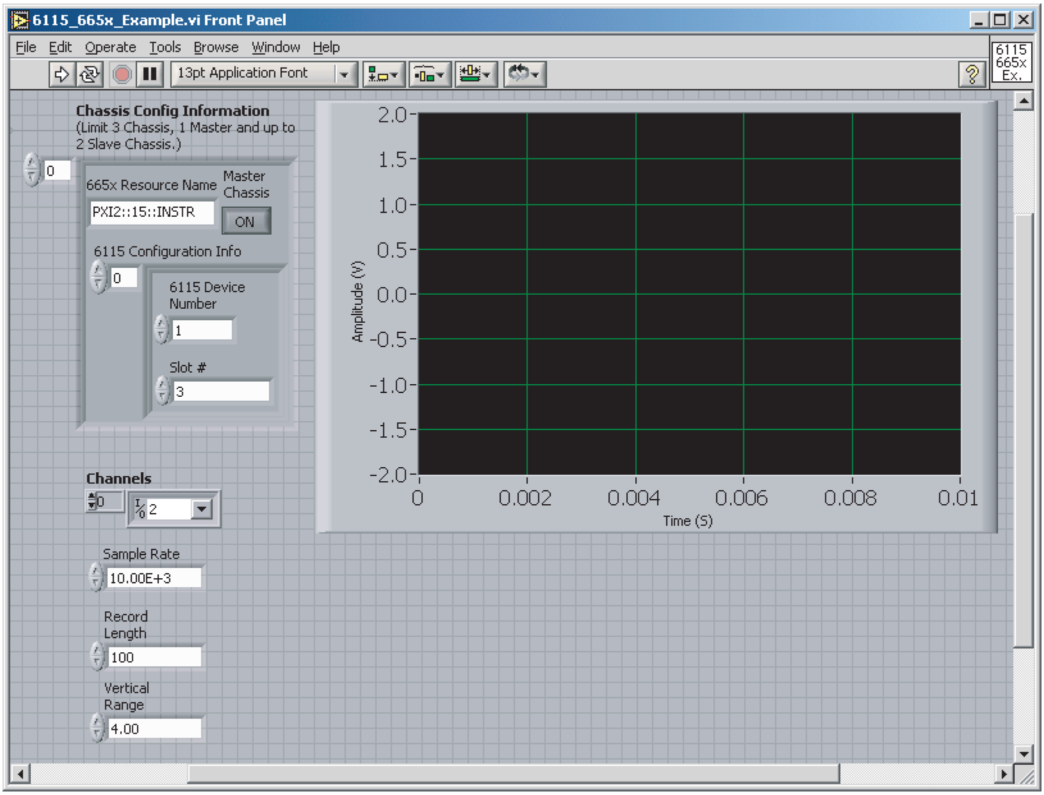
Example Overview

The following section discusses the example **6115 665x Example** VI. This example synchronizes up to three chassis of NI PXI-6115 modules. The NI PXI-665*x* boards share the synchronization signals for the NI PXI-6115 described in the previous section. In this example, the DDS Clock from the NI PXI-665*x* provides the scan clock to each NI PXI-6115 in the master chassis, as well as in the slave chassis. The NI PXI-665*x* master module then uses its software trigger to generate the NI PXI-6115 start trigger. The NI PXI-665*x* modules then route this signal from the master NI PXI-665*x* to all slave chassis and onto a PXI Trigger line where it can be received by all NI PXI-6115 devices.

Figure 5-1 shows the front panel for this example VI. The left side of the panel contains a **Chassis Config Information** array, where each element specifies the configuration information for an individual chassis. Within each element, you can specify the instrument descriptor for the NI PXI-665*x* in the chassis and the device number and slot of each NI PXI-6115 in the chassis.

The graph on the right displays the various signals acquired from each NI PXI-6115 module. The remaining front panel controls specify the details of the acquisition and apply to all NI PXI-6115 devices in the system. These parameters are the analog input channels from which to acquire, the sampling rate (Hertz), the record length (samples), and the vertical range (volts). The graph is scaled automatically to fit the acquisition parameters.

This example is a single-shot acquisition. Each time the VI runs, one record of data is captured and displayed. Refer to your NI-DAQ user documentation for more information about the NI PXI-6115 and continuous acquisition.



Chassis Config Information
(Limit 3 Chassis, 1 Master and up to 2 Slave Chassis.)

0

665x Resource Name: PXI2::15::INSTR

Master Chassis: ON

6115 Configuration Info: 0

6115 Device Number: 1

Slot #: 3

Channels: 0, 2

Sample Rate: 10.00E+3

Record Length: 100

Vertical Range: 4.00

Figure 5-1. 6115 665x Example VI Front Panel

What You Need to Get Started

You need the following hardware and software to get started and set up this example.

- ❑ PC and remote chassis links (such as MXI-3) or PXI embedded controller and remote chassis link
- ❑ Up to three PXI chassis
- ❑ Master NI PXI-665*x*
- ❑ Slave NI PXI-665*x* (one per chassis)

❑ NI PXI-6115 modules

❑ SMB cables

Setting Up the Multichassis PXI System

Complete the following steps to set up the multichassis system and connect the MXI-3 devices:

1. Install the PCI and PXI remote link modules. For detailed instructions, refer to your remote link (such as MXI-3) documentation.
2. If you are using a PC, install a PCI remote link device in your PC. If you are using an embedded controller, install this controller into the first PXI chassis.
3. For each chassis that needs to link to a secondary chassis, install a PXI remote link module in some slot other than Slot 2. For best performance, if you have a PXI chassis that contains a PCI-PCI bridge, install the link to secondary chassis in the first bus segment of the PXI chassis.
4. For each secondary chassis, install a PXI remote link module in Slot 1, or the system controller slot, of the PXI chassis.
5. Connect your remote link cable from each PCI or PXI remote link module to its corresponding PXI remote link module on the secondary side.

For additional help, refer to the documentation for the remote links you are using in your system. If you are using National Instruments remote links such as MXI-3, you can download the documentation from `ni.com/manuals`.

Connecting the NI PXI-665*x* Devices

Complete the following steps to properly connect the trigger and synchronization signals between your master and slave NI PXI-665*x* modules.

1. For each chassis in the multichassis system, install an NI PXI-665*x* module in Slot 2 of the chassis. This allows the NI PXI-665*x* module to access the PXI Star Trigger lines.
2. Use an SMB cable to connect PFI 1 of the master NI PXI-665*x* module to PFI 0 of the master NI PXI-665*x* module. This cable must be the same length as the other SMB cables in the system. This wrapback cable provides the master chassis with a DDS Clock that is in phase with the DDS Clock provided to the slave chassis.

3. Use an SMB cable to connect the outputs of the master NI PXI-665*x* module that carries the DDS Clock (scan clock) to each secondary chassis. For this example, the scan clock is routed out of the odd numbered front panel PFI outputs, PFI 1, PFI 3, and PFI 5. Connect one front panel PFI output for each slave chassis you will support to the front panel PFI 0 of the slave NI PXI-665*x* module for the slave chassis. As previously noted, you can support up to two secondary or slave chassis for a total of three chassis, including the master with this example. Notice that PFI 1wraps back the scan clock for the master chassis, as described above. Be sure that all cables that distribute the scan clock are of equal length.
4. Use SMB cables to connect the outputs of the master NI PXI-665*x* module that carry the start trigger to each secondary chassis. For this example, the start trigger is routed out the even numbered front panel PFI outputs, PFI 2 and PFI 4. Connect one front panel PFI output for each slave chassis you will support to the front panel PFI 1 of the slave NI PXI-665*x* module.

Connecting the NI PXI-6115 Devices

For information about connecting input signals to the NI PXI-6115 devices in the system, refer to the *NI PXI-6115 User Manual* included with the hardware. You also can download this manual from `ni.com/manuals`.

Configuring and Running the Software Example

Complete the following steps to configure and run the **6115 665x Example** VI.

1. Set the configuration information for each chassis. Each element of the **Chassis Config Info** array describes the configuration parameters for the modules in one chassis of the multichassis system. Scroll through the array to enter the information for each chassis, up to three chassis. Designating information in any array element beyond the third in the array results in an error, as this example supports up to only three chassis (that is, two slave chassis).
    a. Type in the NI PXI-665*x* resource name for each chassis (for example, `PXI1::15::INSTR`). An NI PXI-665*x* module should be installed in Slot 2 of each chassis. You can find these resource names using Measurement & Automation Explorer (MAX).
    b. Be sure that one chassis in your array is configured as the master chassis by clicking the **Master Chassis** button. There must be one and only one chassis configured as the master in the array or the example will generate an error.

c. Each element of the **Chassis Config Info** array contains an array to hold the information about the NI PXI-6115 devices in that chassis. Populate the instrument descriptor for each NI PXI-6115 module in the chassis (for example, you can describe one instrument descriptor in the first chassis as `DAQ::1`). Use MAX to determine the instrument descriptors for all NI PXI-6115 modules.

d. For each NI PXI-6115 module you specify in a chassis, you must also specify the slot number where it resides in the chassis.

e. Configure the acquisition parameters for all NI PXI-6115 modules in the system. These include the channels, sampling rate, record length, and vertical range. For more information about these parameters, refer to the NI-DAQ driver documentation.

f. After you make all your signal connections and configure all the boards according to the preceding steps, run the VI. The VI acquires one waveform from each channel of each NI PXI-6115 based on the acquisition parameters you specified. All waveforms are displayed on the front panel graph.

Example Functionality

This section discusses each **6115 665x Example** VI subVI, including the NI-Sync VIs. A significant amount of NI-DAQ programming in this example is not discussed in this manual. Refer to the NI-DAQ driver documentation more information on programming with this software.

Initialization and Error Checking

This section covers error checking included in this example to verify that configuration information is correct. This section also covers opening references to and initializing devices in the system. The two subVIs covered in this section are the **6115 665x Check** VI, which verifies the user has properly configured the example, and the **6115 665x Init** VI, which opens sessions to all devices in the system.

The **6115 665x Check** VI verifies that you have entered the appropriate configuration information for the system. This VI verifies that the number of master chassis in the system is correct and that you have specified at least one and only one master chassis for the system.

The **6115 665x Init** VI walks through each device in the system and opens a session to this device. This VI calls on the NI-DAQ driver to open sessions to the NI PXI-6115 devices. This VI also calls on the **niSync Initialize** VI**,** shown in Figure 5-2, to open sessions to the NI PXI-665*x* devices in the system. This example specifies that the NI PXI-665*x* modules be reset on initialization, which clears any previous routes set on the board.

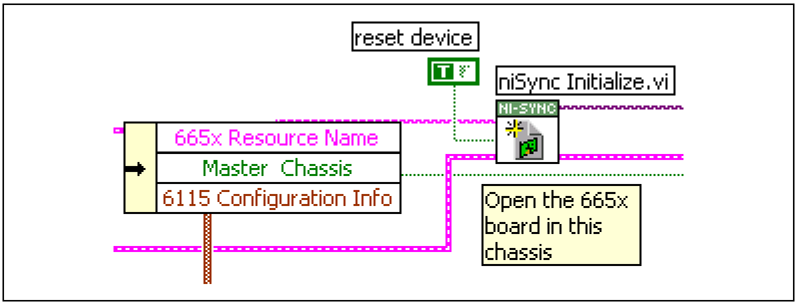


Figure 5-2. niSync Initialize VI Used to Open Sessions to NI PXI-665*x* Devices

If the NI-DAQ driver or the NI-Sync driver cannot open a session to a device, this VI generates an error, and the other example subVIs do not try to communicate with any device.

Configuring the NI PXI-665*x* Clocks

This section covers the configuration of the various system clocks on the NI PXI-665*x* boards in the system. The master and slave chassis NI PXI-665*x* boards must have a number of clock settings configured for the example to work properly.

The section of code in Figure 5-3 shows the clock configuration for the master chassis. The DDS frequency is set based on the user-specified sample clock rate using a property node. The synchronization clock for the front panel PFIs is set to be the DDS Clock, and the synchronization clock for the backplane trigger lines is set to be the PFI0 signal using a pair of property nodes. The PFI synchronization clock outputs the scan clock from the front panel. The backplane synchronization clock synchronizes the start trigger. The first clock divisor on the master NI PXI-665*x* is set to either 2 or 512 depending on whether the sample clock rate is greater than or less than 100 kHz. This setting is necessary because the start trigger needs to be at least 10 µs for the NI PXI-6115 devices to be guaranteed to receive it properly. The NI PXI-665*x* in the master chassis use a divided version of the master clock to create the greater than 10 µs pulse.

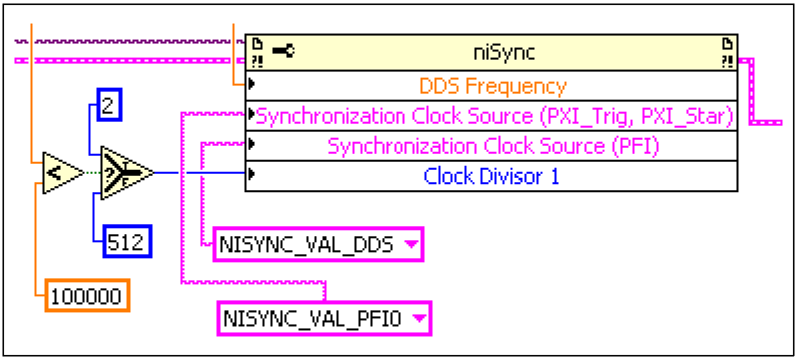

niSync
DDS Frequency
Synchronization Clock Source (PXI_Trig, PXI_Star)
Synchronization Clock Source (PFI)
Clock Divisor 1
2
512
100000
NISYNC_VAL_DDS
NISYNC_VAL_PFI0

Figure 5-3. NI-Sync VI Calls to Configure the Master NI PXI-665*x* Clock Outputs

The section of code in Figure 5-4 shows the clock configuration for the slave chassis. The slave NI PXI-665*x* devices use the signal connected to PFI0 as the synchronization clock source for the board. This PFI0 signal is the scan clock, which is generated on the master NI PXI-665*x* by the DDS and routed to the slave NI PXI-665*x* with an SMB cable. The rear synchronization clock synchronizes the start trigger signal on the PXI backplane in each slave chassis.

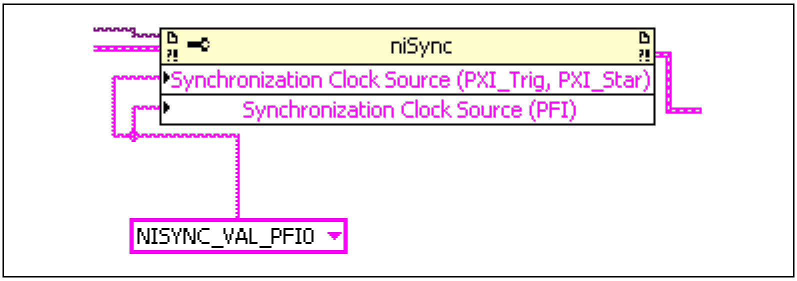
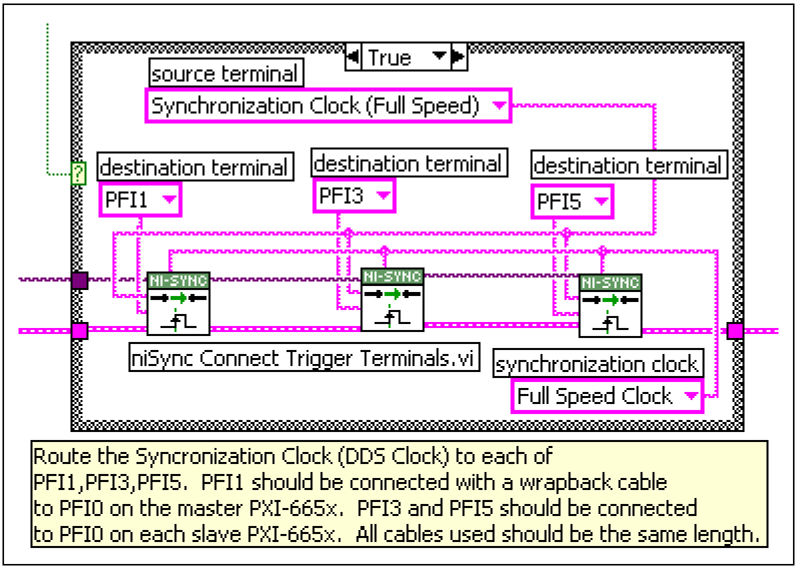

niSync
Synchronization Clock Source (PXI_Trig, PXI_Star)
Synchronization Clock Source (PFI)
NISYNC_VAL_PFI0

Figure 5-4. Clock Configuration for Slave NI PXI-665*x* Devices

Routing the Necessary Synchronization Signals

This section covers the NI PXI-665*x* trigger routing in this example. It discusses two trigger and synchronization signals routed by the NI PXI-665*x*. The first signal is the NI PXI-6115 scan clock. The master NI PXI-665*x* generates this clock using the DDS and passes it to all NI PXI-6115 devices in all chassis. The second signal is the NI PXI-6115 start trigger. The master NI PXI-665*x* generates this trigger and routes it through the NI PXI-665*x* modules to all NI PXI-6115 devices in the system.

The **665x Route** VI executes in three steps. Figure 5-5 shows the first configuration step if the chassis is a master chassis. This step involves setting up the scan clocks sent to the master chassis and both slave chassis. The **niSync Connect Trigger Terminals** VI is called to route the full-speed synchronization clock to PFI1, PFI3, and PFI5. Because the synchronization clock for the PFI lines was specified earlier to be the DDS Clock, this has the effect of outputting the DDS Clock to these three PFI connectors. The NI PXI-665*x* devices in the slave chassis receive no commands, while the master NI PXI-665*x* is setting up these routes. After these three routes are set up, the **665x Route** VI begins the next step.



source terminal: Synchronization Clock (Full Speed)

destination terminal: PFI1

destination terminal: PFI3

destination terminal: PFI5

synchronization clock: Full Speed Clock

Route the Syncronization Clock (DDS Clock) to each of PFI1,PFI3,PFI5. PFI1 should be connected with a wrapback cable to PFI0 on the master PXI-665x. PFI3 and PFI5 should be connected to PFI0 on each slave PXI-665x. All cables used should be the same length.

Figure 5-5. Master NI PXI-665*x* Scan Clock Routing

The next step involves routing the scan clock signal from PFI0 in all chassis to the appropriate star trigger line for each NI PXI-6115 in the chassis, using the **niSync Connect Trigger Terminals** VI. This step is the same whether the chassis is a master or slave chassis. A case structure converts the NI PXI-6115 slot number to the appropriate star trigger number.

Note In all NI chassis, the star trigger number is the number of the slot minus three. Slot 3 in an NI PXI chassis is the first peripheral slot that receives a dedicated star trigger signal, and the star trigger for this slot is PXI_Star0. This is because Slot 1 is reserved for the system controller and Slot 2 is the Star Trigger Controller slot. PXI allows for up to 13 star trigger lines (designated PXI_Star0 through PXI_Star12) per chassis, so in the largest chassis, Slot 15 (PXI_Star12) receives the last available star trigger line.

This route is specified asynchronously because you should not synchronize the incoming clock on PFI0 to any other clock in the system.

Note In general, you should not route external clocks synchronously. Synchronous routing makes sense when you want route a trigger signal synchronous to an edge of the synchronization clock or one of the divided clocks.

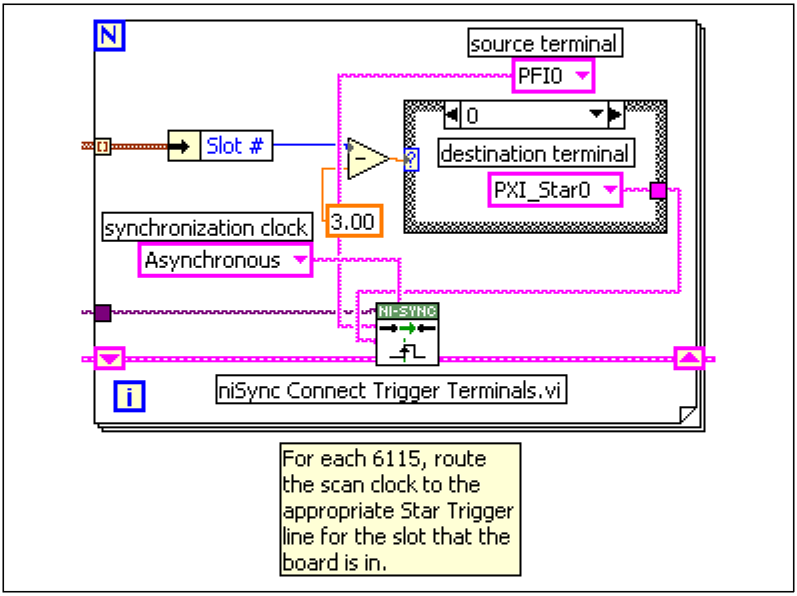
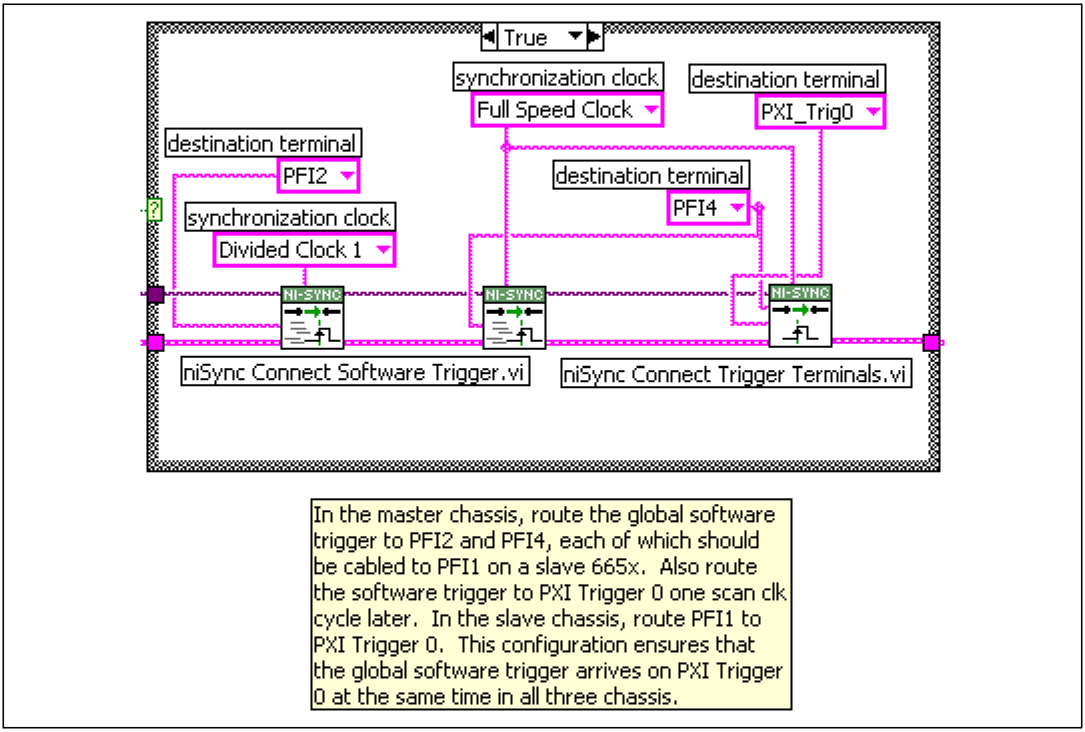

N
source terminal
PFI0
0
Slot #
destination terminal
PXI_Star0
3.00
synchronization clock
Asynchronous
NI-SYNC
i
niSync Connect Trigger Terminals.vi
For each 6115, route the scan clock to the appropriate Star Trigger line for the slot that the board is in.

Figure 5-6. All NI PXI-665*x* Devices Route PFI0 to the Star Trigger Line for Each Slot that Contains an NI PXI-6115

The final routing steps are to generate and distribute the start trigger. The steps required depend on whether the NI PXI-665*x* being configured is in the master chassis or one of the slave chassis. Figure 5-7 shows the master chassis routing required for this step. The **niSync Connect Software Trigger to Terminal** VI routes the software trigger to PFI2 and PFI4, where it is sent to the slave chassis. The synchronization input for this connection is specified as Divided Clock 1. Previously, the Clock Divisor 1 was set to be either 2 or 512, based on the scan clock frequency. The NI PXI-665*x* issues the software trigger on a rising edge of a divided version of the synchronization clock and holds the line high until the next rising edge of this divided clock. This divided clock guarantees that the start trigger is at least 10 µs in duration. (For example, consider a 1 MHz sample rate. The DDS is set to output a 1 MHz clock that is set as the synchronization clock. The Clock Divisor 1 is set to 512, which means that the Divided Clock 1 for the board is a 1 MHz / 512 or 1.953 kHz clock. The period of this clock is 512 µs, and this is also the width of the pulse the software trigger generates, which meets the 10 µs requirement. The full-speed clock would have produced a 1 µs wide pulse, which would not have met the 10 µs requirement.)

Note There are two clock dividers for the synchronization clocks on each NI PXI-665*x*. You can set these dividers to divide by any power of 2 from 2 to 512.

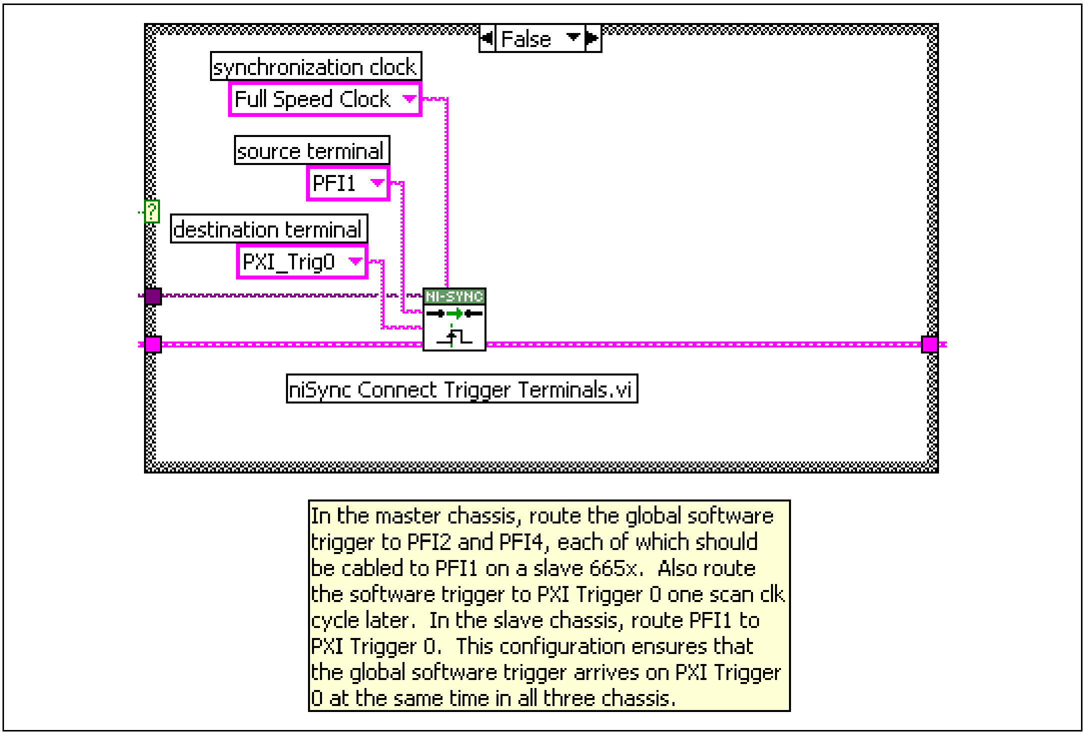
The **niSync Connect Trigger Terminals** VI routes the trigger to the PXI_Trig0 (RTSI0) line on the PXI backplane. For this route, the synchronization input is the Full-Speed Synchronization Clock. This means that when the pulse is sent on the PFI line, the NI PXI-665*x* waits for the next rising edge of the full-speed synchronization clock before allowing the signal to pass onto the PXI_Trig0 line on the backplane. This gives the signal time to travel to the slave chassis.



In the master chassis, route the global software trigger to PFI2 and PFI4, each of which should be cabled to PFI1 on a slave 665x. Also route the software trigger to PXI Trigger 0 one scan clk cycle later. In the slave chassis, route PFI1 to PXI Trigger 0. This configuration ensures that the global software trigger arrives on PXI Trigger 0 at the same time in all three chassis.

Figure 5-7. Master NI PXI-665*x* Routing for the Start Trigger

Figure 5-8 shows the slave chassis routing required for this step. The **niSync Connect Trigger Terminals** VI routes the incoming start trigger on PFI1 to PXI_Trig0 (RTSI0) on the PXI backplane. For this route, the synchronization input is the Full-Speed Synchronization Clock. This means that when the pulse is received on the PFI line, the NI PXI-665*x* waits for the next rising edge of the full-speed synchronization clock before allowing the signal to pass onto the PXI_Trig0 line on the backplane. Therefore, the slave chassis and the master chassis all send the start trigger on the PXI_Trig0 backplane on the same rising edge of the scan clock. All NI PXI-6115 devices receive the start trigger within the same sample clock period and start the acquisition simultaneously.



synchronization clock: Full Speed Clock

source terminal: PFI1

destination terminal: PXI_Trig0

In the master chassis, route the global software trigger to PFI2 and PFI4, each of which should be cabled to PFI1 on a slave 665x. Also route the software trigger to PXI Trigger 0 one scan clk cycle later. In the slave chassis, route PFI1 to PXI Trigger 0. This configuration ensures that the global software trigger arrives on PXI Trigger 0 at the same time in all three chassis.

Figure 5-8. Slave NI PXI-665*x* Routing for the Start Trigger

Configuring the NI PXI-6115 Synchronization and Acquisitions

The next VI called in this example deals primarily with setting up the synchronization on the NI PXI-6115 devices. Refer to the NI-DAQ documentation for more detailed information about configuration and using the NI-DAQ driver software.

The **6115 Sync Config** VI sets up all synchronization parameters for the NI PXI-6115 devices in the system. For this example, all NI PXI-6115 devices receive an external scan clock on the PXI Star Trigger line. In the NI-DAQ driver, **RTSI6** is used to import a signal from the PXI Star Trigger bus. All NI PXI-6115 devices also receive an external start trigger on the PXI_Trig0 line (referred to in NI-DAQ as **RTSI0**).

Beginning the Acquisition

This section discusses initiating the actual acquisition. The VI involved in beginning the acquisition, the **665x Send Sync** VI, fires the NI PXI-665*x* software trigger that all NI PXI-6115 devices use as the start trigger. Figure 5-9 shows the NI-SYNC code that fires the software trigger previously configured in the example by calling the **665x Route** VI.

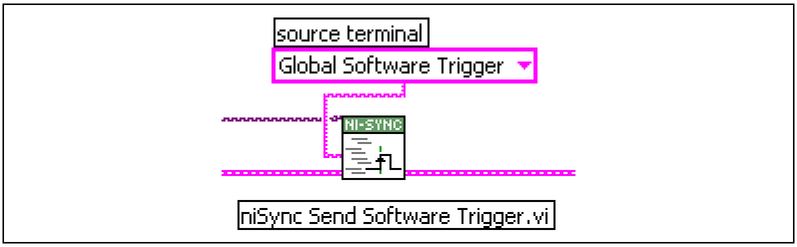


Figure 5-9. Firing the Global Software Trigger on the Master NI PXI-665*x*

Closing and Error Checking

The final functionality in this example is closing references to devices and reporting any errors that occurred during execution. The **6115 665x Close** VI closes all references to the NI PXI-6115 and NI PXI-665*x* devices opened during initialization. Figure 5-10 shows the call to the **niSync Close** VI for each reference to an NI PXI-665*x* device that was opened.

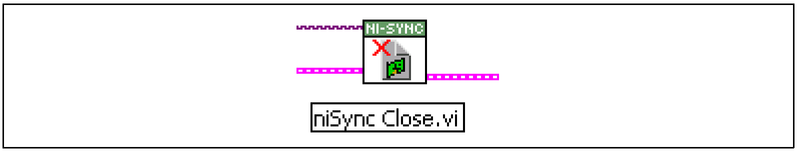


Figure 5-10. Closing Each Reference to an NI PXI-665*x*

The final VI is a standard LabVIEW VI that reports any errors detected during execution. If an error is encountered during execution, no further calls to the driver are executed. The error location and message are passed to the **Simple Error Handler** VI and reported with a pop-up dialog.

6

Synchronizing Multiple NI MIO Modules

Theory of Operation

You can synchronize multiple MIO devices (also known as E Series devices) by sharing a common acquisition clock (scan clock) and having a single start trigger initiate the data acquisition process. The NI PXI-665*x* can supply both the scan clock and the start trigger through the PXI backplane to the MIO devices.

You can synchronize multiple MIO devices using the following PXI triggering and timing features:

- **PXI Trigger Bus**—This bus features eight bidirectional lines that link all PXI slots, providing interdevice synchronization and communication. The skew from slot to slot is less than 10 ns in a typical 8-slot chassis. You can use this to provide the start trigger.
- **PXI Star Trigger**—This special trigger slot provides an independent dedicated bidirectional line for each of up to 13 peripheral slots on a single backplane. All lines are matched in length, which provides a low slot-to-slot skew of less than 1 ns. A star trigger controller plugged into this slot can route triggers and clocks among peripheral slots. You can use this to provide the common acquisition clock.

To synchronize multiple MIO devices, the following must be true:

- All the MIO devices must share a common acquisition clock. The NI PXI-665*x* device provides this clock via the star trigger lines.
- All the MIO devices must wait on a common start trigger. This start trigger is distributed to all devices in the system.

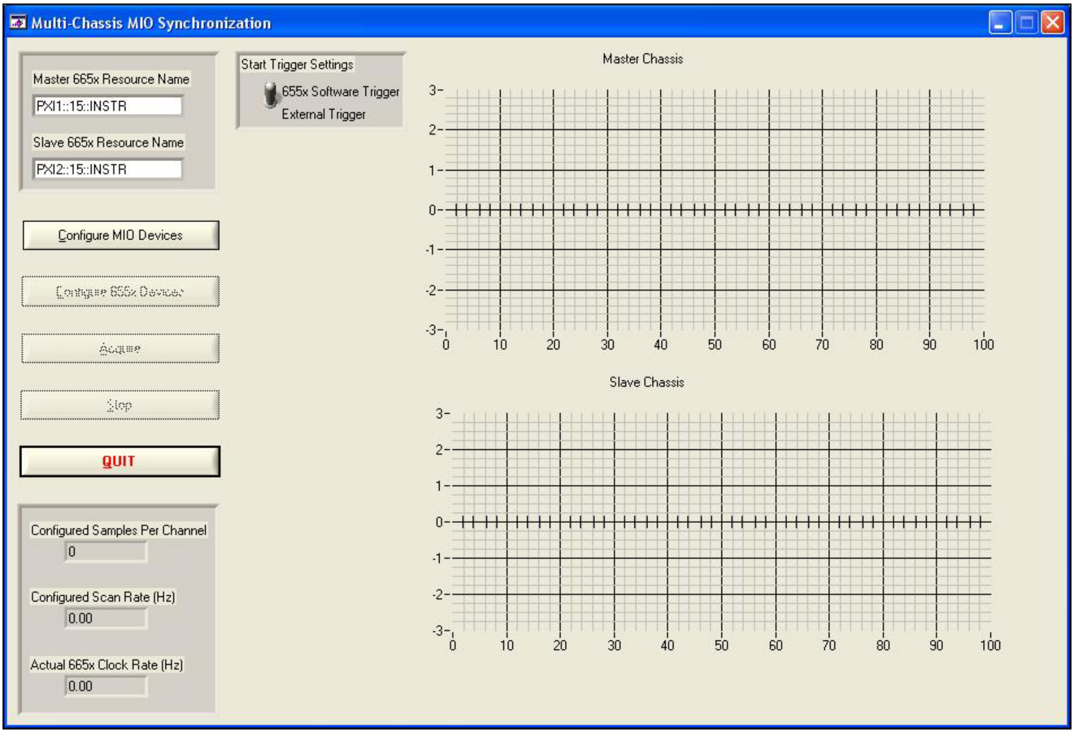
Using the NI PXI-665*x* to Route Synchronization Signals Between Multiple Chassis

Example Overview

The following section discusses the **MIO_665x_Sync** example. This example synchronizes two chassis of MIO devices and was created using LabWindows/CVI 7.0 and NI-DAQmx. The two NI PXI-665*x* modules share the synchronization signals for the MIO devices, as described in the previous section. In this example, the master NI PXI-665*x* generates a scan clock and start trigger. The example provides an option for the MIO devices to start an acquisition using either the global software trigger of the NI PXI-665*x* device or an external trigger.

Figure 6-1 shows the example front panel. The graphs display the data acquired by the devices in each chassis. For the purpose of this example, each MIO device acquires from channel 1 continuously. On the left side of the front panel, you specify the resource names of the master and slave NI PXI-665*x* devices. You must specify these names correctly before any configuration is done. You configure each device in turn, starting with the MIO devices. After you correctly configure the devices, the other disabled buttons are enabled.

The **Acquire** button is enabled after you configure all devices. This initiates the acquisition on the MIO devices. Then, depending on the start trigger selection, the NI PXI-665*x* master either generates the start trigger or you must provide an external trigger. Pressing **Stop** stops the devices from acquiring. You must provide a start trigger every time you start acquiring.



Multi-Chassis MIO Synchronization

Master 665x Resource Name: PXI1::15::INSTR

Slave 665x Resource Name: PXI2::15::INSTR

Start Trigger Settings: 655x Software Trigger / External Trigger

Configured Samples Per Channel: 0

Configured Scan Rate (Hz): 0.00

Actual 665x Clock Rate (Hz): 0.00

Figure 6-1. MIO_665x_Sync Example Front Panel

What You Need to Get Started

You need the following items to get started and set up this example:

- ❑ PC and remote chassis links (such as MXI-3) or PXI embedded controller and remote chassis link
- ❑ Two PXI chassis
- ❑ Master NI PXI-665*x*
- ❑ Slave NI PXI-665*x*
- ❑ MIO devices
- ❑ SMB cables
- ❑ 68-pin cables and breakout boxes

Setting Up the Multichassis PXI System

Complete the following steps to set up the multichassis system and connect the MXI-3 devices:

1. If you are using a PC, install a PCI remote link device in your PC. If you are using an embedded controller, install it in the first PXI chassis.
2. You must install a PXI remote link module in the primary chassis. If you have a PXI chassis with a PCI-PCI bridge, for best performance install the link to the secondary chassis in the first bus segment of the PXI chassis. Otherwise, choose any slot above Slot 2.
3. For the secondary chassis, install a PXI remote link module in Slot 1, or the system controller slot.
4. Connect your remote link cable from the slave chassis to the primary chassis. If you are using a PC, also connect the remote link from the primary chassis to the PC.

For additional help, refer to the documentation for the remote links you are using in your system. For National Instruments remote links such as MXI-3, you can download this documentation from `ni.com/manuals`.

Connecting the NI PXI-665*x* Devices

Complete the following steps to properly connect the trigger and synchronization signals between your master and slave NI PXI-665*x* modules:

1. For each chassis in the multichassis system, install an NI PXI-665*x* module in Slot 2 of the chassis. This allows the NI PXI-665*x* module to replace the PXI_Clk10 reference clock with the onboard high-precision oscillator, and allows the module to control the PXI Star Trigger lines.
2. Use an SMB cable to connect PFI2 of the master NI PXI-665*x* module to a clock splitter. Then wire the output of the splitter to PFI0 of the master and slave NI PXI-665*x* modules. PFI2 generates the scan clock, which must be routed to each input via equal length cables.
3. For the purpose of this example, the master and slave NI PXI-665*x* devices must receive the start triggers at their respective PFI1 inputs. Connect your start trigger signals accordingly. If you intend to use the master NI PXI-665*x* device as the start trigger generator, PFI3 will be the source of the start trigger. Wire PFI3 from the master board to the PFI1 inputs of the master and slave boards using a T-splitter.

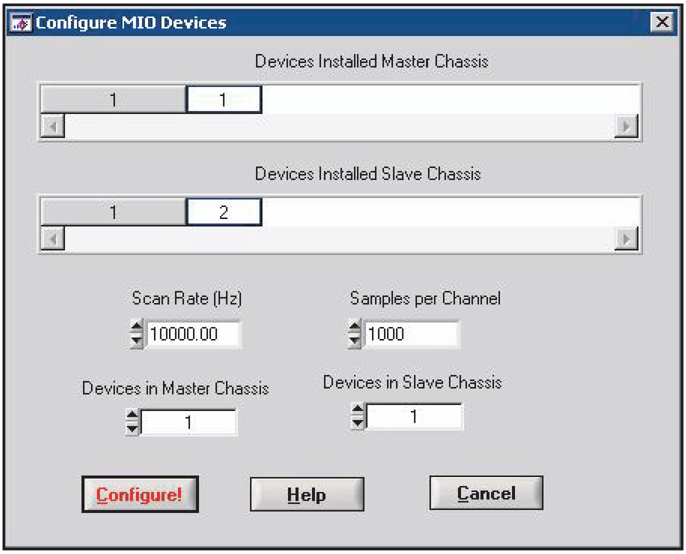
Connecting the MIO Devices

For this example, connect the signals for your data acquisition to channel 1 of each MIO device in the master and slave chassis using the 68-pin cables and breakout boxes. Refer to the your device user manual for more information about the pin numbers needed to access channel 1. The devices must be plugged into consecutive slots, starting from Slot 3.

Configuring and Running the Software Example

Complete the following steps to configure and run the **MIO_665x_Sync** example:

1. Type in the resource name of each NI PXI-665*x* device in the appropriate control (for example, `PXI1::15::INSTR`). An NI PXI-665*x* module should be installed in Slot 2 of each chassis. You can find these resource names using Measurement & Automation Explorer (MAX). Be sure to specify the correct resource name for the master and slave chassis.
2. Click on **Configure MIO Devices**. This brings up another dialog for configuring the acquisition devices in the master and slave chassis (refer to Figure 6-2).
   a. For the configuration to work correctly, you must configure each device string for the MIO Devices in MAX as *Dev#*, where # is the number with which the board is associated. For example, if you have three MIO Devices, the channel strings associated with them in MAX should be Dev1, Dev2, and Dev3. These device numbers are used in the **Configure MIO Devices** window.
   b. In the configuration dialog, indicate the number of MIO devices in each chassis. This updates the number of cells in the appropriate tables. Enter the device numbers in the appropriate table. Each table should have unique device numbers in each cell. Specify the scan rate and the number of Samples per Channel. Click the **Configure!** button.
3. Click **Configure 665x Devices**, which should be enabled now.
4. Before clicking **Acquire**, ensure that your start trigger connections are set up as described above. When you click **Acquire**, the devices are configured to wait 10 s for the start trigger before timing out.
5. Click **Stop** to stop the acquisition. You can restart the acquisition if needed by clicking **Acquire**.
6. Click **Quit** to close resources and exit the example.



Devices Installed Master Chassis: 1 1

Devices Installed Slave Chassis: 1 2

Scan Rate (Hz): 10000.00

Samples per Channel: 1000

Devices in Master Chassis: 1

Devices in Slave Chassis: 1

Figure 6-2. Configure MIO Devices Dialog

Example Functionality

This section discusses the **MIO_665x_Sync** example configuration functions, including the NI-Sync functions. A significant amount of NI-DAQmx programming in this example is not discussed in this manual. Refer to the user documentation for NI-DAQmx for more information about programming with this driver.

Initialization and Error Checking

All configuration functions for this example are in the **MIO_665x_Sync_fn** module.

Two main functions initialize the master and slave NI PXI-665*x* devices, `InitializeMaster665x()` and `InitializeSlave665x()`.

`InitializeMaster665x()` configures the master timing device by first obtaining the instrument handle to the device via the resource name. The front panel synchronization clock is routed to **PFI2**. The DDS Clock is selected as the front panel synchronization clock, and **PFI0** is selected as the source for the backplane clock. Also, **PFI3** is set as the global software trigger source. This configuration is done at an early stage to keep any line activity from triggering the MIO Devices.

InitializeSlave665x() obtains the instrument handle for the slave device. **PFI0** is selected as the source for the front and backplane synchronization clocks.

Error handling is done using a macro ErrChk, defined in MIO_665x_Sync_fn.h. To use this macro in a function, there must be a local variable named error of type int and an Error: label that precedes the error recovery code at the end of the function.

Configuring the Scan Clock

This section covers configuration of the scan clock that drives the MIO device acquisition. For the purpose of this example, the high-precision DDS Clock is the scan clock. The clock frequency is configured in the ConfigureMaster665x() function using the niSync_SetAttributeViReal64() function. Because the DDS Clock is already specified as the front panel synchronization clock, and **PFI2** is connected to the front panel synchronization clock, a clock is now generated at **PFI2**.

Routing the Necessary Synchronization Signals

This section covers the NI PXI-665*x* trigger routing in this example. Two essential routes are set up in each NI PXI-665*x* devices. These routes send the scan clock and start trigger to each MIO device in the chassis. It is assumed that the MIO devices are present in consecutive slots, starting with Slot 3.

In the ConfigureMaster665x() and ConfigureSlave665x() function, the scan clock is routed to each star trigger line from **PFI0**. This is done using niSync_ConnectTrigTerminals(). This is set up for each MIO device in each chassis:

```
sprintf (starTriggerBuffer, "PXI_Star%d",
        deviceCounter);
```

```
ErrChk(niSync_ConnectTrigTerminals (instrumentHandle,
      NISYNC_VAL_PFI0,
      starTriggerBuffer,
      NISYNC_VAL_SYNC_CLK_ASYNC,
      NISYNC_VAL_DONT_INVERT,
      NISYNC_VAL_UPDATE_EDGE_RISING));
```

For more information about which star trigger line is mapped to which slot, refer to the *PXI Specification,* rev. 2.0.

Similarly, the start triggers are routed from **PFI1** to PXI_Trig3 (**RTSI 3** for the MIO Devices). This occurs for the master and slave chassis:

```
ErrChk(niSync_ConnectTrigTerminals (instrumentHandle,
       NISYNC_VAL_PFI1,
       NISYNC_VAL_PXITRIG3,
       NISYNC_VAL_SYNC_CLK_FULLSPEED,
       NISYNC_VAL_DONT_INVERT,
       NISYNC_VAL_UPDATE_EDGE_FALLING));
```

If an error is returned, the NI PXI-665*x* devices are reset, and the instrument handle is closed, using `nisync_reset()` and `nisync_close()`, respectively.

Configuring the MIO Synchronization and Acquisitions

This section describes setting up the MIO devices to receive the external clock and start trigger. For each MIO device in the chassis, you must create and configure a DAQmx task. To keep track of the device names and the number of devices in each chassis, the CHANNEL_ARRAY_STRUCT structure is used.

The MIO device numbers are passed into the `ConfigureChassisSlaves()` function. The device number is used to configure the device for Continuous Acquisition at the user-specified scan rate. The device also is configured to acquire using an external clock via the PXI_Star Trigger (**RTSI6**) and a start trigger from **RTSI3**. The following code configures each MIO device in each chassis:

```
ErrChk (DAQmxCreateTask("",&(*taskHandleArray)
[deviceCounter]));

//Create channel string to create DAQmx task.Reads from
channel 1
sprintf (channelStringBuffer, "Dev%d/ai1",
chassisSlaveChannels.physicalChannelArray[deviceCounter]);

ErrChk (DAQmxCreateAIVoltageChan
((*taskHandleArray)[deviceCounter],
channelStringBuffer,"", DAQmx_Val_Diff, minVoltage,
maxVoltage, DAQmx_Val_Volts, NULL));

//RTSI 6 is star trigger line for E-series devices
sprintf (channelStringBuffer, "/Dev%d/PXI_Star",
chassisSlaveChannels.physicalChannelArray[deviceCounter]);
```

```
//Need to set the scan rate the same as the DDS clock
ErrChk (DAQmxCfgSampClkTiming
((*taskHandleArray)[deviceCounter], channelStringBuffer,
scanRate, DAQmx_Val_Rising,
DAQmx_Val_ContSamps, 100));

//RTSI 3 will receive the start trigger for the device
sprintf (channelStringBuffer, "/Dev%d/RTSI3",
chassisSlaveChannels.physicalChannelArray[deviceCounter]);

ErrChk(DAQmxCfgDigEdgeStartTrig
((*taskHandleArray)[deviceCounter],
channelStringBuffer,DAQmx_Val_Rising));
```

At this point, the devices are configured and ready for an acquisition. The task handles are returned as a parameter.

Beginning the Acquisition

After you configure the MIO devices, you can initiate acquisition using the `DAQmxStartTask()` function. The devices are then waiting to receive the start trigger. They time out if they do not receive the start trigger in 10 s. After they receive the start trigger, the devices use the external clock to acquire data from channel 1.

Closing and Error Checking

Any errors that occur during configuration and acquisition are displayed in a pop-up dialog box. All NI PXI-665*x* instrument handles are closed using `nisync_close()`, and MIO sessions are closed using `DAQmxClearTask()`.

7

Synchronizing Multiple NI PXI-5411 Modules

NI PXI-5411 Theory of Operation

The NI PXI-5411 40 MS/s arbitrary waveform generator can use a phase-locked loop to synchronize its sample clock to the PXI_Clk10. This method is the same method used by the NI PXI-5112 digitizer previously discussed in this manual. The high-precision oscillator on the NI PXI-665*x* can supply this reference frequency to the PXI backplane.

You can synchronize multiple NI PXI-5411 modules with the following PXI timing and triggering features:

- **PXI_Clk10**—This is a 10 MHz clock with at least 100 ppm accuracy. It is independently distributed to each PXI peripheral slot through equal-length traces with a skew of less than 1 ns between slots. Multiple devices can use this common timebase for synchronization. This allows each NI PXI-5411 to phase lock to a clock shared by the entire system.
- **PXI Trigger Bus**—This bus features eight bidirectional lines that link all PXI slots, providing inter-device synchronization and communication. The skew from slot to slot is less than 10 ns.

To synchronize multiple NI PXI-5411 modules, the following must be true:

- All NI PXI-5411 devices must share a common timebase. The NI PXI-5411 modules can achieve this by phase locking all NI PXI-5411 modules in the system to the PXI_Clk10.
- All NI PXI-5411 modules must use a synchronization pulse (sync pulse) to align the clock dividers on each NI PXI-5411 module. In a single-chassis system, the sync pulse is typically generated by a master NI PXI-5411 module in Slot 2 of the PXI chassis.

- To designate the actual acquisition, a start trigger signal must be distributed to all devices in the system. In a single-chassis system, the start trigger is typically generated by a master NI PXI-5411 module in Slot 2 of the PXI chassis.

Note When synchronizing multiple NI PXI-5411 modules, the asynchronous trigger and the synchronization pulse must be routed on different bus lines.

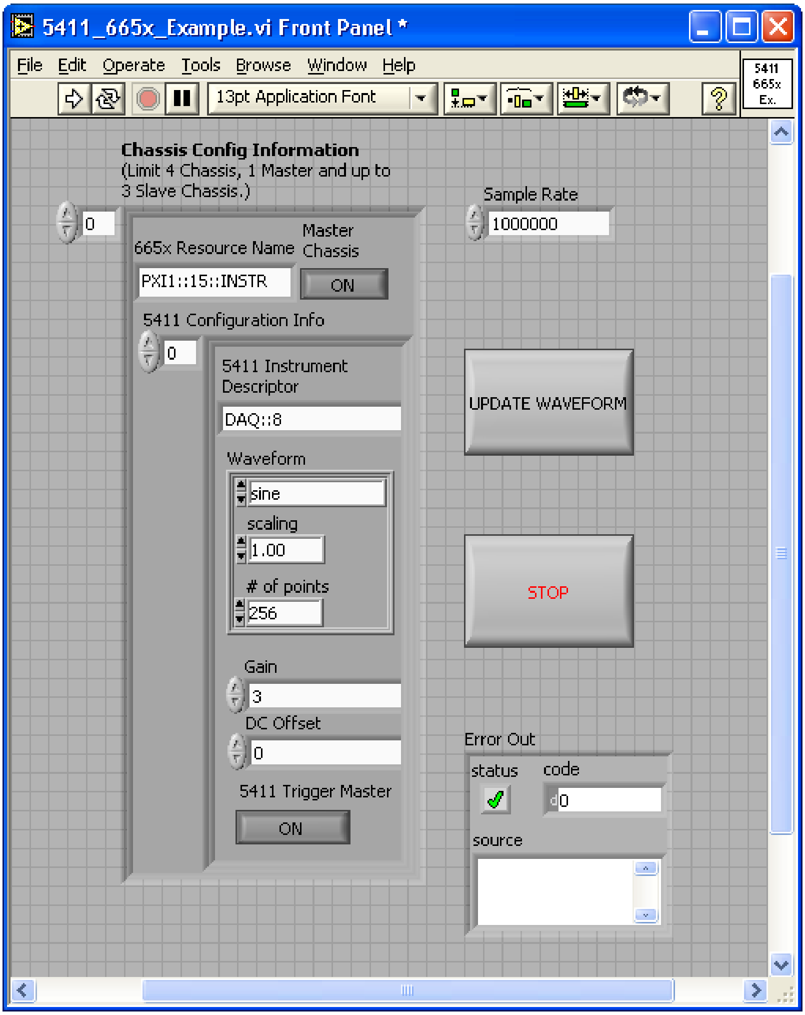
Using the NI PXI-665*x* to Route Synchronization Signals Between Multiple Chassis

Example Overview

The following section discusses the example **5411 665x Example** VI. This example synchronizes up to four chassis of NI PXI-5411 modules. The NI PXI-665*x* shares the synchronization signals for the NI PXI-5411 described in the previous section. In this example, the high-precision oscillator in the NI PXI-665*x* replaces the PXI_Clk10 in the master chassis, as well as the slave chassis. The NI PXI-665*x* master module then uses its software trigger to send the NI PXI-5411 sync pulse to all boards in the system. Finally, the master NI PXI-5411 module triggers when given the “initiate” command in software and sends this start trigger to the PXI backplane. The NI PXI-665*x* modules then route this signal from the master chassis to all slave chassis and slave PXI-5411 modules.

Figure 7-1 shows the front panel for this example VI. The left side of the panel contains a **Chassis Config Information** array, where each element specifies the configuration information for an individual chassis. Within each element, you can specify the acquisition and synchronization characteristics for a chassis.

The graph on the right displays the various signals acquired from each NI PXI-5411 module. The **Stop** button at the bottom of the front panel stops execution of the VI. The **Update** button updates the waveforms each NI PXI-5411 in the system generates. The remaining front panel controls specify the remaining trigger and signal generation parameters. For more information about the NI PXI-5411 modules’ ability to do phase compensation, refer to the *NI-FGEN User Manual*.


5411_665x_Example.vi Front Panel *
File Edit Operate Tools Browse Window Help
13pt Application Font
5411 665x Ex.
UPDATE WAVEFORM
STOP

Chassis Config Information
(Limit 4 Chassis, 1 Master and up to 3 Slave Chassis.)

0

665x Resource Name: PXI1::15::INSTR

Master Chassis: ON

5411 Configuration Info

0

5411 Instrument Descriptor: DAQ::8

Waveform: sine

scaling: 1.00

# of points: 256

Gain: 3

DC Offset: 0

5411 Trigger Master: ON

Sample Rate: 1000000

Error Out

status: ✓

code: 0

source:

Figure 7-1. 5411 665x Example VI Front Panel

What You Need to Get Started

To set up and get started with this example, you need the following hardware and software.

- ❑ PC and remote chassis links (such as MXI-3) or PXI embedded controller and remote chassis link
- ❑ PXI chassis (up to four chassis)

- ❑ Master NI PXI-665*x*
- ❑ Slave NI PXI-665*x* (one per slave chassis)
- ❑ NI PXI-5411 modules
- ❑ SMB cables
- ❑ BNC cables

Setting Up the Multichassis PXI System

Complete the following steps to set up the multichassis system and connect the MXI-3 devices:

1. Refer to your remote link (such as MXI-3) documentation for detailed instructions about installation of PCI and PXI remote link modules.
2. If you are using a PC, install a PCI remote link device in your PC. If you are using an embedded controller, install this controller into the first PXI chassis.
3. For each chassis that needs to link to a secondary chassis, you need to install a PXI remote link module in some slot other than Slot 2. For best performance, if you have a PXI chassis that contains a PCI-PCI bridge, it is best to install the link to secondary chassis in the first bus segment of the PXI chassis.
4. For each secondary chassis, install a PXI remote link module in Slot 1, or the system controller slot, of the PXI chassis.
5. Connect your remote link cable from each PCI or PXI remote link module to its corresponding PXI remote link module on the secondary side.

For additional help, refer to the documentation for the remote links you are using in your system. For National Instruments remote links such as MXI-3, you can download this documentation from `ni.com/manuals`.

Connecting the NI PXI-665*x* Devices

Complete the following steps to properly connect the trigger and synchronization signals between your master and slave NI PXI-665*x* modules.

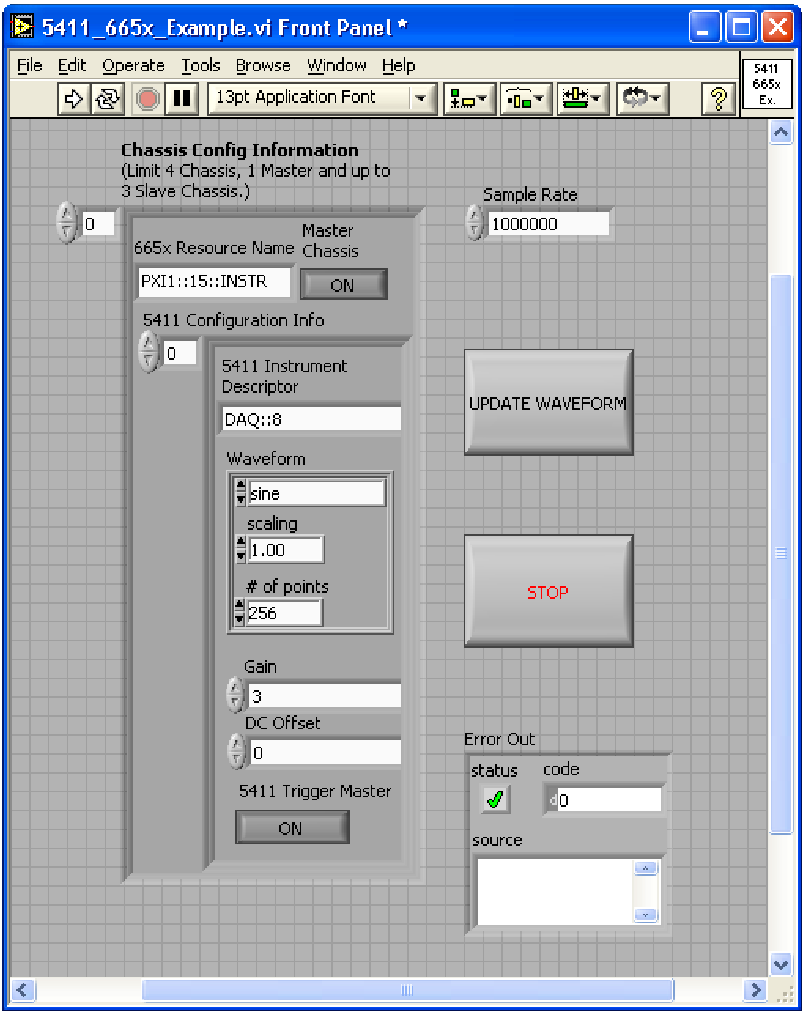
1. For each chassis in the multichassis system, install an NI PXI-665*x* module installed in Slot 2. This allows the NI PXI-665*x* module to replace the PXI_Clk10 reference clock with the onboard high-precision oscillator.
2. Use an SMB cable to connect Clk Out of the master NI PXI-665*x* module to the Clk In of either one slave NI PXI-665*x* module if using a single secondary chassis, or to a clock splitter if using more than one secondary chassis. If using more than one secondary chassis, the output from the clock splitter must be connected to the Clk In of each slave NI PXI-665*x* device.
3. Use SMB cables to connect each secondary chassis to the outputs of the master NI PXI-665*x* module that carries the sync pulse. For this example, the sync pulse is routed out of the even-numbered front panel PFI outputs, PFI 0, PFI 2, and PFI 4. Connect one of these front panel PFI outputs from the master to the front panel PFI 0 of the slave NI PXI-665*x* modules for the slave chassis. As previously noted, you can support up to three secondary or slave chassis (for a total of four chassis) with this example.
4. Use SMB cables to connect each secondary chassis to the outputs of the master NI PXI-665*x* module that carries the start trigger. For this example, the start trigger is routed out of the odd numbered front panel PFI outputs, PFI 1, PFI 3, and PFI 5. Connect one of these front panel PFI outputs from the master to the front panel PFI 1 of the slave NI PXI-665*x* modules for the slave chassis.

Connecting the NI PXI-5411 Devices

Connect the signals from the NI PXI-5411 modules in your system via a BNC cable to the destination for the signal you will generate. For more information about connecting signals generated by the NI PXI-5411 to other devices, refer to the user manual for your NI PXI-5411 hardware.

Configuring and Running the Software Example

Figure 7-2 shows the front panel for the **5411 665x Example** VI.


5411_665x_Example.vi Front Panel *
File Edit Operate Tools Browse Window Help
13pt Application Font
5411 665x Ex.
UPDATE WAVEFORM
STOP

Chassis Config Information
(Limit 4 Chassis, 1 Master and up to 3 Slave Chassis.)

0

665x Resource Name: PXI1::15::INSTR

Master Chassis: ON

5411 Configuration Info: 0

5411 Instrument Descriptor: DAQ::8

Waveform: sine

scaling: 1.00

# of points: 256

Gain: 3

DC Offset: 0

5411 Trigger Master: ON

Sample Rate: 1000000

Error Out

status: ✓

code: d0

source:

Figure 7-2. 5411 665x Example VI Front Panel

Complete the following steps to configure and run the **5411 665x Example** VI:

1. Set the configuration information for each chassis. Each element of the **Chassis Config Information** array describes the configuration parameters for the modules in one chassis of the multichassis system. Scroll through the array to enter the information for each chassis, up to four chassis. Designating information in any array element beyond the fourth in the array results in an error, as this example supports up to four chassis only (that is, three slave chassis).

    a. Type in the NI PXI-665*x* resource name for each chassis (for example, `PXI1::15::INSTR`). An NI PXI-665*x* module should be installed in Slot 2 of each chassis. You can find these resource names using Measurement & Automation Explorer (MAX).

    b. Be sure that one chassis in your array is configured as the master chassis by clicking the **Master Chassis** button. Configure one and only one chassis as the master in the array, or the example generates an error. The chassis configured as the master must contain the master NI PXI-5411 module.

    c. Each element of the **Chassis Config Info** array contains an array to hold the information about the NI PXI-5411 devices in that chassis. Populate the instrument descriptor for each NI PXI-5411 module in the chassis (for example, you may note one instrument descriptor in the first chassis as `DAQ::1`). Use MAX to determine the instrument descriptors for all of your NI PXI-5411 modules. Use the **Waveform** input to specify the waveform each NI PXI-5411 device generates. You can specify the type of waveform, scaling, and the number of data points to be generated. Finally, specify the gain for the waveform amplitude as well as any DC offset necessary for the waveform.

       Be sure one NI PXI-5411 is the trigger master by clicking the **5411 Trigger Master** button. One and only one NI PXI-5411 in the system must be selected as the trigger master.

    d. Set the sample rate. This example uses a global sample rate, so each arb uses the same value taken from the **Sample Rate** front panel input.

    e. After making your signal connections and configuring all modules according to the above steps, run the VI. Use the **Stop** button to stop VI execution safely.

Example Functionality

This section discusses each **5411 665x Example** VI subVI, including the NI-Sync VI functionality. A significant amount of NI-FGEN programming in this example is not discussed in this manual. Refer to the user documentation for the NI-FGEN driver for more information about programming with this software.

Initialization and Error Checking

This section discusses error checking included in this example to verify configuration information is correct. It also discusses opening references to and initializing devices in the system. The two subVIs in this section are the **5411 665x Check** VI, which verifies that the user has properly configured the example, and the **5411 665x Init** VI, which opens sessions to all devices in the system.

The **5411 665x Check** VI verifies that you have entered the appropriate configuration information for the system. First, the VI verifies that only four chassis are specified in the system. If you enter information for more than four chassis, this VI generates an error, and the rest of the example VI does not execute. This VI also verifies that the number of masters in the system is correct. It checks that you have specified at least one and only one master chassis for the system. It also checks that there is at least one and only one master NI PXI-5411 module in the master chassis.

The **5411 665x Init** VI walks through each device in the system and opens a session to the device. This VI calls on the NI-FGEN driver to open sessions to the NI PXI-5411 devices. It also calls on the **niSync Initialize** VI, shown in Figure 7-3, to open sessions to the NI PXI-665*x* devices in the system. This example specifies resetting the NI PXI-665*x* modules on initialization to clear any previous routes on the board.

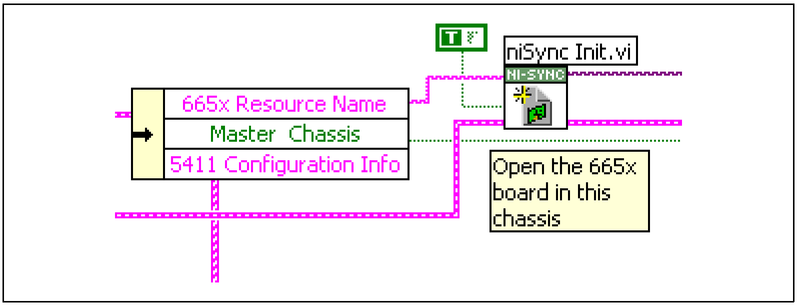


Figure 7-3. The niSync Initialize VI Used to Open Sessions to NI PXI-665*x* Devices

If the NI-FGEN driver or the NI-Sync driver cannot open a session to their devices, this VI passes along the error from their respective drivers, and the example stops execution of any subsequent subVIs.

Configuring the 10 MHz Timebase

This section discusses configuration of the shared 10 MHz timebase in the system. For the purpose of this example, the master NI PXI-665*x* device in the system uses its high-precision oscillator to replace the PXI_Clk10 signal supplied by each chassis. The clock from the high-precision oscillator is routed to the PXI backplane and also out the front panel Clk Out SMB output to be shared, usually via a clock splitter, with other chassis in the system. Each NI PXI-665*x* device in a slave chassis is configured to input this high-precision oscillator signal into its Clk In front panel SMB input, and route this clock to replace the PXI_Clk10 signal the chassis supplies. This allows a single reference clock to be shared with all devices in the system.

Figure 7-4 shows the section of code for master chassis clock configuration. The synchronization clock for both the front panel and backplane are set to be the PXI_Clk10 using a property node. Next, the **niSync Connect Clock Terminals** VI is called to route the onboard **oscillator** to **PXI_Clk10** in the master chassis. Finally, the **niSync Connect Clock Terminals** VI is called again to route the same oscillator signal to the front panel **ClkOut** SMB.

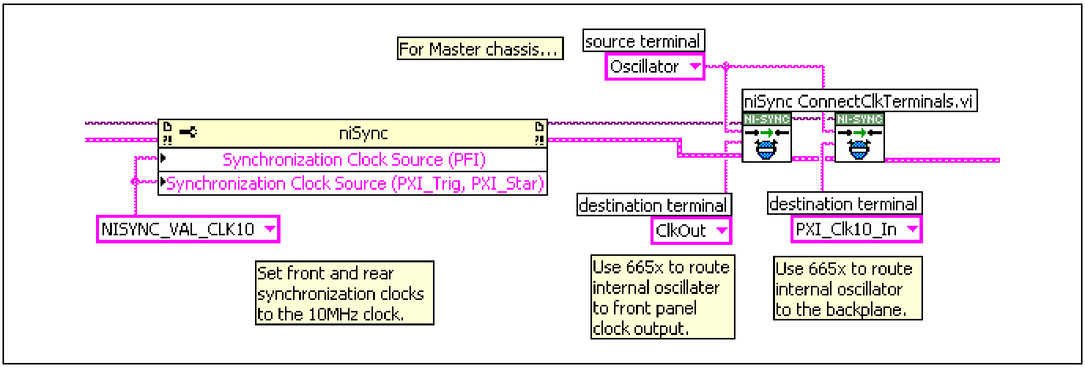

For Master chassis...
source terminal
Oscillator
niSync ConnectClkTerminals.vi
NI-SYNC
NI-SYNC
niSync
Synchronization Clock Source (PFI)
Synchronization Clock Source (PXI_Trig, PXI_Star)
NISYNC_VAL_CLK10
destination terminal
ClkOut
destination terminal
PXI_Clk10_In
Set front and rear synchronization clocks to the 10MHz clock.
Use 665x to route internal oscillater to front panel clock output.
Use 665x to route internal oscillator to the backplane.

Figure 7-4. NI-Sync Calls to Configure the Master NI PXI-665*x* Clock Outputs

Figure 7-5 shows the section of code for the slave chassis clock configuration. The slave NI PXI-665*x* devices are configured to receive a clock at their Clk In SMB and route this clock to replace PXI_Clk10 in the chassis. This time, the **niSync Connect Clock Terminals** VI routes the signal on **ClkIn** to the destination **PXI_Clk10**.

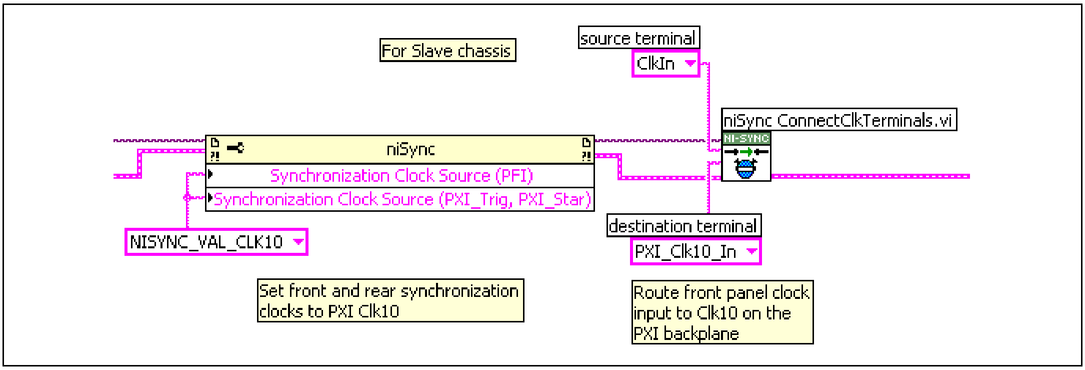
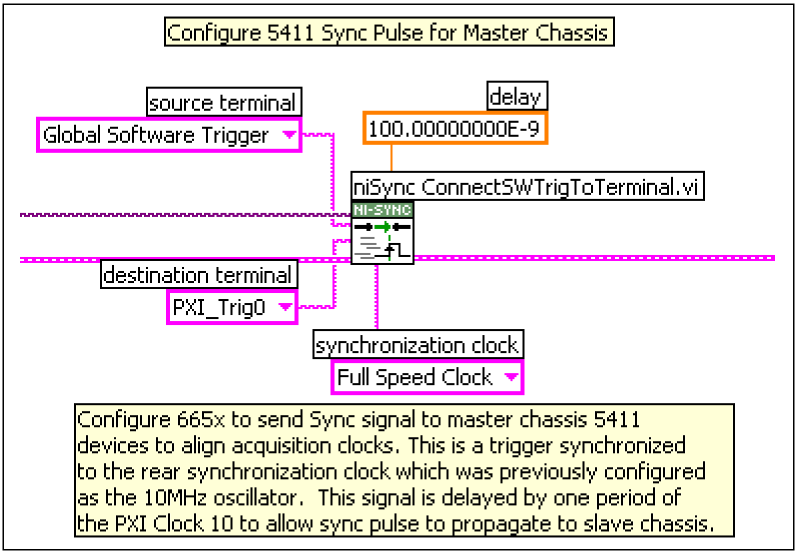

For Slave chassis
source terminal
ClkIn
niSync ConnectClkTerminals.vi
NI-SYNC
niSync
Synchronization Clock Source (PFI)
Synchronization Clock Source (PXI_Trig, PXI_Star)
NISYNC_VAL_CLK10
destination terminal
PXI_Clk10_In
Set front and rear synchronization clocks to PXI Clk10
Route front panel clock input to Clk10 on the PXI backplane

Figure 7-5. Clk10 Routing for Slave NI PXI-665*x* Devices

Routing the Necessary Synchronization Signals

This section discusses the NI PXI-665*x* trigger routing conducted in this example, including two trigger and synchronization signals routed by the NI PXI-665*x*. The first signal is the NI PXI-5411 sync pulse. This pulse is generated by the master NI PXI-665*x* as a software trigger and passed to all NI PXI-5411 devices in all chassis. The second signal is the NI PXI-5411 start trigger. This trigger is generated by the master NI PXI-5411 device and routed through the NI PXI-665*x* modules to all slave NI PXI-5411 devices in the system.

The **665x_Route** VI first determines whether the chassis it is configuring is a master or slave chassis. After determining the appropriate configuration steps to take, it makes the routes for the chassis. Figure 7-6 shows the first configuration step if the chassis is a master chassis. This step involves setting up the synchronization signals to be sent to the master chassis (in this case, only the SW trigger used by the sync pulse). The **niSync Connect Software Trigger** VI routes the SW trigger to PXI Trigger line 0, where the master NI PXI-5411 device expects to receive its sync pulse. This pulse is specified in this VI to be synchronized to the full-speed clock, previously assigned as PXI_Clk10.



Configure 5411 Sync Pulse for Master Chassis

source terminal: Global Software Trigger

delay: 100.00000000E-9

niSync ConnectSWTrigToTerminal.vi

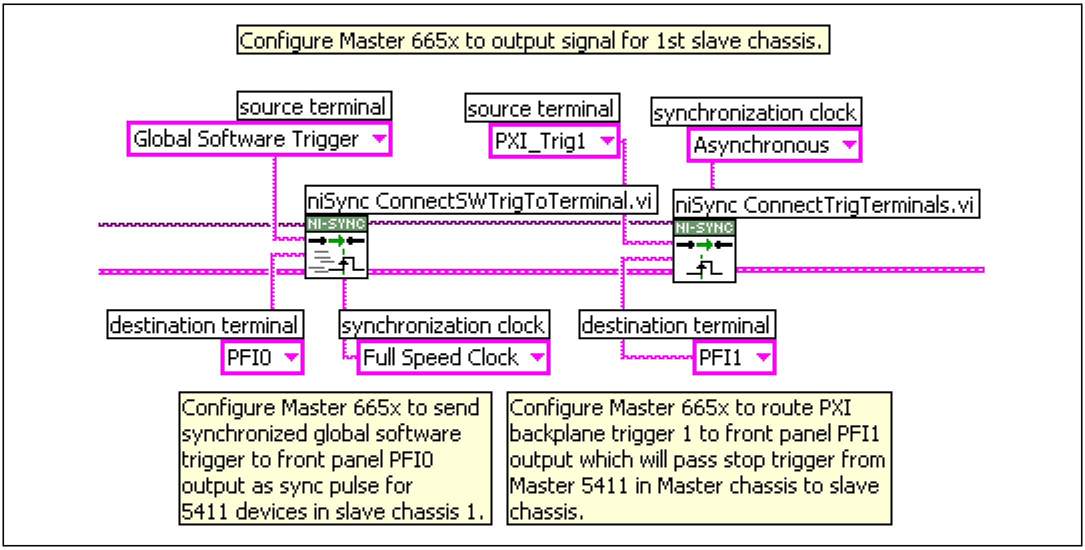
destination terminal: PXI_Trig0

synchronization clock: Full Speed Clock

Configure 665x to send Sync signal to master chassis 5411 devices to align acquisition clocks. This is a trigger synchronized to the rear synchronization clock which was previously configured as the 10MHz oscillator. This signal is delayed by one period of the PXI Clock 10 to allow sync pulse to propagate to slave chassis.

Figure 7-6. Master NI PXI-665*x* Configures Software Trigger Routes for Master Chassis

The next few configuration steps in the master NI PXI-665*x* involve setting up the routes to output the sync pulse and start trigger for each slave chassis. Figure 7-7 shows the code for these routes for the first slave chassis. These same calls are repeated for each slave chassis, although the specific PFI outputs used change for each chassis. The **niSync Connect Software Trigger** VI routes the software trigger for the sync pulse to the first slave chassis. The **niSync Connect Trigger Terminals** VI routes the start trigger the NI PXI-5411 generates on the backplane to the front panel **PFI0** output to be sent to the first slave chassis. In this case, the start trigger is not resynchronized, and therefore the synchronization clock is left **asynchronous**.



Configure Master 665x to output signal for 1st slave chassis.

source terminal: Global Software Trigger

niSync ConnectSWTrigToTerminal.vi

destination terminal: PFI0

synchronization clock: Full Speed Clock

source terminal: PXI_Trig1

synchronization clock: Asynchronous

niSync ConnectTrigTerminals.vi

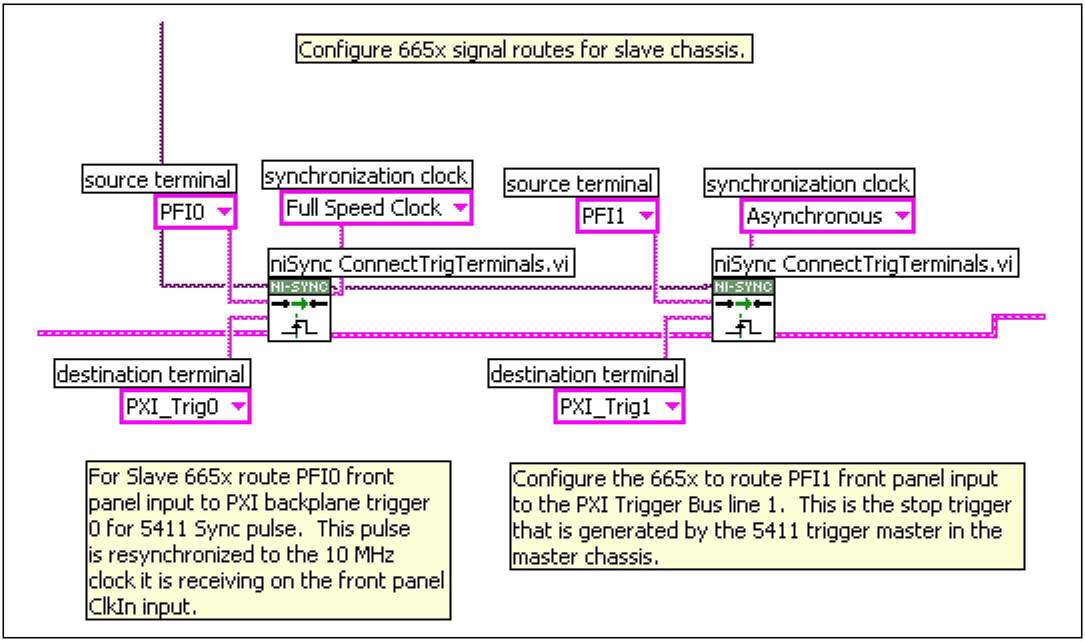
destination terminal: PFI1

Configure Master 665x to send synchronized global software trigger to front panel PFI0 output as sync pulse for 5411 devices in slave chassis 1.

Configure Master 665x to route PXI backplane trigger 1 to front panel PFI1 output which will pass stop trigger from Master 5411 in Master chassis to slave chassis.

Figure 7-7. Master NI PXI-665*x* Configures Software Trigger Routes and Start Trigger Routes for Slave Chassis

Figure 7-8 shows the final routing steps, which concern the NI PXI-665*x* modules in the slave chassis. Each module must be set up to input the sync pulse on the PFI 0 SMB input and route it to PXI Trigger line 0. Then, each slave NI PXI-665*x* must be configured to input the start trigger on the PFI 1 SMB input and route this signal to any star trigger line where a device resides. The first step is to call the **niSync Connect Trigger Terminals** VI with a source of **PFI0** and a destination of **PXI_Trig0**. The sync pulse is set up to be resynchronized to the synchronization clock, which in this example is the PXI_Clk10. For the next route the same function is called, but this time with a source and destination of **PFI1** and the appropriate PXI Star Trigger, respectively. This route passes the trigger asynchronously and is specified as such in the call to the VI.



For Slave 665x route PFI0 front panel input to PXI backplane trigger 0 for 5411 Sync pulse. This pulse is resynchronized to the 10 MHz clock it is receiving on the front panel ClkIn input.

Configure the 665x to route PFI1 front panel input to the PXI Trigger Bus line 1. This is the stop trigger that is generated by the 5411 trigger master in the master chassis.

Figure 7-8. Slave NI PXI-665*x* Configures Routing for Sync Pulse and Start Trigger Inputs to Slave Chassis

Configuring the NI PXI-5411 Synchronization and Generations

The next two VIs called in this example deal primarily with setting up output generations on the NI PXI-5411 devices. Refer to the NI-FGEN documentation for more detailed information about configuration and using the NI-FGEN software.

The first VI, the **5411 Sync Config** VI, sets up all synchronization parameters for the master and slave NI PXI-5411 devices in the system. For this example, all NI PXI-5411 devices receive an external sync pulse on the PXI Trigger bus line 0. The master NI PXI-5411 device generates a start trigger based on receiving a signal of the appropriate level on the channel designated as the Trigger Channel. The master NI PXI-5411 module outputs this trigger to the PXI Trigger bus, where it is then routed to the rest of the chassis in the system. All slave NI PXI-5411 devices receive an external start trigger. All slave NI PXI-5411 modules receive their start trigger on PXI Trigger line 1 on the backplane.

The second VI, the **5411 Gen Config** VI, sets up all generation parameters for the NI PXI-5411 devices in the system, such as the signal rate and sample size. To simplify this example, it abstracts away many of the configuration options the NI-FGEN software provides. As previously described, more detailed information on these capabilities is beyond the scope of this manual; you can find it in the user documentation for your NI-FGEN software.

Beginning the Signal Generation

The next section discusses initiating the actual generation. Before this can occur, the sync pulse must be sent to all NI PXI-5411 devices to align all clock dividers already synchronized to the same system clock. The **665x Send Sync** VI fires the NI PXI-665*x* software trigger which all NI PXI-5411 devices use for their sync pulse. Figure 7-9 shows the NI-Sync code that fires the software trigger previously configured in the example by the **665x Route** VI.

The VI in this example that handles this functionality is the **5411 Start Gen** VI. This VI initiates all NI PXI-5411 devices in the system. For the master NI PXI-5411, initiating the generation means it begins signal generation immediately on receipt of this software call and sends the start trigger to the slave modules. Initiating the generation for slave NI PXI-5411 devices means they begin waiting for a start trigger on the appropriate PXI Trigger Bus line.

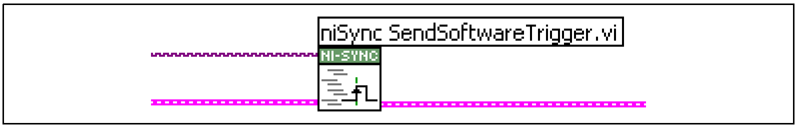


Figure 7-9. Firing the Global Software Trigger on the Master NI PXI-665*x*

Closing and Error Checking

The final functionality in this example is closing references to devices and reporting any errors that occurred during execution. The **5411 665x Close** VI closes all references to the NI PXI-5411 and NI PXI-665*x* devices opened during initialization. Figure 7-10 shows the call to the **niSync Close** VI that is called for each reference to an NI PXI-665*x* module opened.

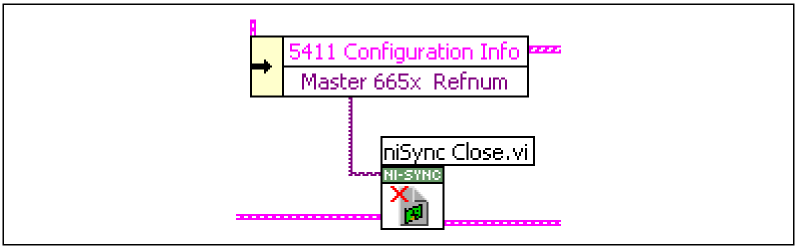


Figure 7-10. Closing Each Reference to an NI PXI-665x

The final VI is a standard LabVIEW VI that reports any errors detected during execution. If an error is encountered during execution, no further calls to the driver are executed, and the error location and message are passed to the **Simple Error Handler** VI and reported with a pop-up dialog.

A

Technical Support and Professional Services

Visit the following sections of the National Instruments Web site at ni.com for technical support and professional services:

- **Support**—Online technical support resources include the following:
  - **Self-Help Resources**—For immediate answers and solutions, visit our extensive library of technical support resources available in English, Japanese, and Spanish at ni.com/support. These resources are available for most products at no cost to registered users and include software drivers and updates, a KnowledgeBase, product manuals, step-by-step troubleshooting wizards, conformity documentation, example code, tutorials and application notes, instrument drivers, discussion forums, a measurement glossary, and so on.
  - **Assisted Support Options**—Contact NI engineers and other measurement and automation professionals by visiting ni.com/support. Our online system helps you define your question and connects you to the experts by phone, discussion forum, or email.
- **Training**—Visit ni.com/training for self-paced tutorials, videos, and interactive CDs. You also can register for instructor-led, hands-on courses at locations around the world.
- **System Integration**—If you have time constraints, limited in-house technical resources, or other project challenges, NI Alliance Program members can help. To learn more, call your local NI office or visit ni.com/alliance.

If you searched ni.com and could not find the answers you need, contact your local office or NI corporate headquarters. Phone numbers for our worldwide offices are listed at the front of this manual. You also can visit the Worldwide Offices section of ni.com/niglobal to access the branch office Web sites, which provide up-to-date contact information, support phone numbers, email addresses, and current events.

Glossary

Symbol	Prefix	Value
p	pico	10^{-12}
n	nano	10^{-9}
μ	micro	10^{-6}
m	milli	10^{-3}
k	kilo	10^{3}
M	mega	10^{6}
G	giga	10^{9}
T	tera	10^{12}

Symbols

%	percent
±	plus or minus
+	positive of, or plus
–	negative of, or minus
/	per
°	degree
Ω	ohm

A

accumulator	a part where numbers are totaled or stored
ADE	application development environment

asynchronous

a property of an event that occurs at an arbitrary time, without synchronization to a reference clock

B

backplane

an assembly, typically a printed circuit board (PCB), with 96-pin connectors and signal paths that bus the connector pins. PXI systems have two connectors, called the J1 and J2 connectors.

backplane synchronization clock

the clock signal that is used to synchronize the RTSI/PXI triggers or PXI_Star triggers on an NI PXI-665*x*

bus

the group of conductors that interconnect individual circuitry in a computer. Typically, a bus is the expansion vehicle to which I/O or other devices are connected. An example of a PC bus is the PCI bus.

C

C

Celsius

Clk In

Clk In is a signal connected to the SMB input pin of the same name. Clk In can serve as PXI_Clk10_IN or be used as a phase lock reference for the OCXO.

Clk Out

Clk Out is the signal on the SMB output pin of the same name. The OCXO clock, DDS clock, or PXI_Clk10 may be routed to Clk Out.

clock

hardware component that controls timing for reading from or writing to groups

CompactPCI

a Eurocard configuration of the PCI bus for industrial applications

D

D/A

digital-to-analog

DAC

digital-to-analog converter—an electronic device that converts a digital number into a corresponding analog voltage or current

DAQ

data acquisition—(1) collecting and measuring electrical signals from sensors, transducers, and test probes or fixtures and inputting them to a computer for processing; (2) collecting and measuring the same kinds of electrical signals with A/D and/or DIO devices plugged into a computer, and possibly generating control signals with D/A and/or DIO devices in the same computer

DC

direct current

DDS

Direct Digital Synthesis—a method of creating a clock with a programmable frequency

E

EEPROM

electrically erasable programmable read-only memory—ROM that can be erased with an electrical signal and reprogrammed

ESD

electrostatic discharge

F

frequency

the basic unit of rate, measured in events or oscillations per second using a frequency counter or spectrum analyzer. Frequency is the reciprocal of the period of a signal.

frequency tuning word

a number that specifies the frequency

front panel

the physical front panel of an instrument or other hardware

H

Hz

hertz—the number of scans read or updates written per second

I

in.

inch or inches

J

jitter — the rapid variation of a clock or sampling frequency from an ideal constant frequency

L

LabVIEW — a graphical programming language

LED — Light-Emitting Diode—a semiconductor light source

M

master — the requesting or controlling device in a master/slave configuration.

Measurement & Automation Explorer (MAX) — a controlled centralized configuration environment that allows you to configure all of your National Instruments DAQ, GPIB, IMAQ, IVI, Motion, VISA, and VXI devices

N

NI-DAQ — National Instruments driver software for DAQ hardware

O

OCXO — oven-controlled crystal oscillator

oscillator — a device that generates a fixed frequency signal. An oscillator most often generates signals by using oscillating crystals, but may also use tuned networks, lasers, or atomic clock sources. The most important specifications on oscillators are frequency accuracy, frequency stability, and phase noise.

output impedance — the measured resistance and capacitance between the output terminals of a circuit

P

PCI

Peripheral Component Interconnect—a high-performance expansion bus architecture originally developed by Intel to replace ISA and EISA. It is achieving widespread acceptance as a standard for PCs and work-stations; it offers a theoretical maximum transfer rate of 132 Mbytes/s.

PFI

Programmable Function Interface

PLL

phase-locked loop

precision

the measure of the stability of an instrument and its capability to give the same measurement over and over again for the same input signal

propagation delay

the amount of time required for a signal to pass through a circuit

PXI

a rugged, open system for modular instrumentation based on CompactPCI, with special mechanical, electrical, and software features. The PXIbus standard was originally developed by National Instruments in 1997, and is now managed by the PXIbus Systems Alliance.

PXI star

a special set of trigger lines in the PXI backplane for high-accuracy device synchronization with minimal latencies on each PXI slot

R

RTSI bus

Real-Time System Integration bus—the NI timing bus that connects DAQ devices directly, by means of connectors on top of the devices, for precise synchronization of functions

S

s

seconds

skew

the actual time difference between two events that would ideally occur simultaneously. Inter-channel skew is an example of the time differences introduced by different characteristics of multiple channels. Skew can occur between channels on one module, or between channels on separate modules (intermodule skew).

slave

a computer or peripheral device controlled by another computer

slot	the place in the computer or chassis in which a card or module can be installed
Slot 2	the second slot in a PXI system which can house a master timing unit
SMB	Sub Miniature Type B—a small coaxial signal connector that features a snap coupling for fast connection
synchronous	a property of an event that is synchronized to a reference clock

T

t_{CtoQ}	clock to output time
t_{hold}	hold time
t_{pd}	propagation delay time
TRIG	trigger signal
trigger	a digital signal that starts or times a hardware event (for example, starting a data acquisition operation)
t_{setup}	setup time

V

V	volts
VI	virtual instrument

Index