PXI

NI-Sync User Manual

July 2009
370926E-01

NATIONAL INSTRUMENTS™

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 43 662 457990-0, Belgium 32 (0) 2 757 0020, Brazil 55 11 3262 3599, Canada 800 433 3488, China 86 21 5050 9800, Czech Republic 420 224 235 774, Denmark 45 45 76 26 00, Finland 358 (0) 9 725 72511, France 01 57 66 24 24, Germany 49 89 7413130, India 91 80 41190000, Israel 972 3 6393737, Italy 39 02 41309277, Japan 0120-527196, Korea 82 02 3451 3400, Lebanon 961 (0) 1 33 28 28, Malaysia 1800 887710, Mexico 01 800 010 0793, Netherlands 31 (0) 348 433 466, New Zealand 0800 553 322, Norway 47 (0) 66 90 76 60, Poland 48 22 328 90 10, Portugal 351 210 311 210, Russia 7 495 783 6851, Singapore 1800 226 5886, Slovenia 386 3 425 42 00, South Africa 27 0 11 805 8197, Spain 34 91 640 0085, Sweden 46 (0) 8 587 895 00, Switzerland 41 56 2005151, Taiwan 886 02 2377 2222, Thailand 662 278 6777, Turkey 90 212 279 3031, United Kingdom 44 (0) 1635 523545

For further support information, refer to the *Technical Support and Professional Services* appendix. To comment on National Instruments documentation, refer to the National Instruments Web site at `ni.com/info` and enter the info code `feedback`.

Important Information

Contents

About This Manual

The *NI-Sync User Manual* is for users of the NI-Sync driver software, an application programming interface (API) for controlling National Instruments 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:

»	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.
✎	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. Italic text 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.

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`
- *NI PCI-1588 User Manual*, available from `ni.com/manuals`
- *NI PXI-6682 User Manual*
- *NI PXIe-6672 User Manual*

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 National Instruments timing modules.

About the NI-Sync Driver Software

NI-Sync is a library of VIs and functions for controlling NI timing modules. You can use NI-Sync to configure the timing and synchronization of your system. This can include signal-based synchronization such as sharing triggers and clocks to be used directly. You can also do time-based synchronization, using time protocols such as IEEE-1588, IRIG, or GPS as a time reference to time stamp or perform synchronized generation of clocks and triggers. Use NI-Sync in conjunction with other measurement software, such as NI-DAQmx, for advanced timing, high channel count, distributed or multiple-instrument applications.

Introduction

The NI-Sync driver software includes the following:

- NI-Sync instrument driver API and device driver
- Example software for signal-based (using clocks and triggers directly) and time-based (using time protocols) synchronization

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.

Supported Devices and Platforms

NI-Sync supports Windows XP (32-bit), Windows Vista x86, Windows Vista x64, LabVIEW RT 8.2.1 and above.

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	8.2.1 or later
LabVIEW RT Module	8.2.1 or later
LabWindows™/CVI™	7.0 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

Installing the Software

The software package that ships with the NI PXI-665*x*, NI PXI-6672, NI PXI-6682(H), and NI PCI-1588 provides the following items:

- NI-Sync driver software
- LabVIEW example code
- LabWindows/CVI example code

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 examples are included to give you a starting point in using the NI timing and synchronization modules. Additional examples for using NI timing modules with other devices are online at `ni.com/examples`.

Note Be sure to install the NI-Sync software *before* installing your device hardware.

Device and System Configuration

Before you begin using your NI timing devices, 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 your PXI system components in Measurement & Automation Explorer (MAX). Refer to your PXI hardware user manual for more information.

Note The NI PCI-1588 and PXI-6682 devices are actually two devices, a timing and synchronization device and a Network Interface Card (NIC). In the Windows Device Manager, the timing and synchronization devices are enumerated in the **Data Acquisition Devices** section as **NI PCI-1588** or **PXI-6682**. The NIC is enumerated in the **Network adapters** section as **AMD PCNET Family PCI Ethernet Adapter**. When configuring your network connections in Windows, the local area connection associated with the AMD PCNET Family PCI Ethernet Adapter is the one associated with the timing and synchronization device.

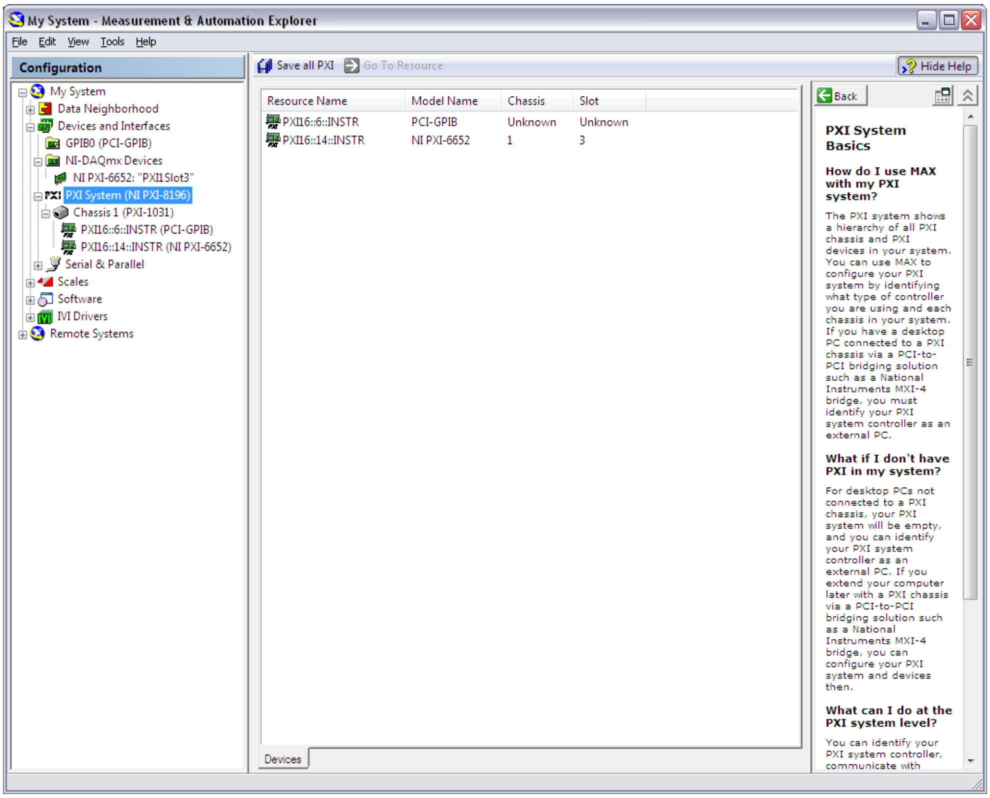
Using Measurement & Automation Explorer

MAX is a Windows-based application for configuring and viewing National Instruments device settings on Windows operating systems.

Locating Your NI Timing and Synchronization Devices

Your NI timing modules appear in MAX under **My System»Devices and Interfaces»NI DAQmx Devices**. From this location, you can launch test panels, perform self tests, and view properties of your devices. Once you have identified your PXI system components, you also can locate your NI timing devices by browsing the PXI System view (**My System»Devices and Interfaces»PXI System**). Refer to Figure 1-1 for an example of the type of device information available in MAX.

Note The DAQmx device name, VISA Resource Name, and VISA alias are all valid inputs for the Resource Name 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.


My System - Measurement & Automation Explorer
File Edit View Tools Help
Configuration
Save all PXI
Go To Resource
Hide Help
Back
Devices

- My System
  - Data Neighborhood
  - Devices and Interfaces
    - GPIB0 (PCI-GPIB)
    - NI-DAQmx Devices
      - NI PXI-6652: "PXI1Slot3"
    - PXI System (NI PXI-8196)
      - Chassis 1 (PXI-1031)
        - PXI16::6::INSTR (PCI-GPIB)
        - PXI16::14::INSTR (NI PXI-6652)
    - Serial & Parallel
  - Scales
  - Software
  - IVI Drivers
- Remote Systems

Resource Name	Model Name	Chassis	Slot
PXI16::6::INSTR	PCI-GPIB	Unknown	Unknown
PXI16::14::INSTR	NI PXI-6652	1	3

PXI System Basics

How do I use MAX with my PXI system?

The PXI system shows a hierarchy of all PXI chassis and PXI devices in your system. You can use MAX to configure your PXI system by identifying what type of controller you are using and each chassis in your system. If you have a desktop PC connected to a PXI chassis via a PCI-to-PCI bridging solution such as a National Instruments MXI-4 bridge, you must identify your PXI system controller as an external PC.

What if I don't have PXI in my system?

For desktop PCs not connected to a PXI chassis, your PXI system will be empty, and you can identify your PXI system controller as an external PC. If you extend your computer later with a PXI chassis via a PCI-to-PCI bridging solution such as a National Instruments MXI-4 bridge, you can configure your PXI system and devices then.

What can I do at the PXI system level?

You can identify your PXI system controller, communicate with

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 timing and synchronization devices. 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 7.1.1 or later to manage your code development and 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 in the LabVIEW Example Finder. Navigate to **Help»Find Examples** and browse **Hardware Input and Output»Timing and Synchronization** or search for the keyword `NI-Sync`.

Creating a Windows Application Using LabWindows/CVI

This section assumes that you are using LabWindows/CVI 7.0 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`.

Note The default installation directory has changed from `C:\VXIPNP` to `C:\Program Files\IVI Foundation\VISA`. Upgrades over previous versions of NI-VISA use the previous installation directory.

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.

Note The default installation directory has changed from `C:\VXIPNP` to `C:\Program Files\IVI Foundation\VISA`. Upgrades over previous versions of NI-VISA use the previous installation directory.

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 signal-based 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 timing devices and other functions.

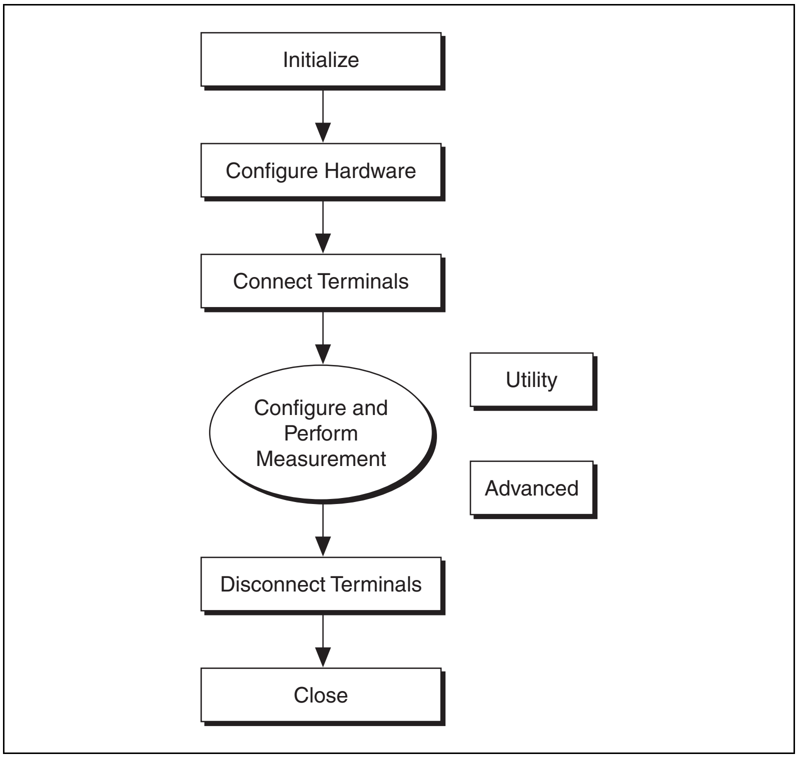


Figure 2-1. Basic Programming Flow of an NI-Sync Application with NI PXI-665*x* Devices

Figure 2-2 shows the basic programming flow of typical time-based NI-Sync applications. NI-Sync VIs and functions are organized under the Initialize, Configure Hardware, Get Time, Create Future Time Event, Enable Time Stamp Trigger, Create Clock, Read Trigger Time Stamp, Clear Future Time Events, Disable Time Stamp Trigger, Clear Clock, 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. These functions perform various tasks such as resetting devices, returning the revision number of the NI-Sync instrument driver and instrument firmware, and other functions.

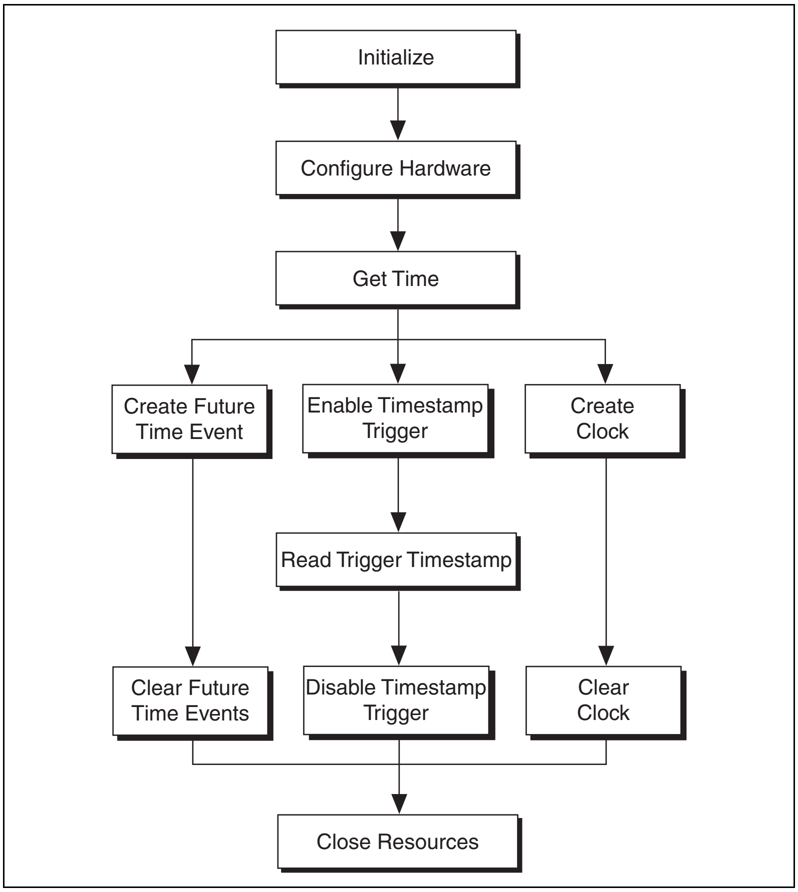


Figure 2-2. Basic Programming Flow of an NI-Sync Application with NI PCI-1588 Devices

Note Using MAX, you can configure the 1588 PTP protocol to start automatically at boot.

Initialize

For any application you write, you must first open a session to establish communication with the NI timing and synchronization device 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 DAQmx Device Name or VISA Resource Name corresponding to an NI timing device and use this information to locate the instrument and create a session to it. You can obtain the DAQmx Device Name and VISA Resource Name for an instance of your device using MAX. Refer to Chapter 1, *Introduction, Installation, and Configuration*, for an example of using MAX.

The Initialize VI and function create a new instrument session. Any session returned from Initialize may be used in multiple program threads.

Configure Hardware

Use Configuration VIs, LabVIEW property nodes, or functions to adjust settings of the timing and synchronization features of the timing module, including ADC input threshold voltage levels, DDS frequency, synchronization clock sources, specific time reference properties, and other settings and features.

Attributes are configured using a LabVIEW property node or the `niSync_SetAttribute` and `niSync_GetAttribute` functions.

Accessing Attributes

In LabVIEW, you can find NI-Sync 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 to the **niSync** palette at **Measurement I/O»NI-Sync** and drag the property node to the block 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

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 module. Refer to your hardware 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 precise onboard or external 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[0:16]), and the front panel PFI lines. Refer to your hardware user manual for a complete discussion of trigger terminals.

Notes You should route clocks directly over point-to-point connections such as PXI_Star to avoid possible bus loading signal integrity issues over shared trigger buses such as PXI_Trig. You can use PXI_Trig if a clean clock such as PXI_CLK10 or an onboard clock governs the ultimate acting on the signal. Refer to your hardware user manual for a complete discussion of trigger terminals.

Star triggers may not correlate to slots as expected. You should refer to your chassis manual for more information on routing star triggers.

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 the following:

- 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 over the matched trace length PXI_Star synchronized lines)

Note Some destination terminals are part of a shared bus and should not be driven by multiple signal sources. NI driver software implements reservation of shared resources including 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. For more information, refer to KnowledgeBase 3TJDOND8 at `ni.com/support`.

Trigger terminal connections are characterized by a source terminal, destination terminal, and route properties such as inversion and synchronization. Check your hardware user manual to see if your hardware supports these additional routing features.

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. The Global Software Trigger terminal can be connected to any other trigger terminal (PXI_Trig, PXI_Star, and PFI). Once connected to destinations, you can initiate a hardware pulse that is then routed to all destinations.

Typical uses of the Global Software Trigger include the following:

- 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 timing modules

Software trigger terminal connections are characterized by a source terminal (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.

Note The PXI-6682(H) and PCI-1588 can implement similar functionality by creating a Future Time Event.

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`

Start 1588

After you have configured the NI PCI-1588 or PXI-6682 device, you can start the Precision Time Protocol (PTP). PTP synchronizes the clock on the NI device according to the IEEE 1588 specification. For more information regarding PTP and the IEEE 1588 specification, refer to Chapter 3, *Timing Protocols*. In general, you should start 1588 and ensure it has been synchronized before performing any other operations with the NI device.

Note Using MAX, you can configure the 1588 PTP protocol to start automatically at boot. This is enabled by default.

Note When an NI device is participating in PTP as a slave device, it may be required to perform a macro phase adjustment. A macro phase adjustment is when the 1588 clock is adjusted by a significant amount and, therefore, the 1588 time no longer atomically increments. This should not occur on a well designed and stable network. If this occurs, future time events, clocks, and time stamps may be affected. If the 1588 time is set forward, future time events and clock transitions that were missed occur immediately. If the 1588 time is set backward, future time events and clock transitions are delayed.

Note If the 1588 clock participating in the PTP enters the faulty state, future time events, clocks, and time stamps will no longer be synchronized with other 1588 devices participating in the PTP. This should not occur on a well designed and stable network. You can check for this condition by monitoring the 1588 clock state property.

The following VI and function deal with starting PTP.

LabVIEW VI	C Function
niSync Start 1588	`niSync_Start1588`

Get Time

This function queries a time-based synchronization board for the current board time.

The following VI and function deal with getting the 1588 time.

LabVIEW VI	C Function
niSync Get Time	`niSync_GetTime`

Create Future Time Event

You can change a signal level of a device terminal at a specific time by creating a future time event. When the time on the time-based device reaches the specified time, the signal level is changed as the future time event specifies. You can create multiple future time events that change the signal levels on different terminals or change the signal at the same terminal to create waveforms.

The following VI and function deal with creating future time events.

LabVIEW VI	C Function
niSync Create Future Time Event	`niSync_CreateFutureTimeEvent`

Enable Time Stamp Trigger

You can generate a time stamp when a signal at any NI terminal changes its level by enabling a time stamp trigger. The time stamp is the board time on the NI time-based device when the specified terminal changed state.

The following VI and function deal with enabling time stamp triggers.

LabVIEW VI	C Function
niSync Enable Time Stamp Trigger	`niSync_EnableTimeStampTrigger`

Create Clock

You can generate a clock at destination terminals. You can start and stop the clock at a specific board time. The clock is synchronized to the board clock.

The following VI and function deal with creating clocks.

LabVIEW VI	C Function
niSync Create Clock	`niSync_CreateClock`

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, NI-Scope, NI-FGEN, or others. For example code to get started with measurement synchronization, refer to `ni.com/examples`.

Read Trigger Time Stamp

After you have enabled a time stamp trigger, you can read all generated time stamps. The time stamp is the1588 clock time on the NI PCI-1588 device when the specified terminal changes its state.

The following VI and function deal with reading trigger time stamps.

LabVIEW VI	C Function
niSync Read Trigger Time Stamp	`niSync_ReadTriggerTimeStamp` `niSync_ReadMultipleTriggerTimeStamp`

Clear Future Time Events

After you no longer want to generate future time events, they should be cleared. This allows the terminal generating time stamps to be used for other operations. Clearing future time events on a specific terminal clears all future time events for that terminal.

The following VI and function deal with clearing future time events.

LabVIEW VI	C Function
niSync Clear Future Time Events	`niSync_ClearFutureTimeEvents`

Disable Time Stamp Trigger

After you no longer want to generate time stamps, they should be disabled. This allows the terminal generating time stamps to be used for other operations.

The following VI and function deal with disabling time stamp triggers.

LabVIEW VI	C Function
niSync Disable Time Stamp Trigger	`niSync_DisableTimeStampTrigger`

Clear Clock

After you no longer want to generate a clock, it should be cleared. This allows the terminal generating the clock to be used for other operations.

The following VI and function deal with clearing clocks.

LabVIEW VI	C Function
niSync Clear Clock	`niSync_ClearClock`

Stop 1588

After you no longer want to synchronize the 1588 clock on the NI PCI-1588, PTP can be stopped. However, PTP should not be stopped if other operations are currently configured.

The following VI and function deal with stopping PTP.

LabVIEW VI	C Function
niSync Stop 1588	`niSync_Stop1588`

Disconnect Terminals

After a measurement has been performed, connected terminals should be disconnected. This returns the PXI system to its pre-measurement state and frees any reserved resources for use. Terminals are disconnected by supplying the connected source and destination terminals to Disconnect VIs or functions.

Terminals are disconnected according to their type used for generating the connection.

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, **niSync Reset** disconnects all connected terminals as part of a board reset.

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.

However, for an NI PCI-1588 or PXI-6682 session, calling Close stops, clears, and disables operations configured within the session. That is, if you started PTP within the session, calling Close stops PTP. Likewise, if you created a future time event within the session, it is cleared; if you enabled a time stamp trigger, it is disabled; if you created a clock, it is cleared.

Utility

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

Instrument Driver Utility Functions

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

LabVIEW VIs	C Function
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

Some hardware may be able to measure the frequency of a signal. 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 measurement terminal and calling the Measure Frequency VI or function.

FPGA Reconfiguration

Caution FPGA Reconfiguration is a sensitive operation that can damage your module. Do *not* use this operation unless you are confident about what you are doing.

Your hardware may support an on-demand update of the field-programmable gate array (FPGA) with a new bitstream. The following VI and function support this operation.

LabVIEW VI	C Function
niSync Configure FPGA	`niSync_ConfigureFPGA`

3 Timing Protocols

Distributed Time Technology Overview

Measurement and automation systems involving multiple devices often require accurate timing for event synchronization and data correlation. For example, an industrial automation application may need to synchronize distributed motion controllers, or a test and measurement application may need to correlate data acquired from sensors distributed across a device under test. You can achieve this synchronization through signal-based or time-based synchronization. Signal-based synchronization involves sharing signals such as clocks and triggers directly between nodes that need to be synchronized. Time-based synchronization involves nodes independently synchronizing their time to a time reference. There are advantages and disadvantages to both methods of device synchronization.

In systems where the devices are near each other, sharing a common timing signal is generally the easiest and most accurate method of synchronization. For example, modular instruments in a PXI chassis all share a common 10 MHz clock signal from the PXI backplane, enabling synchronization to less than 1 ns. To accurately use a common timing signal, a device must be calibrated to account for the signal propagation delay from the timing source to the device. Sharing a common timing signal becomes unfeasible when the distance between devices increases, or when devices frequently change location. Even at moderate distances, a common timing signal may require significant costs for cabling and configuration.

In these situations, time-based synchronization may be necessary. Using this approach, devices act on timing signals originating from a local clock that is synchronized to the other clocks in the system. Examples of distributed clock synchronization include devices synchronized to GPS satellites, a PC's internal clock synchronized to an NTP time server, a group of devices participating in the IEEE 1588 protocol, or devices synchronized to a common IRIG-B source. Instead of sharing timing signals directly, these devices periodically exchange information and adjust their local timing sources to match each other.

The synchronization of distributed time requires a continuous process. A clock is essentially a two-part device, consisting of a frequency source and an accumulator. In theory, if two clocks were set identically and their frequency sources ran at the exact same rate, they would remain synchronized indefinitely. In practice, however, clocks are set with limited precision, frequency sources run at slightly different rates, and the rate of a frequency source changes over time and temperature. Most modern electronic clocks use a crystal oscillator as a frequency source. The frequency of a crystal oscillator varies due to initial manufacturing tolerance, temperature and pressure changes, and aging. Because of these inherent instabilities, distributed clocks must be synchronized continually to match each other in frequency and phase.

Time Reference

The niSync timing family of devices, including the NI PCI-1588 and NI PXI-6682, use *Time Referencing* to synchronize frequency and phase with a *Time Reference*. A Time Reference is an external time source that provides periodic time updates. Some examples of Time References are GPS satellites, IEEE 1588 masters, or IRIG-B sources. Each of these sources provides periodic time updates. GPS satellites, for example, broadcast the current time once per second, on the second's boundary. When used as a Time Reference on the niSync timing family of devices, the niSync timing device uses this once per second update as a reference time. The niSync timing device uses a sequence of these reference times to match the source of the reference times in frequency and phase as closely as possible. The niSync timing device has an onboard clock used to provide clock holdover between reception of reference times, and the previously received reference times are used to adjust the onboard clock frequency and phase. The end result is that the niSync timing device can provide a continuous time source synchronized to the device's Time Reference as closely as possible, and makes it possible to tightly synchronize multiple distributed clocks using a single Time Reference technology.

Regardless of the Time Reference in use, the niSync timing family of devices adjusts its board time to the TAI timescale. Therefore, regardless of Time Reference, all events and time stamps occur in the TAI timescale. (Refer to KnowledgeBase 4C6CKR8P for more information about timescales.)

1588

IEEE 1588-2008 Protocol

IEEE 1588 provides a standard protocol for synchronizing clocks connected through a multicast capable network, such as Ethernet. IEEE 1588-2008 provides fault-tolerant synchronization among heterogeneous networked clocks requiring little network bandwidth overhead, processing power, and administrative setup. IEEE 1588 provides this by defining a protocol known as the precision time protocol (PTP).

GPS

Global Positioning System (GPS) is a system of satellites funded and controlled by the US Department of Defense. While GPS is typically considered a technology used to determine location, GPS is also an extremely accurate time source. Every GPS satellite contains multiple atomic clocks. The atomic clocks are controlled and referenced to the Master Clock (MC) at the United States Naval Observatory (USNO), called UTC (USNO). The RMS difference between each individual satellite and UTC (USNO) is generally between 2 and 20 ns. GPS receivers use signals from multiple satellites and use averaging algorithms to determine time, so individual satellite drift is not as significant as the average drift of the entire satellite constellation. The RMS difference of the averaged constellation and UTC (USNO) is routinely maintained to be no greater than 10 ns. Therefore, you can expect the error the satellite distribution medium introduces into the GPS receiver to be within 10 ns of UTC (USNO) globally.

Synchronizing to GPS Time

The NI PXI-6682 can use GPS technology as a Time Reference. The device uses the time updates received by the onboard GPS receiver every second, derives from it the current TAI time, and applies this time as the current board time.

When you initially connect the GPS antenna to the NI PXI-6682, the onboard GPS receiver searches for visible satellites. After detecting at least four satellites, the GPS receiver performs a *self survey*. During a self survey, GPS can be used as a Time Reference, but is less accurate than after the self survey has completed. The self survey is a process of performing measurements of the visible satellites once per second and averaging those measurements so that the current position can be determined as accurately as possible. Once the accurate position is determined, time data received from the GPS satellites can be precisely applied.

A self survey is applied only if the NI PXI-6682 is not configured for *mobile mode*. Use mobile mode if the antenna is moving while the device has power. If the antenna is moving and mobile mode is not enabled, you may get unexpected and invalid timing results. However, using mobile mode degrades onboard GPS receiver accuracy, and you should not use it unless the antenna is moving. For most accurate results, disable mobile mode and maintain the antenna in a fixed position.

Factors Affecting GPS Synchronization Accuracy

You can obtain the best GPS timing results by having an ideally located, long-term, stable GPS antenna installation. Ideally, the GPS antenna should be mounted in a location where it has an unobstructed, clear view of the entire sky. This means that from the location of the GPS antenna, every horizon is visible. This orientation allows the GPS receiver to detect additional satellites and perform additional averaging while discarding the worst signals. It also helps to alleviate effects of multipath, where the GPS receiver does not receive the direct signal from the satellite and instead receives a signal reflected off an object or surface. Multipath signals are delayed in reception and therefore degrade the average timing performance.

Additionally, it is best to ensure the antenna is in a fixed location through the self survey process and throughout use. The self survey improves accuracy by performing long-term averaging of location during the self survey. Any small movement of the antenna during this process or during use reduces accuracy. Even a fixed position antenna may be subject to movement caused by wind or vibration, and should be minimized.

Antenna cable latency also adds constant error. For most accurate results, you must calculate the latency of the GPS antenna cable in use and apply a correction. The niSync timing family of devices supports the Clock Adjustment Offset property to allow this source of error to be removed. For example, if the antenna cable in use has a published latency of 5 ns/m, and the antenna installation uses 30 m of cable, the total delay that the antenna installation causes is 150 ns. You can correct this by setting the Clock Adjustment Offset to 150. Remember to account for all sources of delay in your GPS installation, including cable, lightning arrestors, or amplifiers.

The niSync timing family of devices supports querying the number of visible satellites through the Satellites Available property and determining if any fatal GPS errors are present through the Status property. A minimum of four satellites should be visible for stable GPS clock operations, and GPS clock accuracy and stability increase as the number of visible satellites increases. Fatal GPS errors, such as less than four satellites visible, are reported through the Status property.

IRIG

IRIG Protocol Overview

The Inter Range Instrumentation Group (IRIG) currently defines six serial protocols for distributing time codes. Each of the six versions described in the IRIG specification describes a data frame format containing time/date information and the means for signaling and encoding the data.

IRIG-B is probably the most common IRIG format and is the one the NI PXI-6682 supports. IRIG-B specifies that a 100-bit time frame is transmitted once per second, with each bit having a duration of 10 ms. Data in the time frame includes Binary Coded Decimal (BCD) time of year, year, and straight binary seconds (SBS). The data can be DC biased (DC) or amplitude modulated (AM) with a 1 kHz sine wave.

The reception of the first bit of an IRIG-B frame causes a time stamp to be generated for the event. The time stamp cannot be read or used as a Time Reference until the entire IRIG-B frame has been received and decoded. After a successful decode of the frame, it drives the Time Reference engine if IRIG-B is configured as the Time Reference. Both the time stamp generated by receiving the first frame bit and the time/date encoded in the IRIG-B frame can be read using the Read Last IRIG Time Stamp function.

PPS

Synchronizing to a Pulse Per Second (PPS)

The NI-Sync timing family of devices can use an external pulse per second (PPS) signal as a Time Reference. Configuring PPS as the Time Reference configures the device to interpret a rising edge on the configured input as representing a second’s boundary. As the PPS signal cannot indicate an absolute time, you can configure the device to use either a manual start time or its current time, and use the PPS signal only to correct frequency.

If configured to use a manual start time, the first pulse received on the configured PPS input terminal is interpreted to represent the start time configured. Every subsequent pulse is interpreted as having occurred one second after the previous pulse. This configuration allows for easy synchronization of multiple systems instrumented with niSync timing family devices, if absolute time is not a concern. You can configure the systems to be synchronized to use PPS as the Time Reference, with the same manual start time configured. You then can connect the PPS signal to the systems and start the PPS output. As the systems are connected to the same signal, they are closely synchronized.

If configured not to use a manual start time, the first pulse received is interpreted to represent the time equal to the device's current time. Therefore, no correction is applied when the first pulse is received. Every subsequent pulse is interpreted as having occurred one second after the previous pulse. This configuration allows for distributing frequency corrections to multiple systems without concern for actual time values.

For best results when using PPS Time Reference, ensure that the device supplying the PPS signal can provide a stable, consistent 1 Hz signal. You can achieve optimal results when an Oven Controlled Crystal Oscillator (OCXO) or better drives the source signal. You can introduce error into the system if the reference signal contains significant jitter or the reference frequency strays from 1 Hz.

Application of Timing Technologies

Because distributed clocks using Time Referencing have precise synchronization capabilities, they are being used for many applications, including:

- Test and measurement
- Factory automation
- Power plants
- Telecommunications
- Robotic control

You can use the National Instruments niSync timing devices to perform the following synchronized distributed measurement and automation tasks:

- Read the current time
- Create future time events
- Time stamp triggers and pulse trains
- Create synchronized clocks

A Technical Support and Professional Services

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

- **Support**—Technical support at `ni.com/support` includes the following resources:
  - **Self-Help Technical Resources**—For answers and solutions, visit `ni.com/support` for software drivers and updates, a searchable KnowledgeBase, product manuals, step-by-step troubleshooting wizards, thousands of example programs, tutorials, application notes, instrument drivers, and so on. Registered users also receive access to the NI Discussion Forums at `ni.com/forums`. NI Applications Engineers make sure every question submitted online receives an answer.
  - **Standard Service Program Membership**—This program entitles members to direct access to NI Applications Engineers via phone and email for one-to-one technical support as well as exclusive access to on demand training modules via the Services Resource Center. NI offers complementary membership for a full year after purchase, after which you may renew to continue your benefits.

    For information about other technical support options in your area, visit `ni.com/services`, or contact your local office at `ni.com/contact`.
- **Training and Certification**—Visit `ni.com/training` for self-paced training, eLearning virtual classrooms, interactive CDs, and Certification program information. 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, National Instruments Alliance Partner members can help. To learn more, call your local NI office or visit `ni.com/alliance`.

- **Declaration of Conformity (DoC)**—A DoC is our claim of compliance with the Council of the European Communities using the manufacturer's declaration of conformity. This system affords the user protection for electromagnetic compatibility (EMC) and product safety. You can obtain the DoC for your product by visiting `ni.com/certification`.
- **Calibration Certificate**—If your product supports calibration, you can obtain the calibration certificate for your product at `ni.com/calibration`.

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}

Numbers/Symbols

1588 epoch	A period of absolute time defined by the IEEE 1588 specification. The current 1588 epoch is assigned the number 0 and started at 0 hours 1 January 1970. The length of a 1588 epoch is 2^{32} seconds.
1588 grandmaster clock	The 1588 clock to which all other 1588 devices in a specific PTP subdomain are synchronized.
1588 master clock	The 1588 clock to which other 1588 devices are synchronized if they are directly connected to it (that is, they are not connected through a boundary clock).
1588 time	The time format specified by IEEE 1588. IEEE 1588 represents time as a 32-bit unsigned integer for the number of seconds and a 32-bit unsigned integer for the number of nanoseconds since the 1588 epoch. From 1 January 1972 onward, 1588 time follows TAI time with an offset of 10 seconds.
%	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.

G

GPS	Global Positioning System—a system of satellites that broadcast accurate times. GPS receivers acquire these times, which you can use to establish geographic position. You can also use the time received as an accurate clock source.

H

Hz	Hertz—the number of scans read or updates written per second.

I

IEEE 1588	The IEEE specification that describes a synchronization protocol for clocks of multiple devices connected through a network.
in.	Inch or inches.
IP	Internet Protocol—a packet-based protocol used to communicate between multiple computer systems on a network. The IP is a low-level protocol on top of which other, more reliable, protocols are often defined.

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.
NIC	Network Interface Card—a device that connects a computer system to a network.
NTP	Network Time Protocol—a protocol that synchronizes the clocks of computers connected through an IP network. You may use NTP to synchronize computer clocks over a very wide geographical area.

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.
PTP	Precision Time Protocol—the IEEE 1588-defined network protocol used to synchronize the clocks of multiple devices connected through a network.
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.
t_{setup}	Setup time.
TAI	International Atomic Time. Unlike UTC, TAI does not account for leap seconds. Therefore, TAI is the time system employed by network standards for which leap seconds may be problematic.
TRIG	Trigger signal.
trigger	A digital signal that starts or times a hardware event (for example, starting a data acquisition operation).

U

UTC	Coordinated Universal Time—the time system that accounts for leap seconds and is employed by many network standards, including NTP.

V

V	Volts.
VI	Virtual instrument.

Index